Deep learning for churn prediction

Louis Geiler

Université Paris Cité

École doctorale Informatique, Télécommunications et Electronique (EDITE, ED 130)

Laboratoire : Centre Borelli (UMR 9010)

Équipe : Intelligence artificielle pour la science des données et la cybersécurité (AI-DSCy)

Apprentissage Profond pour la Prédiction de l'Attrition

Spécialité : Science des données

Louis GEILER

Thèse présentée en vue de l'obtention du titre de Docteur en Informatique à Université Paris Cité

Date de soutenance : 07.12.2022

Directeur de thèse : Prof. Mohamed Nadif (Université Paris Cité)

Jury composé de :

Mustapha LEBBAH, Prof,	Université de Versailles, Rapporteur
Rafik ABDESSELAM, Prof,	Université Lyon 2, Rapporteur
Blaise HANCZAR, Prof,	Université d'Évry, Examinateur
Ndèye NIANG, MC-HDR,	CNAM, Examinateur
Séverine AFFELDT, MC,	Université Paris Cité, Co-encadrante
Mohamed NADIF, Prof,	Université Paris Cité, Directeur de thèse

Université Paris Cité
45, rue des Saints-Pères
75006 Paris

Université Paris Cité

École doctorale Informatique, Télécommunications et Electronique (EDITE, ED 130)

Laboratory : Centre Borelli (UMR 9010)

Team : Artificial Intelligence for Data Science and Cybersecurity (AI-DSCy)

Deep Learning for Churn Prediction

Speciality : Data Science

Louis GEILER

This Dissertation is submitted for the degree of Doctor of Computer Science at the Université Paris Cité

Defense Date : 07.12.2022

Thesis director : Prof. Mohamed Nadif (Université Paris Cité)

Dissertation committee :	Mustapha LEBBAH, Prof,	Université de Versailles, Rapporteur
	Rafik ABDESSELAM, Prof,	Université Lyon 2, Rapporteur
	Blaise HANCZAR, Prof,	Université d'Évry, Examiner
	Ndèye NIANG, MC-HDR,	CNAM, Examiner
	Séverine AFFELDT, MC,	Université Paris Cité, Co-supervisor
	Mohamed NADIF, Prof,	Université Paris Cité, Thesis director

Université Paris Cité
45, rue des Saints-Pères
75006 Paris

Short Abstract

The problem of churn prediction has been traditionally a field of study for marketing. However, in the wake of the technological advancements, more and more data can be collected to analyze the customers behaviors. This manuscript has been built in this frame, with a particular focus on machine learning. Thus, we first looked at the *supervised learning* problem. We have demonstrated that logistic regression, random forest and XGBoost taken as an ensemble offer the best results in terms of Area Under the Curve (AUC) among a wide range of *traditional machine* learning approaches. We also have showcased that the re-sampling approaches are solely efficient in a local setting and not a global one. Subsequently, we aimed at fine-tuning our prediction by relying on *customer segmentation*. Indeed, some customers can leave a service because of a cost that they deem to high, and other customers due to a problem with the customer's service. Our approach was enriched with a novel deep neural network architecture, which operates with both the auto-encoders and the k-means approach. Going further, we focused on *self-supervised learning* in the tabular domain. More precisely, the proposed architecture was inspired by the work on the `SimCLR` approach, where we altered the architecture with the *Mean-Teacher model* from semi-supervised learning. We showcased through the win matrix the superiority of our approach with respect to the state of the art. Ultimately, we have proposed to apply what we have built in this manuscript in an industrial setting, the one of Brigad. We have alleviated the company churn problem with a random forest that we optimized through grid-search and threshold optimization. We also proposed to interpret the results with `SHAP` (*SHapley Additive exPlanations*).

`Keywords:` Supervised Learning, Self-supervised learning, Deep learning, Autoencoder, Clustering and Churn.

Résumé court

Le problème de la prédiction de l'attrition est généralement réservé aux équipes de marketing. Cependant, grâce aux avancées technologiques, de plus en plus de données peuvent être collectés afin d'analyser le comportement des clients. C'est dans ce cadre que cette thèse s'inscrit, plus particulièrement par l'exploitation des méthodes d'apprentissages automatiques. Ainsi, nous avons commencés par étudier ce problème dans le cadre de l'*apprentissage supervisé*. Nous avons montré que la combinaison en ensemble de la régression logistique, des forêt aléatoire et de XGBoost offraient les meilleurs résultats en terme d'Aire sous la courbe (Are Under the Curve, AUC). Nous avons également montré que les méthodes du type ré-échantillonage jouent uniquement un rôle *local* et non pas global. Ensuite, nous avons enrichi nos prédictions en prenant en compte la *segmentation des clients*. En effet, certains clients peuvent quitter le service à cause d'un coût qu'ils jugent trop élevés ou suite à des difficultés rencontrés avec le service client. Notre approche a été réalisée avec une nouvelle architecture de réseaux de neurones profonds qui exploite à la fois les autoencodeur et l'approche des k-means. De plus, nous nous sommes intéressés à l'*apprentissage auto-supervisé* dans le cadre tabulaire. Plus précisément, notre architecture s'inspire des travaux autour de l'approche `SimCLR` en modificant l'architecture *mean-teacher* du domaine du semi-supervisé. Nous avons montré via la *win matrix* la supériorité de notre approche par rapport à l'état de l'art. Enfin, nous avons proposé d'appliquer les connaissances acquises au cours de ce travail de thèse dans un cadre industriel, celui de Brigad. Nous avons atténué le problème de l'attrition à l'aide des prédictions issues de l'approche de forêt aléatoire que nous avons optimisés via un grid search et l'optimisation des seuils. Nous avons également proposé une interprétation des résultats avec les méthodes `SHAP` (*SHapley Additive exPlanations*).

`Mots-Clefs :` Apprentissage supervisé, Apprentissage auto-supervisé, Deep learning, Autoencodeur, Clustering et Attrition.

Résumé substantiel

Le problème de la prédiction de l'attrition est généralement réservé aux équipes de marketing. Cependant, grâce aux avancées technologiques, de plus en plus de données peuvent être collectées afin d'analyser le comportement des clients. C'est dans ce cadre que cette thèse s'inscrit, plus particulièrement par l'exploitation des méthodes d'apprentissages automatiques.

La première étape de cette thèse a été la réalisation dun état de lart sur lensemble des méthodes de lapprentissage supervisé pour résoudre le problème du churn. La motivation initiale était que létat de lart est majoritairement basé sur des jeux de données privées ce qui rend la reproductibilité des expériences irréalisables. En contraste, nous proposons des jeux de données libres ainsi quune pipeline totalement reproductible. Cette pipeline est constituée de trois étapes successives : le ré-échantillonnage optionnel qui se base sur le sur-échantillonnage, sous-échantillonnage ou les méthodes hybrides. Puis une phase dapprentissage des modèles avec trois grandes familles dalgorithmes, ceux dit classique, les réseaux de neurones et les approches semi-supervisés. Enfin, nous poursuivons sur le calcul des scores qui est basé sur une validation croisée stratifié k-fold. Nous avons fait le choix du ROC-AUC pour le score, ce choix peut être débattu en effet, le ROC-AUC noffre pas un score significatif lorsque les données sont déséquilibrées. Malgré cette contrainte, nous avons fait le choix du ROC-AUC comme une solution générique. Une fois la pipeline mise en place, nous proposons plusieurs visualisations, comme celle avec test de nemenyi qui permet de comparer les rangs des classificateurs pour savoir si ils sont statistiquement similaires. On conclut que sans ré-échantillonnage, LR, RF, XGB et GEV-NN sont similaires en termes de rang calculé via lAUC. Et finalement nous visualisons sur un plan par analyse des correspondance les relations entre classificateurs et jeux de données.

Dans le second chapitre, nous poursuivons nos analyses en montrant sur une boîte à moustache que les méthodes ensembles constituées de trois ou quatre classificateurs avec LR, XGB, RF et NN donnent les meilleurs résultats. Cette première intuition est appuyée sur notre table dAUC qui compare lensemble des combinaisons ensemble ainsi que les combinaisons de ré-échantillonneurs. Dans lensemble le trio LR, XGBoost et RF obtiennent le score de 0.8577 qui est le meilleur. Enfin, nous comparons notre proposition densemble sur chaque jeux de données et on note que notre approche est la plus générique. Autrement lutilisateur devrait pour chaque jeux de données trouver la combinaison la plus efficace ce qui peut consommer beaucoup de temps.

Dans le second sous-chapitre nous avons enrichi nos prédictions en prenant en compte la segmentation des clients. En effet, certains clients peuvent quitter le service à cause d'un coût qu'ils jugent trop élevés ou suite à des difficultés rencontrés avec le service client. Pour ce faire nous avons construit une architecture de réseaux de neurones à base d'auto encodeur et de k-means ce qui permet doptimiser à la fois la qualité de la réduction de la dimension et du clustering. Une fois que cette approche a été réalisée nous avons exploité lapproche ensemble générique du précédent sous-chapitre pour apprendre nos modèles sur chacun des sous-clusters. Cette combinaison de classificateurs est réalisée à partir dune moyenne pondérée par la corrélation des classificateurs. Cette étape préliminaire nous permet de construire un benchmark et montre la supériorité de notre approche en termes dAUC par rapport à nos deux baselines LLM et Rf-based. Nous avons également profité du fait davoir des clusters pour construire une importance des features par forêts aléatoires. Ceci nous permet par exemple dans le cas du jeux de données banque, de voir que dans un cluster les churner sont surtout allemand avec un score de crédit significatif et il soppose à un groupe despagnol.

Ensuite dans le chapitre 3, nous nous sommes attaqué aux approches auto supervisées dans le cadre tabulaire. Plus précisément, notre approche sappuie sur lapprentissage contrastif et larchitecture SimCLR. Nous nommons notre solution Mean teacher Architecture using Contrastive learning (MAC). Nous commençons par la phrase de pré-entrainement qui va séparer les données en catégorielles et continu. Les données catégorielles passeront par une matrice dembedding et celle continue par un MLP classique afin de faire correspondre les dimensions des deux features. Ensuite les données sont divisées en deux branches : celle du haut ne sera pas perturbé et celle du bas le sera soit par un processus de diffusion soit par MixUp. Ensuite les données sont envoyées dans un réseau de neurones SAINT puis divisé en deux branches le Student et le Teacher. La fonction de coût est basée sur InfoNCE. La rétropropagation se fait normalement pour la branche du Student mais par contre pour celle du Teacher elle se réalise via une moyenne mobile exponentielle. Une fois le pré-entraînement réalisé, il reste la phase de fine-tuning qui va exploiter la mise à jour des poids du réseau pour effectuer une nouvelle tâche. Cette nouvelle tâche est lapprentissage semi-supervisé. Nos expériences ont été calculées

par le ROC-AUC en utilisant la matrice de Win. Nous montrons que lorsque 20% des données sont labellisés, lapproche ema+mixup donne le meilleur résultat. Lorsque les données sont à 70% labellisées on obtient le même retour.

Enfin, nous avons proposé d'appliquer les connaissances acquises au cours de ce travail de thèse dans un cadre industriel, celui de Brigad. La première étape a été de créer le vecteur de churn qui est calculé par analyse de cohorte. C'est-à-dire que nous suivons pendant trois mois un groupe de clients actifs, puis nous regardons si les trois prochains mois ils seront toujours actifs (non churner) ou si ils seront inactifs (churner). Nous avons ensuite créé la matrice X de 2481 entreprises, avec huit variables numériques et une variable catégorielle. Dans les variables numériques certaines ont une importance significative comme le no-show qui va compter le nombre de fois où le business a subi le fait quun talent na pas effectué sa mission et quil na pas prévenu lentreprise. On a également le nombre dannulation de moins de 4h qui suit le même processus mais lentreprise a été prévenue. Cette introduction nous a permis de créer une pipeline complète pour la résolution du problème du churn. On a entre autres, une partie importante constituée dextraction de données par requête SQL puis du feature engineering qui revient à transformer les features en utilisant des connaissances dentreprises. Puis nous traitons nos données en les dummifiant pour les catégorielles et en les fixant entre 0 et 1 pour les features continues. Ceci nous permet de filtrer nos entreprises, par exemple nous ne prendrons pas en compte celles qui sont très récentes. On va ensuite apprendre sur nos données avec les approches LR, RF, XGB et ensemble. Nos expériences ont montré la supériorité de lapproche RF en termes dAUC médian sur une validation croisée stratifiée avec 5 folds. Enfin le problème des faux positifs et assez critique dans le cadre du churn lobjectif était de le réduire. En effet, un taux de faux positifs élevés implique de contacter des clients qui ne vont pas churners pour leurs donner potentiellement des coupons, chose que nous voulons éviter. La phase de réduction des faux positifs passe par un grid search, puis une optimisation du seuil. Enfin, nous avons réalisé des graphes SHAP afin de vérifier la significativité des résultats.

Pour conclure, cette thèse a eu pour but de trouver de nouvelles solutions pour la résolution du problème de la prédiction du churn. Nous avons commencé par un état de lart qui nous a permis d'envisager une approche ensemble constitué de LR, RF et XGBoost qui a été efficace. Ensuite, nous avons rajouté une segmentation clients par un auto encodeur et du clustering appris conjointement. Puis nous avons abordé lapprentissage auto supervisé dans le cadre tabulaire nous avons proposé lapproche MAC. Enfin, nous avons présenté nos méthodes dans un cadre industriel pour la prédiction du churn entreprise chez Brigad.

Summary

1

Introduction

1.1 Motivation

Retaining customers is at a fundamental importance for the stability of a company which must define an efficient retention strategy to secure its longevity. The cost in terms of time and money of seeking new customers is customarily higher than retaining them (Reinartz and Kumar, 2003 ; Reichheld and Sasser, 1990). In this sense, gauging the risk related to the churn associated with each customers is crucial to prevent the loss of profitable clients.

Attrition or *churn* detection is a process enabling the prediction of the customers that will most probably quit or leave a service. At `Brigad`*, we are generally interested in predicting the probability for a customer – which can be either a *freelancer* or a *business* – to stop using the `Brigad` recruitment platform in a near future. Such information can enable the customer service to plan efficient preventive actions, e.g. offering a discount, to retain the customers. Several supervised learning approaches can be used to predict the risk of customer loss (*customer churn models*) based solely on the data (Larivière and Van den Poel, 2005) and initiate retention marketing campaign (Mozer et al., 2000). However, as the heterogeneity of clients' profiles and the diversity of services grow, offering robust predictions remains a challenge, even though the amount of data increases. An interesting strategy to improve classification results is to combine the advantages of several efficient algorithms that are diverse enough within an ensemble approach. In particular, our experiments revealed that combining Logistic Regression (`LR`), eXtreme Gradient Boosting (`XGBoost`), Random Forest (`RF`) and MultiLayer Perceptron (`MLP`) by averaging their predicted probabilities induced a much more performant approach on a wide range of real-world benchmark datasets than each of these approach taken independently (Chapter 2). We first demonstrated the strength of our ensemble proposal as compared to baselines models (Geiler et al., 2022) (Chapter 3) and then extended our framework to the realm of *tabular deep learning* which has been experiencing recently novel promising ideas (Chapter 4).

One way to better handle a churn analysis is to address the inherent customer's segmentation issue. Common solutions draws their strategies from clustering (Athanassopoulos, 2000 ; Kuo et al., 2006 ; Chan, 2008). The aim of customer's segmentation is to *target* several customers for a specific marketing campaign and hence concentrate the marketing efforts to one or a few key segments. Another churn prediction hurdle is the amount of unlabeled data. Labeling manually a very large dataset is usually unfeasible and error prone (Northcutt et al., 2021). Recent research combining self-supervised learning SimCLR (Chen et al., 2020) and semi-supervised learning demonstrate great potential when tackling tabular data (Bahri et al., 2021 ; Somepalli et al., 2021).

*This thesis project was conducted in parallel within the Prof. Nadif's team at the Université Paris Cité and at the `Brigad` company. `Brigad` creates the connection for hospitality and healthcare businesses to propose temporary missions to professional freelance workers.

For a company such as `Brigad`, the demand for churn prediction is twofold. Indeed, the online platform `Brigad` connect freelancers (also called *Talents*) and companies, within the scope of short term contract. Hence, both freelancers and companies may exhibit a substantial risk of choosing another mission's/freelancers' provider (Keaveney and Parthasarathy, 2001). It is thus critical to continuously ensure a sufficient number of *stable Talents* with a wide range of qualifications to meet the companies demand. Undoubtedly, an accurate and reliable churn prediction model would be a strong asset for `Brigad` to simultaneously monitor the loyalty of businesses and freelancers.

1.2 Table of content

- **Chapter 1 : Preliminaries and Baselines** reviews traditional machine learning techniques and algorithms for the churn prediction issue. It also provides baseline scores by summarizing various *area-under the curve* (AUC) results for these most common models.
- **Chapter 2 : Ensemble for churn prediction** is an extension of the previous chapter to the framework of ensemble classification methods. Based on our experimental results, we propose to combine `LR`, `XGBoost`, `RF` and `MLP` to benefits from their respective strengths regarding imbalanced datasets. Furthermore, customers profile being naturally diverse, we enrich our proposal with a segmentation step prior to classification. Our final extension hinges on a deep-clustering architecture based on *autoencoder*.
- **Chapter 3 : Deep learning for Tabular data** Dealing with a vast quantity of unlabeled data remains a challenge. As such we propose a novel *deep-learning framework* that is built on the student-teacher connection from semi-supervised learning in a SimCLR like architecture.
- **Chapter 4 : An industrial application** is the opportunity of building a complete *customer churn model* in an industrial setting. As such the ensemble classification approach have been evaluated in our `Brigad` private dataset. Additionally, our model has been fine-tuned to lower the number of false positives. Indeed, a high false positive rate would imply for the company to spend time and money on contacting a great number of non-churners.
- **Conclusion and perspective** reviews the main contributions from this thesis in terms of churn and deep-learning in the tabular context, and lays the foundation for future works.

1.3 Publications

Publications of this thesis that have been published

(a) Louis Geiler et al. 2022 *A Technical Survey and Comparative Machine Learning Study for Churn Prediction*, Louis Geiler, Séverine Affeldt et Mohamed Nadif. International Journal of Data Science and Analytics (JDSA)

(b) Louis Geiler et al. 2022 *Apprentissage machine pour la prédiction de l'attrition : une étude comparative, Louis Geiler, Séverine Affeldt et Mohamed Nadif. Extraction & Gestion des Connaissances (EGC)*

(c) Louis Geiler et al. 2022 Machine Learning for churn prediction and customer profiling. *Louis Geiler, Séverine Affeldt et Mohamed Nadif. Data & Knowledge Engineering (DKE)*

Publications that are currently under review

(a) Louis Geiler et al. 2022 Mean-teacher using Contrastive learning (MAC). Louis Geiler, Séverine Affeldt et Mohamed Nadif. *European Conference on Information Retrieval (ECIR2023)*

1.4 Notations

Throughout our papers, we use bold uppercase characters to denote vectors, uppercase characters to denote random variable and lowercase characters to denote variable values. Let $\mathbf{X} = (x_{ij})$ be a data matrix of $n \times d$ dimension. We assume that Y is the random variable indicating the class y_i of an observation $\mathbf{x}_i = [x_{i1}, \ldots, x_{id}]^\top$ which denotes the i^{th} instance of $\mathbf{X}$. The total number of observations is noted n, and G is the number of classes $C_1, \ldots, C_G$. The churn prediction problem can be modeled as a standard binary classification task. Formally, it is an assignment task that amounts to estimate the conditional probability of $Y = y_i$ given $\mathbf{x}_i$, $P(Y = y_i|\mathbf{x}_i)$, so-called *class posterior*. Note that in a binary or churn prediction context, $G = 2$ and we consider the two classes $+,-$ that correspond to the churn and non churn classes respectively.

2

Investigating Machine Learning techniques for Churn

The *churn prediction* is at the source of a wide area of research. In this chapter, we address the churn prediction issue in a supervised machine learning context. Our investigations and experimental results gave rise to a survey on machine learning for churn prediction (Geiler et al., 2022). In a nutshell, we have considered sixteen public churn-like datasets to perform experiments along with eight classical machine learning techniques from the supervised learning framework. In addition, we applied the data pre-processing strategies usually proposed in imbalanced data contexts, namely *oversampling*, *undersampling* and *hybrid* approach. This chapter is thus investigating the following question : which algorithm is the most suitable for the churn task at hand ? All in all, using default hyperparamters, our experiments demonstrate the superiority of `GEV-NN` (Munkhdalai et al., 2020), an anomaly detection algorithm, as the most advisable churn prediction technique.

2.1 Context and background

2.1.1 Introduction

Building a strong Customer Relationship Management (CRM) has become a crucial topic for many companies in recent years. In particular, management and marketing services are focusing their attention on the customer retention, as it clearly appeared that the acquisition costs of a new customer can be much more higher than the retention costs of an existing one (Reinartz and Kumar, 2003 ; Siber, 1997 ; Yang and Peterson, 2004). Besides, retained customers can be of great help for the company by spreading positive word of mouth (Reichheld and Sasser, 1990), which would subsequently lower the marketing costs of new customers acquisition (Bolton and Bronkhorst, 1995). The ever-rising competition in industry has therefore pushed forward companies to carefully control the switch of customers or subscribers to another company, also known as customer *churn*, customer *attrition* or customer *defection*. The customer churn can be particularly damaging for subscription-based service firms, such as insurance (Günther et al., 2014), banking (Kumar et al., 2008), online gambling (Coussement and De Bock, 2013), online video games (Kawale et al., 2009), music streaming (Chen et al., 2018), online services Tan et al. (2018) or telecommunication (Effendy et al., 2014 ; Abdillah et al., 2016 ; Hudaib et al., 2015 ; Hosein et al., 2021). As such companies are expecting fixed and regular membership fees, customer switching behavior should be tempered to ensure sustainable profits. Therefore, accurately predicting the customers who are prone to churn has become a priority in industry.

In addition to the systematic prediction of customers with switching intentions, firms also seek to determine the causes of churn behavior. Knowing the reasons for customers defection would both provide support for the profiling of defection-prone customers and help fostering efficient pro-active campaigns for customers retention (Leung et al., 2021). The customer data generally contains service usage (e.g. frequency, duration), billing information (e.g. regularity of payments, contract term) and support service usage and satisfaction. Among the most probable antecedents of customer churn, several prior studies have reported the satisfaction and the service quality (Anderson and Sullivan, 1993 ; Zeithaml et al., 1996). Finding the most significant churn behavior causes (or *features*) also bring a valuable technical advantage for the prediction model formulation. Indeed, the number of features in churn datasets is usually large and dimentionality reduction helps reducing overfitting and improving the generalization of the prediction models.

Marketing and financial industry services preferentially focused on statistic modeling methods to tackle the churn analysis and prediction task. A well-known approach is the *survival analysis* that proposes to model the occurrence and timing of events (Van den Poel and Lariviere, 2004 ; Bhattacharya, 1998 ; Bolton, 1998). In the context of customer attrition, the time to failure corresponds to the churn behavior. The potential churner behavior has also been analyzed using *structural equation modeling* (Varki and Colgate, 2001 ; Nguyen and LeBlanc, 1998 ; Ganesan, 1994). Such approach can be of great interest for managerial decisions, as it evaluates the effect of suspected influential features on a specific customer decision, such as churn. The *analysis of variance* was also widely used in marketing and business areas to uncover customer behavior (Maxham III, 2001 ; Mittal and Kamakura, 2001 ; Zeithaml et al., 1996). Financial and retail services also rely on *T-test* and *Chi square* statistics to forecast customer behavior and perceptions (Hitt and Frei, 2002 ; Paulin et al., 1998 ; Mittal and Lassar, 1998).

The primary objective of our investigations is not to explore these traditional approaches and rather focuses on machine learning techniques that are being increasingly encountered in the customer churn

context. These techniques include supervised and semi-supervised approaches. K-nearest neighbors, Naive Bayes classifiers, Linear Regression, Logistic Regression, Linear Discriminant Analysis (Xie and Li, 2008), Decision Tree learning (Hadden et al., 2006; Mozer et al., 2000) and Support Vector Machine are among the widely used supervised algorithms in the context of churn prediction. Algorithmic modifications (Zadrozny and Elkan, 2001) and cost-sensitive learning variants (Domingos, 1999; Zadrozny et al., 2003) of the aforementioned learning methods have also been proposed in the context of imbalanced classes, as encountered in churn datasets. Finally, several studies proposed to rely on ensemble approaches such as Random Forest, AdaBoost (Xie and Li, 2008), Gradient Boosting (Mozer et al., 2000; Lemmens and Croux, 2006) or XGBoost (Gregory, 2018) to tackle the churn prediction task. Successful semi-supervised methods have been proposed (Li et al., 2016), as well as deep learning approaches that offer promising results (Tan et al., 2018; Gregory, 2018; Hadden et al., 2006; Mozer et al., 2000).

The churn prediction problem relates to the broader issue of class imbalance from which the anomaly or outlier detection is an extreme case (Kong et al., 2020). Efficient anomaly detection systems provide valuable information in a wide range of diverse domains, such as medical diagnostic systems (Cabral and Oliveira, 2014), fraud detection (Kamaruddin and Ravi, 2016) or industrial fault detectors (Xiao et al., 2016). Many approaches have been proposed to tackle the outlier detection task (Chandola et al., 2009; Alam et al., 2020; Pang et al., 2017; Taha and Hadi, 2019). In particular, semi-supervised approaches regularly provide state-of-the-art results (Alam et al., 2020; Villa-Pérez et al., 2021). Among the well-known semi-supervised techniques for anomaly detection, one could cite Local Outlier Factor (LOF) (Breunig et al., 2000), One-Class SVM (ocSVM) (Schölkopf et al., 1999), Isolation Forest (iForest) (Liu et al., 2012) and Support Vector Data Description (SVDD) (Tax and Duin, 1999) methods. The deep learning research field enabled also the emergence of a large number of deep anomaly detection methods (Pang et al., 2021). In particular, GEV-NN (Generalized Extreme Value Neural Network) which proposes to use Gumbel distribution as an activation function, reaches state-of-the-art results in the context of imbalanced data (Munkhdalai et al., 2020). DevNet (Deviation Network) also demonstrates efficiency and competing results for anomaly detection (Pang et al., 2019).

2.1.2 Related works

In recent years, churn prediction triggered novel strategies for which machine learning approaches were used and adapted. The strong interest in churn prediction led to various surveys related to machine learning in the fields of telecommunication industry, human resources, bank subscription or financial services. Saradhi and Palshikar (2011) reviewed three machine learning techniques in the *employee churn* context, a problem similar to customer churn prediction. They provide comparative results on a private dataset using a cross-validation procedure. Similarly, Śniegula et al. (2019) compare three machine learning techniques on a single churn dataset in the context of telecommunication industry. Keramati et al. (2014) proposed a literature and comparative experimental study with four models on a private dataset. Other comparative studies based on ensemble machine learning approaches were also proposed by Risselada et al. (2010), Lemmens and Croux (2006) and Wang et al. (2017). Umayaparvathi and Iyakutti (2016) literature survey, which focuses on customer churn prediction in telecommunication, provides a list of regularly encountered models in churn analysis. The authors indicate four publicly available churn datasets and briefly discuss the possible metrics. A more thorough literature review was proposed by García et al. (2017). Several steps of the churn prediction analysis are discussed by the authors, among which the data gathering, the features selection, the model implementation and the possible evaluation procedures and metrics. Their survey concludes with recommendations based on literature. Several deep learning approaches have been investigated for churn prediction. In Seymen et al. (2020), the authors proposed a novel deep learning model which is compared to logistic regression and artificial neural network models. Their study encloses a detailed literature review of deep learning methods in churn prediction. Beyond this domain, several reviews dedicated to anomaly detection, which can be seen as an extreme case of churn prediction, have been proposed. In Ruff et al. (2021), the authors highlight connections between classic *shallow* and novel deep approaches applied to anomaly detection. A thorough deep anomaly detection review, recently proposed by Pang et al. (2021), provides a comprehensive taxonomy of deep learning techniques for anomaly detection and discusses the associated challenges and perspectives.

Although interesting, these surveys compare very few machine learning techniques in the churn context and hardly include any experimental study. Furthermore, comparative results usually involve private datasets, making the experiments not reproducible and extrapolation to novel datasets difficult. Beyond discussion on the models themselves, these reviews typically omit the techniques for classes rebalancing, which is an important issue for churn prediction. Finally, churn prediction surveys rarely raised the topic of evaluation procedures that impact the validity and robustness of the evaluations. Part of this thesis work tries to remedy partially this lack by proposing a large comparative machine learning study on public churn-like datasets exclusively.

2.2 Our contribution

Our goal is to compare multiple alternatives within a machine learning churn analysis pipeline that involves *(i)* a sampling stage, *(ii)* a model fitting phase and *(iii)* a robust evaluation procedure (Fig. 2.1). An exhaustive analysis of all existing algorithmic variants and cost-sensitive approaches within this pipeline would not be reasonably feasible. Hence, we rather focus on base learning algorithms in combination with widespread sampling approaches to finally propose a pipeline that is successful on a wide range of churn-like datasets. In the churn context, several data issues have been pointed out in relation with classes imbalance (López et al., 2013 ; Błaszczyński and Stefanowski, 2018 ; Stefanowski, 2016), among which the existence of small *disjuncts* (Weiss, 2010 ; Weiss and Hirsh, 2000 ; Holte et al., 1989), the overlap between classes (Denil and Trappenberg, 2010 ; García et al., 2008), the noisy data (Seiffert et al., 2014) or the borderline instances (Napierała et al., 2010). For this study, we do not try to correct for these specific issues and rather focus on the balancing of the classes distribution as it was shown to play a significant role in the performance of standard classifiers (García et al., 2012). Several deep learning approaches were proposed to tackle the churn prediction problem (Umayaparvathi and Iyakutti, 2017 ; Dingli et al., 2017 ; Yang et al., 2018 ; Castanedo et al., 2014). We propose to compare traditional machine learning approaches to a simple feed-forward neural network and also to more recent and sophisticated deep learning methods which have been shown to be particularly efficient for imbalanced data or in the context of outliers detection (Pang et al., 2019 ; Munkhdalai et al., 2020).

In this Chapter, we first provide an overview of publicly available churn datasets (Section 2.2.2). Then, we introduce the imbalance class distribution issue and describe seven widespread balancing techniques (Section 2.3). The description of supervised, ensemble supervised, semi-supervised and deep learning techniques are given in Section 2.4. We also discuss three evaluation procedures (Section 2.5) and four metrics (Section 2.5.2) before providing the exhaustive experimental results of our pipeline variants (Section 2.6). Our experiments are performed on sixteen publicly available *churn-like* datasets that range from human resources, to telecommunication, internet subscription and music streaming industry. Our results reveal interesting complementary behaviors between machine learning techniques (Section 2.6.2) and ultimately indicate an advisable churn analysis pipeline which can be successfully applied to various churn-like datasets (Chapter 3). We summarized our experimental findings with Nemenyi tests and Correspondence Analysis visualizations (Section 2.6.3). The overall conclusion is given in Section 3.4.

All our experiments are performed with freely accessible Python packages (Appendix A.1.2) and publicly available datasets exclusively (Table 2.1 & Appendix A.1.1). Thus, our results are fully reproducible and the proposed procedure can be easily applied to novel datasets.

2.2.1 Churn prediction pipeline

This section introduces the machine learning churn prediction pipeline used for our experiments and the associated variants that we evaluated (Figure 2.1). This pipeline unfolds in three parts, namely *(i)* Sampling, *(ii)* Model fitting and *(iii)* Evaluation, through which we sequentially combine several techniques. For the sampling, we explore seven different approaches that either correspond to *oversampling*, *undersampling* or *hybrid* (Section 2.3). The sampling objective is to transform the original churn dataset into a similar dataset with a better class balance, either by reducing the majority class, expanding the minority class or both. For the model fitting, we consider eleven supervised and semi-supervised techniques, some of which are *ensemble* approaches. Finally, we discuss in the evaluation step three different procedures and four evaluation metrics.

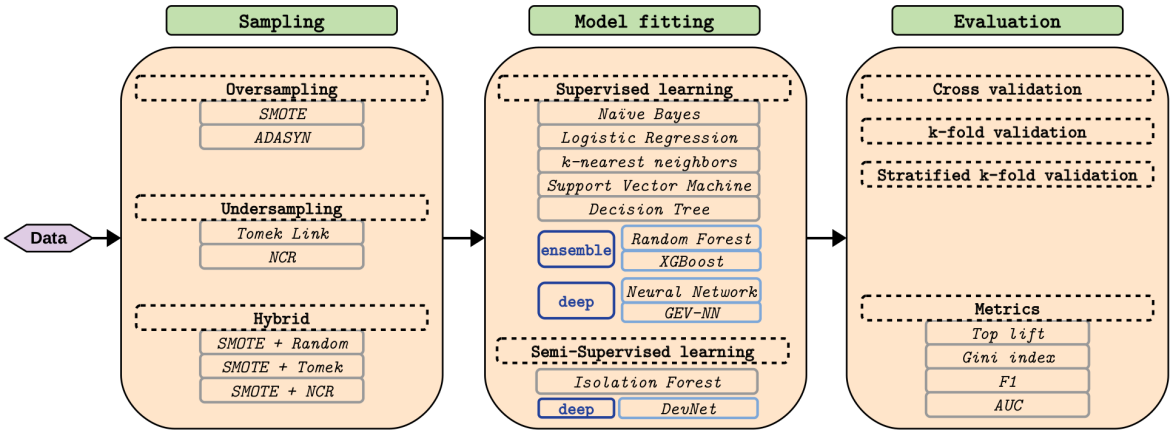


Figure 2.1 : Machine learning pipeline for churn prediction and analysis

Customer defection is an *infrequent* event that is inevitably associated with a class imbalance hassle that impedes the quality of customer churn prediction. This is particularly true when the classes are highly overlapping and when the minority class is divided into sub-clusters. The class rarity issue is widespread throughout a broad range of contexts beyond churn prediction such as fraudulent credit card usage, telecommunication equipment failure or patient survival prediction. In such contexts, instances of the minority or *positive* class induce a great cost when they are not well classified.

2.2.2 Public datasets

Several studies have evaluated machine learning approaches for churn modeling on various datasets. However, these studies typically include private datasets that prevent from reproducibility and extrapolation to novel datasets. In this thesis, we performed a comparative evaluation of multiple churn analysis techniques on publicly available datasets only. A churn dataset usually comprises features of different types that reflect customers behavior. It also generally exhibits a strong class imbalance, as the proportion of churners is typically lower than the proportion of customers that remain with the company. Our benchmark datasets are also enriched with three datasets that are usually found in anomaly detection contexts, namely *Fraud*, *Thyroid* and *Campaign*.

Table 2.1 : Publicly available churn and *churn-like (*)* datasets with online access link

Link to Data	#Instances	#Features	#Dum.Feat.	$\#churn$	$\#non-churn$	$\%churn$	$\frac{\#churn}{\#non-churn}$
*Fraud**	284,807	29	29	492	284,315	0.0017	0.0017
K2009	50,000	230	1,039	3,672	46,327	0.07	0.08
*Thyroid**	7,200	21	21	534	6,666	0.07	0.08
KKbox	970,960	49	56	87,330	883,630	0.09	0.10
UCI	5,000	20	21	707	4,293	0.14	0.16
*Campaign**	41,188	17	63	4,640	36,548	0.12	0.13
HR	1,470	34	86	37	1,233	0.16	0.19
TelE	190,776	19	26	29,884	160,892	0.16	0.19
News	15,855	18	307	3,037	12,818	0.19	0.23
Bank	10,000	12	16	2,037	7,963	0.20	0.25
Mobile	66,469	65	65	13,907	52,562	0.21	0.27
TelC	7,043	20	34	1,869	5,174	0.27	0.37
C2C	71,047	71	75	20,609	50,438	0.29	0.41
Member	10,362	14	26	3,143	7,219	0.30	0.43
SATO	2,000	13	29	1,000	1,000	0.50	1
DSN	1,401	15	32	700	700	0.50	1

Table 2.1 lists the public churn datasets that are considered in this work and provides their online access (see also Appendix A.1.1). These datasets have diverse number of instances, number of features and *dummified* features *, and percentage of churners. The Figure 2.2 gives the distribution of these

*Before fitting a model, categorical variables are converted to their numerical representation through a *dummification*

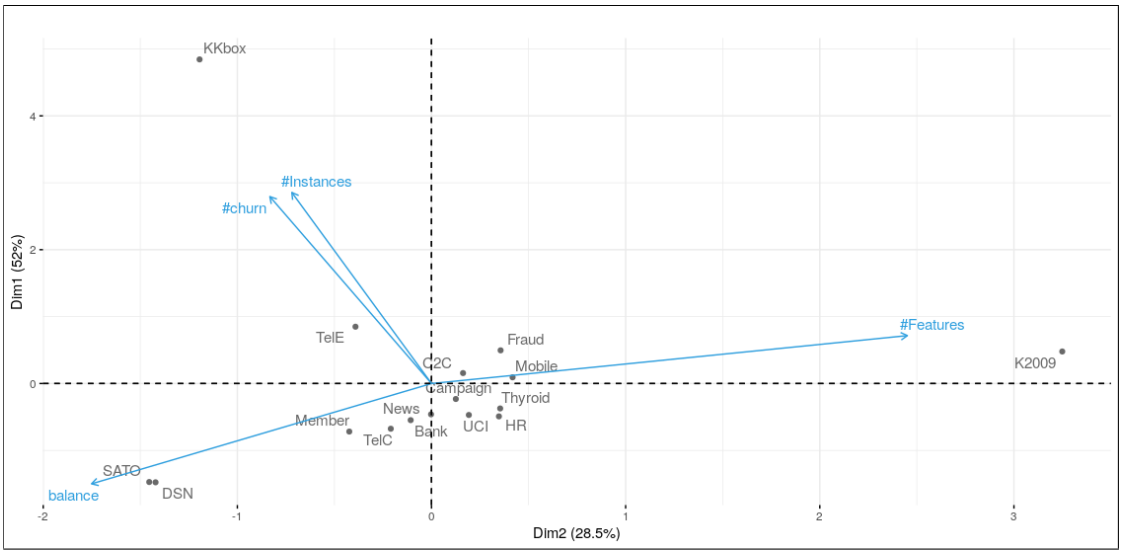


Figure 2.2 : Datasets distribution on the two first PCA components of Table 2.1

datasets in the 2D space obtained with the two first PCA (Principal Component Analysis) components based on the Table 2.1. Although the Figure 2.2 suggests similarities between several datasets, it is important to remind that multiple intrinsic data properties might impact the prediction in the churn context, such as the existence of small *disjuncts*, the overlap between classes, the noisy data or the borderline instances (see Section 2.2). Hence, directly drawing conclusions on the most suitable machine learning based on the general characteristics given in Table 2.1 remains challenging.

Churn datasets call for the use of various sampling methods (Batista et al., 2004 ; Batuwita and Palade, 2010) to change the class distribution. These methods consist in either introducing data points within the minority class (*oversampling*), removing datapoints from the majority class (*undersampling*) or applying both sampling strategies (*hybrid*). Basic and advanced sampling methods have been proposed (Chawla et al., 2002 ; Deville and Tillé, 2004), and several studies showed that undersampling tends to overtake oversampling (Chen et al., 2004 ; Drummond et al., 2003).

2.3 Data sampling

2.3.1 Oversampling

The oversampling methods generally consist in duplicating instances in the minority class or synthesizing new examples from the available instances. A straightforward oversampling approach is the *random oversampling* that randomly selects the instances to be replicated (Ling and Li, 1998). However, random replication can impede the decision boundary performance by for instance repeating outliers. We describe in the following two more sophisticated and widely used oversampling approaches, namely the *Synthetic Minority Oversampling Technique* (SMOTE) (Chawla et al., 2002 ; Fernández et al., 2018) and the *Adaptative Synthetic Sampling* (ADASYN) (He, H., Bai, Y., Garcia, E., & Li, 2008).

(i) Synthetic Minority Oversampling Technique The **SMOTE** technique consists in oversampling the minority class by generating *synthetic* instances along the line segments created by a k-nearest neighbors approach. Specifically, a sample $\mathbf{x}$ is taken at random from the minority class. Then, its k-nearest neighbors $\{\mathbf{x}_i\}_{i\in\{1\dots n\}}$ are considered and used to generate a new synthetic instance following the formula,

$$\mathbf{x}_i^{new} = \mathbf{x} + \mathcal{U}([0,1]) \times (\mathbf{x}_i - \mathbf{x}).$$

While the simple duplication of random instances won't bring any information, new SMOTE instances are plausible observations, similar to original instances from the minority class. However, while SMOTE helps avoiding the overfitting problem, its synthetic instances might be ambiguous in case of strongly overlapping classes.

process where each category becomes a binary variable.

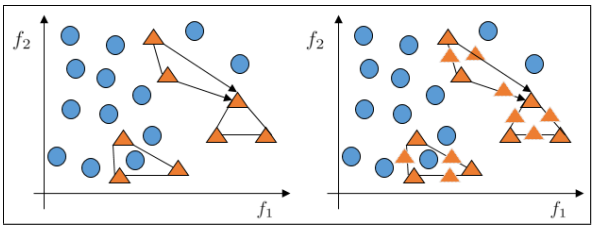


Figure 2.3 : SMOTE Algorithm

To address this issue, three extensions have been proposed, namely *Borderline SMOTE* (Han et al., 2005), *Borderline Oversampling SVM* (Nguyen et al., 2011) and ADASYN (He, H., Bai, Y., Garcia, E., & Li, 2008). The *Borderline SMOTE* focuses on generating instances based on observations that are difficult to classify, according to a k-nearest neighbors classifier while *Borderline Oversampling SVM* uses a SVM classifier to generate new instances. In the following, we focus on the third SMOTE extension, ADASYN.

(ii) Adaptive Synthetic method **ADASYN**, which is based on SMOTE, adaptively generates minority data instances according to their distributions. Specifically, more synthetic instances are generated in the features space regions where the observations density is low, and conversely, fewer synthetic instances are generated from the high density regions. Hence, ADASYN focuses on the class separation boundary region. As for *Borderline SMOTE* and *Borderline Oversampling SVM*, it would be advisable to remove outliers before applying ADASYN.

2.3.2 Undersampling

Undersampling techniques delete instances from the majority class or select a subset of examples. A straighforward approach is to randomly delete instances. However, this can be hazardous and make the classification task more complex as it could lead to the removal of important observations. *Tomek Links* (Tomek, 1976) and *Neighborhood Cleaning rule* (NCR) (Laurikkala, 2001) are more advanced undersampling strategies.

(i) Neighborhood Cleaning rule The **NCR** technique combines two methods that remove from the majority class the instances that are *(i)* redundant and *(ii)* noisy or ambiguous. The first technique is the *Condensed Nearest Neighbor (CNN) Rule* (Hart, 1968), that selects a *minimal consistent set* which is a subset of observations from the majority class that cannot be correctly classified. These samples are considered more relevant for learning. The second approach is the *Edited Nearest Neighbors (ENN) Rule* (Wilson, 1972). It finds and removes noisy and ambiguous instances using a k-`nearest neighbors` approach. With ENN, if a majority class instance is misclassified by its neighbors, it is removed from the dataset. Besides, if a minority class instance is misclassified by its majority class neighbors, the majority class neighbors are also deleted. As shown in Laurikkala (2001), NCR is useful to learn a model upon difficult small classes.

(ii) Tomek links This technique builds on the *Condensed Nearest Neighbor (CNN) Rule* (Hart, 1968) and proposes to identify all *cross-class* pairs of datapoints, i.e. pairs that have a sample from the majority and the minority class that are closest neighbors. Hence, majority samples that belong to *Tomek links* are either boundary instances or noisy instances and should be removed. It is also common to combine CNN and **Tomek links**, as the former will remove redundant samples, while the later deletes noisy/borderline instances.

2.3.3 Hybrid

Over problems beyond the class distribution skewness are usually encountered with churn-like datasets, such as classes overlapping where majority class examples invade the minority class space and conversely. To create a better class separation while balancing the data, various combinations of upsampling and undersampling methods have been proposed. A straightforward hybrid method is

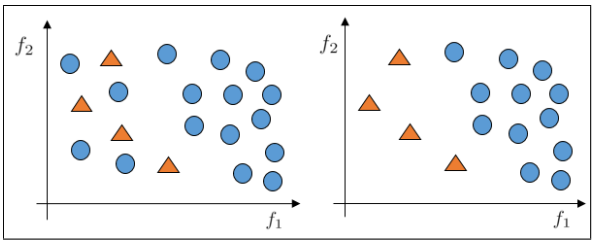


Figure 2.4 : Tomek Algorithm

to combine SMOTE and Random Undersampling approaches. Chawla et al. (2002) shown that this combination performs better than plain undersampling. A more sophisticated combination, proposed by Batista et al. (2003), combines SMOTE with Tomek Links. It has been successfully applied on an imbalanced genomics dataset.

(i) SMOTE and Random Undersampling As detailed in Section 2.3.1, SMOTE selects instances that are similar in the features space and synthesizes new instances in between. This technique increases the size of the minority class. A random deletion of instances from the majority class, in combination with this approach, helps to improve the data balancing and the class clusters separation. However, an obvious limitation with the random undersampling stage is that information-rich samples might be deleted from the majority class.

(ii) SMOTE and Tomek Links This combination has been proposed in Batista et al. (2003). It first uses SMOTE to oversample the minority class by creating synthetic samples. However, as class clusters are generally not well defined, synthetic minority class examples can invade the majority class leading to overfitting. Applying Tomek links undersampling procedure on the over-sampled dataset by removing the *cross-class pairs* finally produces a balanced dataset with well defined class clusters.

(iii) SMOTE and NCR For this technical survey, we also propose to combine SMOTE with NCR. Our experimental results (Section 2.6) show that these two sampling approaches tend to improve some machine learning techniques. NCR has a positive effect on non ensemble approaches. SMOTE preferentially improves LR. By combining SMOTE and NCR, we expect an improvement of several machine learning techniques compared in this survey.

Our experiments demonstrate that churn prediction performance is only slightly impacted by sampling strategies. More precisely, when considering a particular machine learning approach for a given dataset, a significant improvement can be *locally* observed. However, a global improvement of all the machine learning approaches cannot be observed. Rebalancing performance strongly depends on the *couple* ML & sampling approaches, as well as on the dataset (see results in Section 2.6.2).

2.4 Machine learning techniques

We detail in this section the most widespread data mining techniques that have been proposed to tackle the customer churn prediction task. In the following, we mainly focus on *base* machine learning approaches that do not embed any weight correction for the imbalance nature of churn datasets. For our experiments, we rather choose to alleviate the class imbalance using sampling approaches. We invite the reader to refer to the literature which is abundant on the variants of machine learning methods in the context of imbalanced data (López et al., 2012 ; Haixiang et al., 2017 ; Zadrozny and Elkan, 2001 ; Domingos, 1999 ; Zadrozny et al., 2003). We also introduce several machine learning techniques which are suitable for strongly imbalanced data and usually applied in anomaly detection. Hence, Section 2.4 reports several supervised and semi-supervised learning algorithms and supervised ensemble methods. It also briefly covers some aspects of semi-supervised techniques.

2.4.1 Supervised learning

(I) K-NEAREST NEIGHBORS The ***k*-nearest neighbors** (*k*-NN) is a non parametric *memory-based* algorithm. It assigns to an instance $\mathbf{x}_i$ the label that corresponds to the majority label among its k closest training samples Ω_k. Formally,

$$p(C_i = g \mid \mathbf{x_i}) = \frac{1}{K} \sum_{j \in \Omega_k} \mathbb{1}\{\mathbf{x_j}\}$$

where the indicator function $\mathbb{1}$ is defined as being equal to one when $\mathbf{x}_i \in +$, zero otherwise. *k*-NN depends on two main parameters, namely *(i)* the number of neighbors k and *(ii)* a pairwise metric distance function. For continuous data, the following distance is commonly used $dist(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j||$ with $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$ ($||.||$ denotes the Frobenius norm). The simplicity and efficiency of *k*-NN have made this algorithm very attractive in the field of machine learning. Yet, it has several significant drawbacks when used on churn-like data, as shown in (Dubey and Pudi, 2013; Tan, 2005).

(II) NAIVE BAYES CLASSIFIER The **Gaussian Naive Bayes** (Gnb) classifier (John and Langley, 1995; Hand and Yu, 2001) is appropriate in a high feature space context, when the density estimation is difficult. The term *naive* results from a simplifying assumption that posits the conditional independence of the d features $\mathbf{x}^j$ given the class value k. This leads to

$$f_k(\mathbf{x}_i) = \prod_{j=1}^{d} f_{kj}(x_{ij}|k). \tag{2.1}$$

Note that from Eq. 2.1, we can formally write the Gnb classifier function as a *generalized additive model*. The Gnb classifier is simple, scalable and often outperforms more complex approaches. Although, it appears to be sensitive to the class imbalance issue (Chowdhury and Alspector, 2003; Rennie, 2001; Bermejo et al., 2011) - in particular due to the strong bias in the prior estimation -, good results can also be achieved for the churn prediction problem (Huang et al., 2012).

(III) LOGISTIC REGRESSION The **logistic regression** (LR) models the posterior probability of the classes via a linear function in $\mathbf{x}$. In a binary context, such as churn prediction, the posterior probability of the positive class simply amounts to,

$$P(C = +|\mathbf{x}) = \frac{\exp(\beta_{+0} + \beta_{+}\mathbf{x})}{1 + \exp(\beta_{+0} + \beta_{+}\mathbf{x})}$$

and sum to 1 with $P(C = -|\mathbf{x})$. This model is usually fitted by the maximization of the likelihood $L(\theta)$. The maximization can be made with the Newton-Raphson algorithm, which requires the second derivative of $L(\theta)$. Hence, fitting the LR model amounts to solve,

$$\frac{\partial L(\beta)}{\partial \beta} = \mathbf{X}^\top (Y - \mathbf{p}) \text{ and } \frac{\partial^2 L(\beta)}{\partial \beta \partial \beta^\top} = -\mathbf{X}^\top \mathbf{W} \mathbf{X}$$

where $\mathbf{p}$ is the vector of fitted probabilities, $p_i = P(C_i = +|\mathbf{x}_i)$, and $\mathbf{W}$ is a $n \times n$ diagonal matrix with $w_{ii} = p_i(1 - p_i)$. These equations can get solved repeatedly, following the IRLS algorithm (*iteratively reweighted least squares*) Burrus et al. (1994). In the context of unbalanced datasets, it has been shown that the bias of the regression vector intercept tends to be stronger with the unbalanced ratio (Owen, 2007; Salas-Eljatib et al., 2018). This issue can be overcome with a *prior* correction that takes into account the minority class or with a penalized likelihood where the maximum likelihood formula is weighted by the fraction of ones in the target variable (King and Zeng, 2001). The good performance of LR was previously pointed out in Burez and Van den Poel (2009).

(IV) SUPPORT VECTOR MACHINE The **Support Vector Machine** (SVM) was introduced by Vapnik (1998) as a kernel based machine learning model for classification and regression task. A recent survey is available in Cervantes et al. (2020). The SVM classifier aims to construct an optimal separating hyperplane between two linearly separable classes, and can be extended to the non-separable case. The hyperplane can be defined as,

$$\{\mathbf{x}_i | \sum_{j=1}^{d} x_{ij}\beta_j + \beta_0 = \mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0 = 0\}$$

where the coefficients β_j are defined up to a multiplicative factor. Thereby the SVM classification problem can be formally written as,

$$min_{\boldsymbol{\beta},\beta_0}||\boldsymbol{\beta}||^2 \text{ subject to } y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq 1, i \in \{1 \dots n\}.$$

In the case of overlapping classes, the SVM classifier can be optimized by allowing for some points to be on the wrong side of the margin, with a *cost* of $\xi = (\xi_1, \dots, \xi_n)$. Hence, bounding the $\sum_i \xi_i$ by a constant $\mathcal{C}$ leads to bounding the total number of misclassifications, and the standard SVM classifier problem can finally be expressed as,

$$min_{\boldsymbol{\beta},\beta_0}||\boldsymbol{\beta}||^2 \; subject \; to \begin{cases} y_i(\mathbf{x}_i^\top \boldsymbol{\beta} + \beta_0) \geq 1 - \xi_i \; \forall i \\ \xi_i \geq 0, \sum_i \xi_i \leq \mathcal{C}. \end{cases} \tag{2.2}$$

The SVM as described above, uncovers linear boundaries in the input feature space. Based on a quadratic programming solution using Lagrange multipliers, we can re-express the SVM classifier problem of Eq. 2.2 as the following Lagrangian dual objective function,

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,i'=1}^{n} \alpha_i \alpha_{i'} y_i y_{i'} \mathbf{x}_i^\top \mathbf{x}_{i'}. \tag{2.3}$$

We then maximize L_D subject to $0 \leq \alpha_i \leq \mathcal{C}$, $\sum_{i=1}^{n} \alpha_i y_i = 0$ and the Karush-Kuhn-Tucker conditions to find the solution for β.

Note that we can easily enlarge the feature space by using basis expansions h to identify nonlinear boundaries in the original space. This only requires the use of a kernel function, $K(\mathbf{x}, \mathbf{x}') = \langle h(\mathbf{x}), h(\mathbf{x}') \rangle$ at the inner product position of Eq. 2.3. Three widespread kernel functions are regularly encountered in the SVM literature, namely *Radial basis*, *Neural network* and d^{th}*-Degree polynomial* functions.
Since SVM only takes into account the *support vectors*, i.e. the points that are closed to the boundary, it is an interesting candidate for moderately imbalanced datasets (Akbani et al., 2004 ; Coussement and Van den Poel, 2008), although it performs poorly when the class distribution is too skewed (Tian et al., 2011).

(v) Decision Tree The **Decision Tree** (DT) method iteratively partitions the feature space into a set of *rectangles*, for which split-points achieve the best fit, until a stopping rule is reached. Within each partition, or *region* R_m, the target variable Y can be modeled as a constant c_m (Breiman et al., 1984 ; Friedman et al., 2001). A major advantage of tree-based methods is that the recursive binary partition is highly interpretable, and somehow mimics a logical human thinking. For classification purpose, the best split point s is obtained with an impurity measure Q_m that is based on the proportion $\hat{p}_{mk}$ of class k in the region R_m with N_m observations,

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k) \tag{2.4}$$

where I is an indicator function. Hence, at node m, observations are classified at the class $k(m)$ that maximizes the proportion in Eq. 2.4. Three impurity measures are usually encountered in DT classification, namely *Misclassification error*, *Gini index* and *Cross-entropy*, the two last measures being generally preferred as they are differentiable and more sensitive to changes in the node probabilities. In a binary classification problem, such as churn, the Gini index and the Cross-entropy measures simple amount to $2p(1-p)$ and $-p\log(p) - (1-p)\log(1-p)$ respectively, weighted by the number of observations in the obtained regions at split. In the context of imbalance datasets authors argue that decision trees are not viable (Weiss, 2004 ; Branco et al., 2016), while others propose an insensitive splitting strategies based, for instance, on the Hellinger distance (Yin et al., 2013 ; Branco et al., 2016).

(vi) Deep neural networks Deep neural techniques have led to state-of-the-art results in various application domains. While generally efficient on datasets with balanced class distribution, **deep neural networks** performance can be severely impede by imbalanced classes (Wang et al., 2016b ; Zhou et al., 2020). To overcome this issue, some authors focused on specific loss function (Wang et al., 2016b) or cost-sensitive learning (Zhou and Liu, 2005) on neural networks.

Recently, Munkhdalai et al. (2020) proposed an end-to-end deep neural network architecture using the Gumbel distribution as an activation function to tackle the class imbalance issue. Their proposal, so-called GEV-NN (Generalized Extreme Value distribution), outperforms the state-of-the-art baselines while giving a beneficial advantage to interpret variable importance. GEV-NN framework decomposes in three components : *(i)* a feed-forward weighting neural network which provides variable scores to adaptively control input variables (Munkhdalai et al., 2019), *(ii)* an auto-encoder to generate encoded representation and extract efficient features for the minority class (Zong et al., 2018) and *(iii)* a prediction network that receives a concatenation of scored input variables, encoded representation and features.

A key element of the GEV-NN approach is the Gumbel distribution which is used as an activation function (Cooray, 2010). Also known as Generalized Extreme Value distribution, it is widely used to model the distribution of extreme samples and has been extensively applied to characterize, for instance, age at death or risk assessment in financial context. Its cumulative distribution function is given by $F(x) = e^{-e^{-x}}$. The Gumbel function asymmetry naturally provides a different misclassification penalization on both classes.

Among the above mentioned approach, our experiments revealed that GEV-NN was on average, over all considered churn-like datasets, the best performing machine learning technique (without sampling pre-processing, GEV-NN being by design intended for imbalanced data). Interestingly, this sophisticated deep approach, usually dedicated to anomaly detection, is closely followed by the well-known logistic regression (see results in Section 2.6.2).

2.4.2 Ensemble Supervised Learning

Ensemble methods are meta-algorithms that combine several models into one predictive model in order to decrease variance (bagging) or bias (boosting).

(VII) Bagging and Random Forest **Bagging**, which stands for bootstrap aggregation, is an ensemble method for improving unstable estimation or classification schemes. In Breiman (1996) the author motivated bagging as a variance reduction technique for a given base classifier, such as decision tree. This approach stands out from basic ensemble algorithms by fitting a new model to a bootstrap resample of size less than n. As M models are trained, the final decision $\hat{f}_{bag}$ averages the M decision rules $\hat{f}_m(\mathbf{X})$ obtained from the boostrapped training sets. The **Random Forest** approach applies bagging to decision trees while sampling the variables (Breiman, 2001 ; Friedman et al., 2001). Specifically, the `DT` algorithm creates subpartitions by choosing a variable among the available features and splitting following an *impurity* criterion such as *Gini*. With `RF`, the choice of the variable is done within a random subset of features. This ensemble strategy produces more accurate predictions than `DT`. The easily interpretable decision rules are not available anymore, by contrast with `DT`, however `RF` can provide a measure of feature importance for the model accuracy. Previous studies highlighted the good performance of `RF` on imbalanced datasets (see for instance Chen et al. (2004)).

(VIII) eXtreme Gradient Boosting The **boosting** method is similar to bagging in that it combines the results of several classifiers, which are commonly decision trees. Yet, in the boosting strategy, each model tries to minimize the errors of the previous model, by contrast with bagging. The well-known variants of boosting are *Adaboost*, *gradient boosting* and **stochastic gradient boosting** which is the most general and widely used boosting technique. The key ingredient of *Adaboost* is the observation weights w_i, that are larger for misclassified instances. Hence, the approach forces the model $\hat{f}_m$ to train harder on the data for which it performs poorly and iteratively updates the weights. Each model seeks to minimize the weighted error e_m, which corresponds to the sum of the weights for the misclassified observations. Finally, the boosted estimate is given by $\hat{F} = \sum_{m=1}^{M} \alpha_i \hat{f}_i$ where the $\alpha_i = \frac{\log(1-e_m)}{e_m}$ ensure that the models with less errors have a larger weight in the final decision. Instead of adjusting weights, the *gradient boosting* variant optimize a cost function, while the *stochastic gradient boosting* strategy adds observations and variables sampling at each iteration. The most widely used implementation for boosting is `XGBoost`, a computationally efficient implementation of stochastic gradient boosting (Chen and Guestrin, 2016). It is interesting to note that with certain parameters setting, the boosting algorithm can emulate `RF`. When dealing with imbalanced dataset, `XGBoost` has been shown to outperform other types of methods Zhao et al. (2018). Yet, some studies are less optimistic and

suggest that `XGBoost` should be combined with other ensemble methods to achieve state-of-the-art performance (Ruisen et al., 2018).

When considered the ensemble machine learning techniques in our benchmark algorithms, our experiments revealed that random forest and XGBoost were on average, over all considered churn like datasets, the third and fourth best performing machine learning technique. When combined with a sampling pre-processing step, we found that a straightforward neural network performed better than these ensemble techniques (see results in Section 2.6.2).

2.4.3 Semi-supervised learning

Although very few churn prediction and analysis studies focus on semi-supervised techniques, we briefly address this type of approaches as they could be of great interest for future innovative developments in the field.

Semi-supervised techniques have been widely studied in the context of anomaly detection, an extreme case of churn prediction. These approaches combine unsupervised learning - which does not require labeled data - and supervised learning - which learns from labeled data. Semi-supervised techniques can be either generative, discriminative or a combination of both. Generative models attempt to model the joint probabilities of examples and their labels. Once this joint probability is modeled, one can generate new examples for a particular class, as well as determine the most likely class for a given example. Discriminative models restrict themselves to determining the most likely class for a given example by estimating the probability of each class given the data example. Discriminative models do not model the classes, so generation of new class examples is difficult. An example of semi-supervised learning in the context of churn for telecommunication area can be found in Benczúr et al. (2007). More recently, Xiao et al. (2018) propose to combine a semi-supervised approach with `Metacost`, a cost-sensitive model, in an ensemble strategy.

In the context of anomaly detection, One-Class Support Vector Machine (`ocSVM`) (Schölkopf et al., 1999) and Isolation Forest (iForest) (Liu et al., 2012) are among the most widely used semi-supervised anomaly detection algorithms. `ocSVM` identifies the smallest hypersphere containing the majority class datapoints (Tax and Duin, 1999). As for `SVM` (Section 2.4.1), `ocSVM` supports the introduction of a kernel function to allow for more flexibility. Although interesting, this approach does not perform well on large databases (Villa-Pérez et al., 2021). Indeed, `ocSVM` introduces significant memory requirements and is computationally expensive when the number of instances increases. By contrast, `iForest` (Liu et al., 2012) has a low linear time complexity and a small memory requirement. This approach posits that outlier datapoints can be isolated more easily than normal datapoints. `iForest` is based on a recursive 2D partitioning that can be represented by a tree structure (Section 2.4.1), so-called *Isolation Tree*. Anomalies or outliers correspond to leaf node with the smaller path length in the tree. This approach has been shown to perform well on imbalanced datasets in several studies (Villa-Pérez et al., 2021 ; Pang et al., 2019).

Recently, Pang et al. (2019) proposed a semi-supervised *deep anomaly detection* framework, so-called `DevNet`, which outperforms state-of-the-art methods. `DevNet` relies on neural deviation learning, requires few labeled anomalies and uses a prior probability that enforces statistically significant deviations of the anomaly scores. Specifically, `DevNet` decomposes as follows : *(i)* assigning an anomaly score to each training data object, *(ii)* providing a reference score based on the mean of the anomaly scores of normal data objects based on a prior probability and *(iii)* defining a loss function (*deviation loss*) to enforce statistically significant deviations of the anomaly scores as compared to normal data objects. A strength of `DevNet` framework is that it can naturally accommodate anomalies with different anomalous behaviors.

2.5 Model validation

2.5.1 Validation strategies

Model validation aims at estimating how effective is the model for the predictions of *unseen* instances. A straightforward validation principle is the *holdout set*, where some data subset that was not used for the training is used for evaluating the predictions of the trained model. We describe and discuss in the following subsections two validation approaches that build on and improve the *holdout set* idea.

Cross-validation A clear disadvantage of the *holdout set* strategy is that a portion of the data is *lost* for the model training. This especially becomes an issue when the dataset is small. The *cross-validation* addresses this issue by defining a training set and a validation set, and then switching the sets before combining the two validation scores.

K-fold validation The aforementioned cross-validation idea can be expanded to more subsets or *folds*, which is of great interest when data are scarce. The dataset is split in K subsets of equivalent sizes and the model is fitted on $K-1$ folds. The prediction error of the fitted model is then calculated on the k^{th} *unseen* subset. This strategy is repeated K times while taking another subset as validation set. Finally, the K estimates are combined. This is known as *K-fold cross-validation*. A typical value for K is 5 or 10 (Kohavi et al., 1995 ; Breiman and Spector, 1992 ; Burman, 1989). The K-fold cross validation is not appropriate as is for evaluating models on churn-like datasets which are typically imbalanced (He and Ma, 2013). Indeed, as the data is split into K-fold with a uniform probability distribution, it is likely that one or more folds will have few or no examples from the minority class, which in turn severely impedes the model training.

Stratified K-fold validation The dataset imbalance issue can be addressed with a *stratified* sampling, where the target variable y is used to control the sampling process. Hence, for a K-fold cross validation procedure, each fold will roughly contain the same distribution of class labels as the whole dataset.

The stratified K-fold validation is the validation strategy retained for our experiments, as it is the validation procedure that would be applicable in both balance and imbalance class contexts.

2.5.2 Evaluation metrics

The assessment procedure of a predictive model can rely on different metrics. Several metrics have been proposed in marketing and machine learning areas. We present in the following the most common metrics and emphasize their strengths and drawbacks when tackling churn-like data.

Metrics based on probability The **Top decile-lift** is one of the oldest evaluation metric among marketers to evaluate and compare predictive models. It is also a widespread measure in the churn literature (Burez and Van den Poel, 2009 ; Lemmens and Croux, 2006). The lift measure considers the observations/customers in order of their predicted probability of being churners. Specifically, when focusing on the 10% riskiest customers, the top decile-lift gives the ratio between the proportion of churners in the risky segment, $\pi_{10\%}$, and the whole proportion of churners in the validation set, π, $lift_{10\%} = \hat{\pi_{10\%}}/\hat{\pi}$. Hence, this measure evaluates if churners predicted as risky are actually at risk. The top decile-lift is directly related to the profitability or *gain* (Neslin et al., 2006) which is formally defined as,

$$GAIN = n\alpha\hat{\pi}(\Delta lift_{10\%})\left[\gamma LVC - \delta(\gamma - \psi)\right]$$

where n is the number of customers, α is the number of customers under study (here, 10%), $\Delta lift_{10\%}$ is the top decile-lift increase, γ is the success rate of the incentive among the churners, LVC is the lifetime value of a customer Gupta et al. (2004), δ is the incentive cost among customers and ψ is the success rate of the incentive among the non-churners.

Gini coefficient While the top decile-lift measure focuses on the 10% riskiest customer, the **Gini coefficient** takes also into account the less risky customers. This coefficient is formally defined as follows,

$$Gini = \frac{2}{M}\sum_{\ell=1}^{M}(\pi_\ell^c - \pi_\ell)$$

where M is the size of the validation set, π_ℓ^c is the fraction of actual churners above the threshold $\hat{f}(\mathbf{x}_i)$, π_ℓ the fraction of customers above the same threshold $\hat{f}(\mathbf{x}_i)$ and $\hat{f}(\mathbf{x}_i)$ corresponds to a predicted churn probability. In the same way as for the top decile-lift, the Gini coefficient takes advantage of the predicted churn probabilities. It is also a complementary measure as it considers the ability to predict less risky customers.

Metrics from confusion matrix Let TP be the *True Positive*, the number of customers predicted as churners who actually churned, and FP, *False Positive*, the number of customers predicted as churners who did not churn. Similarly, we can define TN, *True Negative*, the number of customers predicted as non churners who did not resign, and FN, *False Negative*, the number of customers predicted as non churner who actually churned. Hence, the number of correct predictions would be (TP+TN). By dividing with the total number of predictions (TP+TN+FP+FN), we obtain the *accuracy* that can summarize the classification performance of a model. However, using accuracy for churn predictive model evaluation is not appropriate as the data is strongly imbalanced (Weiss, 2004). We present below two metrics that are advisable in the churn context.

F_1 score This score summarizes the *Precision* and *Recall* metrics. The *Precision* estimates the ability of the model to obtain TP among its positive predictions, i.e. $Precision = \frac{TP}{TP+FP}$. It is a complementary measure to the *Recall*, that evaluates the ability of the model to recover TP, i.e. $Recall = \frac{TP}{TP+FN}$. The F_1 score proposes an harmonic mean of these two metrics, $F_1 = 2 \times \frac{Precision \cdot Recall}{Precision+Recall}$.

Area Under the Curve (AUC) The AUC measure first requires to express the performance of the model with a *Receiver Operating Characteristic* (ROC) curve. This curve gives the True Positive Rate ($TPR = \frac{TP}{TP+FN}$) as a function of the False Positive Rate ($FPR = \frac{FP}{FP+TN}$) for a series of decision thresholds. The AUC corresponds to the *Area Under the Curve*. Hence, it provides an aggregated performance measure for all possible ranking thresholds. This measure can be interpreted as the probability that the model correctly classifies an instance as positive as compared to a negative instance.

The Area Under the Curve (AUC) metric is the metric strategy retained for our experiments evaluations, as it is the metric that is the most advisable in imbalance class contexts.

2.6 Experiments

This section presents the churn prediction evaluations for several variants of our pipeline (Fig. 2.1). The retained datasets cover a range of domains where churn is regarded as a core issue (Table 2.1). We first summarize the experiments settings and necessary preprocessing steps. We then detail the machine learning performance on these datasets when associated to a sampling approach or not.

2.6.1 Experimental settings

We consider nine popular supervised algorithms - namely K-Nearest Neighbors (k-`NN`), Gaussian Naive Bayes (`Gnb`), Logistic Regression (`LR`), Support Vector Machine with Radial Basis Function kernel (`SVM-rbf`) and without kernel (`SVM`)[†], Decision Tree (`DT`), Random Forest (`RF`), `XGBoost`, a feed-forward neural network (`NN`) and `GEV-NN` - in association with different undersampling, oversampling and hybrid sampling strategies. Two semi-supervised techniques are also considered, namely `iForest` and `DevNet`[‡]. All the implementations are freely available from python packages. We mainly kept default parameters (Appendix A.1.2).

In this comparative study, we focus on the association between several machine learning techniques, sampling strategies and datasets in a churn prediction context. Hence, we do not resort to hyperparameters tuning. We adjusted the sampling so as to obtain a balance distribution as suggested by the AUC results presented in Weiss and Provost (2003), where the authors show that the best class distribution for learning tends to be near the balanced class distribution. Our evaluations follow a stratified K-fold cross validation procedure where $K = 5$ ($K \in [5, 10]$ is typically advised in the literature ; see for instance Kohavi et al. (1995) ; Breiman and Spector (1992) ; Burman (1989)).

Several preprocessing steps where performed on all datasets. First, we exclude features that take a unique value for each observation (e.g. customer ID, phone number, address). Besides, only observations with less than 20% missing feature values are retained. All numeric variables are standardized.

[†]In our experiments, we consider both the linear `SVM` and the `SVM-rbf`, which is a kernel `SVM` using the *Radial basis* function, following Amnueypornsakul et al. (2015)

[‡]`GEV-NN`, `iForest` and `DevNet` being specifically designed for imbalance binary classification or anomaly detection, these approaches are only evaluated without sampling.

The missing values are replaced by the feature mean for numeric variables and the majority category for a categorical variable (see Appendix A.1.1 for details).

2.6.2 Experimental results

We evaluate the churn prediction for all the pipeline alternative as given in Figure 2.1. The evaluation procedure follows a stratified 5-fold cross-validation. Results are given in AUC without sampling (Table 2.2), and with various oversampling (Table 2.3), undersampling (Table 2.4) and hybrid sampling approaches (Tables 2.5 & 2.6). The mean rank and the median AUC ($\widetilde{AUC}$) for each algorithm are given in the last two rows of each table.

Table 2.2 : AUC Classification results *(No Sampling approach).*

Dataset	k-NN	Gnb	LR	SVM	SVM-rbf	DT	RF	XGBoost	NN	GEV-NN	iForest	DevNet
Fraud	0.8990	0.9217	**0.9766**	0.9465	0.9441	0.8660	0.9466	0.9456	0.9573	0.9707	0.9459	0.9621
K2009	0.5004	0.5002	**0.5135**	0.5052	0.4989	0.4993	0.5114	0.5112	0.4999	0.5058	0.4975	0.4997
Thyroid	0.7598	0.5876	0.8645	0.9821	0.9786	0.9834	**0.9996**	0.9994	0.6223	0.9941	0.7551	0.7924
KKBox	0.5835	0.6468	0.6763	0.5022	0.4983	0.5302	0.6442	0.6800	0.6994	**0.7054**	0.5757	0.6184
UCI	0.7731	0.8477	0.8244	0.5963	0.7528	0.8447	**0.9182**	0.9174	0.8033	0.9137	0.6711	0.8139
Campaign	0.7596	0.8271	0.9331	0.5971	0.6451	0.7290	**0.9395**	0.9322	0.9134	0.9362	0.7338	0.7687
HR	0.6575	0.7442	**0.8596**	0.8091	0.4984	0.6053	0.7867	0.7993	0.6310	0.8558	0.6243	0.7677
TelE	0.8226	0.7505	0.7584	0.5335	0.6098	0.8514	0.9380	**0.9411**	0.8924	0.9320	0.5883	0.6769
News	0.7484	0.5655	0.8369	0.5958	0.6227	0.6754	**0.8615**	0.8323	0.8266	0.8525	0.5364	0.7003
Bank	0.7768	0.7166	0.8322	0.6645	0.7248	0.6908	0.8506	0.8216	0.8295	**0.8583**	0.6969	0.7686
Mobile	0.7567	0.7201	0.9030	0.4605	0.5463	0.6660	0.8095	0.7816	**0.9118**	0.8916	0.7963	0.8576
TelC	0.7822	0.8245	**0.8458**	0.6498	0.6548	0.6555	0.8210	0.7983	0.8357	0.8404	0.4542	0.7897
C2C	0.4387	0.5181	**0.5222**	0.4578	0.4656	0.4440	0.3518	0.3862	0.4541	0.3698	0.4985	0.4878
Member	0.5827	0.5914	0.6146	0.4874	0.5088	0.5462	0.6130	0.5987	0.6084	0.6243	0.5606	**0.6283**
SATO	0.6900	0.7272	0.7594	0.7116	0.7153	0.6365	**0.7882**	0.7396	0.7367	0.7600	0.6321	0.7030
DSN	0.6576	0.6671	0.7319	0.6868	0.6293	0.7350	**0.8590**	0.8516	0.6537	0.7493	0.6282	0.6941
$\widetilde{AUC}$	0.7526	0.7184	0.8283	0.5967	0.6260	0.6707	0.8358	0.8104	0.7700	**0.8542**	0.6262	0.7353
$\overline{Rank}$	8.06	7.19	3.19	9.00	9.56	8.62	3.38	4.44	5.69	**2.88**	9.69	6.31

The median AUC ($\widetilde{AUC}$) given in Tables 2.2 to 2.6 indicates only small $\widetilde{AUC}$ variations over sampling strategies. We can notice that the sampling methods generally degrade $\widetilde{AUC}$ for `RF` as compared to results obtained without sampling (from $\widetilde{AUC} = 0.8358$ to $\widetilde{AUC} = 0.8020$). Only SMOTE combined with NCR strongly increases `RF` $\widetilde{AUC}$ (0.8404). On average, `XGBoost` performance is slightly improved when using NCR and SMOTE combined with NCR (+0.0188 and +0.0186).

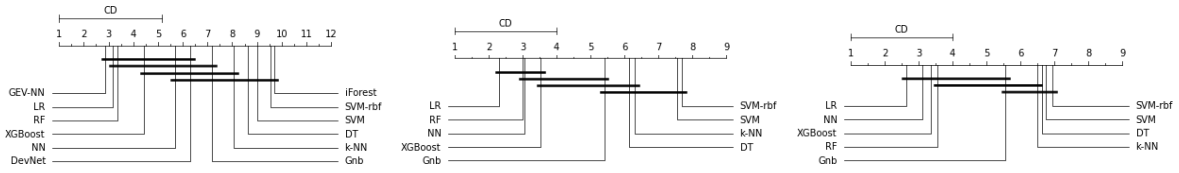
The approach that benefits the most from the sampling strategies is `NN`, with a maximum $\widetilde{AUC}$ increase of 0.0728 with SMOTE + Tomek Links. The top approaches over all datasets and sampling strategies are `LR`, `RF`, `XGBoost` and `NN`, with a mean rank of 2.61, 3.21, 3.33 and 3.66 respectively. When considering particular methods and datasets, greater improvement can be observed. For instance, combining `SVM` with NCR increases AUC of 0.1081 on *C2C*. The performance of `XGBoost` is also increased when using the hybrid sampling SMOTE & Tomek Links (from 0.8516 to 0.8694) on *DSN*. We notice an AUC increase of 0.0124 when using SMOTE in combination with NCR on *Member* with `LR`. While a global improvement of *all* the machine learning approaches cannot be observed, *local* improvements can be observed for given methods and samplings, depending on the datasets.

It is important to highlight the almost systematic complementary behaviors of `LR`, `RF`, `XGBoost` and `NN` overall datasets. As can be seen from Table 2.3 to Table 2.6, whenever `LR` is not the best approach, `XGBoost`, `RF` or `NN` outperforms the other machine learning techniques, and conversely (see for instance bold values of Table 2.4, Tomek Links or Tables 2.5, SMOTE & Random Undersampling). This finding suggests the use of an *ensemble* method based on the top four approaches, `LR`, `XGBoost`, `RF` and `NN` (see Chapter 3).

Table 2.3 : Oversampling methods : AUC Classification results *(top, SMOTE; bottom, ADASYN).*

SMOTE	k-NN	Gnb	LR	SVM	SVM-rbf	DT	RF	XGBoost	NN	Max-Min
Fraud	0.9054	0.9238	**0.9751**	0.7062	0.3136	0.8408	0.9693	0.9462	0.9648	0.6615
K2009	0.5001	0.4991	**0.5135**	0.4965	0.4993	0.5022	0.5023	0.4991	0.5054	0.0170
Thyroid	0.8006	0.5644	0.9039	0.8394	0.7128	0.9846	**0.9995**	0.9992	0.8624	0.4351
KKBox	0.5918	0.6430	0.6763	0.5590	0.4370	0.5272	0.6129	0.6414	**0.6851**	0.2481
UCI	0.7871	0.8273	0.8278	0.5327	0.7729	0.8490	0.9130	**0.9154**	0.8701	0.3827
Campaign	0.7657	0.7712	0.9311	0.6063	0.5761	0.7521	**0.9406**	0.9318	0.9258	0.3645
HR	0.6631	0.7168	**0.8501**	0.7066	0.5040	0.6309	0.7304	0.7905	0.7412	0.3461
TelE	0.8277	0.7497	0.7626	0.5470	0.5692	0.8482	0.9373	**0.9421**	0.9094	0.3951
News	0.7452	0.5664	0.8336	0.5651	0.6337	0.6881	0.8136	0.8333	**0.8428**	0.2777
Bank	0.7744	0.7861	0.8325	0.5830	0.7204	0.6940	0.8255	0.8234	**0.8422**	0.2592
Mobile	0.6479	0.6993	0.8942	0.6185	0.4404	0.6570	0.8138	0.7835	**0.9124**	0.4720
TelC	0.7650	0.8224	**0.8451**	0.5098	0.6881	0.6656	0.8007	0.7941	0.8439	0.3353
C2C	0.4375	0.5033	**0.5160**	0.4965	0.4751	0.4415	0.3944	0.3878	0.4348	0.1282
Member	0.5865	0.5936	**0.6213**	0.5176	0.5187	0.5489	0.6122	0.5959	0.6203	0.1037
SATO	0.6900	0.7272	0.7594	0.7116	0.7152	0.6385	**0.7601**	0.7396	0.7393	0.1216
DSN	0.6576	0.6671	0.7319	0.6868	0.6298	0.7314	0.8166	**0.8516**	0.6584	0.2218
$\widetilde{AUC}$	0.7176	0.7081	0.8302	0.5740	0.5726	0.6768	0.8137	0.8088	**0.8425**	
$\overline{Rank}$	6.31	5.38	**2.31**	7.56	7.69	6.12	3.00	3.56	3.06	
ADASYN	k-NN	Gnb	LR	SVM	SVM-rbf	DT	RF	XGBoost	NN	Max-Min
Fraud	0.8990	0.9217	**0.9766**	0.9466	0.9428	0.8621	0.9514	0.9456	0.9635	0.1145
K2009	0.5007	0.4987	**0.5137**	0.5032	0.5053	0.4985	0.4945	0.5013	0.5013	0.0192
Thyroid	0.7598	0.5876	0.8645	0.9821	0.9786	0.9806	**0.9995**	0.9994	0.6381	0.4119
KKBox	0.5899	0.6421	0.6777	0.5491	0.5239	0.5268	0.6107	0.6468	**0.6923**	0.1684
UCI	0.7791	0.8293	0.8276	0.5512	0.7601	0.8483	0.9112	**0.9156**	0.8712	0.3644
Campaign	0.7596	0.8271	0.9331	0.5971	0.6505	0.7269	**0.9398**	0.9322	0.9156	0.3427
HR	0.6612	0.7241	**0.8476**	0.6768	0.5026	0.5814	0.7597	0.7978	0.7566	0.3450
TelE	0.8248	0.7551	0.7634	0.4678	0.5559	0.8382	0.9364	**0.9418**	0.9097	0.4740
News	0.7377	0.5661	0.8309	0.5467	0.6419	0.6876	0.8107	0.8328	**0.8384**	0.2917
Bank	0.7647	0.7865	0.8315	0.6403	0.7123	0.6865	0.8197	0.8225	**0.8408**	0.2005
Mobile	0.6203	0.6814	0.8848	0.1398	0.4864	0.6644	0.7970	0.7937	**0.9100**	0.7702
TelC	0.7515	0.8311	**0.8444**	0.4093	0.6822	0.6546	0.8003	0.7968	0.8429	0.4351
C2C	0.4408	0.5031	0.5171	**0.5271**	0.4734	0.4401	0.3971	0.3905	0.4606	0.1366
Member	0.5791	0.5958	**0.6266**	0.5015	0.5304	0.5479	0.6092	0.5973	0.6153	0.1251
SATO	0.6900	0.7272	0.7594	0.7116	0.7153	0.6375	0.7494	0.7396	**0.7613**	0.1238
DSN	0.6576	0.6671	0.7319	0.6869	0.6297	0.7336	0.8038	**0.8516**	0.6602	0.2219
$\widetilde{AUC}$	0.7138	0.7028	**0.8292**	0.5502	0.6358	0.6754	0.8020	0.8101	0.7998	
$\overline{Rank}$	6.50	5.56	**2.62**	6.75	6.94	6.62	3.56	3.31	3.12	

We propose to visualize the machine learning performance similarities and ranking with Critical Difference (CD) diagrams (Demšar, 2006) based on statistical pairwise comparisons computed from the AUC results (Table 2.2 to Table 2.6). For these comparisons, we consider the post-hoc Nemenyi test ($\alpha = 0.05$) for which Figures 2.5, 2.6 and 2.7 provide the CD diagrams (Demšar, 2006) for each sampling strategy. Horizontal lines connect the approaches for which we cannot exclude the hypothesis that the average AUC rank is equal. As can be seen, the sampling strategies have a weak effect on the machine learning approaches ranking.

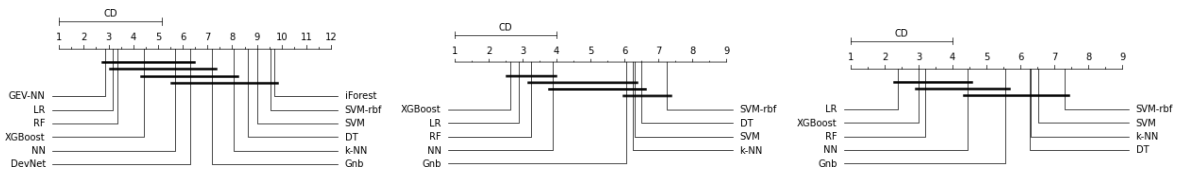


(a) No sampling (with AD methods) **(b)** SMOTE **(c)** ADASYN

Figure 2.5 : Approaches similarities based on Critical Difference diagrams *(Oversampling)*

Table 2.4 : Undersampling methods : AUC Classification results (top, NCR ; bottom, Tomek).

NCR	k-NN	Gnb	LR	SVM	SVM-rbf	DT	RF	XGBoost	NN	Max-Min
Fraud	0.9000	0.9226	**0.9762**	0.9472	0.9423	0.8803	0.9496	0.9405	0.9664	0.0959
K2009	0.5061	0.5004	0.5146	0.5017	0.5033	0.5027	0.5105	**0.5149**	0.5065	0.0145
Thyroid	0.7650	0.5887	0.8574	0.9726	0.9548	0.9824	**0.9993**	0.9992	0.6557	0.4106
KKBox	0.6099	0.6483	0.6762	0.5353	0.4797	0.5488	0.6397	0.6824	**0.7002**	0.2205
UCI	0.8052	0.8512	0.8234	0.6309	0.6288	0.8500	0.9145	**0.9200**	0.8118	0.2912
Campaign	0.7789	0.8150	0.9287	0.6751	0.6828	0.7934	**0.9374**	0.9353	0.9017	0.2623
HR	0.6761	0.7350	**0.8580**	0.8332	0.4984	0.6194	0.7430	0.7918	0.6803	0.3596
TelE	0.8295	0.7468	0.7615	0.4438	0.6260	0.8583	0.9394	**0.9417**	0.8922	0.4979
News	0.7804	0.5672	0.8371	0.6727	0.6745	0.7306	0.8298	**0.8399**	0.8189	0.2727
Bank	0.7994	0.7460	0.8313	0.6647	0.7938	0.7327	0.8361	**0.8369**	0.8335	0.1722
Mobile	0.7274	0.7255	0.8867	0.4912	0.6077	0.6710	0.7862	0.7745	**0.8883**	0.3971
TelC	0.8028	0.8205	**0.8438**	0.8007	0.7920	0.7136	0.8201	0.8216	0.8380	0.1302
C2C	0.4069	0.4890	0.4985	**0.5659**	0.4533	0.4146	0.3527	0.3668	0.4360	0.2132
Member	0.5915	0.5886	0.6209	0.4915	0.5512	0.5693	0.6129	0.6104	**0.6218**	0.1303
SATO	0.7028	0.7348	0.7645	**0.7741**	0.7089	0.6615	0.7631	0.7685	0.7198	0.1126
DSN	0.6634	0.6328	0.7311	0.7186	0.6308	0.7214	0.8173	**0.8672**	0.6952	0.2364
$\widetilde{AUC}$	0.7462	0.7302	0.8274	0.6687	0.6298	0.7175	0.8187	**0.8292**	0.7658	
$\overline{Rank}$	6.25	6.06	2.88	6.31	7.25	6.50	3.25	**2.62**	3.88	
Tomek	k-NN	Gnb	LR	SVM	SVM-rbf	DT	RF	XGBoost	NN	Max-Min
Fraud	0.8990	0.9217	**0.9766**	0.9457	0.9445	0.8793	0.9477	0.9446	0.9629	0.0973
K2009	0.4999	0.5002	**0.5138**	0.5007	0.4961	0.5044	0.5106	0.5017	0.4944	0.0194
Thyroid	0.7607	0.5879	0.8638	0.9825	0.9769	0.9825	**0.9996**	0.9994	0.6779	0.4117
KKBox	0.5873	0.6470	0.6761	0.5335	0.4762	0.5337	0.6189	0.6805	**0.6994**	0.2232
UCI	0.7773	0.8487	0.8252	0.6336	0.7540	0.8431	0.9134	**0.9150**	0.8241	0.2814
Campaign	0.7628	0.8252	0.9324	0.5985	0.6502	0.7449	**0.9391**	0.9341	0.9141	0.3406
HR	0.6671	0.7426	**0.8585**	0.8260	0.4990	0.6152	0.7481	0.7997	0.6281	0.3595
TelE	0.8236	0.7501	0.7589	0.5695	0.6031	0.8543	0.9379	**0.9412**	0.8906	0.3717
News	0.7533	0.5653	**0.8376**	0.6010	0.6395	0.6909	0.8132	0.8365	0.8263	0.2723
Bank	0.7797	0.7196	**0.8321**	0.5793	0.7500	0.6963	0.8243	0.8253	0.8314	0.2528
Mobile	0.7514	0.7182	0.8991	0.3813	0.5211	0.6619	0.7880	0.7868	**0.9061**	0.5248
TelC	0.7882	0.8240	**0.8459**	0.7019	0.7055	0.6683	0.8001	0.8017	0.8375	0.1776
C2C	0.4359	0.5164	**0.5208**	0.4803	0.4567	0.4427	0.3863	0.3855	0.4488	0.1353
Member	0.5890	0.5924	**0.6170**	0.4801	0.5162	0.5474	0.6036	0.6033	0.5960	0.1369
SATO	0.6891	0.7247	**0.7573**	0.7253	0.7029	0.6415	0.7483	0.7514	0.7034	0.1158
DSN	0.6535	0.6632	0.7286	0.7000	0.6241	0.7293	0.8294	**0.8655**	0.6518	0.2414
$\widetilde{AUC}$	0.7524	0.7189	**0.8286**	0.5998	0.6318	0.6796	0.8067	0.8135	0.7638	
$\overline{Rank}$	6.31	5.56	**2.38**	6.50	7.31	6.31	3.19	3.00	4.44	

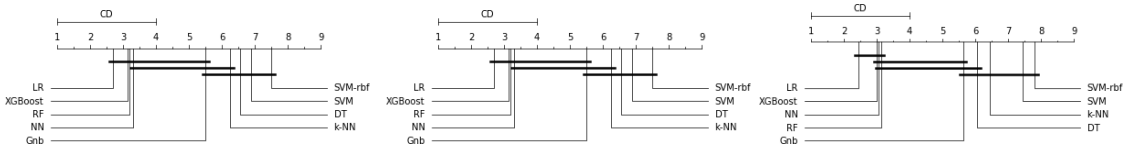


(a) No sampling **(b)** Neighborhood Cleaning rule **(c)** Tomek

Figure 2.6 : Approaches similarities based on Critical Difference diagrams *(Undersampling)*

Table 2.5 : Hybrid methods : AUC Classification results

	k-NN	Gnb	LR	SVM	SVM-rbf	DT	RF	XGBoost	NN	Max-Min
Dataset	SMOTE + Random undersampling									
Fraud	0.9054	0.9238	**0.9751**	0.7758	0.3237	0.8357	0.9694	0.9462	0.9746	0.6514
K2009	0.5001	0.4991	**0.5135**	0.4967	0.5012	0.5023	0.5055	0.4991	0.5038	0.0168
Thyroid	0.8006	0.5644	0.9039	0.8394	0.7224	0.9835	**0.9995**	0.9992	0.8548	0.4351
KKBox	0.5918	0.6430	0.6763	0.5654	0.4628	0.5277	0.6199	0.6480	**0.6997**	0.2369
UCI	0.7871	0.8273	0.8278	0.5326	0.7727	0.8499	**0.9168**	0.9154	0.8715	0.3842
Campaign	0.7657	0.7712	0.9311	0.6063	0.5761	0.7500	**0.9403**	0.9318	0.9279	0.3642
HR	0.6631	0.7168	**0.8501**	0.7065	0.5031	0.6295	0.7560	0.7905	0.7601	0.3470
TelE	0.8275	0.7497	0.7626	0.5756	0.5677	0.8486	0.9373	**0.9421**	0.9084	0.3744
News	0.7454	0.5664	0.8337	0.5652	0.6337	0.6871	0.8117	0.8333	**0.8415**	0.2763
Bank	0.7744	0.7861	0.8325	0.5830	0.7204	0.6936	0.8240	0.8234	**0.8430**	0.2600
Mobile	0.6586	0.6993	0.8942	0.5304	0.5588	0.6586	0.7953	0.7835	**0.9080**	0.3776
TelC	0.7650	0.8224	**0.8451**	0.5785	0.6881	0.6675	0.7947	0.7941	0.8419	0.2666
C2C	0.4375	0.5033	**0.5160**	0.5097	0.4783	0.4429	0.3964	0.3878	0.4557	0.1282
Member	0.5866	0.5936	0.6213	0.5179	0.5169	0.5426	0.5985	0.5959	**0.6235**	0.1066
SATO	0.6900	0.7272	**0.7594**	0.7117	0.7152	0.6375	0.7491	0.7396	0.7405	0.1219
DSN	0.6576	0.6671	0.7319	0.6868	0.6293	0.7343	0.8156	**0.8516**	0.6677	0.2223
$\widetilde{AUC}$	0.7177	0.7081	0.8302	0.5771	0.5719	0.6773	0.8035	0.8088	**0.8417**	
$\overline{Rank}$	6.38	5.56	**2.38**	7.44	7.69	6.19	3.00	3.62	<u>2.75</u>	



(a) SMOTE + Random undersampling **(b)** SMOTE + NCR **(c)** SMOTE + Tomek

Figure 2.7 : Approaches similarities based on Critical Difference diagrams *(Hybrid sampling)*

Table 2.6 : Hybrid methods : *AUC* Classification results *(top, SMOTE-Tomek ; bottom, SMOTE-NCR)*

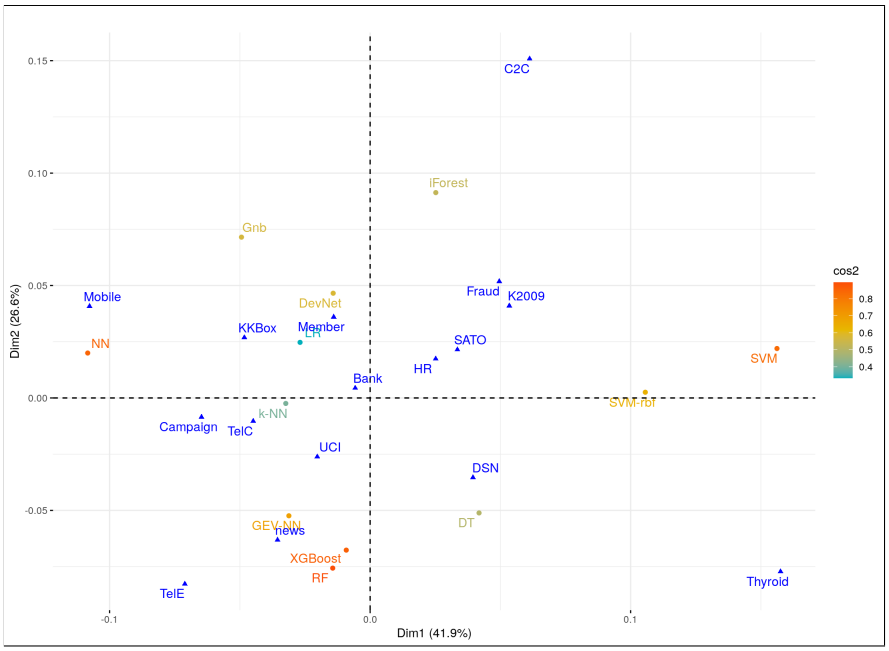
ST-T.L.	k-NN	Gnb	LR	SVM	SVM-rbf	DT	RF	XGBoost	NN	Max-Min
Fraud	0.9054	0.9238	**0.9751**	0.7893	0.3207	0.8377	0.9679	0.9462	0.9698	0.6544
K2009	0.5001	0.4991	**0.5135**	0.4999	0.4985	0.5050	0.5088	0.5084	0.5047	0.0150
Thyroid	0.8006	0.5656	0.9035	0.8683	0.7173	0.9826	**0.9995**	0.9993	0.8685	0.4339
KKBox	0.5926	0.6432	0.6764	0.5098	0.4378	0.5291	0.6142	0.6494	**0.7017**	0.2639
UCI	0.7871	0.8273	0.8278	0.5685	0.7700	0.8457	**0.9189**	0.9150	0.8750	0.3504
Campaign	0.7633	0.7708	0.9304	0.5914	0.5887	0.7491	**0.9399**	0.9335	0.9294	0.3512
HR	0.6631	0.7168	**0.8501**	0.7065	0.5018	0.6298	0.7533	0.7905	0.7378	0.3483
TelE	0.8270	0.7496	0.7628	0.5042	0.5492	0.8482	0.9359	**0.9402**	0.9098	0.4360
News	0.7450	0.5690	0.8335	0.5414	0.6363	0.6882	0.8124	0.8273	**0.8435**	0.3021
Bank	0.7746	0.7860	0.8325	0.5952	0.7295	0.6958	0.8232	0.8273	**0.8420**	0.2468
Mobile	0.6351	0.6995	0.8941	0.2132	0.5761	0.6639	0.7951	0.7939	**0.9073**	0.6941
TelC	0.7708	0.8223	**0.8449**	0.5011	0.7051	0.6717	0.7980	0.7960	0.8447	0.3438
C2C	0.4370	0.5034	**0.5158**	0.4691	0.4705	0.4419	0.3894	0.3846	0.4574	0.1312
Member	0.5852	0.5925	0.6201	0.4627	0.5118	0.5470	0.6007	0.6029	**0.6206**	0.1579
SATO	0.6986	0.7219	0.7581	0.7438	0.7122	0.6375	0.7565	**0.7602**	0.7388	0.1227
DSN	0.6531	0.6644	0.7304	0.7125	0.6257	0.7314	0.8066	**0.8694**	0.6691	0.2437
$\widetilde{AUC}$	0.7218	0.7082	0.8302	0.5550	0.5824	0.6800	0.8023	0.8116	**0.8428**	
$\overline{Rank}$	6.44	5.62	**2.44**	7.44	7.81	6.06	3.12	3.00	3.06	

ST-NCR	k-NN	Gnb	LR	SVM	SVM-rbf	DT	RF	XGBoost	NN	Max-Min
Fraud	0.9054	0.9238	**0.9751**	0.8562	0.3237	0.8358	0.9681	0.9452	0.9642	0.6514
K2009	0.5003	0.4995	**0.5153**	0.4972	0.5044	0.4984	0.4944	0.4974	0.5063	0.0209
Thyroid	0.8004	0.5672	0.9032	0.8399	0.7201	0.9865	**0.9994**	0.9991	0.8587	0.4322
KKBox	0.6054	0.6485	0.6801	0.5243	0.4790	0.5479	0.6665	0.6705	**0.7004**	0.2214
UCI	0.7856	0.8341	0.8274	0.5683	0.7524	0.8537	0.9144	**0.9187**	0.8726	0.3504
Campaign	0.7536	0.7706	0.9284	0.6180	0.5952	0.7495	**0.9402**	0.9311	0.9223	0.3450
HR	0.6569	0.7080	**0.8274**	0.7500	0.4992	0.6620	0.7911	0.8031	0.7334	0.3282
TelE	0.8178	0.7465	0.7633	0.5954	0.5967	0.8524	0.9364	**0.9413**	0.9095	0.3459
News	0.7495	0.5936	0.8388	0.6342	0.7010	0.7323	**0.8537**	0.8477	0.8404	0.2601
Bank	0.7781	0.7827	0.8320	0.6542	0.7773	0.7232	**0.8495**	0.8423	0.8414	0.1953
Mobile	0.6260	0.6984	**0.8799**	0.6541	0.5329	0.5825	0.6210	0.6689	0.8747	0.3470
TelC	0.7754	0.8176	**0.8435**	0.6038	0.7778	0.7139	0.8312	0.8156	0.8425	0.2397
C2C	0.4225	0.4963	**0.5022**	0.4692	0.4468	0.4101	0.3153	0.3638	0.4563	0.1869
Member	0.5860	0.5791	0.6270	0.4485	0.5654	0.5590	0.6218	0.6125	**0.6354**	0.1869
SATO	0.7053	0.7387	0.7575	0.7556	0.7138	0.6850	**0.7811**	0.7671	0.7371	0.0961
DSN	0.6513	0.6515	0.7392	0.7334	0.6393	0.6986	0.8556	**0.8661**	0.6909	0.2268
$\widetilde{AUC}$	0.7274	0.7032	0.8274	0.6261	0.5960	0.7062	0.8404	0.8290	**0.8409**	
$\overline{Rank}$	6.25	5.50	**2.69**	6.88	7.50	6.56	3.19	3.12	3.31	

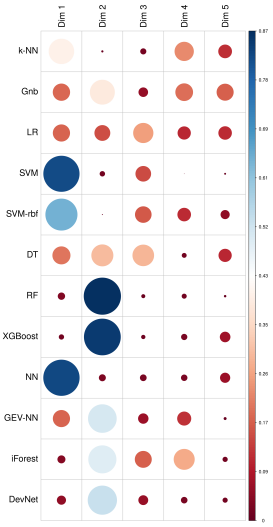
2.6.3 Models and datasets CA

To go beyond the analyses in Section 2.6.2, we propose to visualize the relationships between the machine learning techniques and the churn-like datasets in a two-dimensional plot based on the AUC results. To this end, we perform a Correspondence Analysis (`CA`) - a geometric approach that extends principal component analysis - on an AUC results table (Table 2.2). The Figure 2.8 provides a `CA` result overview that is useful for interpretation.

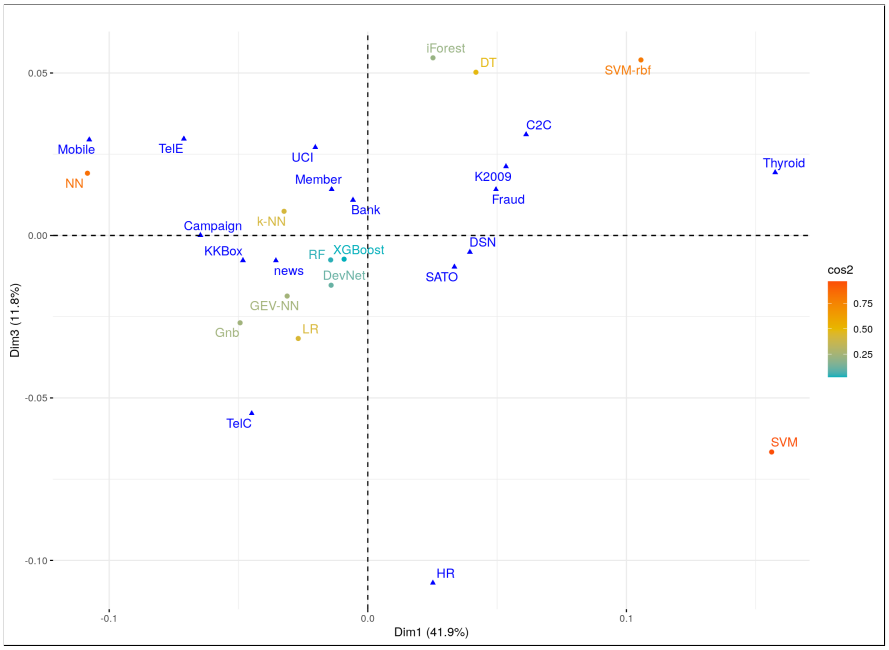
As can be seen from correlation plots in Figures 2.8(b) and 2.8(d), `SVM`, and `NN` are well represented by the first dimension, `RF` and `XGBoost` by the second dimension and `LR` by the third dimension. Similarly, not all datasets are well represented by the two first components and some of them are found on the third and the fourth dimensions. Hence, we provide in Figures 2.8(a) and 2.8(c) two CA biplots based either on the two first components, or on the first or third dimensions.



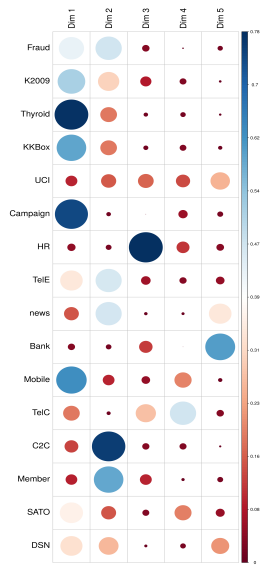
(a) *CA* biplot, dimensions 1 and 2, no sampling



(b) Representation Quality



(c) *CA* biplot, dimensions 1 and 3, no sampling



(d) Representation Quality

Figure 2.8 : (a & c) Visualization of associations between machine learning approaches and churn-like datasets without sampling using Correspondance Analysis. (b & d) Quality of representations on the factor map.

The Figure 2.8(a) suggests a similar behavior between `RF` and `XGBoost`. It also highlights the difference with these approaches and `SVM` and `SVM-rbf`. *News* appears associated with `RF`, `XGBoost` and `GEV-NN`, in agreement with the AUC Table 2.2. We also visualize the *Mobile* dataset in the vicinity of `NN` which is the most suitable technique without sampling. Similarly, *TelE* is found near `XGBoost`. The Figure 2.8(b) uses the third dimension instead of the second dimension, bringing a better representation of `LR`. We notice the positioning of *News* between `RF` and `GEV-NN`, as expected from the AUC table. Interestingly, *SATO* has shifted towards `RF`, `GEV-NN` and `LR`. This is in agreement with Table 2.2, as these machine learning techniques provide the best top three AUC results. Similarly, *KKBox* stands towards `LR` and `GEV-NN`.

2.7 Conclusion

In this Chapter, we included a background of the churn analysis research, an introduction to widespread data sampling and classifier approaches and a presentation of advisable evaluation metrics and strategies. First, we described publicly available churn-like datasets and provide links for an easy access. Then, we introduced data sampling approaches, which unfold in three categories, namely oversampling, undersampling and hybrid. We also detailed several machine learning classifiers encountered in the churn research field and discussed their reported success in the literature. The validation strategies and metrics are then discussed. Finally, machine learning approaches are combined and evaluated on sixteen publicly available churn-like datasets. We summarized our results in terms of AUC score.

The experimental investigations given in this Chapter also offer original analyses and visualizations. Ultimately, this part of my thesis project provides a general recommendation on a churn prediction pipeline based on an ensemble approach, as detailed in Chapter 3. It is interesting to highlight that the proposed visualizations easily emphasize behavioral relationships between classifiers, sampling methods and their association with churn-like benchmark datasets. In this comparative study, we only consider the default parameters for each approach. However, the supervised context would also allow for boosting versions of some of these techniques. This could significantly improve their classification results, in particular for `SVM` (Vafeiadis et al., 2015). The boosting strategy has been successfully applied to the prediction of customer churn in retail (Clemente et al., 2010) and telecom companies (Lemmens and Croux, 2006).

If the advantage of supervised learning is that all input labels are typically meaningful and serve as basis for an explainable discriminative classifier, the need for labels collection is however by itself a strong limitation. First of all, when the volume of the data is too large, it becomes prohibitively expensive to collect all labels. Furthermore, when distinctive labels are hard to find, it implies noise or uncertainties in the supervision which can lead to inaccurate results (Cabral and Oliveira, 2014 ; Taha and Hadi, 2019). In addition, in an imbalanced or strongly imbalanced classes distribution context, accessing high quality labels for the minority class is generally challenging. Indeed, the existence of different instance profiles within the positive class strongly impedes the training phase (Taha and Hadi, 2019).

Unsupervised or semi-supervised learning can be used to overcome these issues. While unsupervised learning requires no class label, semi-supervised learning only requires a small number of labeled samples. A key idea is to learn a model for the class associated with the normal behavior and then use this model to identify abnormal behaviors (Chandola et al., 2009). Hence, semi-supervised or unsupervised approaches can handle, during the test phase, abnormal behaviors that did not appear in the training dataset. This is a clear advantage as compared to supervised learning strategy.

3

Ensemble for churn prediction

In Chapter 2, we provided a large comparative study of customer churn prediction models. The strengths and drawbacks of each model appear to be strongly dependent on the latent dataset characteristics. To alleviate this issue, we propose in this Chapter an ensemble strategy that relies on the machine learning techniques identified as the most efficient in Chapter 2. Then, we enriched this ensemble approach with a *deep-clustering* that aims to uncover the underlying *customer's segmentation*.

Our ensemble approach embeds `LR`, `RF`, `XGBoost` and `NN` (Neural Network), as suggested from our experimental results. We jointly use their predictions for composing an average churn probability. Our empirical results, on several public datasets, provide a strong support for the benefits of our approach and illustrates the improvement of the supervised learning approach for binary classification (Section 3.1.2).

Churn behaviors are usually driven by multiple motivations, giving rise to a customers population with an heterogeneous set of profiles. Clients having a disappointing experience with the product or a service of a company, are usually prone to churn. Yet, the reasons underlying their disappointment might be diverse. Similarly, their can be multiple reasons behind customers loyalty. Furthermore, some customers might exhibit a churner prone behavior but still remain with the company. Hence, we might benefit from a customer segmentation in order to identify these heterogeneous profiles. To do so, we propose a deep-churn model that relies on Deep AutoEncoders (DAEs). Based on the obtained clusters, we propose to build one model for each subgroup of clients and combine them to make new predictions on unseen observations. Our results show that this proposal achieves good AUC score with respect to our vanilla ensemble while providing valuable information on the customers profile.

3.1 Controlling churn behaviors with an ensemble strategy

3.1.1 Motivation and Methods

Several traditional machine learning approaches evaluated in Chapter 2 have shown good results with their default settings on the churn prediction issue. Thorough evaluations could have been done through hyperparameters tuning. However, such tuning is not fully safe from the risk of overfitting. As we are motivating by finding an approach that could manage a wide range of churn-like datasets, we resort to rather investigate an *ensemble strategy*. Ensemble learning is an ML paradigm where multiple learners are trained to solve the same problem. This technique is not restricted to binary classification, researchers have extended its framework to regression and clustering (Zhou, 2012).

Ensemble learning experienced its golden age in the ninety's when some of the most popular papers of machine learning were released such as bagging, boosting and stacking methods. They usually rely on weak learners which are slightly better than random guess. Their predictions are averaged to improve the standalone prediction. Straightforward aggregation have been proposed such as *majority voting* or *weighted majority voting* (Sagi and Rokach, 2018). Ensemble strategies are nowadays regularly encountered in machine learning competitions, such as the Netflix prize (Bennett et al., 2007) or competitions on Kaggle (Martínez-Usó et al., 2015 ; Oza and Tumer, 2008).

One key component of the multiple classifier systems is the *diversity*. The individual classifiers must be different from one another to trigger performance improvement. However, the individual learners are usually trained for the same task, on the same training data, and are thus highly correlated. Besides, correctly quantifying the diversity is still under investigation. Previous studies relied on pairwise measures such as correlation coefficient (Sneath and Sokal, 1973), along with non-pairwise measures using for instance entropy (Cunningham and Carney, 2000). This proposal has been questioned by Shipp and Kuncheva (2002), who showcased the lack of correlation between those metrics and the classifiers. A more recent approach is based on information theoretic measures, and more precisely on the *interaction information* (Brown, 2009 ; Meynet and Thiran, 2010 ; Gupta and Bhavsar, 2021). These approaches have been recently extended to deep learning with the Ensembling Loss (EL) technique (Zaid et al., 2021).

Our contribution operates on what we have previously built : our benchmark of models. Providing the most performant models with respect to AUC, we propose to jointly learn them to perform a mean prediction in order to improve the outcomes of the churn forecast. We have selected `LR`, `RF`, `XGBoost` and `NN` as our top models.

3.1.2 Ensemble comparative experiments

In this Section, we combine `LR`, `XGBoost`, `RF` and `NN` for the churn prediction. Specifically, we average predicted probabilities for each instance, over two, three or four methods among `LR`, `XGBoost` `RF` and `NN`. The Figure 3.1 shows, for each sampling strategy, and over all datasets, the AUC for `LR`, `XGBoost`, `RF` and `NN` (light gray), their pairwise ensembles (light orange), the combination of three methods (dark orange) and the combination of all four methods (dark blue).

Figure 3.1 : AUC ensemble results on the three top machine learning approaches and all datasets

As can be seen from Figure 3.1, `LR|XGBoost|RF|NN` ensemble mostly outperforms the other methods, closely followed by `LR|XGBoost|RF` (Table 3.1). The best ensemble approach is obtained when combining the three approaches (`LR|XGBoost|RF`) and without sampling strategy (Table 3.1, $\widetilde{AUC} = 0.8577$).

Table 3.1 : $\widetilde{AUC}$ for *ensemble* and *non ensemble* approaches and all datasets.

Sampling	no	SMOTE	ADASYN	NCR	Tomek Links	SMOTE & R.U.	SMOTE & T.L.	SMOTE & NCR	$\widetilde{\widetilde{AUC}}$
LR	0.8283	0.8301	0.8293	0.8274	0.8287	0.8301	0.8302	0.8274	0.8294
XGBoost	0.8104	0.8087	0.8102	0.8292	0.8135	0.8087	0.8117	0.8290	0.8167
RF	0.8358	0.8137	0.8021	0.8187	0.8066	0.8035	0.8023	0.8403	0.8162
NN	0.7700	0.8425	0.7998	0.7658	0.7617	0.8417	0.8428	0.8409	0.8159
LR|XGBoost	0.8479	0.8464	0.8465	0.8457	0.8485	0.8464	0.8466	0.8395	0.8464
LR|RF	0.8516	0.8439	0.8460	0.8457	0.8476	0.8467	0.8466	0.8470	0.8472
LR|NN	0.8383	0.8442	0.8408	0.8378	0.8403	0.8446	0.8449	0.8424	0.8418
XGBoost|RF	0.8325	0.8256	0.8251	0.8374	0.8267	0.8240	0.8255	0.8405	0.8313
XGBoost|NN	0.8388	0.8461	0.8450	0.8412	0.8388	0.8484	0.8448	0.8352	0.8431
RF|NN	0.8533	0.8409	0.8365	0.8411	0.8358	0.8375	0.8395	0.8449	0.8423
LR|XGBoost|RF	**0.8577**	0.8526	0.8489	**0.8500**	0.8529	0.8521	0.8489	0.8466	0.8517
LR|XGBoost|NN	0.8498	0.8457	0.8477	0.8459	0.8452	0.8478	0.8465	0.8462	0.8473
LR|RF|NN	0.8523	0.8462	0.8484	0.8472	0.8491	0.8485	0.8470	0.8479	0.8483
XGBoost|NN|RF	0.8566	0.8512	0.8463	0.8486	0.8533	0.8510	0.8464	0.8464	0.8501
LR|XGBoost|RF|NN	0.8562	**0.8533**	**0.8506**	0.8491	**0.8546**	**0.8537**	**0.8492**	**0.8513**	**0.8526**

The Table 3.2 provides for each dataset, the pipeline that produces the highest AUC (*Best non ensemble pipeline AUC* & *Best non ensemble pipeline* columns). Our recommended ensemble pipeline (`LR|XGBoost|RF` and no sampling) provides an AUC that nearly reaches the best AUC result, for almost all datasets. The only exception is for *C2C*.

Table 3.2 : Our ensemble proposal vs. best non ensemble approach for each dataset.

	LR|XGBoost|RF & no sampling *AUC*	Best *non ensemble* pipeline AUC	Best *non ensemble* pipeline
Fraud	0.9794	0.9766	no sampling & LR
K2009	0.5197	0.5153	SMOTE-NCR & LR
Thyroid	0.9989	0.9996	no sampling & RF
KKBox	0.6890	0.7054	no sampling & GEV-NN
UCI	0.9215	0.9200	NCR & XGBoost
Campaign	0.9440	0.9402	SMOTE-NCR & RF
HR	0.8443	0.8596	no sampling & LR
TelE	0.9435	0.9421	SMOTE & XGBoost
News	0.8636	0.8615	no sampling & RF
Bank	0.8531	0.8583	no sampling & GEV-NN
Mobile	0.8761	0.9124	ADASYN & NN
TelC	0.8340	0.8459	Tomek Links & LR
C2C	0.3852	0.5659	NCR & SVM
Member	0.6201	0.6354	SMOTE-NCR & NN
SATO	0.7765	0.7882	no sampling & RF
DSN	0.8623	0.8694	SMOTE-T.Links & XGBoost
$\widetilde{AUC}$	0.8069	0.8240	

All in all, in practice, we recommend the use of the ensemble `LR|XGBoost|RF` with no sampling for analyzing novel churn-like datasets.

3.1.3 Discussion

Ensemble approach should be considered for the classification task in a churn-like context, as they repeatedly performed better than individual classifiers in the field of data mining. Ahmed et al. (2018b) even proposed nested ensemble learners models that outperform traditional ensemble when applied to churn prediction in telecom industry. The finance industry has gradually adapted various machine lear-

ning techniques. In particular, detecting economic crimes (eg., accounting fraud, money laundering) triggered successful applications of machine learning. `LR`, `Gnb` and `SVM` are among the most classic methods exploited in this area. The emergence of new kinds of fraud with the growth of electronic market has also popularized deep learning methods in finance. Ensemble strategies and boosting also remain a valuable option in this area. An enhanced hybrid ensemble approach, named *RS-MultiBoosting* Zhu et al. (2019) has been proposed; it incorporates *random subspace* and *MultiBoosting* to improve the accuracy of forecasting credit risk.

As already mentioned, the existence of small disjuncts within the minority class – corresponding in the churn context to the customer profile heterogeneity – can significantly impede the classifier performance. Hence, it would be advisable to segment the minority class upstream of, or during, the model training phase. The *Logit Leaf Model* proposed by De Caigny et al. (2018) (`LLM`) is a successful example of this strategy; it is an hybrid classification algorithm that combines `DT` and `RF` over a dataset whose partitioning is in agreement with the heterogeneity between customers. Hence, `LLM` is an ensemble approach that takes into account specific group characteristics that remained disregarded when a single classifier is trained over the whole dataset. In the following Section 3.2, we conduct investigations along this direction.

3.2 Augmenting churn prediction with customer profiling

3.2.1 Introduction

Management and marketing services are trying to cope with the ever-rising competition in industry by focusing their efforts on a strong Customer Relationship Management (CRM). In particular, customer retention has attracted interest as it clearly appeared that retained customers can be of great help for the company by spreading positive word of mouth (Reichheld and Sasser, 1990). Such behavior can subsequently lower the marketing costs of new customers acquisition (Bolton and Bronkhorst, 1995). Besides, it has become clearer that the acquisition costs of a new customer can be much more higher than the retention costs of an existing one (Reinartz and Kumar, 2003; Siber, 1997; Yang and Peterson, 2004). Hence, preventing customer *churn* or *attrition* can be vital for subscription-based service firms, that rely on fixed and regular membership fees, in numerous areas among which insurance (Günther et al., 2014), banking (Kumar et al., 2008), online gambling (Coussement and De Bock, 2013), online video games (Kawale et al., 2009), music streaming (Chen et al., 2018), online services (Tan et al., 2018) or telecommunication (Effendy et al., 2014; Abdillah et al., 2016; Hudaib et al., 2015; Hosein et al., 2021). Therefore, accurately predicting the customers who are prone to churn has become a priority in many industries.

Beyond the churn prediction, the study of the dynamic relationship between the customer satisfaction, the service quality and the customer behavior – loyalty or switching – is today a lively field of research. Indeed, a better understanding of customers experience offers valuable information for marketers. As an example, satisfied customers will be more tolerant to price increases which will in turn bring greater profits (Garvin, 1988). However, certain customer groups may have different perceptions of service providers (Gilmour et al., 1994). For instance, many studies propose to describe the customer satisfaction as a composite of factors such as the corporate image, the internal organization, the physical environment, the staff service and the customer-personal interaction (LeBlanc and Nguyen, 1988). In the Banking industry, Laroche et al. (1986) decompose the customer satisfaction into the speed service, the convenience of the location, the staff competence and the bank friendliness (Laroche et al., 1986). The amalgamation of the multiplicity and divergence of customer expectations and perceptions naturally calls for customer base segmentation to optimize churn behavior management.

While the negative effects of customer churn can be easily observed – lack of revenues or supplementary costs of attracting new customers –, the churn causes are under continuous study, as these causes generally vary across economical fields and customer groups. For service industries, Cronin Jr and Taylor (1992) relied on the effects of time, money constrains, lack of credible alternatives, switching costs, habit, price, convenience and availability to explain customers switching. Similarly, Keaveney (1995) identified eight main causal variables for churn, namely price, inconvenience, core service failures, service encounter failures, competitive issues, ethical problems and involuntary factors. Following on these proposals, Athanassopoulos (2000) proposed, based on Confirmatory Factor Analysis, five dimensions to describe different customer satisfaction profiles in retail banking services. These dimensions are staff service, business profile, innovativeness, convenience and price. The author also

validated the interest in dividing customers into segments market that correspond to their preferences regarding particular aspects of service. The motivation behind customers segmentation – which is one of the most significant methods used in marketing studies – is to select appropriate customers for a campaign. This typically increase customer profitability through adapted customer targeting (Tsai and Chiu, 2004; Vellido et al., 1999). In fact, a large amount of segmentation methods are developed each year (Kuo et al., 2006; Chan, 2008), making hard any exhaustive comparison between them.

In this project, one of our main motivations was to evaluate several machine learning techniques on the churn prediction task (Chapter 2. In order to take into account the multifactorial aspect of attrition, we also studied the performance of ensemble learning approaches to improve the attrition prediction (Section3.1. In this Section, we aim at taking *explicitly* into account the underlying customer segmentation. To this end, we rely on a deep unsupervised clustering method before exploiting an ensemble machine learning approach. Hence, our global objective *shifted* now (as compared to Figure 2.1), to evaluate the variants of the processing chain for churn analysis as given in Figure 3.2.

This chain includes (Figure 3.2),

(i) a class rebalancing step or a clustering step,
(ii) a supervised or a meta ensemble learning phase,
(iii) a robust evaluation procedure.

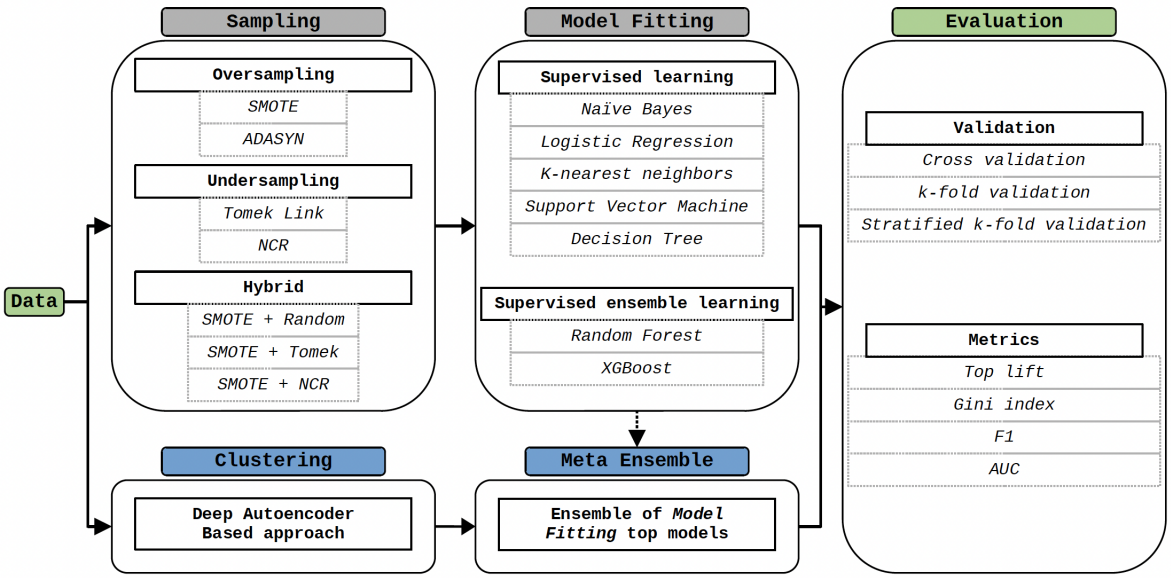


Figure 3.2 : Machine learning pipeline for churn prediction and analysis.

As all the variants of the algorithms in the proposed pipeline can not be exhaustively studied, we only consider the algorithms in their original version. Furthermore, the benchmark datasets for our experiments have a relatively important class imbalance between the minority class – unsubscribed individuals – and the majority class (Table 3.3). This decreases the performance of standard classifiers (García et al., 2012) which can be aggravated by an overlap of the classes or a fragmentation of the minority class into subsets corresponding to different customer profiles. As detailed in Section 3.1, this motivated the idea to combine the model fitting step with class rebalancing approaches. In this section, beyond the good performance obtained with our ensemble proposal, we also make the customer segmentation explicit via a deep autoencoder-based clustering. This clustering reveals the features associated to each underlying customer group.

In the following, we provide a brief overview of unsupervised machine learning techniques (Sections 3.2.4.a). We also enrich our ensemble proposal with a data segmentation that respect the underlying customer behavior patterns. The corresponding prediction results are given in Section 3.3.1 and are compared with the recent `LLM` (De Caigny et al., 2018) and `RF`-based (Ullah et al., 2019) models. We discuss the benefits of our approach in terms of churn prediction in 3.3.2 and customer profiling in Section 3.3.3.

3.2.2 Public datasets

This work relies on publicly available datasets only. A churn dataset usually comprises features of different types – numerical and categorical variables – that reflect customers behavior. It also generally exhibits a strong class imbalance, as the proportion of churners is typically lower than the proportion of customers that remain with the company.

Table 3.3 : Publicly available churn datasets with online access link

Dataset	#Instances	#Features	#Cont.Feat.	#Cat.Feat.*	mix.Score	$\frac{churn}{nonchurn}$
K2009	50,000	230	37	1,001	7.28×10^{-4}	0.08
KKbox	970,960	49	12	43	1.48×10^{-2}	0.10
UCI	5,000	20	0	20	2.66×10^{-1}	0.16
HR	1,470	34	14	71	1.03×10^{-1}	0.19
TelE	190,776	19	15	10	3.75×10^{-2}	0.19
News	15,855	18	2	304	1.56×10^{-1}	0.23
Bank	10,000	12	5	10	2.30×10^{-1}	0.25
Mobile	66,469	62	57	5	1.07×10^{-1}	0.27
TelC	7,043	20	3	30	3.69×10^{-1}	0.37
C2C	71,047	71	32	42	1.89×10^{-2}	0.41
Member	10,362	14	4	21	6.26×10^{-2}	0.43
SATO	2,000	13	9	19	1.72×10^{-1}	1
DSN	1,401	15	10	21	2.68×10^{-2}	1

(1) Categorical variables with more than two levels are converted to their numerical representation by *dummification* where each category becomes a binary variable.

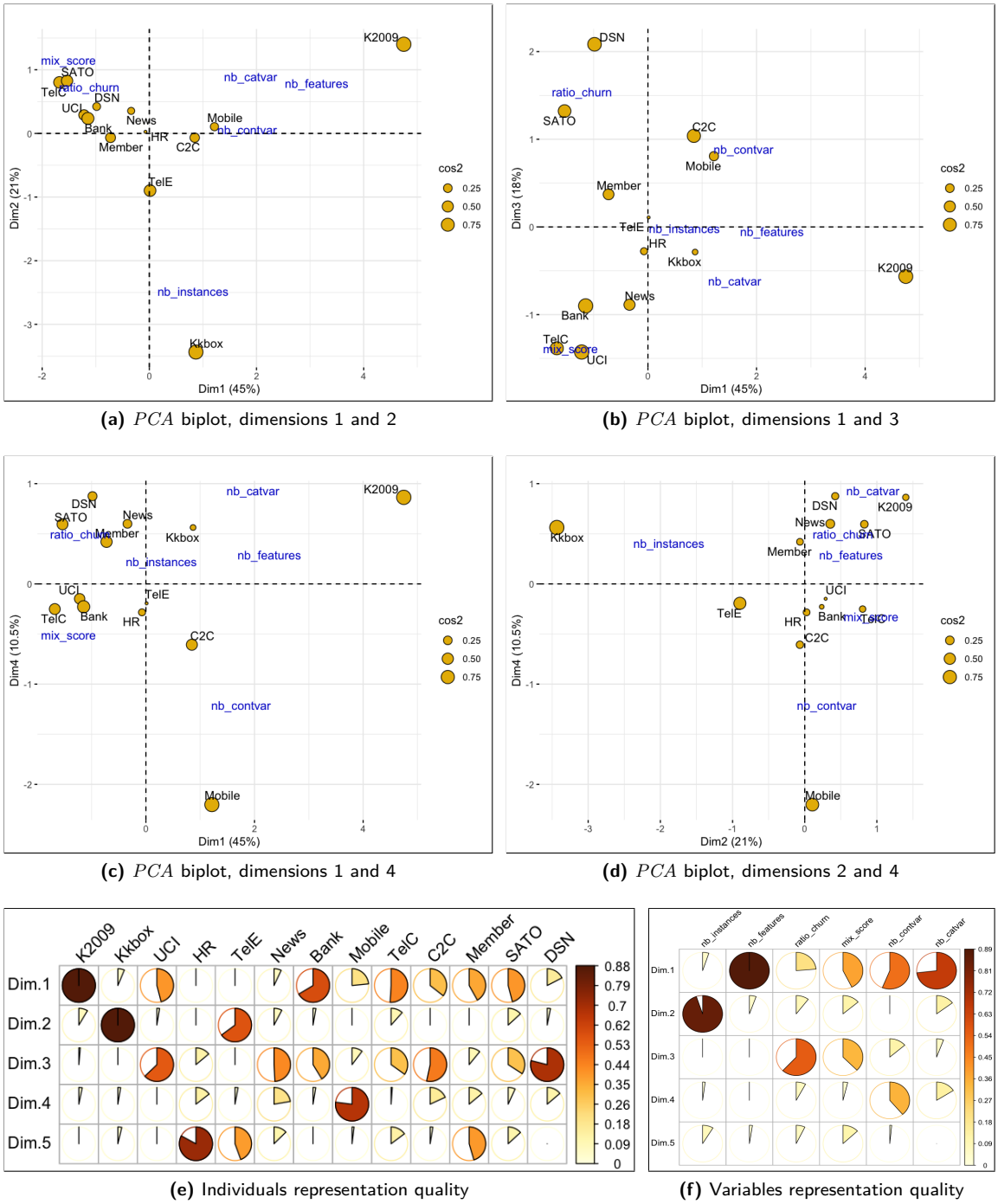
The Table 3.3 gives the public churn datasets that are considered in this work dedicated to customer profiling and provides their online access (see also Appendix A.1.1 for details). These datasets have diverse number of instances, number of features, and percentage of churners and *dummified* features. Specifically, before fitting a model, categorical variables are converted to their numerical representation through a *dummification* process where each category becomes a binary variable. We also provide the number of continuous and categorical variables after *dummification*.

Although the general data characteristics given in Table 3.3 suggest similarities between several datasets, it is important to remind that multiple intrinsic data properties can impact the prediction in the churn context. This includes in particular the existence of small *disjuncts*, the overlap between classes, the noisy data or the borderline instances (see Section 3.2.4). To establish the extent to which the classes may be intertwined, we propose a *mixture score*, which is defined as follows,

$$mix.Score = (\mu_+ - \mu_-)^\top \left(\frac{\Sigma_+ + \Sigma_-}{2}\right)^{-1} (\mu_+ - \mu_-), \tag{3.1}$$

where μ_i is the mean vector and Σ_i the covariance matrix of the cluster i respectively. Note that as we deal with mixed data (continuous and categorical variables) we perform the Factor Analysis for Mixed Data (Bécue-Bertaut and Pagès, 2008) on the original dataset, and derive μ_i and Σ_i. Thereby, the higher the mixture score, the more separable the classes.

To get a better overview of the multiple datasets facets, we provide in Figure 3.3, PCA (Principal Component Analysis) biplot representations of the datasets distribution over the characteristics identified in Table 3.3. As different dimensions can provide different information, we give biplots for the 4 first PCA components explaining 94.5% of the total variance. However, what is important is above all to observe the diversity of these data by the characteristics that describe them. To this end, we rely on the quality of representations of datasets depicted in Fig. 3.3 (e) and the correlation between the variables and the components depicted in Fig. 3.3 (f). Thus in Fig. 3.3 (a,e,f), we note the opposition between very balanced and mixed datasets, with many categorical variables (about 27 times of categorical variables than continuous) such as K2009 and more balanced and less mixed datasets, with fewer variables and only about twice categorical variables than continuous such as SATO, UCI and TeIC. In Fig. 3.3 (b,e,f), we observe that dimension 2 is mainly characterized by the KKbox dataset with a very high number of instances followed by TeIE the closest dataset. The 3^{rd} component in Fig. 3.3 (c,e,f) characterized mainly by the ratio-churn contrasts highly balanced and less well separated data such as DSN, SATO and less balanced and better separated datasets such as UCI and TeIC. Finally,



(a) *PCA* biplot, dimensions 1 and 2

(b) *PCA* biplot, dimensions 1 and 3

(c) *PCA* biplot, dimensions 1 and 4

(d) *PCA* biplot, dimensions 2 and 4

(e) Individuals representation quality

(f) Variables representation quality

Figure 3.3 : (a, b, c & d) Biplots visualization for publicly available churn-like datasets (*individuals*) and their characteristics (*variables*) for different PCA components. (e & f) Quality of representations on the factor map.

the 4^{th} component makes it possible to show the opposition between datasets with a very high ratio of continuous variables, compared to the number of categorical variables, such as the Mobile dataset and the rest of the datasets with opposite characteristics. The other datasets not mentioned before share the same interpretations, according to their proximity with the other datasets cited, while taking into account their quality of representation.

3.2.3 Machine learning for churn profiling

Recently, several studies focused on churn prediction models that can reach a good trade-off between the prediction performance and the results interpretability in terms of customers profile. As an example, De Caigny et al. (2018) designed the Logit Leaf Model (`LLM`), which consists in two phases, namely a segmentation phase followed by a prediction step. For `LLM`, the segmentation is based on

the partitioning obtained at the leaves of a decision tree that exploits the churn label from the input data. Then, for each data subset a logistic regression model is fitted which offers prediction and interpretability capabilities. `LLM` also include a random undersampling and a features selection phase. The authors provide experimental results on fourteen datasets ranging from the Financial Service to Telecommunication industry.

Following on `LLM` proposal, Ullah et al. (2019) designed a churn prediction model using Random Forest which aims at providing both interpretability and prediction efficiency. The authors performed customers profiling using k-means and partition the data into three groups labeled as *Low*, *Medium* and *Risky* churners. As `LLM`, Ullah *et al.*'s `RF`-based model includes features selection. Customer churn data have usually a complex structure which reflects a strong class imbalance and also an intrinsic data segmentation due to the multiplicity of customer behavior patterns. Let us remember that the standard k-means algorithm considers the uniform spherical Gaussian mixture model with equal proportions. Hence, when the clusters are not easily separable, one should depart from the standard k-means assumptions by using novel representations that takes into account the non linearity of the underlying data structure.

Successful clustering strategies have proposed to rely on Deep AutoEncoders (DAE) (Hinton and Salakhutdinov, 2006 ; Bengio et al., 2013) to handle data that require weak assumptions regarding the clusters shapes and filter out irrelevant features (Song et al., 2013 ; Alkhayrat et al., 2020). DAEs can generate a more cluster-friendly representation of the data (or *encoding*) in an unsupervised manner while automatically learning important features. This type of *self-supervised* neural network is trained to replicate its input at output while optimizing a cost function. Several works have proposed to combine deep embeddings and clustering in a sequential way or within a joint optimization. Stacked DAEs were successfully used to learn the representation of an affinity graph before running k-means on the learned representations in order to identify clusters Tian et al. (2014). In Guo et al. (2017), the authors incorporate a DAE into the Deep Embedded Clustering (DEC) framework (Xie et al., 2016) to jointly learn features and clustering. A novel ensemble method was introduced in Affeldt et al. (2020) that uses *landmarks* and DAE to perform an efficient deep spectral clustering.

Customer data typically involved continuous and categorical features which should both be taken into account by the embeddings. In this work, we propose the use of a DAE loss function that jointly optimizes the novel representations based on categorical *and* continuous variables, which avoids the usual *dummification* pre-processing that can be damaging for the underlying data structure (Section 3.2.4.a).

3.2.4 Our contribution

In this Section, we first evaluate multiple alternatives within a machine learning churn prediction pipeline composed of a sampling stage, a model fitting phase and a robust evaluation procedure (Figure 3.2 ; green and gray parts). We choose to focus on *traditional* machine learning techniques as reviewing in depth of the existing algorithmic variants and cost-sensitive approaches would not be feasible in the scope of this article.To tackle the imbalance issue (López et al., 2013 ; Błaszczyński and Stefanowski, 2018 ; Stefanowski, 2016), we associate each learning method with widespread sampling approaches to balance the classes distribution as it was shown to play a significant role in the performance of standard classifiers (García et al., 2012). Based on theses experiments results, we can identify the most effective machine learning techniques and propose an ensemble method that can be successfully applied to a wide range of *churn-like* datasets. Finally, following on recent developments in machine learning customer profiling (De Caigny et al., 2018 ; Ullah et al., 2019) and the promising results obtained with deep clustering approaches (Section 3.2.3), we demonstrate the effectiveness of our ensemble proposal on a segmented version of several churn benchmark datasets which makes it possible to directly draw conclusions on customer profile (Figure 3.2 ; green and blue parts).

3.2.4.a Unsupervised machine learning techniques

An autoencoder is a neural network that is trained in an unsupervised or *self-supervised* manner. Its parameters are learned in such a way that the output values tend to replicate the input training samples. The internal hidden layer can be used as a low dimensional representation of the input which captures the more salient features. We can decompose an autoencoder in two parts, namely an *encoder* f_θ, followed by a *decoder* g_ψ. The first part provides the *encoding* of the input dataset by computing a feature vector $\mathbf{y}_i = f_\theta(\mathbf{x}_i)$ for each input training sample. Then, the encoding is transformed back

to its original representation by the *decoder* part, following $\hat{\mathbf{x}}_i = g_\psi(\mathbf{y}_i)$. The sets of parameters for the encoder f_θ and the decoder g_ψ are learned simultaneously during the reconstruction task while minimizing the *loss*, referred to as $\mathcal{J}$ and given by,

$$\mathcal{J}_{AE}(\theta, \psi) = \sum_{i=1}^{n} \mathcal{L}(\mathbf{x}_i, g_\psi(f_\theta(\mathbf{x}_i))), \tag{3.2}$$

where $\mathcal{L}$ is a cost function for measuring the divergence between the input training sample and the reconstructed data. The encoder and decoder parts can have several shallow layers, yielding a deep autoencoder (DAE) that enables to learn higher order features. The network architecture of these two parts usually mirrors each other.

Churn data typically contain numerical *and* categorical data. A straightforward manner for a neural network to process categorical input is by using the *one-hot encoding* strategy. However, as shown in Guo and Berkhahn (2016), embeddings should be preferred to one-hot encoding vectors, as they reduce memory usage and speed up the neural network learning. Besides, embeddings can capture intrinsic properties of the categorical variables and reveal relationship between them.

Inspired by Guo *et al.* (Guo and Berkhahn, 2016) proposal, we adapt the *entity embedding* in a unsupervised context to automatically learn the representation of categorical features in multi-dimensional spaces which puts the feature's values with similar effect close to each other.

Such an approach reveals the inherent continuity of the categorical data. Practically, it consists in *transforming* categorical columns (vectors of size n) into an embedding matrix (of size $n_{instances} \times embedding_{dim}$) taken from a neural network trained with those categories (Fig. 3.4). In this study, we set $embedding_{dim}$ to be 2 when the categorical variables have only two unique values, and to be $ceil(n_{unique} \times compression)$, where $compression = \frac{1}{2}$. We provide in Table 3.4 a toy example of entity embeddings obtained for two categorical variables cat^a ($n_{unique} = 2$) and cat^b ($n_{unique} = 4$), as done in our experiments.

Table 3.4 : Toy example of an entity embeddings for 2 categorical variables

instance	cat^a	cat^b	$\mathbf{x}^{cat}$ *(entity embeddings)*			
			cat_0^a	cat_1^a	cat_0^b	cat_1^b
$i = 0$	1	2	0.002598	−0.012928	0.036055	−0.003408
$i = 1$	1	1	−0.015642	0.016857	0.036055	−0.019931
...	...	...	...	...	...	...
$i = n$	2	4	−0.015642	0.016857	0.013035	−0.019931

Thus, to optimize the customer segmentation while learning a combination of numerical and categorical features within a unique embedding, we train the parameters of a DAE as given in Fig. 3.4[†]. Inspired by Guo et al. (2017) we propose to combine embedding and clustering simultaneously as depicted in Fig. 3.4. This respects the idea of improving embedding taking into account local structure preservation. Thereby the loss function to be minimized amounts to the sum of a reconstruction loss noted $\mathcal{J}_{DAE}$ and a clustering loss noted $\mathcal{J}_{clust}$ given by

$$\mathcal{J}_{AE}(\theta, \psi) = \sum_{i=1}^{n} ||y_i - g_\psi(f_\theta(\mathbf{x}_i^{num}))||_2^2 - \sum_{i=1}^{n} y_i \log(g_\psi(f_\theta(\mathbf{x}_i^{cat}))), \tag{3.3}$$

and

$$\mathcal{J}_{clust}(\theta, \psi) = \sum_{i=1}^{n} \sum_{k=1}^{G} r_{ik} ||g_\psi(\mathbf{x}_i) - \mu_k||_2^2, \tag{3.4}$$

with n the number of samples, G the number of clusters, $r_{ik} = 1$ if sample i belongs to cluster k, and the concatenation of the vectors $\mathbf{x}_i^{num}$ and $\mathbf{x}_i^{cat}$ gives $\mathbf{x}_i$. Ultimately, for our experiments, each customers' segments is then split in train and test embeddings subsets, before the machine learning models are fit on the train part (see Section 3.3.1).

[†]**Dropout** refers to cutting the connection to a set of random neurons in order to reduce overfitting ; **LinBnDrop** is a sequence of linear layer and batch normalization that aims at standardizing the input to improve training and dropout (Loffe and Normalization, 2014).

Additionally the neural network architecture rely on several components : - *Dropout* (Srivastava et al., 2014) : refers to cutting the connection to a set of random neurons in order to reduce overfitting - *LinBnDrop* (Ioffe and Szegedy, 2015) : Which is a sequence of Linear layer, Batch normalization aims at standardizing the input to a layer to improve training and Dropout. Finally we have chosen a sigmoid for the continuous variables providing that they are standardized between zero and one.

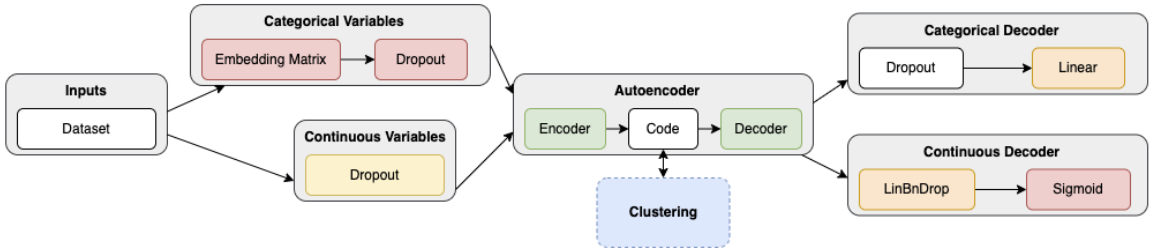


Figure 3.4 : Deep network pipeline for the joint learning of instances embeddings and customer segmentation. *Adapted from the online course walkwithfastai.com.*

3.2.4.b An effective soft voting approach

Our ensemble strategy involves a set of $\{M_i\}$ models for which we propose to apply a *soft voting*, as describe in Mohandes et al. (2018), before computing the metrics. One advantage of this type of vote is related to the increase weight given to the most different pairs of models. Specifically, if we consider an ensemble of three models, using the soft voting, the expected score $\hat{y}_{ens}$ is then expressed as a weighted sum of the individual scores as given by Eq.3.5,

$$\hat{y}_{ens} = \omega_1 \hat{y}_{M_1} + \omega_2 \hat{y}_{M_2} + \omega_3 \hat{y}_{M_3}, \tag{3.5}$$

where

$$\omega_1, \omega_2, \omega_3 = softmax(\widetilde{\omega_1}, \widetilde{\omega_2}, \widetilde{\omega_3}), \tag{3.6}$$

with

$$\widetilde{\omega_i} = \frac{1}{\rho(\hat{\mathbf{Y}}_{M_i}, \hat{\mathbf{Y}}_{M_j}) + \rho(\hat{\mathbf{Y}}_{M_i}, \hat{\mathbf{Y}}_{M_k})}. \tag{3.7}$$

where ρ denotes the Pearson correlation.

3.3 Experiments on public datasets

3.3.1 Ensemble method for prediction and profiling

In this section, we propose to associate our churn prediction ensemble method (`LR`|`XGBoost`|`RF` ; Chapter 3.1) to a deep customer data profiling. Data are segmented based on the approach described in Section 3.2.4.a (see also Fig. 3.4) that jointly learns a k-means partitioning (G clusters) with DAE encodings and entity embeddings. Each cluster C_i is split into a C_i^{train} train set and a C_i^{test} test set, in a stratified manner (80%/20%). The aggregated score of the models $\{M^j\}_{j=1..m}$ is then used to predict churn behavior on each segment C_i, and the average of all the test sets AUC_i provides the overall AUC prediction result (Fig. 3.5). In a supervised churn prediction context, labels are already known for our benchmark datasets and can be used for the model evaluation. In practice, novel observations for which the company requires a label would correspond to our test subsets.

3.3.2 Quantitative evaluation of churn prediction

We compare our approach to state-of-the-art methods in the context of churn, namely `LLM` and Ullah's `RF` model. For `LLM`, we used the implementation provided by the `LLM` R package (V1.1.0) [‡]. Ullah's `RF`-based model was implemented following the author's description and based on *scikit-learn* Python package. We performs 50 runs on all benchmark datasets [§] for the compared approaches. The Table 3.5 summarizes AUC results for different number of clusters (from $k = 2$ to $k = 6$).

[‡] `https://cran.r-project.org/web/packages/LLM/LLM.pdf`
[§] for the largest datasets (*K2009*, *KKBox* and *C2C*), 20 runs were done

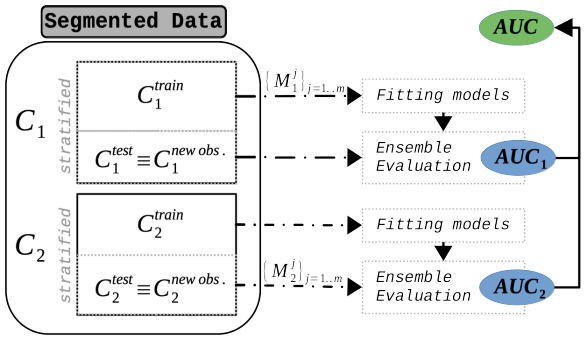


Figure 3.5 : Evaluation of the ensemble profiling and prediction approach. Two customers subgroups are identified in an unsupervised manner ($G = 2$; see Section 3.2.4.a). In practice, test subsets would correspond to novel observations for which the company expects a label

As can be seen, our ensemble proposal combined with the DAE data segmentation outperforms the competitive profiling approaches, with $\widetilde{AUC}$ between 0.8516 and 0.8546, while `LLM` and the `RF`-based model reach 0.8450 and 0.8317 respectively. It should be highlighted that `LLM` encounters difficulty to handle several datasets, for which its execution could exceed 3 hours (vs. less than half an hour on average for our proposal) and our experiments should be stopped before the convergence of these approaches (Table 3.5, *overtime* labels).

The AUC value embeds two metrics which are the *Precision* and the *Recall*. While the Precision estimates the ability of the model to obtain actual churners among its predicted churners, the *Recall* estimates the ability of the model to recover actual churners. Usually, the cost of a false positive in the churn context is considered as less damaging than the non identification of an actual churner. Indeed, contacting loyal customers to propose them with several advantages such as discounts generally reinforce their loyalty at a fixed cost.

By contrast, missing an actual churner could induce a significant loss of profit. Hence, *Recall* is, along with AUC, an important metric to be considered when building a churn prediction model. The Table 3.6 summarizes the Recall of our ensemble approach combined with data segmentation (best value is given in bold, second best value is underlined). Our proposal outperforms the *Recall* of `LLM` and Ullah's `RF`-based model.

Table 3.5 : AUC results for our ensemble proposal vs. LLM (De Caigny et al., 2018) and RF-based (Ullah et al., 2019).

	LR|XGBoost|RF & DAE-based Segmentation			**LLM**	**RF-based**
	k=2	k=4	k=6	*(De Caigny et al. (2018))*	*(Ullah et al. (2019))*
Bank	**0.8620** ± 0.0088	0.8612 ± 0.0096	0.8605 ± 0.0086	0.8501 ± 0.0089	0.8422 ± 0.0119
C2C	**0.6719** ± 0.0062	0.6671 ± 0.0048	0.6708 ± 0.0051	*overtime*	0.6558 ± 0.0046
DSN	**0.8923** ± 0.0157	0.8867 ± 0.0185	0.8852 ± 0.0193	0.8589 ± 0.0278	0.8885 ± 0.0171
HR	0.8402 ± 0.0267	**0.8449** ± 0.0294	0.8335 ± 0.0367	*overtime*	0.7491 ± 0.0355
K2009	0.5063 ± 0.0083	0.5059 ± 0.0113	**0.5091** ± 0.0105	*overtime*	**0.5091** ± 0.0112
KKBox	0.8764 ± 0.0009	**0.8765** ± 0.0012	0.8756 ± 0.0014	*overtime*	0.8749 ± 0.0013
Member	**0.7048** ± 0.0094	0.7028 ± 0.0122	0.6985 ± 0.0106	0.6708 ± 0.0118	0.6858 ± 0.0121
Mobile	0.9071 ± 0.0026	0.9074 ± 0.0028	0.9069 ± 0.0036	*overtime*	0.8985 ± 0.0039
SATO	**0.8175** ± 0.0213	0.8142 ± 0.0199	0.8133 ± 0.0189	0.7835 ± 0.0188	0.8171 ± 0.0182
TelC	0.8465 ± 0.0097	0.8480 ± 0.0098	**0.8490** ± 0.0096	0.8399 ± 0.0114	0.8212 ± 0.0107
TelE	0.9360 ± 0.0023	0.9341 ± 0.0022	0.9319 ± 0.0024	*overtime*	**0.9409** ± 0.0016
UCI	0.9120 ± 0.0215	**0.9152** ± 0.0209	0.9084 ± 0.0197	0.8732 ± 0.0323	0.9095 ± 0.0213
News	**0.8639** ± 0.0076	0.8620 ± 0.0076	0.8541 ± 0.0071	*overtime*	0.8554 ± 0.0084
$\widetilde{AUC}$	**0.8620**	0.8612	0.8541	0.8450	0.8219
$\overline{AUC}$	**0.8182**	0.8174	0.8151	0.8127	0.7978

3.3.3 Qualitative evaluation of churn profiling

Beyond the performance in churn prediction performance for our ensemble approach, it is important to highlight the benefit of the data segmentation in terms of customers profiling. Indeed, the parti-

Table 3.6 : Recall results for our ensemble proposal vs. LLM (De Caigny et al., 2018) and RF-based (Ullah et al., 2019).

	LR|XGBoost|RF & DAE-based Segmentation			**LLM**	**RF-based**
	k=2	k=4	k=6	*(De Caigny et al. (2018))*	*(Ullah et al. (2019))*
Bank	**0.7551** ± 0.0329	0.7548 ± 0.0346	0.7490 ± 0.0410	0.7398 ± 0.0376	0.7162 ± 0.0396
C2C	0.6663 ± 0.0490	0.6578 ± 0.0658	**0.6750** ± 0.0466	overtime	0.6006 ± 0.0400
DSN	0.8356 ± 0.0459	0.8065 ± 0.0489	0.8027 ± 0.0408	0.8118 ± 0.0608	**0.8376** ± 0.0507
HR	**0.7488** ± 0.0591	0.7297 ± 0.0590	0.7311 ± 0.0789	*overtime*	0.6629 ± 0.0887
K2009	**0.5376** ± 0.2220	0.4416 ± 0.2668	0.4669 ± 0.2608	*overtime*	0.4403 ± 0.1780
KKBox	0.7466 ± 0.0147	0.7507 ± 0.0195	**0.7569** ± 0.0162	*overtime*	0.7440 ± 0.0144
Member	**0.7482** ± 0.0635	0.7426 ± 0.0751	0.7276 ± 0.0867	0.7140 ± 0.0921	0.6996 ± 0.0776
Mobile	0.8365 ± 0.0152	**0.8415** ± 0.0116	0.8388 ± 0.0111	*overtime*	0.8219 ± 0.0140
SATO	0.7371 ± 0.0797	0.7283 ± 0.0797	0.7106 ± 0.0712	0.7079 ± 0.0733	**0.7631** ± 0.0417
TelC	**0.7985** ± 0.0528	0.8026 ± 0.0443	0.7940 ± 0.0452	0.8060 ± 0.0450	0.7671 ± 0.0416
TelE	0.9313 ± 0.0092	0.9320 ± 0.0076	0.9322 ± 0.0091	*overtime*	**0.9361** ± 0.0067
UCI	0.8155 ± 0.0289	0.8227 ± 0.0335	0.8208 ± 0.0424	0.7727 ± 0.0470	**0.8257** ± 0.0385
News	0.7729 ± 0.0408	$mathbf{0.7772 \pm 0.0362}$	0.7675 ± 0.0389	*overtime*	0.6715 ± 0.0681
$\widetilde{AUC}$	0.7551	0.7548	**0.7569**	0.7563	0.7440
$\overline{AUC}$	**0.7638**	0.7529	0.7518	0.7587	0.7297

tioning of the customer data puts forward the most important features on which the M_i models are fitted. These features can be further assigned to subgroups of churners and non churners within each cluster. Hence, proactive marketing campaigns could be designed to target a group of both churners and non churners – reinforcing the loyalty for the former while potentially retaining the latter – or focus only on several churners subgroups.

The Tables 3.7 to 3.9 provide the 3 most important features for three datasets ; *Bank*, *Member* and *TelC*. The features are ranked based on their *importance score* which is computed from the mean impurity decrease of each split during class prediction (Section 3.3.1). This score is further multiplied by the average standardized mean value of each segment in each class. The top most important features are obtained on these final importance values.

Bank dataset (Table 3.7) As an example, the *tenure*[¶] feature helps to discriminate churners and non churners in clusters C_2 to C_4 for *Bank*, while geographical aspects and credit type information are more important in cluster C_1. The *creditscore* variable also plays a discriminative role in clusters C_1 and C_3. This is understandable given that a customer with higher credit score would tend to remain with the same bank. A plausible interpretation would be that a customer with higher credit score would tend to remain with the same bank. Hence, it would be interesting for the company to conduct investigations along this line in order to build efficient proactive marketing campaigns.

Table 3.7 : Top 3 features for our ensemble proposal with DAE-based segmentation ($k = 4$; Bank data).

	Bank	
	churner	*non churner*
C_1	creditscore	geography_spain
	numproducts_2	numproducts_2
	geography_germany	hascrcard
C_2	estimatedsalary	age
	gender	gender
	tenure	numproducts_2
C_3	age	creditscore
	balance	balance
	hascrcard	tenure
C_4	estimatedsalary	estimatedsalary
	age	balance
	tenure	gender

Member dataset (Table3.8) Another example is given by *Member*, where only the cluster C_3 is not concerned by the *annual_fees*[‖] variable. It is rather impacted by the *member_gender* information. This is indicative of a particular customer subgroup. We also notice for this cluster the impact of the

[¶] *tenure* refers to the number of years that the customer has been a client of the bank

[‖] *annual_fees* are paid in return for using the exclusive facilities offered by this club

membership_package on the non churner subgroup. This variable indicates whether fees are customized for members personal package, suggesting straightforward manner to improve member loyalty.

Table 3.8 : Top 3 features for our ensemble proposal with DAE-based segmentation ($k = 4$, Member data).

	Member	
	churner	*non churner*
C_1	annual_fees	membership_term_years
	additional_member_3	member_annual_income
	member_occupation_cd_2	annual_fees
C_2	annual_fees	member_age_at_issue
	member_age_at_issue	payment_mode_semi-annual
	payment_mode_annual	member_occupation_cd_2
C_3	member_annual_income	membership_package
	membership_term_years	member_occupation_cd_1
	member_gender	payment_mode_annual
C_4	member_age_at_issue	member_age_at_issue
	membership_term_years	additional_member_0
	annual_fees	member_occupation_cd_2

TelC dataset (Table 3.9) With *TelC*, we can notice that clusters C_1 to C_3 decomposes into churners subgroups that are concerned by different payment method (C_1, bank transfer ; C_2, credit card ; C_3, electronic check). The C_4 *TelC* cluster stands out from the rest of the clusters in terms of most important features, both for churners and non churners. Interestingly, the C_3 cluster seems to indicate a non churner subgroup that is satisfied with the technical support.

Table 3.9 : Top 3 features for our ensemble proposal with DAE-based segmentation ($k = 4$, TelC data).

	TelC	
	churner	*non churner*
C_1	monthlycharges	totalcharges
	totalcharges	techsupport_no
	paymentmethod_bank transfer	onlinebackup_no
C_2	paymentmethod_credit card	gender
	partner	monthlycharges
	gender	paymentmethod_elec. check
C_3	monthlycharges	monthlycharges
	tenure_group_tenure_24-48	totalcharges
	paymentmethod_elec. check	techsupport_yes
C_4	totalcharges	seniorcitizen
	seniorcitizen	dependents
	paperlessbilling	streamingmovies_yes

All in all, these qualitative evaluations put forward the intrinsic multidimensionality and multiplicity of the customer behavior patterns.

3.4 Conclusion

In this Chapter, we propose to review, evaluate, and compare several widespread machine learning approaches in the context of churn prediction and profiling. We also provide insightful general recommendations for the choice of a processing pipeline for churn prediction and profiling based on an *ensemble* approach.

Note that we only consider the default parameters for each approach while the supervised context would also allow for boosting versions of some of these techniques (Clemente et al., 2010 ; Lemmens and Croux, 2006). This could significantly improve classification results, in particular for `SVM` (Vafeiadis et al., 2015). Furthermore, churn prediction issue pertains to the broader class of imbalance

data problem. It is therefore related to the extreme case of anomaly or *outlier* detection (Kong et al., 2020) for which many approaches have been proposed (Chandola et al., 2009 ; Alam et al., 2020 ; Pang et al., 2017 ; Taha and Hadi, 2019). In particular, semi-supervised approaches regularly provide state-of-the-art results (Alam et al., 2020 ; Villa-Pérez et al., 2021). Among the well-known semi-supervised techniques for anomaly detection, one could cite Local Outlier Factor (`LOF`) (Breunig et al., 2000), One-Class SVM (`ocSVM`) (Schölkopf et al., 1999), Isolation Forest (`iForest`) (Liu et al., 2012) and Support Vector Data Description (`SVDD`) (Tax and Duin, 1999) methods. These type of techniques should be the object of our future works.

Another type of approaches for which a particular interest should be taken in the context of attrition are the deep learning methods. We can observe that the finance industry is gradually adapting various machine learning techniques. In particular, detecting economic crimes (eg., accounting fraud, money laundering) triggered successful applications of machine learning to this area, where `LR`, `Gnb` and `SVM` are among the most classic methods exploited. However, the occurence of new kinds of fraud, with the growth of electronic market, has popularized deep learning methods which enable the emergence of numerous and innovative deep anomaly detection methods (Pang et al., 2021). In particular, `GEV-NN` (Generalized Extreme Value Neural Network) which proposes to use Gumbel distribution as an activation function, reaches state-of-the-art results in the context of imbalanced data (Munkhdalai et al., 2020).

It is also important to notice that most of the churn-like prediction frameworks typically consider only *structured data*. However, as a large proportion of big data consists of diverse *unstructured data* (Gandomi and Haider, 2015), it is important to find strategies that enable the incorporation of the information that they contain. Indeed, online communication means between customers and companies or banks are expanding rapidly. Previous studies demonstrate that textual data can improve the churn prediction performance. Examples can be found with the use of highly unstructured data coming from social networks (Tang et al., 2014 ; Benoit and Van den Poel, 2012 ; Coussement and Van den Poel, 2008). Recently De Caigny et al. (2020) proposed the incorporation of textual information based on Convolutional Neural Network. This last consideration should be part of an interesting short-term study.

Finally, while our study does not analyze the customers' churn decision through time, it is important to mention that multivariate times series data have triggered innovative techniques last years in the context of churn. Indeed, it is reasonable to hypothesis that the modifications of customers' behavior can be detected during the time leading to a churn decision. To deal with multivariate times series, several techniques were proposed that are based either on the featurization of the time series data to construct a tabular dataset or on dimension reduction combined with a binary classifier (Orsenigo and Vercellis, 2010 ; He et al., 2014). More recently, Wang *et al.* (Wang et al., 2016a) propose to use recurrent neural networks to tackle the time series data classification task. Finally, Óskarsdóttir *et al.* (Óskarsdóttir et al., 2018) designed extensions of the similarity forest method and successfully applied them for classifying multivariate time series data for churn prediction.

4

Deep Learning for Tabular data

In this Chapter we propose to enrich our churn prediction model with recent deep learning approaches. More specifically, we propose a novel *semi-supervised deep learning architecture*, which is built on top of the recent *contrastive learning* framework (Chen et al., 2020).

4.1 Context and background

4.1.1 Introduction

Labeling humongous amount of data is a bottleneck for building generalist models that acts on complex tasks. Indeed, labeling a large amount of data is costly and typically error prone when the task is performed by a human annotator (Northcutt et al., 2021). Self-supervised learning (SSL) has recently emerged has a novel paradigm to deal with this issue. Broadly speaking, SSL aims at framing the unsupervised problem as a supervised one in order to build a good representation. It can then be used in turn for downstream tasks such as classification or semi-supervised learning. Two strategies are generally encountered, namely *pretext task(s)* and *contrastive learning*. *Pretext tasks* learn representations of the data using pseudo labels. These pseudo-labels are generated automatically based on the characteristics found in the data. An example of a pseudo-supervised task could be to predict whether an image is rotated by a certain amount of degree (Gidaris et al., 2018). *Contrastive learning* is built on positive pairs which is formed from two perturbed views of the same image that are kept close. By contrast, negative pairs comprise the rest of the dataset along with their respective perturbed image. They are pushed away as being two different images. In terms of manual task, it would be equivalent to ask if two images are similar or dissimilar (Chen et al., 2020).

Self-supervised learning is regularly encountered in several areas of machine learning. Natural Language Processing (NLP) was one of the first field to leverage the power of self-supervised learning with the `Word2Vec` technique which aims at predicting a *target* word from the surrounding *context words* (Goldberg and Levy, 2014). It has been extended with the family of `Transformers`, one instance of it being `BERT` (*Bidirectional Encoder Representations from Transformers*) which relies on the *Mask Language Modeling* task. Specifically, it randomly mask 15% of the tokens from each sequence while the encoder is trained to predict those missing tokens, given all the other words of the sequence (Devlin et al., 2018). Other models have been proposed (Lewis et al., 2019 ; Alrowili and Vijay-Shanker, 2021). In vision, several pre-text tasks were incorporated into the framework such as context prediction (Doersch et al., 2015), image clustering where the clusters are used as class (Caron et al., 2018) and several others variations (Misra et al., 2016 ; Doersch et al., 2015 ; Zhang et al., 2017)].

In this work, we propose `MAC` (*Mean-teacher Architecture using Contrastive learning*), a contrastive pre-training procedure along with a novel noise generator based on a *diffusion process* (Song et al., 2020). We test `MAC` on several datasets extracted from the *OpenML-CC18*[a] benchmark data (Bischl et al., 2017), which is a collection of 72 real-world classification datasets. We focus on 20 datasets with a binary target, in line with the attrition issue. We show that `MAC` pre-training improve classification win ratio in the semi-supervised setting.

[a] https://www.openml.org

4.1.2 Related Works

Self-supervised learning for tabular dataset has naturally arose in the aftermath of its success in vision and NLP. Even though, in the tabular paradigm, there is still a contest between traditional methods like `XGBoost`, `Random Forest` and `Neural networks`. Some author argues that deep learning approches are not *all you need* (Shwartz-Ziv and Armon, 2022) in the tabular domain.

4.1.2.a SimCLR approach for Contrastive Learning

In the tabular setting, we will concentrate around the contrastive learning paradigm. In particularly, we focus on `SimCLR` (*a Simple framework for Contrastive Learning of visual Representations*), a contrastive self-supervised learning approach which was designed to deal with images (Chen et al., 2020). Regarding its structure, an image is taken randomly. Then, two transformations are applied to get two different views of the image. Each image is passed through an encoder to build representations. Subsequently, a non-linear fully connected layer is applied to get two representations, z_i and z_i'. The

pre-training task is to maximize the similarity between those two representations and push away the negative elements.

First, such approaches incorporated an autoencoder in the tabular framework, such has encountered with `VIME` (Yoon et al., 2020) which builds on the *denoising autoencoder* architecture. The authors propose to perform two novel pre-text tasks, rather than injecting a gaussian noise : *(i)* estimating mask vectors from corrupted tabular data and *(ii)* a reconstruction pretext task of the input. The `TabNet` approach (Arık and Pfister, 2021) uses the *sequential attention* mechanism to select the top most important features for each decision node. For this approach, the P pretraining involves feeding to the `TabNet` encoder a corrupted data frame, where cells are randomly deleted, and then reconstructing it through the decoder. In `SubTAB` (Ucar et al., 2021) the authors proposed to turn the task of learning from tabular data into a multi-view representation learning problem by dividing the input features to multiple subsets. The architecture is an autoencoder-like with two outputs : *(i)* one reconstruction loss and *(ii)* one contrastive and divergence loss.

The `SAINT` method (Somepalli et al., 2021) is the first model to leverage `SimCLR` in the tabular domain. They proposed two pre-training task : *(i)* reconstructing the input and *(ii)* maximizing the similarity between two views of the data, z_i and z_i'. Yet, it should be noted that, by contrast with the image paradigm, only one view of the data can be corrupted in the tabular setting. The authors have shown that the results become unstable otherwise. In the appraoch contrastive `Mixup` (Darabi et al., 2021), the authors proposed to alter the data with a `MixUp` augmentation. It is then mapped to a low dimension latent space where samples with the same label class are pushed closer. With the `SCARF` approach (Bahri et al., 2021), the data are divided into two views ; one of which being perturbed by drawing from the feature's marginal. This is then sends out to the encoder to obtain the z_i and z_i' representations, before being optimized through an `InfoNCE` (*Noise-Contrastive Estimation*) loss. The benchmark built for `SCARF` represent a significant amount of work as the authors propose a benchmark built on 69 tabular dataset along with 7 baselines and a range from 25% to 100% of labeled training data.

4.1.2.b The semi-supervised mean-teacher approach

In our extended model, we propose to incorporate a semi-supervised *mean-teacher architecture* (Tarvainen and Valpola, 2017) to connect the two projections heads of the contrastive learning architecture. This implies that only the student head will experience the *backpropagation* ; the teacher head will be updated by *Exponential Moving Average* (EMA) every p iterations of the student. More formally, the student has weights θ_t and the teacher has weights θ'_t which are updated in an EMA fashion weighted by an hyperparameter α, as given by Equation 4.1,

$$\theta'_t = \alpha\theta'_{t-1} + (1-\alpha)\theta_t \tag{4.1}$$

As detailed in Section 4.2, we enrich the mean-teacher architecture with a novel perturbation which is inspired from a score-based generative model as proposed by Song et al. (2020). Specifically, the authors inject noise from several possible diffusion processes during the forward phase to map the distribution to a Gaussian distribution. Providing the static nature of data, they performed an *antithetic* sampling, which consists in perturbing the observations at different levels of the stochastic process. The perturbation is based on the *Diffusion Processes* which originated from statistical physics as a way of modeling the trajectory of a particle in a flowing fluid while being subjected to collisions with other particles. In a more abstract setting, it is a subset of a Markov process (Pavliotis, 2015) where the state space is $S = \mathbb{R}$, where jumps are not allowed, and which is framed as an Itô Stochastic Differential Equation, as described in Equation 4.2,

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + G(x, t)\mathrm{d}\mathbf{w} \tag{4.2}$$

where $f(\cdot, t)$ is the drift coefficient, $G(x, t)$ is called the diffusion coefficient and w is the standard Brownian motion.

4.2 Mean-Teacher Architecture using Contrastive learning (MAC)

In this Section, we provide the different elements that compose our `MAC` architecture and their relationships. The Figure 4.1 provides an overview of the proposed approach.

4.2.1 MAC architecture overview

The *pre-training* is performed in two views, with one branch being perturbed by injecting a diffusion type noise (Fig. 4.1, top). The *teacher* path does not incorporate any backpropagation of the loss. Only the EMA (Exponential Moving Average) of the *student* branch is concerned by the backpropagation. As opposed to the *teacher* branch, the *student* branch does not experience any noise ; it just performs a sequence of operations to build a better representation.

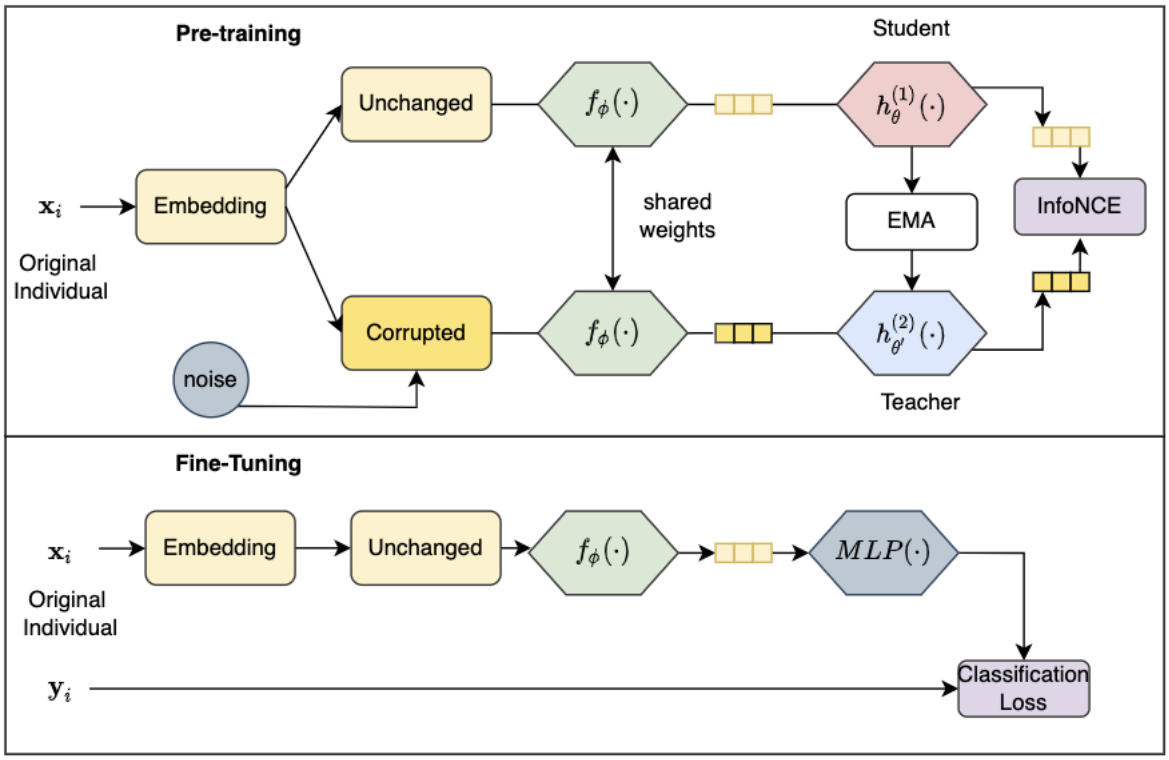


Figure 4.1 : The MAC architecture, The pre-training stage is divided into two branches the top branch backpropagate the weights of the InfoNCE loss ; the bottom branch is corrupted and is backpropagated not directly but through the Exponential Moving Average (EMA) of the student head ; Fine-Tuning is accomplished after the representation has been learned in order to perform a fine-tuning task.

The `MixUp` strategy is one of the possible way to augment the input data. The approach is similar to what has been proposed with the `SAINT` architecture (Somepalli et al., 2021). The *diffusion noise* is derived from one of the four perturbations kernel from discrete markov chains propose by Song et al. (2020) (see Section 4.2.2). Given that they all have an affine drift coefficient, their perturbations kernel are all Gaussian's. The *Loss function* is the `InfoNCE` loss (Oord et al., 2018) which pushes positive pairs from the two different representations z_i and z_i' of the same dataset to be as close as possible while negative pairs are pushed apart whenever $i \neq j$ such as given in Equation 4.3,

$$L_{pre-train} = -\sum_{i=1}^{n} log \frac{exp(z_i \cdot z_i'/\tau)}{\sum_{k=1}^{m} exp(z_i \cdot z_k'/\tau)} \tag{4.3}$$

The *fine-tuning* follows the pre-training stage, which proposed a proper representation of the data (Fig. 4.1). This final step aims at learning the weights of an `MLP` in order to perform a semi-supervised classification problem. This task requires a novel loss which is the cross-entropy between the predicted labels and the true class. In particular, the data is passed either through the Embedding layer for the categorical data, or either through an `MLP` for the numerical data. Then both of them are fed to the `SAINT` layer $f_\phi(\cdot)$ and finally to the $MLP(\cdot)$ to be learnt along with the cross entropy loss.

4.2.2 Perturbation kernel descriptions

The transition kernels aim at perturbing the data distribution p_0 to a prior distribution p_T which is in our case a Gaussian distribution.

- Variance Explosion SDE (VESDE) yields a process with exploding variance when $t \to \infty$.

Its kernel is derived as being :

$$p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0)) = \mathcal{N}\left(\mathbf{x}(t); \mathbf{x}(0), \sigma_{\min}^2 \left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{2t} \mathbf{I}\right), \quad t \in (0, 1]$$

To pick the hyperparameters we follow the instructions of the paper Song and Ermon (2020). More particularly the technique 1 ; it states that we should choose σ_{max} to be as large as the maximum Euclidean distance between all pairs of training data points.

- Variance Preserving SDE (VPSDE) yields a process with bounded variance

$$p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0)) = \mathcal{N}\left(\mathbf{x}(t); e^{-\frac{1}{4}t^2(\bar{\beta}_{\max}-\bar{\beta}_{\min})-\frac{1}{2}t\bar{\beta}_{\min}}\mathbf{x}(0), \mathbf{I} - \mathbf{I}e^{-\frac{1}{2}t^2(\bar{\beta}_{\max}-\bar{\beta}_{\min})-t\bar{\beta}_{\min}}\right), \quad t \in [0, 1]$$

In their paper the authors specifies that $\bar{\beta}_{\min} = 0.1$ and $\bar{\beta}_{\max} = 20$ to match the paper of Ho et al. (2020).

- The sub-Variance Preserving SDE (sub-VP SDE) is a modification of VPSDE which achieves better performance on images.

$$p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0)) = \mathcal{N}\left(\mathbf{x}(t); e^{-\frac{1}{4}t^2(\bar{\beta}_{\max}-\bar{\beta}_{\min})-\frac{1}{2}t\bar{\beta}_{\min}}\mathbf{x}(0), \left[1 - e^{-\frac{1}{2}t^2(\bar{\beta}_{\max}-\bar{\beta}_{\min})-t\bar{\beta}_{\min}}\right]^2 \mathbf{I}\right), \quad t \in [0, 1]$$

We follow the same criteria to set the $\bar{\beta}_{\min}$ and $\bar{\beta}_{\max}$ as the VPSDE transition kernel.

- Gaussian perturbations SDE (vanilla SDE) Lastly we harness the following SDE that comes from the blog of Yang Song and more specifically to his notebook *Tutorial of score-based generative modeling with SDEs in PyTorch*

$$d\mathbf{x} = \sigma^t d\mathbf{w}, \quad t \in [0, 1]$$

In this case the transition kernel becomes

$$p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0)) = \mathcal{N}\left(\mathbf{x}(t); \mathbf{x}(0), \frac{1}{2\log\sigma}\left(\sigma^{2t} - 1\right)\mathbf{I}\right), \quad t \in [0, 1]$$

This approach offers a simple type of noise that increase depending on the time variable t.

4.3 Experiments

We evaluate our novel method `MAC` pre-training on test AUC after semi-supervised fine-tuning in two distinct settings : on the full dataset but where only 70% of samples have labels and the remaining 30% do not, and on the full dataset where 20% of the samples have labels.

4.3.1 Experimental settings

4.3.1.a Datasets and data preprocessing

The table 4.1 lists the twenty binary dataset from the *OpenML-CC18* datasets. The table yields information's about the number of instances, of features, continuous features, categorical features and we also calculated a ratio of the class 1 over the class 0. We form a 70%/10%/20% of train/validation/test splits, where a different split is generated for every trial and all methods use the same splits.

Table 4.1 : Twenty OpenML-CC18 datasets with online access link

Dataset	#Instances	#Features	#Cont.Feat.	#Cat.Feat.	$\frac{class1}{class0}$
sick	3772	30	7	23	0.065
pc1	1109	22	21	1	0.07
pc3	1563	38	37	1	0.11
pc4	1458	38	37	1	0.14
online-shoppers-intention	12,330	18	14	4	0.18
kc1	2109	22	21	1	0.18
jm1	10,885	22	21	1	0.24
kc2	522	22	21	1	0.26
Churn-for-Bank-Customers	10,000	14	11	3	0.26
blood-transfusion-service-center	748	5	4	1	0.31
telco-customer-churn	7043	20	3	17	0.36
ilpd	583	11	9	2	0.40
spambase	4601	58	57	1	0.65
breastTumor	286	10	1	9	0.72
electricity	45,312	9	7	2	0.74
madelon	2600	501	500	1	1
balance-scale	625	5	4	1	1
kr-vs-kp	3196	37	0	37	1.09
PhishingWebsites	11,055	31	0	31	1.26
credit-g	1000	21	7	14	2.33

Regarding the data pre-processing, the data were split into categorical and continuous types. The categorical variables are passed through an embedding layer of dimension m. The continuous variables are mapped through an $MLP(\cdot)$ from a one dimension to an m dimension given that they have been previously standardized in range of zero to one by max normalization.

4.3.1.b Model architecture and training

After the data preprocessing, the observations are mapped either to the `MAC` top branch or to the `MAC` low branch, where a noise is injected (Fig. 4.1). We evaluate five types of noise, namely `MixUp` (Berthelot et al., 2019), Variance Explosion (`VP`), Sub-variance preserving (`Sub-VP`), the variance preserving (`VP`) and Stochastic Differential Equation (`SDE`). The data is then fed to a neural network – `SAINT` in our case –, which is followed by two heads : the student with the backpropagation and the teacher, without the backpropagation. The teacher learns thanks to the Exponential Average block (`EMA`). Finally we consider the loss function `InfoNCE`, as recommended by previous studies*. For the fine-tuning, the observations pass through the top branch (Fig. 4.1) and after the first encoder $f_\theta(\cdot)$, it is fed to an $MLP(\cdot)$ and optimized through a classification loss (i.e. cross entropy) based on the target value. The architecture `MAC` is pre-trained with 10 iterations and 5 iterations for the fine-tuning task. Increasing the number of iterations, led to two issues : *(i)* either we do not observe any significant improvement, *(ii)* the AUC decreases.

4.3.1.c Evaluation metric

We use the *Win matrix* on the AUC matrix of our models (including `SAINT`). To construct the *Win matrix* $\mathbf{W}$, given M methods, we compute the coefficients (i, j) following Equation 4.4,

$$W_{i,j} = \frac{\sum_{d=1}^{5} \mathbf{1}[\text{ method } i \text{ beats } j \text{ on dataset } d]}{\sum_{d=1}^{5} \mathbf{1}[\text{method } i \text{ beats } j \text{ on dataset } d] + \mathbf{1}[\text{method } i \text{ loses to } j \text{ on dataset } d]}. \tag{4.4}$$

where 'beat' means being greater or equal. Hence, if two noises have the same AUC score they will be considered as being beaten by one another.

*__Alternatives losses.__ We have investigated one alternative loss the so-called Barlow Twins (Zbontar et al., 2021). However we did not observe any improvement even sometimes a decrease in the scores whereas `InfoNCE` still remains a good asset.

4.3.2 Results

This Section provide our experimental results using the win matrices computed from the AUC scores. Each coefficient ranges from zero to one, where zero stands for the model being unable to beat its opponent. We compare six models, a baseline `SAINT` (Somepalli et al., 2021), our model with EMA and MixUp (*ema+mixup*) ; and four varinats with diffusion noises (*ema+sde*, *ema+VE*, *ema+VP* and *ema+subVP*). By contrast with Bahri et al. (2021) proposal, our win matrices are not symmetric. Indeed a great deal of competitors yields the same score and with our definition of "beat" as being greater or equal we obtain this asymmetry.

4.3.2.a Results with unbalanced labeled data

Our first evaluation scenario is constrained by a small sample size of labeled data, 20%. As shown in Figure 4.2, the baseline `SAINT` does not beat *ema+mixup* on the twenty datasets with a win score of 0.45 (Fig. 4.2, first row). It outperforms three of our diffusion noises, however *ema+VE* appears as a strong challenger for `SAINT`. It could be explained by the stochastic behavior of the model *ema+VE*. Furthermore, *ema+mixup* is overall the best model as it beats all of its competitors by a large margin above the threshold of 0.5 (Fig. 4.2, second row). Finally regarding the diffusion noises no model really stands out regarding the baseline and our proposed models.

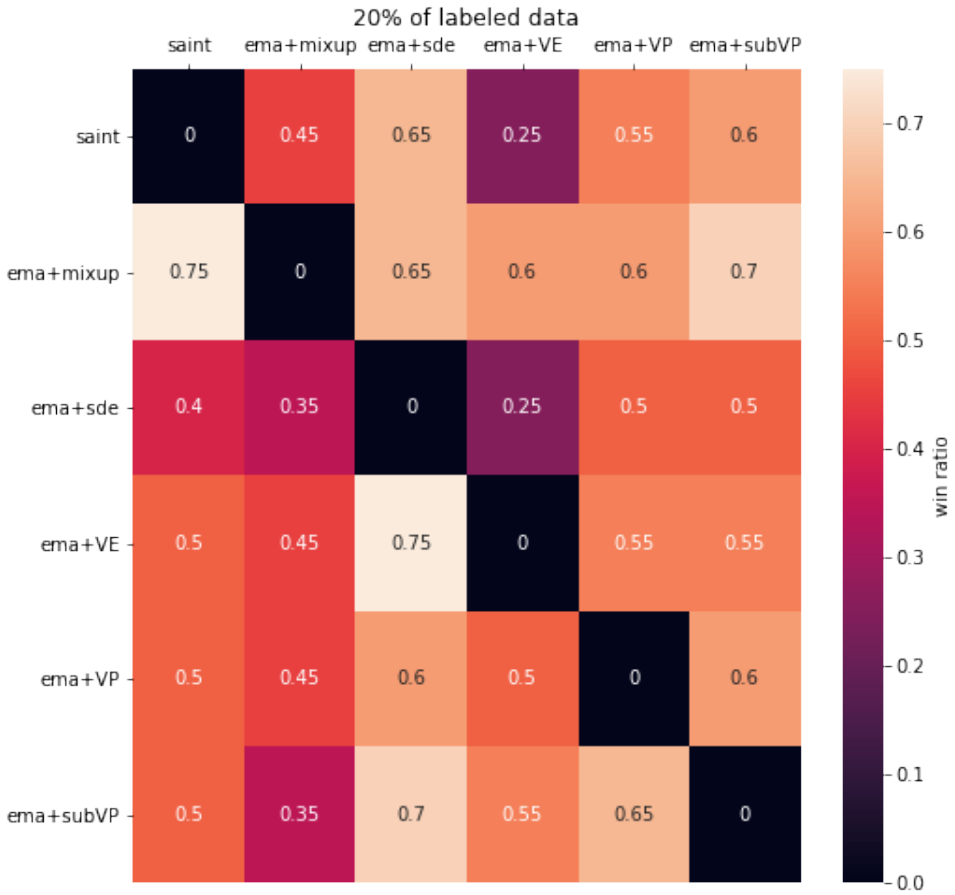


Figure 4.2 : Win matrix comparing pre-training methods against each other, and their improvement to existing solutions in a semi-supervised learning setup with only 20% of labeled data.

4.3.2.b Results with quasi-balanced of labeled data

In the second scenario we consider datasets with 70% of labeled data. By contrast with the results in Section 4.3.2.a, `SAINT` and *ema+mixup* exhibit similar win scores. It suggests that both models converge whenever the data is fully supervised. Additionally contrary to the previous case the diffusion group of diffusion noises *ema+*** is way below average which makes them ineffective in the quasi balanced case.

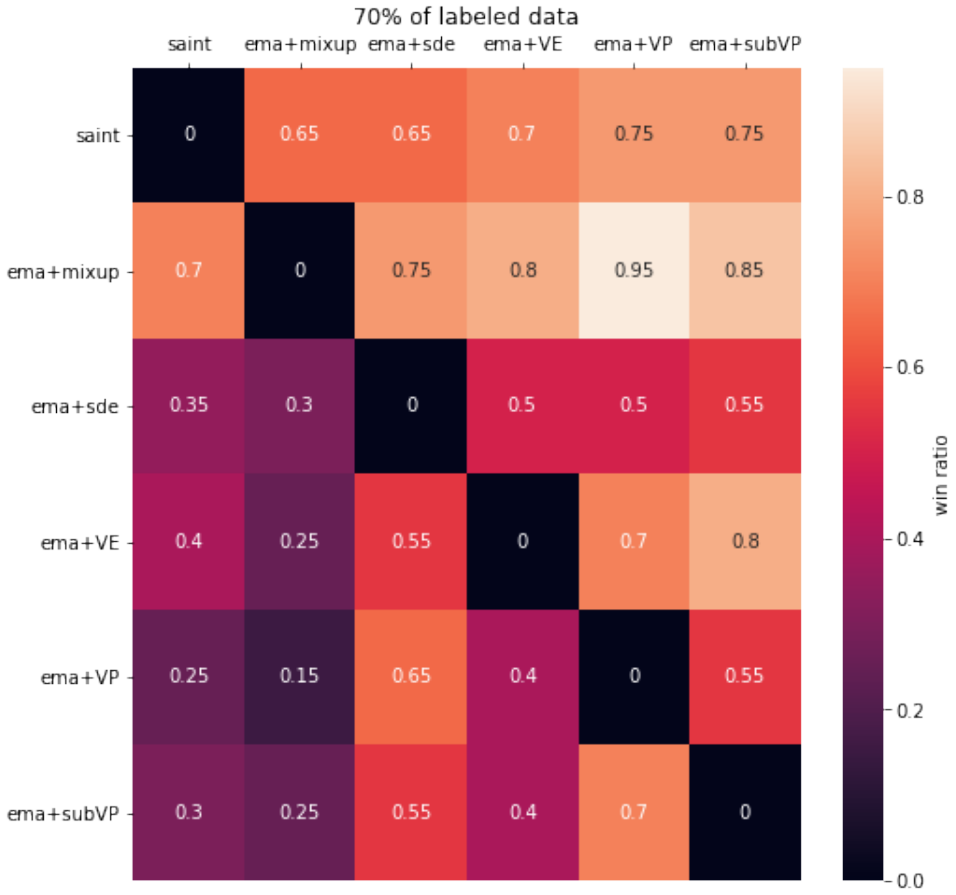


Figure 4.3 : Win matrix comparing pre-training methods with 70% of labeled data

Our experiments in the quasi-balance and unbalanced case have showcased the strength of our `MAC` architecture. Unfortunately the diffusion noise do not perform well even in the unbalanced case in regards to the win ratio. Ultimately the architecture armed with *ema+mixup* is the architecture that should be favored. Indeed we have experimentally demonstrated its superiority in comparison with our baseline and the diffusion noises.

4.4 Conclusion

Building on the self-supervised learning framework, we proposed `MAC` (Mean-teacher Architecture using Contrastive learning), a novel architecture that greatly improve classification accuracy under the win matrix primarily, when labeled data is scarce. Whenever the data is almost balanced we have observed that the baseline (`SAINT`) and our proposal model tends to coincide. All in all the mean-teacher architecture from semi-supervised learning along with `MixUp` corruption enable an improvement over the `SAINT` neural network.

5

An industrial application

During my work as a PhD candidate, I was continuously willing to bring research into the industry, which is naturally expected in a CIFRE* context. In particular, my main objective was to comprehend machine learning approaches around the attrition issue and develop novel churn prediction and analysis methods. Providing that churn is a key issue for my hist company, Brigad, I was collaborating with the Marketing, Finance and the IT departments to elaborate the first draft of the project.

5.1 Brigad a staffing and recruiting company

Brigad was created in November 2015 at the initiative of Jean Lebrument, Florent Malbranche and Alexandre Rovetto. Brigad is specialized in the development of a novel solution to connect freelancers and businesses which ranges from Hospitality to Healthcare. More precisely, Brigad is specialized on short term contract between both actors. On the long run, Brigad aims at being present on all the *just-in-time* jobs, which requires a large number of employees to be efficient. Nowadays, many industry and service areas are concerned, such as construction, healthcare, sales, education or hospitality.

The Brigad's online platform relies on algorithms that enable professionals to immediately match their request with the available *Talent* which is the most adequate employee for the mission. The service is available in France and in the top ten largest french cities along with the UK (London and Manchester). The service is accessible through the mobile application Brigad and from the web site Brigad.co.

Whenever a professional propose a mission, the platform, automatically build a set of features – the skills of the *Talent*, his localization with respect to the mission, if the *Talent* and the *Establishment* have already worked together or the grade they have given to each other – to return the most relevant candidates. The candidates are immediately informed of their mission. The connection between the professionals and the candidates is always setup solely through the mobile app ; Brigad's aim at finding someone available in less than thirty minutes.

In an effort to always make the platform more effective and in tune with its users it was important to pinpoint the Business on which we should listened to. Indeed by analyzing their behavior across time we were expecting to better know the companies that could stop using our services. So that we could intervene and stop or at least dampen the process by understanding how we could improve our product and solution to retain those customers.

5.2 Business Churn Prediction

5.2.1 Introduction

I would like to present in this chapter how my PhD work answered Brigad's needs. At Brigad the churn prediction is an important issue, as well as in many service company. The cost of retaining a customer is way lower than chasing a new one. For this project I was tasked to predict the potential Brigad churners in the near-future. So that the business expert can ponder and propose an offer to a customer that we wish to retain. I organized my work as follow :

1. Several meetings with the business experts were set to define the most suitable data.
2. I define with Brigad the *churn* variable, which is *industry-dependent*.
3. I explore the data, as detailed in Section 5.3.1
4. I preprocessed Brigad's data and fitted the Ensemble of Logistic regression, Random Forest and XGBoost. Rather than measuring the score with AUC, I relied on the LIFT (Section 5.3.2).
5. I performed feature importance with `SHAP` to pinpoint critical features for churn (Section 5.3.3).

5.2.2 Methods & Feature Engineering

5.2.2.a Introduction

At first, it is important to define the *churn* or *attrition* rate in the Brigad business context. The straightforward definition of attrition comes up in the *subscription based companies* e.g. the telecommunications, the banking, insurance, online music or online game industry. Indeed clients are allowed

*Conventions industrielles de formation par la recherche

to quit at the end of their contract. At Brigad, defining churn is somewhat more challenging. The customers are free to stop using the platform whenever they wish. Additionally, the offered missions on our platform depends on seasonality, localization or areas. And the depth of each individual history is relatively low given the young age of the Brigad company. As such, how to calculate the probability of churn, given a small volume of data that depends on seasonality, the mission localization's and the area ? How to foresee the non-use of a business in order to setup preventive actions and motivate them to carry on ordering staff on Brigad's platform ?

5.2.2.b Defining churn

Brigad being in a non-contractual setting, we analyze the behavior of groups of customers. Such an analysis is called a *cohort analysis*, where the cohorts are a group of clients that we track for a contiguous period of time. We decided to go for a monthly period of time. We consider the interval $[t-3,..,t]$ as our training set. To build the target, we look $[t+1, t+2, t+3]$ three months in the future to know if the clients has churned or not (i.e. the client hasn't used the platform).

5.3 Experiments

5.3.1 Description of the dataset

5.3.1.a Brigad's data

To begin with, I introduce the dataset that was extracted from the database. For this study, the following information were available :

- $n = 2841$ Businesses
- A churn variable $\mathbf{y} \in \{0, 1\}$ with an imbalance rate of 33.19%
- 3 numerical variables
  - **seniority.** The distance between the first and last order of the Business on the platform.
  - **total_minutes.** The accumulated number of minutes of missions ordered
  - **count_missions.** The total number of missions ordered
- 5 categorical variables
  - **count_unique_jobs.** The unique number of jobs ordered (a Business who would order only cooks would have its value set to one).
  - **distance_today_last.** The distance in days between the begin date and the last mission
  - **account_type.** Is the Business a Key account or a non-key account user ?
  - **nb_of_cancellations_less_than_4h_before_start.**
  - **nb_of_cancellations_more_than_4h_before_start. nb_of_no_shows.**

5.3.1.b Descriptive analysis

Given that the dataset is setup, the next step is to better understand the data. To do so, I first obtained the distribution of Businesses by their category, as shown in Figure 5.1. We observe that the Traditional Restaurants and Brasseries are the main customer's of Brigad. This Business type is followed by an almost uniform set of other categories, among which Care home, Event, Catering or Hospitality.

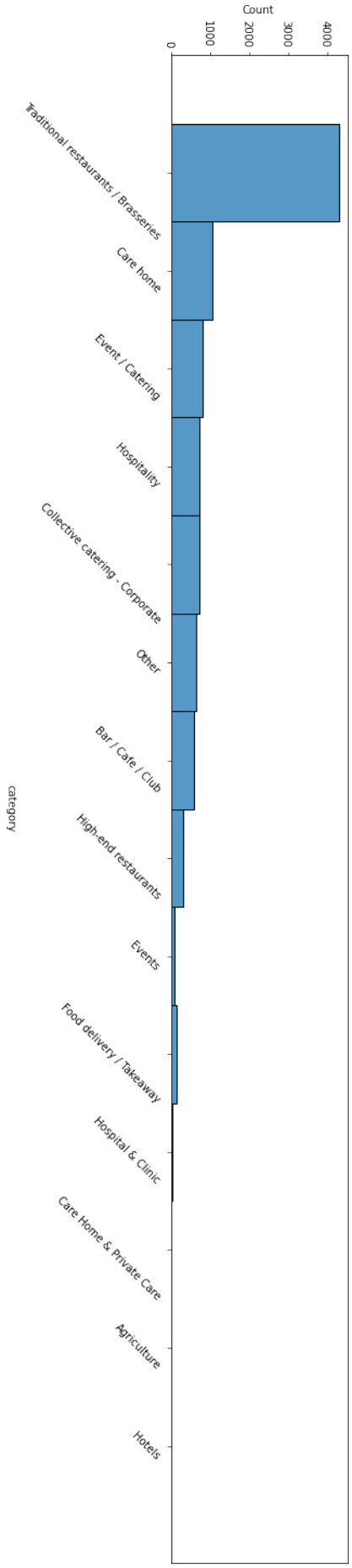


Figure 5.1 : Distribution of businesses types at Brigad

Brigad has two types of users, namely the *Key account* which is generally a large company, and the *regular user*, which is for instance a Brasserie. We note that the vast majority of users are regular, as shown in Figure 5.2.

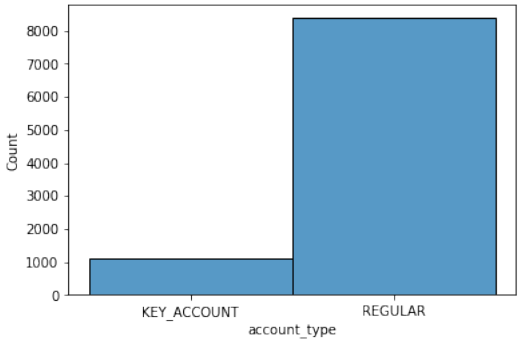
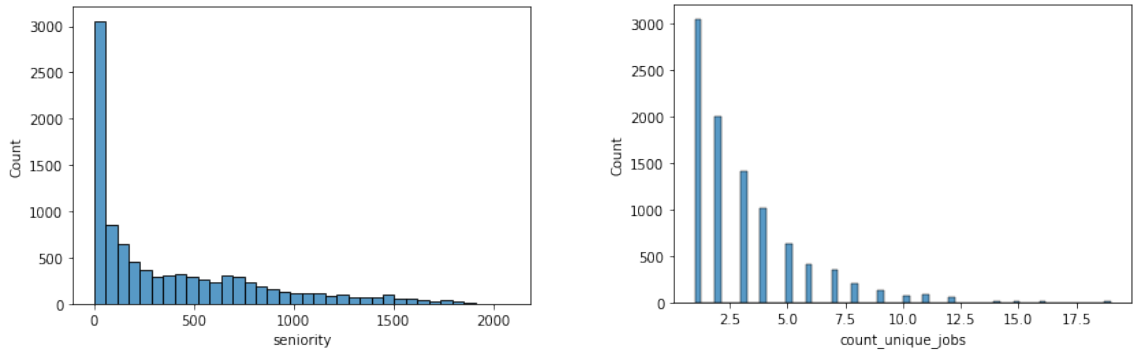


Figure 5.2 : Business category

Brigad also considers the *Seniority*, which is defined as the distance between the first order and the last order made by a Business. Its distribution is highly skewed to the left (Fig. 5.3, (a)). It shows that most customers are very young. This is an important issue, as young customers are usually expected to churn more easily. Then, the count unique jobs gives us the the unique type of jobs ordered (Fig. 5.3, (a)). For example, a value of one state could be a user that only order cooks. And one is the highest value on the histogram followed closely by two and three.



(a) Seniority **(b)** Count unique jobs

Figure 5.3 : The seniority and count_unique_jobs features

The total minutes ordered plot is highly intriguing. It is the cumulative sum of minutes ordered on the platform by the user. Graphically, we have an outlier close to zero. It signifies that most customers try the platform and grinds to a halt or churn (Fig. 5.4).

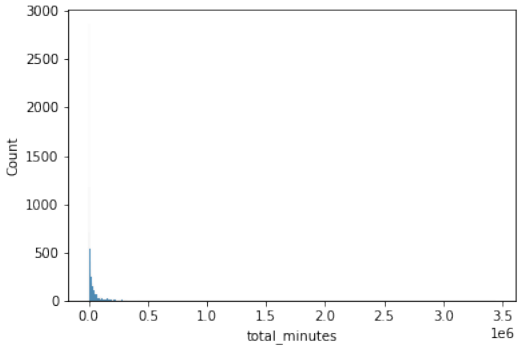


Figure 5.4 : Total minutes ordered

5.3.2 Preventing churn through the use of an Ensemble algorithm

In this chapter I focus on churn prediction. To start off, I motivate the need of an Ensemble approach to improve the churn prediction. Then, I plot the LIFT curve, which is typically used by business experts. And finally, I optimize the model and evaluate different thresholds to minimize the false positive rate.

5.3.2.a Dataset preprocessing

I represent the categorical features by a one-hot encoding vector, and the continuous variables are scaled in the range $[0, 1]$ by performing a min-max normalization. This is done in order to match the discrete variables, which falls into the range $\{0, 1\}$. No missing values have been observed. I split the data by a stratified k-fold with k set to 5. Stratified k-fold is the go-to method when the data is imbalanced.

5.3.2.b Methodology

I build on the churn pipeline from Figure 5.5, which correspond as our global strategy. After querying the data from our database, I perform feature engineering to build the the *seniority*, *total_minutes* and *count_unique_jobs* features. This generates a novel table, so-called *Churn Table*. Next the data is pre-processed following what I previously described in the first Chapters. The last pre-processing state is the filtering which is two-folds :

- On one side, I filter on *seniority*. At Brigad, some businesses churn after a couple of days. This subset is by definition not predictable by Machine Learning. Hence, I remove them from the dataset.
- On the other side, the Businesses behavior in Hospitality and in Healthcare are dramatically different. Hence, I filter on Hospitality.

Regarding the fitting of the model I used an Ensemble of Logistic regression, Random Forest and XGBoost, as I have demonstrated its strength in the first Chapters.

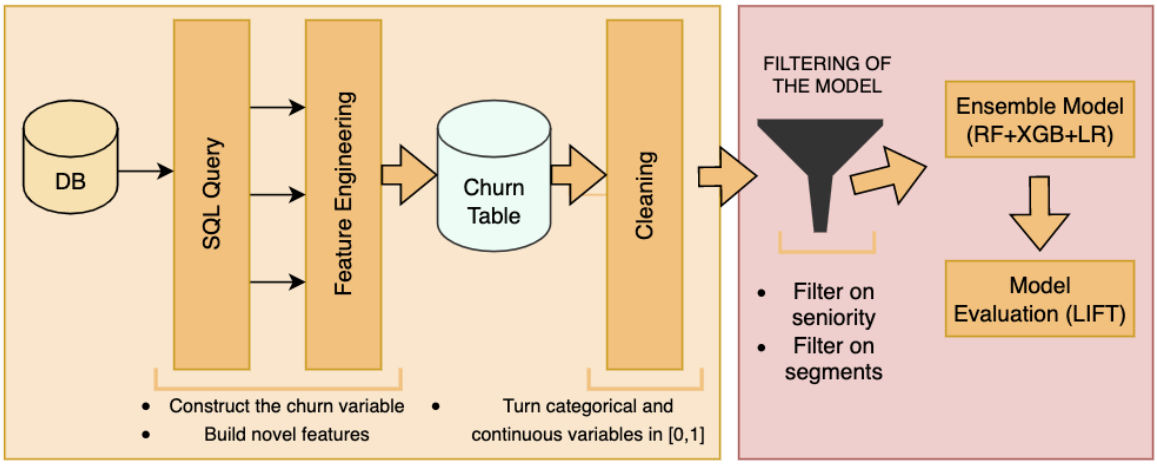


Figure 5.5 : The Churn pipeline is composed of two stages : - In orange we preprocess the data - In pink we fit, predict and evaluates the model.

5.3.2.c Model selection

Regarding the evaluation, I first benchmark the three models individually and the Ensemble of the three, in order to understand which one performs the best. The Accuracy is not directly applicable in the churn context. Indeed the dataset being imbalanced, the Accuracy could be high even though the minority class is not well identified. In Figure 5.6, we note a median $\widetilde{AUC}_{LR} = 0.656$, $\widetilde{AUC}_{XGB} = 0.662$, $\widetilde{AUC}_{RF} = 0.69$ and $\widetilde{AUC}_{Ens} = 0.67$.

Random Forest seems to be the go-to model in this specific Brigad data context.

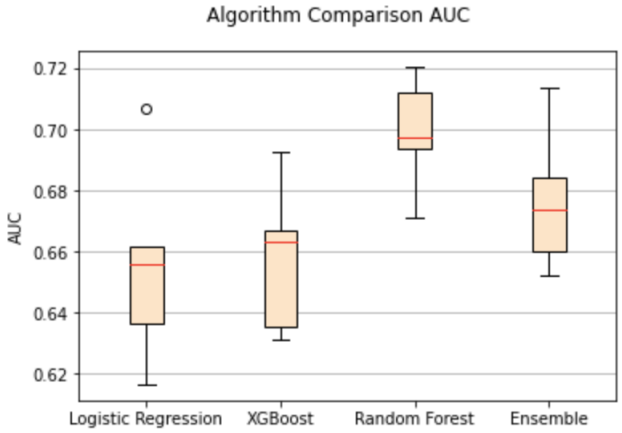


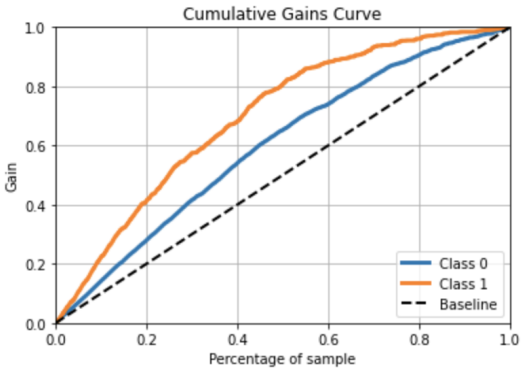
Figure 5.6 : Benchmark of our models for stratified 5-fold

Now let us consider a more *Business expert* metric, namely the Lift. A measure like the AUC is not directly applicable in the context of marketing because a company may not need to contact all prospects. Marketers usually contact only 10% to 20% of the prospects. That is why the lift is typically appropriate.

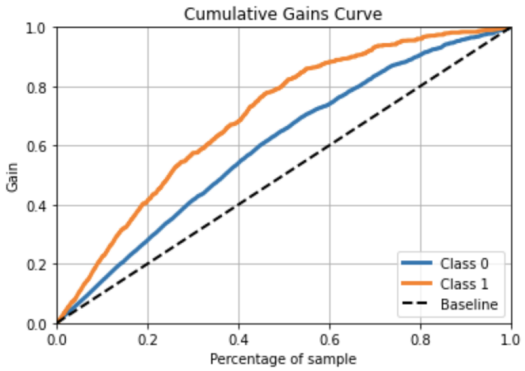
In Figure 5.7 and 5.8 we have on the x-axis the sample size in percentage which ranges from zero to $2,841$. Similarly on the y-axis we have either the percentage of churners, given that 942 businesses are churners, in orange, and the percentage of non-churners in blue. For example, on the plot we can see that for $20\% \times 2,841 \approx 568$ cases the orange curve predicts $0.4 \times 942 \approx 376$ churners. To calculate the lift, the baseline is required which is $20\% \times 942 = 188$. As such the lift at 20% is

$$Lift@20\% = \frac{376}{188} = 2 \tag{5.1}$$

In other words, the approach performs 2 times better than a random guess with the Random Forest technique.

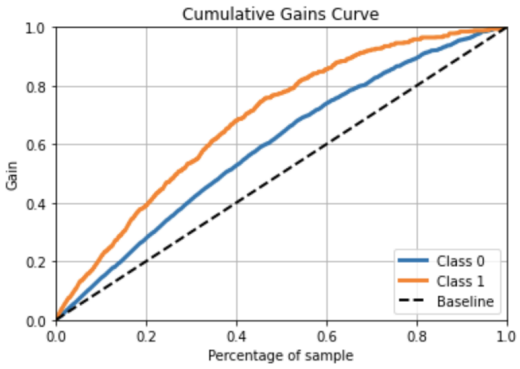


(a) LIFT for RF

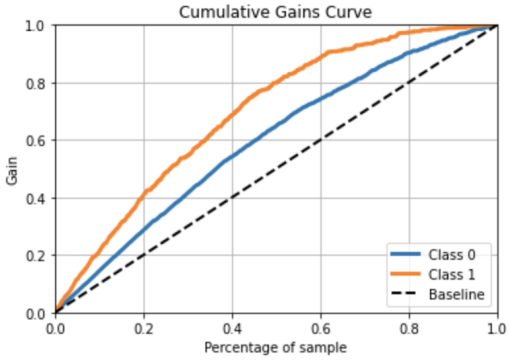


(b) LIFT for LR

Figure 5.7 : Lift of RF and LR



(a) LIFT for XGB



(b) LIFT for Ensemble

Figure 5.8 : Lift of XGB and Ensemble

5.3.2.d Churn prediction

I proceed with the *Random Forest* model, we obtain the confusion matrix given in Table 5.1. One important metric of this table is the number of False Positives, that is the people that Brigad will contact to prevent them from churning, even though there were not planing to churn. This is the group that we want to keep low, as contacting someone for no reason is not suitable and might in certain cases trigger a churn.

		Predicted	
		Churner	Non-churner
Actual	Churner	476	466
	Non-churner	261	1637

Table 5.1 : Confusion matrix

Using a *grid search* strategy, I fine-tuned the model. The Table 5.2 provides the updated confusion matrix. It can be noted that the number of False Positive has decreased, from 261 to 228 individuals. However, the False Negative rate increases, from 466 to 489 customers.

The best parameters found with the fine-tuning strategy are,

- $max\_depth = 90$
- $max\_features = 3$
- $min\_samples\_leaf = 5$
- $min\_samples\_split = 2$
- $n\_estimators = 200$

		Predicted	
		Churner	Non-churner
Actual	Churner	453	489
	Non-churner	228	1670

Table 5.2 : Confusion matrix with optimal parameters

A last important step is to obtain the optimal threshold for the churner detection. If we are interested in maintaining low the number of False Positive, while keeping a reasonable Precision, the threshold should be preferentially set to $0,80$, as can be seen from the Recall and Precision curves in Figure 5.9.

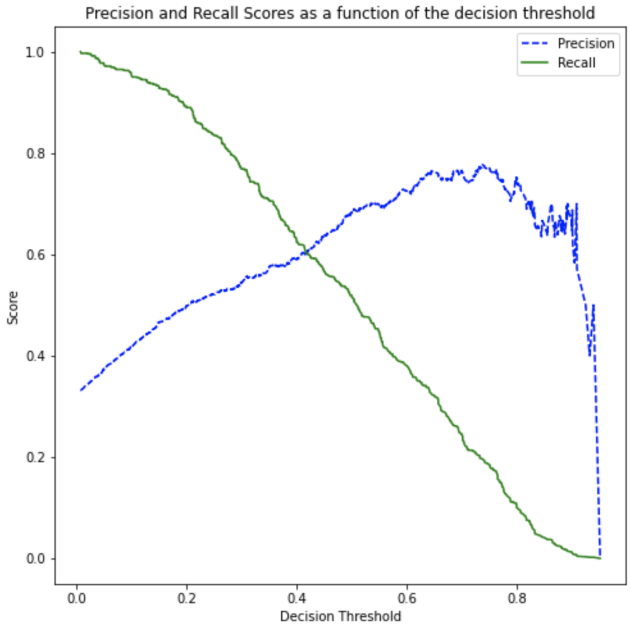


Figure 5.9 : Decision threshold

If we set the threshold at 0.80, we observe a significant decrease in False Positive. We went from 228 churners to 34, which is more suitable for business experts

		Predicted	
		Churner	Non-churner
Actual	Churner	842	100
	Non-churner	34	1864

Table 5.3 : Confusion matrix with a threshold of 0.80

Setting the threshold at 0.70 helps to improve the number of False Negative (from 100 to 71). However, the number of False Positives increases. As such, I prefer a threshold of 0.80.

		Predicted	
		Churner	Non-churner
Actual	Churner	710	71
	Non-churner	71	1827

Table 5.4 : Confusion matrix with a threshold of 0.70

Providing the constraints from the business expert, we decided with Brigad to choose a threshold of roughly 0.80. The main reason is that Brigad's focus is to not contact customers which are not willing to churn. It could have an opposite effect, and trigger a churn reaction.

5.3.3 Interpretation with SHAP

After focusing on churn prediction, I will now try to interpret the model with the techniques available in the python package `SHAP` (*SHapley Additive exPlanations*)[†] `SHAP` is a cooperative game theoretic approach to explain the output of any machine learning model (Ribeiro et al., 2016 ; Štrumbelj and Kononenko, 2014). `SHAP` aims at explaining the outcome of an ML model $f(\cdot)$ by looking at coalitions. For a specific feature we look at all the possible combinations that can be built with the other features in a set S. We then proceed to calculate the contribution of the specific feature a within p features :

$$\phi_a = \frac{1}{p} \sum_S \begin{pmatrix} p-1 \\ |S-1| \end{pmatrix}^{-1} \delta_a^{(S)}$$

Where the $\delta_a^{(S)}$ measure the contribution of the i-th variable to the set S with $S_{-i} = S \setminus \{x_i\}$

$$\delta_i^{(S)} = f_S(S) - f_{S_{-i}}(S_{-i})$$

Unfortunately this model suffers from a combinatorial explosion as the number of feature increases. Several extensions have been proposed e.g. Kernel `SHAP` Lundberg and Lee (2017). We will proceed with Kernel `SHAP`

5.3.3.a The importance plot

An importance plot displays the average SHAP value for every single features. Features with high large absolute Shapley values are important.

[†] `SHAP` package access : `https://shap.readthedocs.io/en/latest/index.html`

$$I_j = \frac{1}{n}\sum_{i=1}^{n}\left|\phi_j^{(i)}\right|$$

From the Figure 5.10, we first note that *dist_today_last* is the most important feature. This intuitively makes sense as a large value means that the Business haven't ordered a mission for quite some times. We observe next *seniority* which is how old the customer is with our service. This feature has also a strong effect on churn behavior. Subsequently, the more often a Talent would cancel its mission in last minutes the larger the impact on churn. Also the total number of minutes ordered by the business have an impact along with the total unique number of missions. Overall, to have a deeper understanding on how positive or negative on churn are those features, we must proceed to analyze the summary plot.

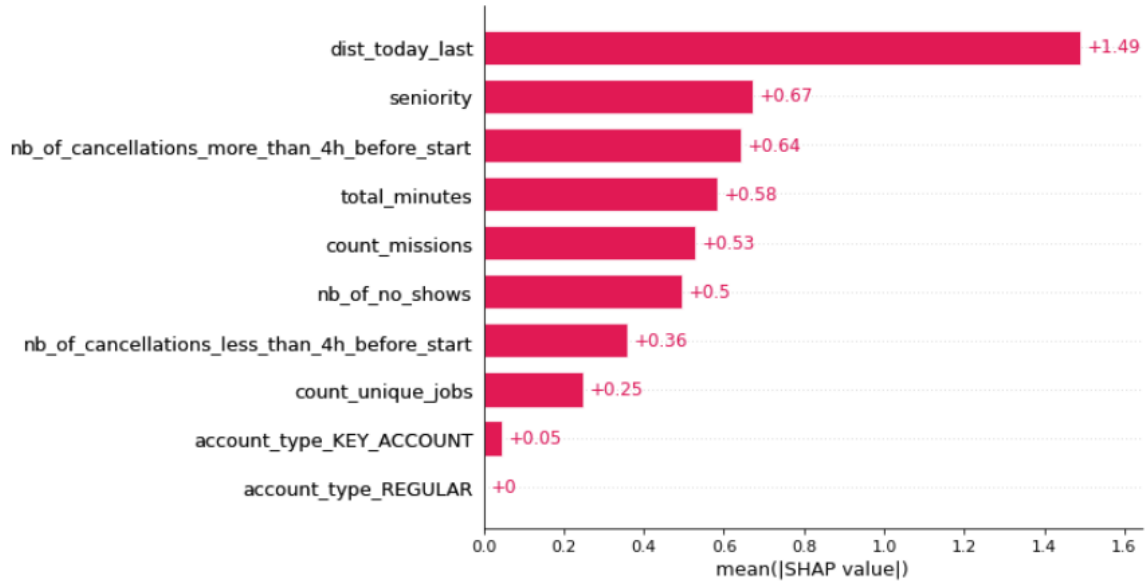


Figure 5.10 : Importance plot

5.3.3.b The summary plot

The summary plot offers somewhat more information.A high feature value signifies that this is related to the "churn" class and a low-value to a "non-churn". We observe a meaningful information on *dist_today_last*, it states that the more delay there is between two missions the more likely the business will churn. In contrast, a high number of *count_mission* has a negative impact on churn, that is if a Business order plenty of missions it will be less likely to churn.

Yet, we observe that the number of cancellations of more than 4h have an undesired effect, if the Talents that the business ordered cancel often, it has a negative impact on churn which is quite surprising. We observe the same unusual effect on *seniority*, a business is more likely to churn if he is a long-term user of the app which contradicts what the data analyst team have observed. We could also discuss about the variable *count_unique_jobs* which states that a business is more likely to churn if it orders a diverse set of jobs. For example it would mean that a Business that orders only waiters is less likely to churn.

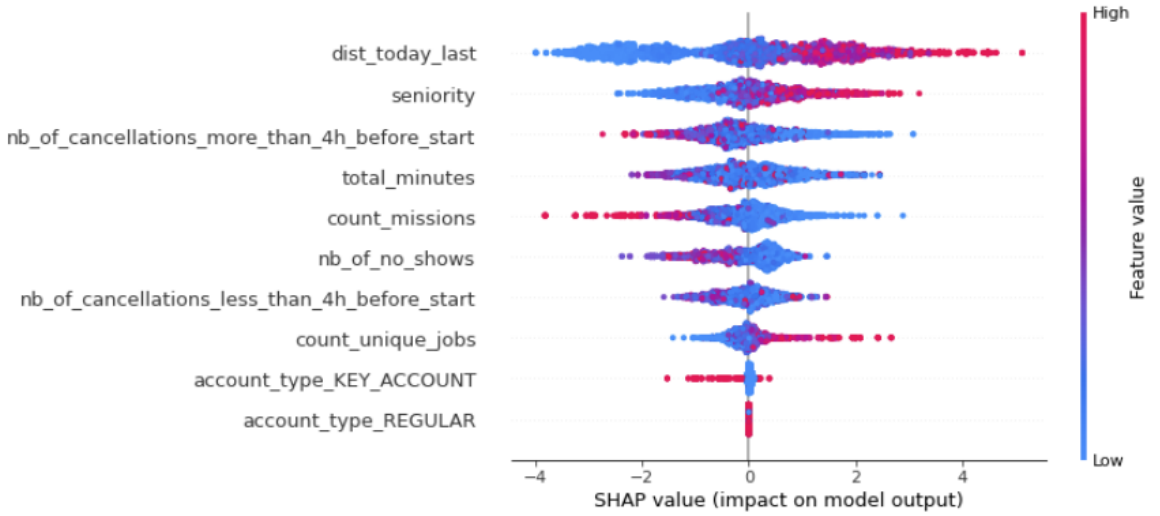


Figure 5.11 : Summary plot for our model

This sub-chapter highlights the value of the importance and summary plots proposed by `SHAP`. They are of a great interest to share our results with the business experts.

5.4 Conclusion

This project was the opportunity of concluding my CIFRE PhD with a real business case in the platform industry. We have proposed a complete pipeline to solve the churn prediction problem. We first had to define churn from a business standpoint, which is defined as the consecutive three months of a business not using the platform. Equipped with the target variable we built our feature set with the business team which ranges from seniority to the number of cancellation that the business has to endure. Subsequently we carried on with performing stratified k-fold on four ML algorithms Logistic Regression (`LR`), XGBoost (`XGB`), Random Forest (`RF`) and Ensemble (`Ens`).

Our experiments have demonstrated that `RF` obtained the greatest median AUC as a consequence it should be the go-to algorithm. Finally we fine-tuned our model by consecutively using grid search and threshold optimization. Above all threshold optimization is critic in the platform industry. Indeed contacting someone that wasn't willing to churn might trigger an opposite effect. Lastly we proposed to better understand `RF` output through `SHAP` with an importance plot and a summary plot.

6

Conclusion and perspectives

In this thesis we have addressed the problem of tabular prediction in the churn paradigm. We have resorted to multiple tools and reached several conclusions on the attrition issue.

First, we looked at the state of the art of the domain which could sequentially be described as the sampling, the model fitting and the evaluation stages. We have observed that the sampling step generally correspond to one of the three tasks among oversampling, undersampling or hybrid sampling. The model fitting revolve around supervised learning along with an evaluation step where we have emphasized the inefficiency of the Accuracy as a metric in the imbalance scenario and in contrast the superiority of the AUC metric. Providing the state of the art, we have proceeded to compare various machine learning techniques on sixteen public churn datasets. At first we compared our models without any resampling methods. It appeared that `LR`, `RF`, `XGBoost` and `GEV-NN` techniques were the best performers for this task on the considered churn benchmark datasets. Secondly, we performed similar experiments with a re-sampling stage, which confirmed the prominent AUC for `LR`, `RF` and `XGBoost`. Additionally, we observed that the resampling methods do not have a *global* effect – i.e., an improvement on any data for any ML technique – but rather a local effect. In other words, they help to improve the prediction AUC but only in some specific cases and combination of techniques/data. Ultimately, we proposed to visualize the datasets and algorithms on a plane by projection with Correspondence Analysis. (Chapter 1)

Secondly, we have demonstrated in the first chapter the strength of the triplet `LR`, `RF` and `XGBoost` without any re-sampling to solve the customer churn problem. Building on those models, we propose to construct an ensemble that will harness the best of each of them. In our experiments, we compared the best non-ensemble pipeline in AUC with respect to our triplet and we demonstrated its superiority in median AUC. Subsequently, we proposed to extend even further our pipeline by adding one additional preprocessing step which is based on a *deep unsupervised neural network*. Indeed, our assumption is that the customers churn for various reasons and we would like to capture them by using clustering. Hence, we fit our triplet on every single clusters and merge their predictions by a novel soft-voting technique that we proposed in this paper. Additionally, we offer a qualitative evaluation of churn profiling by leveraging the feature importance of the `Random Forest`. The customer's being divided into clusters, we can fit one `Random Forest` by cluster and analyze their outcomes. (Chapter 2)

Thirdly, we looked at the tabular prediction problem from a different perspective : the self-supervised learning paradigm. This approach have been well explored in the image and Natural Language Processing paradigm. Yet, it remains nascent in the tabular domain. As a consequence we proposed a novel architecture : *Mean-teacher Architecture using Contrastive learning* (MAC) which leverage techniques from `SimCLR` along with deep semi-supervised learning and diffusion processes. Overall the architecture is built on top of the `SAINT` neural network with a mean-teacher on the heads

of the two branches with five types of noises, either a `MixUp` or one of the four studied diffusion processes. We performed our experiments on 20 datasets from the *OpenML-CC18* benchmark that we constrained to be solely within the binary classification setting. Building on the works on `SCARF`, we chose the *win matrix* to compare our strategies based on the AUC results. Our experiments have showcased the superiority of our architecture with the `MixUp` noise, with respect to our baseline (`SAINT` architecture). (Chapter 3)

Fourth, this thesis was conducted in parallel within Université Paris Cité and at the `Brigad` company. As such the Chapter 4 is dedicated to one internal project within `Brigad`. In particular, `Brigad` aims at matching Talents and Businesses for short term missions within the hospitality or healthcare industry. In this setting, predicting the churn is a main concern for the company and this thesis was the opportunity of connecting academic research and the industry. We decided to build a customer churn prediction model on the business side. The first stage of any ML problem is to define the list of features that our business team think could be coherent to our problem. Subsequently the data was retrieved using SQL from the *Databricks* platform and it was analyzed through basic descriptive statistics plots. Next the data was split by stratified k-fold to enable the comparison of several supervised ML models. In the end, the `Random Forest` was identified as the best fit to our `Brigad` data in median AUC. It follows that we fine-tuned our model by grid search and threshold optimization in order to get the lowest number of false positives. Finally, we proposed some interpretations using the importance plot and summary plot proposed by the `SHAP` techniques (*SHapley Additive exPlanations*). (Chapter 4)

The studies presented in this thesis motivate further investigations. One domain that has a key importance is the causal inference applied to churn which is termed the *uplift model*. The uplift model enables the user to measure the impact of a treatment (e.g. offering a coupon) against churn which is of great importance in the industry. Indeed, detecting churn is not sufficient for a company, and proactive actions must be performed in order to reduce it. Additionally, we could focus solely on the extreme imbalance case which could be solved by using deep learning for anomaly detection. We have briefly talked about this topic in the first chapter. One last domain would be to incorporate time into our problem. Churn is not a static phenomena, as customer behavior varies across time. As an example, the survival analysis aims at calculating the probability of churn in a predefined time horizon. We could propose novel models in this area of research in combination with recent deep learning architectures.

A

Appendix

A.1 Appendix

A.1.1 Datasets complementary information

K2009 (*KDD-Cup 2009 small*) This dataset was proposed in the context of the *KDD Cup 2009 : Churn relationship prediction* and originates from the French telecommunication company *Orange* in order to predict the switch of provider (Guyon et al., 2009). #Dummified Features : 1039.

KKBox's (*WSDM CUP 2018*) This churn dataset was proposed for the 11^{th} ACM International Conference on Web Search and Data Mining (WSDM 2018) and originates from the KKbox Taiwanese music streaming company. The proposed challenge is to predict if a subscriber will churn as soon as the subscription expires (Chen et al., 2018). #Dummified Features : 56.

UCI (*MLC Churn*) This dataset is similar to the *Telecom SingTel*, *CrowdAnalytix* and *UCI* datasets. *MLC Churn* is proposed in the **R** package *modeldata* (Vafeiadis et al., 2015). #Dummified Features : 21.

HR (*IBM Employee Attrition*) This dataset originates from IBM HR and includes $1,470$ records of individuals who left the company or not. It is an artificial dataset created by IBM data scientists from Watson analytics, and has been proposed to uncover the factors that lead to employee attrition (McKinley Stacker, 2015). #Dummified Features : 86.

TelE (*Telco-Europa*) This dataset corresponds to the real data of a small telecommunications company in Oceania that has only 14 months of historical data. It is found in online churn prediction tutorials. #Dummified Features : 26.

News (*Newspaper*) This datasets contains information on Californian newspaper subscribers and an attrition variable. It is found in online churn prediction tutorials. Other newspaper private datasets were analyzed in previous studies ; see (Burez and Van den Poel, 2009 ; Coussement and Van den Poel, 2008 ; Coussement et al., 2010). #Dummified Features : 307.

Bank This data set contains details of a bank's customers and their departure. It is found in online churn prediction tutorials. #Dummified Features : 16.

TelC (*IBM Telco Churn*) This dataset is proposed by IBM and is used in an online tutorial to train a model that predicts if a customer is likely to leave the telecom provider. #Dummified Features : 34.

C2C (*Cell2Cell*) The data sets is provided by the Teradata Center for CRM (Customer Relationship Management). Data were provided by the Cell2Cell company, which is one of the largest wireless company in the USA (Kim, 2006). #Dummified Features : 75.

Member (*Membership Woes*) This dataset is cited in online tutorials. #Dummified Features : 26.

SATO (*South-asian*) This dataset is provided by a South Asian Telecom Operator, also called SATO. Data were collected between August 2015 and September 2015 (Ahmed et al., 2018a). #Dummified Features : 29.

DSN (*DSN-telecom 'Nigerian Telecom'*) This dataset has been proposed in the context of the *DSN Telecoms Churn Prediction 2018* challenge, which is one of the pre-qualification to the *2018 Data Science Nigeria hackathon*. #Dummified Features : 32.

Fraud (*Credit Card Fraud Detection*) The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. It is an anomaly detection dataset.

Thyroid (*Thyroid Disease*) This data are from the Garavan Institute. The problem is to determine whether a patient referred to the clinic is hypothyroid. 92 percent of the patients are not hyperthyroid

in this dataset which contains 7,200 instances. It is an anomaly detection dataset.

Campaign (*Bank Marketing*) The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. It is an anomaly detection dataset.

A.1.2 Python package and functions

All experiments in this survey were performed on public datasets using freely available Python packages. Hence, results are entirely reproducible. Table A.1 summarizes information on packages, functions and parameters used for our experiments. It also provides links to the online description of each function.

Table A.1 : Packages, functions and parameters summary for the churn pipeline

	Approach	*Function*	*parameters*	*version*	*online details*
		Sampling			
over.	SMOTE	`SMOTE`	*default*	0.7.0	imblearn.over_sampling.SMOTE
	ADASYN	`ADASYN`	*'not minority'*	0.7.0	imblearn.over_sampling.ADASYN
under.	Tomek links	`TomekLinks`	*default*	0.7.0	imblearn.under_sampling.TomekLinks.html
	NCR	`NeighbourhoodCleaningRule`	*default*	0.7.0	imblearn.under_sampling.NeighbourhoodCleaningRule
hybrid	SMOTE+Random	`SMOTE` `RandomUnderSampler`	*default*	0.7.0	imblearn.over_sampling.SMOTE imblearn.under_sampling.RandomUnderSampler
	SMOTE+Tomek links	`SMOTETomek`	*default*	0.7.0	imblearn.combine.SMOTETomek
	SMOTE+NCR	`SMOTE` `NeighbourhoodCleaningRule`	SMOTE : *default* NCR : *'minority'*	0.7.0	imblearn.over_sampling.SMOTE imblearn.under_sampling.NeighbourhoodCleaningRule
		Model Fitting			
Supervised	k-nearest neighbors	`KNeighborsClassifiere`	*default*	0.23.2	neighbors.KNeighborsClassifier
	Naïves Bayes	`GaussianNB`	*default*	0.23.2	sklearn.naive_bayes.GaussianNB
	Logistic Regression	`LogisticRegression`	*default*	0.23.2	sklearn.linear_model.LogisticRegression
	Support Vector Machine	`SVC`	*default*	0.23.2	svm.SVC
	Decision Tree	`DecisionTreeClassifier`	*default*	0.23.2	sklearn.tree.DecisionTreeClassifier
	Feed Forward Neural Network	`NN`	*default*	–	Neural-Network-Churn-Prediction
	Generalize Extreme Value-NN	`GEV-NN`	*default*	–	GEV-NN
Semi-supervised	Isolation Forest	`IsolationForest`	*default*	0.23.2	sklearn.ensemble.IsolationForest
	Deep AD with Deviation Networks	`DevNet`	*default*	–	deviation-network
Ensemble Supervised	Random Forest	`RandomForestClassifier`	*default*	0.23.2	sklearn.ensemble.RandomForestClassifier
	XGBoost	`XGBClassifier`	*default*	1.0.2	xgboost.readthedocs.io
		Evaluation			
Strategy	Cross Validation	`train_test_split`	*default*	0.23.2	sklearn.model_selection.train_test_split
	K-fold validation	`KFold`	K=5	0.23.2	sklearn.model_selection.KFold
	Stratified k-fold validation	`StratifiedKFold`	K=5	0.23.2	sklearn.model_selection.StratifiedKFold
Metric	Top-lift	`plot_lift_curve`	*default*	0.3.7	rasbt.github.io – lift_score
	F1-score	`f1_score`	*default*	0.23.2	sklearn.metrics.f1_score
	AUC	`roc_auc_score`	*default*	0.23.2	sklearn.metrics.roc_auc_score

Table of contents

Bibliographie

M. F. Abdillah, J. Nasri, and A. Aditsania. Using deep learning to predict customer churn in a mobile telecomunication network. *eProceedings of Engineering*, 3(2), 2016.

S. Affeldt, L. Labiod, and M. Nadif. Spectral clustering via ensemble deep autoencoder learning (SC-EDAE). *Pattern Recognition*, 108 :107522, 2020.

M. Ahmed, H. Afzal, I. Siddiqi, M. F. Amjad, and K. Khurshid. Exploring nested ensemble learners using overproduction and choose approach for churn prediction in telecom industry. *Neural Computing and Applications*, 8, 2018a. ISSN 09410643. doi : 10.1007/s00521-018-3678-8.

M. Ahmed, I. Siddiqi, H. Afzal, and B. Khan. MCS : Multiple classifier system to predict the churners in the telecom industry. *2017 Intelligent Systems Conference, IntelliSys 2017*, 2018-January (September) :678–683, 2018b. doi : 10.1109/IntelliSys.2017.8324367.

R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *European conference on machine learning*, pages 39–50. Springer, 2004.

S. Alam, S. K. Sonbhadra, S. Agarwal, and P. N. Nagabhushan. One-class support vector classifiers : A survey. *Knowl. Based Syst.*, 196 :105754, 2020.

M. Alkhayrat, M. Aljnidi, and K. Aljoumaa. A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA. *Journal of Big Data*, 7(1) :1–23, 2020.

S. Alrowili and K. Vijay-Shanker. Arabictransformer : Efficient large arabic language model with funnel transformer and electra objective. In *Findings of the Association for Computational Linguistics : EMNLP 2021*, pages 1255–1261, 2021.

B. Amnueypornsakul, S. Bhat, and P. Chinprutthiwong. Predicting Attrition Along the Way : The UIUC Model. *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, October :55–59, 2015. doi : 10.3115/v1/w14-4110.

E. W. Anderson and M. W. Sullivan. The antecedents and consequences of customer satisfaction for firms. *Marketing science*, 12(2) :125–143, 1993.

S. O. Arık and T. Pfister. Tabnet : Attentive interpretable tabular learning. In *AAAI*, volume 35, pages 6679–6687, 2021.

A. D. Athanassopoulos. Customer satisfaction cues to support market segmentation and explain switching behavior. *Journal of business research*, 47(3) :191–207, 2000.

D. Bahri, H. Jiang, Y. Tay, and D. Metzler. Scarf : Self-supervised contrastive learning using random feature corruption. *arXiv preprint arXiv:2106.15147*, 2021.

G. E. Batista, A. L. Bazzan, M. C. Monard, et al. Balancing training data for automated annotation of keywords : a case study. In *WOB*, pages 10–18, 2003.

G. E. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1) :20–29, 2004.

R. Batuwita and V. Palade. Efficient resampling methods for training support vector machines with imbalanced datasets. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2010.

M. Bécue-Bertaut and J. Pagès. Multiple factor analysis and clustering of a mixture of quantitative, categorical and frequency data. *Computational Statistics & Data Analysis*, 52(6) :3255–3268, 2008.

A. A. Benczúr, K. Csalogány, L. Lukács, and D. Siklósi. Semi-supervised learning : A comparative study for web spam and telephone user churn. In *In Graph Labeling Workshop in conjunction with ECML/PKDD*. Citeseer, 2007.

Y. Bengio, L. Yao, G. Alain, and P. Vincent. Generalized denoising auto-encoders as generative models. *Advances in neural information processing systems*, 26, 2013.

J. Bennett, S. Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. Citeseer, 2007.

D. F. Benoit and D. Van den Poel. Improving customer retention in financial services using kinship network information. *Expert Systems with Applications*, 39(13) :11435–11442, 2012.

P. Bermejo, J. A. Gámez, and J. M. Puerta. Improving the performance of naive bayes multinomial in e-mail foldering by introducing distribution-based balance of datasets. *Expert Systems with Applications*, 38(3) :2072–2080, 2011.

D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch : A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

C. Bhattacharya. When customers are members : Customer retention in paid membership contexts. *Journal of the academy of marketing science*, 26(1) :31–44, 1998.

B. Bischl, G. Casalicchio, M. Feurer, F. Hutter, M. Lang, R. G. Mantovani, J. N. van Rijn, and J. Vanschoren. Openml benchmarking suites. *arXiv preprint arXiv:1708.03731*, 2017.

J. Błaszczyński and J. Stefanowski. Local data characteristics in learning classifiers from imbalanced data. In *Advances in Data Analysis with Computational Intelligence Methods*, pages 51–85. Springer, 2018.

R. N. Bolton. A dynamic model of the duration of the customer's relationship with a continuous service provider : The role of satisfaction. *Marketing science*, 17(1) :45–65, 1998.

R. N. Bolton and T. M. Bronkhorst. The relationship between customer complaints to the firm and subsequent exit behavior. *ACR North American Advances*, 22 :94–100, 1995.

P. Branco, L. Torgo, and R. P. Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys*, 49(2) :1–50, 2016. ISSN 15577341. doi : 10.1145/2907070.

L. Breiman. Bagging predictors. *Machine learning*, 24(2) :123–140, 1996.

L. Breiman. Random forests. *Machine learning*, 45(1) :5–32, 2001.

L. Breiman and P. Spector. Submodel selection and evaluation in regression. the x-random case. *International statistical review/revue internationale de Statistique*, 60(3) :291–319, 1992.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees, belmont, california : Wadsworth, 1984.

M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof : Identifying density-based local outliers. *SIGMOD Rec.*, 29(2) :93104, may 2000. doi : 10.1145/335191.335388.

G. Brown. An information theoretic perspective on multiple classifier systems. In *International Workshop on Multiple Classifier Systems*, pages 344–353. Springer, 2009.

J. Burez and D. Van den Poel. Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 36(3) :4626–4636, 2009.

P. Burman. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3) :503–514, 1989. ISSN 00063444. URL http://www.jstor.org/stable/2336116.

C. S. Burrus, J. Barreto, and I. W. Selesnick. Iterative reweighted least-squares design of fir filters. *IEEE Transactions on Signal Processing*, 42(11) :2926–2936, 1994.

G. G. Cabral and A. Oliveira. One-class classification for heart disease diagnosis. *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2551–2556, 2014.

M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

F. Castanedo, G. Valverde, J. Zaratiegui, and A. Vazquez. Using deep learning to predict customer churn in a mobile telecommunication network, 2014.

J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez. A comprehensive survey on support vector machine classification : Applications, challenges and trends. *Neurocomputing*, 408 : 189–215, 2020.

C. C. H. Chan. Intelligent value-based customer segmentation method for campaign management : A case study of automobile retailer. *Expert systems with applications*, 34(4) :2754–2762, 2008.

V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection : A survey. *ACM Comput. Surv.*, 41 (3), jul 2009. ISSN 0360-0300. doi : 10.1145/1541880.1541882.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote : synthetic minority oversampling technique. *Journal of artificial intelligence research*, 16 :321–357, 2002.

C. Chen, A. Liaw, L. Breiman, et al. Using random forest to learn imbalanced data. *University of California, Berkeley*, 110(1-12) :24, 2004.

T. Chen and C. Guestrin. Xgboost : A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Y. Chen, X. Xie, S.-D. Lin, and A. Chiu. Wsdm cup 2018 : Music recommendation and churn prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 8–9. ACM, 2018.

A. Chowdhury and J. Alspector. Data duplication : an imbalance problem ? In *ICML2003 Workshop on Learning from Imbalanced Data Sets (II), Washington, DC*, 2003.

M. Clemente, V. Giner-Bosch, and S. San Matías. Assessing classification methods for churn prediction by composite indicators. *Manuscript, Dept. of Applied Statistics, OR & Quality, UniversitatPolitècnica de València, Camino de Vera s/n*, 46022, 2010.

K. Cooray. Generalized gumbel distribution. *Journal of Applied Statistics*, 37(1) :171–179, 2010.

K. Coussement and K. W. De Bock. Customer churn prediction in the online gambling industry : The beneficial effect of ensemble learning. *Journal of Business Research*, 66(9) :1629–1636, 2013.

K. Coussement and D. Van den Poel. Churn prediction in subscription services : An application of support vector machines while comparing two parameter-selection techniques. *Expert systems with applications*, 34(1) :313–327, 2008.

K. Coussement, D. F. Benoit, and D. Van den Poel. Improved marketing decision making in a customer churn prediction context using generalized additive models. *Expert Systems with Applications*, 37 (3) :2132–2143, 2010.

J. J. Cronin Jr and S. A. Taylor. Measuring service quality : a reexamination and extension. *Journal of marketing*, 56(3) :55–68, 1992.

P. Cunningham and J. Carney. Diversity versus quality in classification ensembles based on feature selection. In *European Conference on Machine Learning*, pages 109–116. Springer, 2000.

S. Darabi, S. Fazeli, A. Pazoki, S. Sankararaman, and M. Sarrafzadeh. Contrastive mixup : Self-and semi-supervised learning for tabular domain. *arXiv preprint arXiv:2108.12296*, 2021.

A. De Caigny, K. Coussement, and K. W. De Bock. A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European Journal of Operational Research*, 269(2) :760–772, 2018. ISSN 0377-2217. doi : https://doi.org/10.1016/j.ejor.2018.02.009.

A. De Caigny, K. Coussement, and K. W. De Bock. A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European Journal of Operational Research*, 269(2) :760–772, 2018.

A. De Caigny, K. Coussement, K. W. De Bock, and S. Lessmann. Incorporating textual information in customer churn prediction models based on a convolutional neural network. *International Journal of Forecasting*, 36(4) :1563–1578, 2020. ISSN 0169-2070. doi : https://doi.org/10.1016/j.ijforecast.2019.03.029.

J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7 :1–30, 2006.

M. Denil and T. Trappenberg. Overlap versus imbalance. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6085 LNAI :220–231, 2010. ISSN 03029743. doi : 10.1007/978-3-642-13059-5_22.

J.-C. Deville and Y. Tillé. Efficient balanced sampling : the cube method. *Biometrika*, 91(4) :893–912, 2004.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

A. Dingli, V. Marmara, and N. S. Fournier. Comparison of deep learning algorithms to predict customer churn within a local retail industry. *International journal of machine learning and computing*, 7(5) :128–132, 2017.

C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

P. Domingos. Metacost : A general method for making classifiers cost-sensitive. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 155–164, 1999.

C. Drummond, R. C. Holte, et al. C4. 5, class imbalance, and cost sensitivity : why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11, pages 1–8. Citeseer, 2003.

H. Dubey and V. Pudi. Class based weighted K-Nearest neighbor over imbalance dataset. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7819 LNAI(PART 2) :305–316, 2013. ISSN 03029743. doi : 10.1007/978-3-642-37456-2_26.

V. Effendy, Z. A. Baizal, et al. Handling imbalanced data in customer churn prediction using combined sampling and weighted random forest. In *2014 2nd International Conference on Information and Communication Technology (ICoICT)*, pages 325–330. IEEE, 2014.

A. Fernández, S. García, F. Herrera, and N. V. Chawla. SMOTE for Learning from Imbalanced Data : Progress and Challenges, Marking the 15-year Anniversary. *Journal of Artificial Intelligence Research*, 61 :863–905, 2018. ISSN 10769757. doi : 10.1613/jair.1.11192.

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

A. Gandomi and M. Haider. Beyond the hype : Big data concepts, methods, and analytics. *International journal of information management*, 35(2) :137–144, 2015.

S. Ganesan. Determinants of long-term orientation in buyer-seller relationships. *Journal of marketing*, 58(2) :1–19, 1994.

D. L. García, À. Nebot, and A. Vellido. Intelligent data analysis approaches to churn as a business problem : a survey. *Knowledge and Information Systems*, 51(3) :719–774, 2017.

V. García, R. A. Mollineda, and J. S. Sánchez. On the k-nn performance in a challenging scenario of imbalance and overlapping. *Pattern Analysis and Applications*, 11(3) :269–280, 2008.

V. García, J. S. Sánchez, and R. A. Mollineda. On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowledge-Based Systems*, 25(1) :13–21, 2012.

D. A. Garvin. *Managing quality : The strategic and competitive edge*. Simon and Schuster, 1988.

L. Geiler, S. Affeldt, and M. Nadif. A survey on machine learning methods for churn prediction. *International Journal of Data Science and Analytics*, pages 1–26, 2022.

S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

P. Gilmour, G. Borg, P. A. Duffy, N. D. Johnston, B. Limbek, and M. R. Shaw. Customer service : differentiating by market segment. *International Journal of Physical Distribution & Logistics Management*, 24(4) :18–23, 1994.

Y. Goldberg and O. Levy. word2vec explained : deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

B. Gregory. Predicting customer churn : Extreme gradient boosting with temporal data. *arXiv preprint arXiv:1802.03396*, , 2018.

C.-C. Günther, I. F. Tvete, K. Aas, G. I. Sandnes, and Ø. Borgan. Modelling and predicting customer churn from an insurance company. *Scandinavian Actuarial Journal*, 2014(1) :58–71, 2014.

C. Guo and F. Berkhahn. Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*, 2016.

X. Guo, L. Gao, X. Liu, and J. Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI*, pages 1753–1759, 2017.

S. Gupta, D. R. Lehmann, and J. A. Stuart. Valuing customers. *Journal of marketing research*, 41 (1) :7–18, 2004.

V. Gupta and A. Bhavsar. Heterogeneous ensemble with information theoretic diversity measure for human epithelial cell image classification. *Medical & Biological Engineering & Computing*, 59(5) : 1035–1054, 2021.

I. Guyon, V. Lemaire, M. Boullé, G. Dror, and D. Vogel. Analysis of the kdd cup 2009 : Fast scoring on a large orange customer database. In *Proceedings of the 2009 International Conference on KDD-Cup 2009-Volume 7*, pages 1–22. JMLR. org, 2009.

J. Hadden, A. Tiwari, R. Roy, and D. Ruta. Churn prediction : Does technology matter. *International Journal of Intelligent Technology*, 1(2) :104–110, 2006.

G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing. Learning from class-imbalanced data : Review of methods and applications. *Expert Systems with Applications*, 73 : 220–239, 2017. ISSN 09574174. doi : 10.1016/j.eswa.2016.12.035.

H. Han, W.-Y. Wang, and B.-H. Mao. Borderline-smote : a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.

D. J. Hand and K. Yu. Idiot's Bayes - Not so stupid after all ? *International Statistical Review*, 69 (3) :385–398, 2001. ISSN 03067734. doi : 10.1111/j.1751-5823.2001.tb00465.x.

P. Hart. The condensed nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 14(3) :515–516, 1968.

G. He, Y. Duan, G. Zhou, and L. Wang. Early classification on multivariate time series with core features. In *International Conference on Database and Expert Systems Applications*, pages 410–422. Springer, 2014.

H. He and Y. Ma. *Imbalanced learning : foundations, algorithms, and applications.* John Wiley & Sons, 2013.

S. He, H., Bai, Y., Garcia, E., & Li. ADASYN : Adaptive synthetic sampling approach for imbalanced learning. In IEEE International Joint Conference on Neural Networks, 2008. *IJCNN 2008.(IEEE World Congress on Computational Intelligence) (pp. 1322 1328)*, (3) :1322– 1328, 2008.

G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786) :504–507, 2006.

L. M. Hitt and F. X. Frei. Do better customers utilize electronic distribution channels ? the case of pc banking. *Management Science*, 48(6) :732–748, 2002.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33 :6840–6851, 2020.

R. C. Holte, L. Acker, B. W. Porter, et al. Concept learning and the problem of small disjuncts. In *IJCAI*, volume 89, pages 813–818. Citeseer, 1989.

P. Hosein, G. Sewdhan, and A. Jailal. Soft-churn : Optimal switching between prepaid data subscriptions on e-sim support smartphones. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–6. IEEE, 2021.

B. Huang, M. T. Kechadi, and B. Buckley. Customer churn prediction in telecommunications. *Expert Systems with Applications*, 39(1) :1414–1425, 2012. ISSN 09574174. doi : 10.1016/j.eswa.2011.08.024.

A. Hudaib, R. Dannoun, O. Harfoushi, R. Obiedat, and H. Faris. Hybrid data mining models for predicting customer churn. *International Journal of Communications, Network and System Sciences*, 8(05) :91, 2015.

S. Ioffe and C. Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc., 1995.

S. Kamaruddin and V. Ravi. Credit card fraud detection using big data analytics : Use of psoaann based one-class classification. In *Proceedings of the International Conference on Informatics and Analytics*, ICIA-16, New York, NY, USA, 2016. Association for Computing Machinery. doi : 10.1145/2980258.2980319.

J. Kawale, A. Pal, and J. Srivastava. Churn prediction in MMORPGs : A social influence based approach. In *2009 International Conference on Computational Science and Engineering*, volume 4, pages 423–428. IEEE, 2009.

S. M. Keaveney. Customer switching behavior in service industries : An exploratory study. *Journal of marketing*, 59(2) :71–82, 1995.

S. M. Keaveney and M. Parthasarathy. Customer switching behavior in online services : An exploratory study of the role of selected attitudinal, behavioral, and demographic factors. *Journal of the academy of marketing science*, 29(4) :374–390, 2001.

A. Keramati, R. Jafari-Marandi, M. Aliannejadi, I. Ahmadian, M. Mozaffari, and U. Abbasi. Improved churn prediction in telecommunication industry using data mining techniques. *Applied Soft Computing*, 24 :994–1012, 2014.

Y. Kim. Toward a successful crm : variable selection, sampling, and ensemble. *Decision Support Systems*, 41(2) :542–553, 2006.

G. King and L. Zeng. Logistic regression in rare events data. *Political analysis*, 9(2) :137–163, 2001.

R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

J. Kong, W. Kowalczyk, S. Menzel, and T. Bäck. Improving imbalanced classification by anomaly detection. In T. Bäck, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich, and H. Trautmann, editors, *Parallel Problem Solving from Nature – PPSN XVI*, pages 512–523, Cham, 2020. Springer International Publishing.

D. A. Kumar, V. Ravi, et al. Predicting credit card customer churn in banks using data mining. *International Journal of Data Analysis Techniques and Strategies*, 1(1) :4–28, 2008.

R. Kuo, Y. An, H. Wang, and W. Chung. Integration of self-organizing feature maps neural network and genetic k-means algorithm for market segmentation. *Expert systems with applications*, 30(2) : 313–324, 2006.

B. Larivière and D. Van den Poel. Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications*, 29(2) :472–484, 2005.

M. Laroche, J. A. Rosenblatt, and T. Manning. Services used and factors considered important in selecting a bank : an investigation across diverse demographic segments. *International Journal of bank marketing*, 1986.

J. Laurikkala. Improving identification of difficult small classes by balancing class distribution. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 63–66. Springer, 2001.

G. LeBlanc and N. Nguyen. Customers' perceptions of service quality in financial institutions. *International Journal of Bank Marketing*, 1988.

A. Lemmens and C. Croux. Bagging and boosting classification trees to predict churn. *Journal of Marketing Research*, 43(2) :276–286, 2006.

C. K. Leung, A. G. Pazdor, and J. Souza. Explainable artificial intelligence for data science on customer churn. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE, 2021.

M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart : Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

W. Li, M. Gao, H. Li, Q. Xiong, J. Wen, and Z. Wu. Dropout prediction in MOOCs using behavior features and multi-view semi-supervised learning. *Proceedings of the International Joint Conference on Neural Networks*, 2016-Octob :3130–3137, 2016. doi : 10.1109/IJCNN.2016.7727598.

C. X. Ling and C. Li. Data mining for direct marketing : Problems and solutions. In *Kdd*, volume 98, pages 73–79, 1998.

F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. Data*, 6(1), mar 2012. doi : 10.1145/2133360.2133363.

S. Loffe and C. Normalization. Accelerating deep network training by reducing internal covariate shift. *arXiv*, 2014.

V. López, A. Fernández, J. G. Moreno-Torres, and F. Herrera. Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. open problems on intrinsic data characteristics. *Expert Systems with Applications*, 39(7) :6585–6608, 2012.

V. López, A. Fernández, S. García, V. Palade, and F. Herrera. An insight into classification with imbalanced data : Empirical results and current trends on using data intrinsic characteristics. *Information sciences*, 250 :113–141, 2013.

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

A. Martínez-Usó, J. Mendes-Moreira, L. Moreira-Matias, M. Kull, and N. Lachiche. Proceedings of the ecml/pkdd 2015 discovery challenges : co-located with european conference on machine learning and principles and practice of knowledge discovery in databases (ecml-pkdd 2015). 2015.

J. G. Maxham III. Service recovery's influence on consumer satisfaction, positive word-of-mouth, and purchase intentions. *Journal of business research*, 54(1) :11–24, 2001.

I. McKinley Stacker. Ibm waston analytics. sample data : Hr employee attrition and performance [data file], 2015.

J. Meynet and J.-P. Thiran. Information theoretic combination of pattern classifiers. *Pattern Recognition*, 43(10) :3412–3421, 2010.

I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn : unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016.

B. Mittal and W. M. Lassar. Why do customers switch ? the dynamics of satisfaction versus loyalty. *Journal of services marketing*, 12(3) :177–194, 1998.

V. Mittal and W. A. Kamakura. Satisfaction, repurchase intent, and repurchase behavior : Investigating the moderating effect of customer characteristics. *Journal of marketing research*, 38(1) : 131–142, 2001.

M. Mohandes, M. Deriche, and S. O. Aliyu. Classifiers combination techniques : A comprehensive review. *IEEE Access*, 6 :19626–19639, 2018.

M. C. Mozer, R. Wolniewicz, D. B. Grimes, E. Johnson, and H. Kaushansky. Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. *IEEE Transactions on neural networks*, 11(3) :690–696, 2000.

L. Munkhdalai, T. Munkhdalai, K. H. Park, T. Amarbayasgalan, E. Batbaatar, H. W. Park, and K. H. Ryu. An end-to-end adaptive input selection with dynamic weights for forecasting multivariate time series. *IEEE Access*, 7 :99099–99114, 2019.

L. Munkhdalai, T. Munkhdalai, and K. H. Ryu. Gev-nn : A deep neural network architecture for class imbalance problem in binary classification. *Knowledge-Based Systems*, 194 :105534, 2020.

K. Napierała, J. Stefanowski, and S. Wilk. Learning from imbalanced data in presence of noisy and borderline examples. In *International Conference on Rough Sets and Current Trends in Computing*, pages 158–167. Springer, 2010.

S. A. Neslin, S. Gupta, W. Kamakura, J. Lu, and C. H. Mason. Defection detection : Measuring and understanding the predictive accuracy of customer churn models. *Journal of marketing research*, 43(2) :204–211, 2006.

H. M. Nguyen, E. W. Cooper, and K. Kamei. Borderline over-sampling for imbalanced data classification. *International Journal of Knowledge Engineering and Soft Data Paradigms*, 3(1) :4–21, 2011.

N. Nguyen and G. LeBlanc. The mediating role of corporate image on customers retention decisions : an investigation in financial services. *International journal of bank marketing*, 16(2) :52–65, 1998.

C. G. Northcutt, A. Athalye, and J. Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Proceedings of the 35th Conference on Neural Information Processing Systems Track on Datasets and Benchmarks*, December 2021.

A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

C. Orsenigo and C. Vercellis. Combining discrete svm and fixed cardinality warping distances for multivariate time series classification. *Pattern Recognition*, 43(11) :3787–3794, 2010.

M. Óskarsdóttir, T. Van Calster, B. Baesens, W. Lemahieu, and J. Vanthienen. Time series for early churn detection : Using similarity based classification for dynamic networks. *Expert Systems with Applications*, 106 :55–65, 2018.

A. B. Owen. Infinitely imbalanced logistic regression. *Journal of Machine Learning Research*, 8(Apr) : 761–773, 2007.

N. C. Oza and K. Tumer. Classifier ensembles : Select real-world applications. *Information fusion*, 9 (1) :4–20, 2008.

G. Pang, H. Xu, L. Cao, and W. Zhao. Selective value coupling learning for detecting outliers in high-dimensional categorical data. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 807–816, 2017.

G. Pang, C. Shen, and A. van den Hengel. Deep anomaly detection with deviation networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 353–362, 2019.

G. Pang, C. Shen, L. Cao, and A. V. D. Hengel. Deep learning for anomaly detection : A review. *ACM Comput. Surv.*, 54(2), mar 2021. doi : 10.1145/3439950.

M. Paulin, J. Perrien, R. J. Ferguson, A. M. A. Salazar, and L. M. Seruya. Relational norms and client retention : external effectiveness of commercial banking in canada and mexico. *International Journal of Bank Marketing*, 16(1) :24–31, 1998.

G. A. Pavliotis. Stochastic processes and applications. *Informe técnico*, 2015.

F. F. Reichheld and W. E. Sasser. Zero defections : Quality comes to services. *Harvard business review*, 68(5) :105–111, 1990.

W. J. Reinartz and V. Kumar. The impact of customer relationship characteristics on profitable lifetime duration. *Journal of marketing*, 67(1) :77–99, 2003.

J. D. Rennie. Improving multi-class text classification with naive bayes. *Technical Report AITR*, 4, 2001.

M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you ?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

H. Risselada, P. C. Verhoef, and T. H. Bijmolt. Staying power of churn prediction models. *Journal of Interactive Marketing*, 24(3) :198–208, 2010.

L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 2021.

L. Ruisen, D. Songyi, W. Chen, C. Peng, T. Zuodong, Y. YanMei, and W. Shixiong. Bagging of xgboost classifiers with random under-sampling and tomek link for noisy label-imbalanced data. In *IOP Conference Series : Materials Science and Engineering*, volume 428, page 012004. IOP Publishing, 2018.

O. Sagi and L. Rokach. Ensemble learning : A survey. *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, 8(4) :e1249, 2018.

C. Salas-Eljatib, A. Fuentes-Ramirez, T. G. Gregoire, A. Altamirano, and V. Yaitul. A study on the effects of unbalanced data when fitting logistic regression models in ecology. *Ecological Indicators*, 85 :502–508, 2018.

V. V. Saradhi and G. K. Palshikar. Employee churn prediction. *Expert Systems with Applications*, 38 (3) :1999–2006, 2011.

B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. NIPS'99, page 582588, Cambridge, MA, USA, 1999. MIT Press.

C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Folleco. An empirical study of the classification performance of learners on imbalanced and noisy software quality data. *Information Sciences*, 259 : 571–595, 2014.

O. F. Seymen, O. Dogan, and A. Hiziroglu. Customer churn prediction using deep learning. In *International Conference on Soft Computing and Pattern Recognition*, pages 520–529. Springer, 2020.

C. A. Shipp and L. I. Kuncheva. Relationships between combination methods and measures of diversity in combining classifiers. *Information fusion*, 3(2) :135–148, 2002.

R. Shwartz-Ziv and A. Armon. Tabular data : Deep learning is not all you need. *Information Fusion*, 81 :84–90, 2022.

R. Siber. Combating the churn phenomenon-as the problem of customer defection increases, carriers are having to find new strategies for keeping subscribers happy. *Telecommunications-International Edition*, 31(10) :77–81, 1997.

P. H. Sneath and R. R. Sokal. *Numerical taxonomy*. San Francisco, 1973.

A. Śniegula, A. Poniszewska-Marańda, and M. Popović. Study of machine learning methods for customer churn prediction in telecommunication company. In *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services*, pages 640–644, 2019.

G. Somepalli, M. Goldblum, A. Schwarzschild, C. B. Bruss, and T. Goldstein. Saint : Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*, 2021.

C. Song, F. Liu, Y. Huang, L. Wang, and T. Tan. Auto-encoder based data clustering. In *Iberoamerican congress on pattern recognition*, pages 117–124. Springer, 2013.

Y. Song and S. Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33 :12438–12448, 2020.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout : a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1) : 1929–1958, 2014.

J. Stefanowski. Dealing with data difficulty factors while learning from imbalanced data. In *Challenges in computational statistics and data mining*, pages 333–363. Springer, 2016.

E. Štrumbelj and I. Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3) :647–665, 2014.

A. Taha and A. S. Hadi. Anomaly detection methods for categorical data : A review. *ACM Comput. Surv.*, 52(2), may 2019. doi : 10.1145/3312739.

F. Tan, Z. Wei, J. He, X. Wu, B. Peng, H. Liu, and Z. Yan. A Blended Deep Learning Approach for Predicting User Intended Actions. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2018-Novem :487–496, 2018. ISSN 15504786. doi : 10.1109/ICDM.2018.00064.

S. Tan. Neighbor-weighted k-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, 28(4) :667–671, 2005.

L. Tang, L. Thomas, M. Fletcher, J. Pan, and A. Marshall. Assessing the impact of derived behavior information on customer attrition in the financial service industry. *European Journal of Operational Research*, 236(2) :624–633, 2014.

A. Tarvainen and H. Valpola. Mean teachers are better role models : Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

D. M. J. Tax and R. P. W. Duin. Support vector domain description. *Pattern Recogn. Lett.*, 20(1113) : 11911199, nov 1999. doi : 10.1016/S0167-8655(99)00087-2.

F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu. Learning deep representations for graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

J. Tian, H. Gu, and W. Liu. Imbalanced classification using support vector machine ensemble. *Neural computing and applications*, 20(2) :203–209, 2011.

I. Tomek. Tomek Link : Two Modifications of CNN. *IEEE Trans. Systems, Man and Cybernetics*, SMC-6 :769–772, 1976. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp={&}arnumber=4309452.

C.-Y. Tsai and C.-C. Chiu. A purchase-based market segmentation methodology. *Expert Systems with Applications*, 27(2) :265–276, 2004.

T. Ucar, E. Hajiramezanali, and L. Edwards. Subtab : Subsetting features of tabular data for self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34, 2021.

I. Ullah, B. Raza, A. K. Malik, M. Imran, S. U. Islam, and S. W. Kim. A churn prediction model using random forest : analysis of machine learning techniques for churn prediction and factor identification in telecom sector. *IEEE access*, 7 :60134–60149, 2019.

V. Umayaparvathi and K. Iyakutti. A Survey on Customer Churn Prediction in Telecom Industry : Datasets, Methods and Metrics. *International Research Journal of Engineering and Technology*, 3 : 2395–56, 2016. ISSN 2395-0072.

V. Umayaparvathi and K. Iyakutti. Automated feature selection and churn prediction using deep learning models. *International Research Journal of Engineering and Technology (IRJET)*, 4(3) : 1846–1854, 2017.

T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis, and K. C. Chatzisavvas. A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*, 55 : 1–9, 2015.

D. Van den Poel and B. Lariviere. Customer attrition analysis for financial services using proportional hazard models. *European journal of operational research*, 157(1) :196–217, 2004.

V. Vapnik. Statistical learning theory wiley-interscience. *New York*, 1998.

S. Varki and M. Colgate. The role of price perceptions in an integrated model of behavioral intentions. *Journal of Service Research*, 3(3) :232–240, 2001.

A. Vellido, P. Lisboa, and K. Meehan. Segmentation of the on-line shopping market using neural networks. *Expert systems with applications*, 17(4) :303–314, 1999.

M. E. Villa-Pérez, M. Á. Álvarez-Carmona, O. Loyola-González, M. A. Medina-Pérez, J. C. Velazco-Rossell, and K.-K. R. Choo. Semi-supervised anomaly detection algorithms : A comparative summary and future research directions. *Knowledge-Based Systems*, page 106878, 2021. doi : https://doi.org/10.1016/j.knosys.2021.106878.

L. Wang, Z. Wang, and S. Liu. An effective multivariate time series classification approach using echo state network and adaptive differential evolution algorithm. *Expert Systems with Applications*, 43 : 237–249, 2016a.

S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy. Training deep neural networks on imbalanced data sets. In *2016 international joint conference on neural networks (IJCNN)*, pages 4368–4374. IEEE, 2016b.

W. Wang, H. Yu, and C. Miao. Deep model for dropout prediction in MOOCs. *ACM International Conference Proceeding Series*, Part F1306 :26–32, 2017. doi : 10.1145/3126973.3126990.

G. M. Weiss. Mining with rarity : a unifying framework. *ACM Sigkdd Explorations Newsletter*, 6(1) : 7–19, 2004.

G. M. Weiss. The impact of small disjuncts on classifier learning. In *Data Mining*, pages 193–226. Springer, 2010.

G. M. Weiss and H. Hirsh. A quantitative study of small disjuncts. *AAAI/IAAI*, 2000 :665–670, 2000.

G. M. Weiss and F. Provost. Learning when training data are costly : The effect of class distribution on tree induction. *Journal of artificial intelligence research*, 19 :315–354, 2003.

D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2(3) :408–421, 1972.

J. Xiao, L. Huang, and L. Xie. Cost-sensitive semi-supervised ensemble model for customer churn prediction. In *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*, pages 1–6. IEEE, 2018.

Y. Xiao, H. Wang, W. Xu, and J. Zhou. Robust one-class svm for fault detection. *Chemometrics and Intelligent Laboratory Systems*, 151 :15–25, 2016. doi : https://doi.org/10.1016/j.chemolab.2015.11.010.

J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

Y. Xie and X. Li. Churn prediction with linear discriminant boosting algorithm. In *2008 International Conference on Machine Learning and Cybernetics*, volume 1, pages 228–233. IEEE, 2008.

C. Yang, X. Shi, L. Jie, and J. Han. I know you'll be back : Interpretable new user clustering and churn prediction on a mobile social application. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 914–922, 2018.

Z. Yang and R. T. Peterson. Customer perceived value, satisfaction, and loyalty : The role of switching costs. *Psychology & Marketing*, 21(10) :799–822, 2004.

L. Yin, Y. Ge, K. Xiao, X. Wang, and X. Quan. Feature selection for high-dimensional imbalanced data. *Neurocomputing*, 105 :3–11, 2013. ISSN 09252312. doi : 10.1016/j.neucom.2012.04.039.

J. Yoon, Y. Zhang, J. Jordon, and M. van der Schaar. Vime : Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33 :11033–11043, 2020.

B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 204–213, 2001.

B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Third IEEE international conference on data mining*, pages 435–442. IEEE, 2003.

G. Zaid, L. Bossuet, A. Habrard, and A. Venelli. Efficiency through diversity in ensemble models applied to side-channel attacks:–a case study on public-key algorithms–. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 60–96, 2021.

J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins : Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

V. A. Zeithaml, L. L. Berry, and A. Parasuraman. The behavioral consequences of service quality. *Journal of marketing*, 60(2) :31–46, 1996.

R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders : Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017.

Z. Zhao, H. Peng, C. Lan, Y. Zheng, L. Fang, and J. Li. Imbalance learning for the prediction of n 6-methylation sites in mrnas. *BMC genomics*, 19(1) :574, 2018.

F. Zhou, S. Yang, H. Fujita, D. Chen, and C. Wen. Deep learning fault diagnosis method based on global optimization gan for unbalanced data. *Knowledge-Based Systems*, 187 :104837, 2020.

Z.-H. Zhou. *Ensemble methods : foundations and algorithms*. CRC press, 2012.

Z.-H. Zhou and X.-Y. Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on knowledge and data engineering*, 18(1) :63–77, 2005.

Y. Zhu, L. Zhou, C. Xie, G.-J. Wang, and T. V. Nguyen. Forecasting smes' credit risk in supply chain finance with an enhanced hybrid ensemble machine learning approach. *International Journal of Production Economics*, 211 :22–33, 2019. ISSN 0925-5273. doi : https://doi.org/10.1016/j.ijpe.2019.01.032.

B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.