Master's Thesis

An Efficient Stacking Ensemble Learning Method for Customer Churn Prediction

Department of Computer Engineering

Graduate school, Chonnam National University

Mukhammadiev Komiljon Jahongir ugli

February 2023

An Efficient Stacking Ensemble Learning Method for Customer Churn Prediction

Department of Computer Engineering

Graduate school, Chonnam National University

Mukhammadiev Komiljon Jahongir ugli

Supervised by Professor Chang Gyoon Lim

A dissertation submitted in partial fulfillment of the requirements

for the Master of Computer engineering

Committee in Charge:

Kang Chul Kim _____

Kim Gwang Jun _____

Chang Gyoon Lim _____

February 2023

Table of Contents

List of Figures

List of Tables

An Efficient Stacking Ensemble Learning Method for Customer Churn Prediction

Mukhammadiev Komiljon Jahongir ugli

Department of Computer Engineering

Graduate School, Chonnam National University

(Supervised by Chang Gyoon Lim)

Abstract

In recent years, Customer churn has been a significant problem and one of the essential concerns for telecom companies. Focusing on retaining existing customers rather than acquiring new customers is a crucial strategy for reducing costs and increasing revenues in the telecom industry. This study proposes an efficient customer-churn prediction model that uses an ensemble-learning technique consisting of the Synthetic Minority Over-Sampling (SMOTE) method for imbalanced data, stacking models, and soft voting. Random Forest, Extreme Gradient Boosting (XGBoost), CatBoost, and Multilayer Perceptrons (MLPs) machine-learning algorithms are selected to build a stacking ensemble model, and the results of the four algorithms are used for soft voting. Compared to other prediction models, the proposed model showed the best accuracy of 78.79% and 88.20% for the original imbalance dataset and the new balanced dataset, respectively. Our proposed model can provide early detection of customer churn in the telecom industry.

I. Introduction

1.1 Background of the study

Most telecom services view customers as their most valuable asset. As a result, one of the most challenging problems that telco companies face nowadays is when clients switch to another service provider for whatever reason. Because consumers can easily change services, in many cases, churn can significantly affect company profitability [1]. Customer churn forecasting enables you to identify the causes of relationship breakdowns and develop strategies to reduce churn while improving profitability. Therefore, it is crucial and seen as a competitive advantage for Telco to anticipate customers' desire to cancel their subscriptions. Various approaches have been proposed for customer churn prediction cases. Linear Regression and Support Vector Machine (SVM) models were adopted and showed promising results for churn prediction problems in the early millennium. Recently, ensemble learning methods have become popular in customer churn prediction [2]. Ensemble methods are meta-algorithms that combine multiple machine learning models to make predictions better. There are two types of ensemble methods: Bagging and Boosting. Random Forest is the most popular Bagging method, with XGBoost, LightGBM, and CatBoost being to Boosting methods that have attracted much attention in recent studies on binary classification.

Previous studies of customer churn prediction have shown superior performance of Random Forest compared to traditional categorization methods. For example, showed that Random Forest achieved higher scores than Naïve Bayes, SVM, Decision Tree, Bagging, and Boosting [3]. The author of showed higher Random

Forest rates than SVM, KNN, and Logistics Regression [4]. Gradient Boosted Decision Trees (GBDTs) achieved the highest performance, followed by a Decision Tree and a Random Forest. KNN (0,575) and Neve Bayes (0,646) achieved low results [5]. Boosting-based algorithms as ensemble methods have shown outstanding classification effectiveness in various research areas but have yet to establish a broad advantage in research on customer churn prediction. XGBoost (Extreme Gradient Boosting) is a recently proposed ensemble method. This is an advanced way of gradient boosting [6]. In this study, we first compare the most popular supervised machine learning algorithms with this problematic situation in the telecom industry. Then, an effective stacking ensemble model is proposed to overcome the problems associated with churn prediction.

1.2 Problem statement

Retaining and purchasing users is one of the significant challenges in the telecommunications industry. Rapid market growth in any business leads to an increase in the subscriber base. Accordingly, companies have recognized the importance of keeping customers on hand. Service providers must reduce customer failure because negligence can negatively influence a company's profitability. Release forecasting helps identify users who can replace the company with another. Telecom is enduring the growing problem of failure. The machine learning algorithm helps protect these telecom firms with practical approaches to reduce churn rate. Customer churn is one of the types that is considered difficult to predict because such users may appear soon. The goal of decision-makers and advertisers should be to reduce consumption, as it is a recognized fact that relatively existing customers

are the most beneficial resources for companies rather than acquiring new ones. Annual churn rates for telecommunications companies are typically higher than 10%. Customer churn is an unavoidable problem for telecommunications companies. Churn affects all businesses. Large companies collapsed within a few years due to poor churn management.

1.3 Organization of the thesis

The thesis consists of five chapters. The structure of the idea is arranged as below:

Section II – This section includes the concepts to understand the thesis, Comparison of models, Resampling technique, Ensemble methods, and Voting Ensemble.

Section III – In this section, we processed our dataset and then went for candidates of model selection to create an ensemble model classifier.

Section IV – This section analyzes different classification algorithms and ensemble methods with some fundamental approaches and explains the chosen algorithms. This part discusses the tools and techniques used to check performance and shows the results of the proposed efficient stacking ensemble model for churn prediction.

Section V – Finally, the conclusion and future work are presented.

II. Related works and methodology

The industry has used various methods to understand and estimate the churn rate of telecommunication services. Data mining, machine learning (ML), and deep learning (DL) algorithms have been widely used. While most relevant studies have focused on using only one machine learning method for data extraction, some have compared multiple methods for classification disorders. Compared linear regression, decision trees, and MLPs to predict customer churn based on features associated with customer complaints. They split the data set evenly between non-churners and churners with a ratio of 50%. The authors demonstrated that MLPs are capable of accurately predicting churners [7]. Authors of presented an efficient data mining model for predicting customer turnover, utilizing a dataset of 3333 call details and 21 characteristics with two values of churn labels: Yes or No. They adopted the Principal Component Analysis (PCA) method to reduce the feature dimension before implementing Bayes Networks, SVM, and MLPs for churn prediction modeling. The author used the AUC metric score to evaluate the performance of the algorithms. There were no missing values in the dataset utilized in this study since it was tiny [8]. Focused on assessing and analyzing the performance of a set of tree-based machine learning methods and algorithms for predicting churn in telco companies based on the big data platform. After implementing data preprocessing, feature engineering, and feature selection techniques, the authors have experimented with several algorithms, namely Random Forest, Decision Tree, XGBoost, and Gradient Boosting, to build the predictive model of customer churn. Working on a massive dataset obtained by processing enormous raw data from the Syriatel telecom firm,

the model was constructed and validated using the Spark environment. The dataset was utilized for training, testing, and evaluating the system at Syriatel, and it includes all of the customers' information throughout nine months [9]. Data imbalance is also vital in churn prediction tasks, similar to other machine learning problems. In an imbalanced dataset, the number of churned customer labels is lower than that of current customer labels. Several studies have looked into the issue of data imbalance. More details about how class imbalance can influence who can find different classification algorithms in [10]. Many other solutions have tried to solve this problem; we can find them in three categories: data-level, algorithm-level, and ensemble solutions. Intelligent sampling methods are aimed at solving this imbalanced data problem, including synthetic minority; oversampling techniques (SMOTE) are probably the most popular ones. SMOTE is commonly used as a benchmark for oversampling algorithms [11, 12]. Therefore, evaluated how well Weighted Random Forests, Gradient Boosting Model, Advanced Undersampling, and Random Sampling performed in churn prediction models using unbalanced datasets. The under-sampling strategy paid off other methods considered according to the results [13]. To solve the problem of telco outage prediction, the authors in considered six different sampling strategies. According to the results, Mega-Trend-Diffusion Function (MTDF) and rules generation based on genetic algorithms outperformed the other oversampling methods [14]. Recently, designed factor analysis to analyze telco business characteristics building a discriminant model and a logistic regression model for forecasting customers and telecommunications Customer attrition using customer segmentation data from three major Chinese

telcos [15]. The authors implemented a logistic regression model to predict the telecommunications customer churn in a new way. CatBoost is a new boosting algorithm compared to XGBoost and LightGBM. CatBoost is an algorithm based on Gradient Boosted Decision Trees (GBDTs) used for regression and classification tasks [16]. One of its strengths is its superiority in handling categorical features [17]. Recent studies have shown the algorithm's superior performance in binary format classification problems. Used CatBoost to predict corporate failure and found that CatBoost achieved higher accuracy and Area Under Curve Receiver Operator Characteristic (ROC-AUC) scores than discriminant analysis, Logistic Regression, SVM, Artificial Neural Network, Random Forest, Gradient Boosting, Deep Neural Network (DNN), and XGBoost [18].

2.1 Introduce machine learning

Machine Learning (ML) is a branch of AI and autonomous artificial intelligence that allows machines to learn from experience with large amounts of data without being programmed. It synthesizes and interprets data for human understanding according to predefined parameters, helping to save time, reduce errors, create preventive measures and automate processes in large operations and companies.

Machine learning algorithms can be classified into four types:

2.1.1 Supervised learning

Every dataset used in machine learning contains instances. The instances are represented or defined using features [19]. These features can be either 'binary/categorical' or 'continuous'. The cases which correspond to the correct

output/answer are called labeled. Supervised machine learning algorithms are employed when the data (training data and testing data) is labeled [20]. These algorithms gain knowledge and forecast with the help of labeled past and present data.

Classification

Classification is a supervised learning problem where the output space has a set of classes. To quote an example, the output feature of the data, determining whether it would rain today or not, is represented with either "YES", "NO," or "MAYBE" [21].

Regression

Regression is a supervised learning problem where the output space is a set of serial numbers [21]. The estimation of the relationship between output and features that influence the result is determined by the regression algorithms, which are based on statistical methods [22].

2.1.2 Unsupervised learning

Unlike supervised learning, what will employ unsupervised learning for unlabeled states? Unsupervised algorithms analyze unlabeled data and generate a function that explains the data example. Output using these algorithms is non-identical; it is known. However, these algorithms help in opening and writing observations about hidden patterns in data. Clustering is association-type unsupervised learning [20].

2.1.3 Reinforcement learning

The learning algorithm is not provided or dictated about the action it must take; instead, a learner takes steps by himself that will result in the best rewards based on

the environment. Chess games would be an excellent example of where what can adopt this type of learner [19].

2.2 Algorithms

To build the ensemble model, below are the Supervised Machine learning algorithms and ensemble methods implemented in this thesis. A brief explanation of the selected Supervised Machine learning algorithms implemented in this thesis is given below:

Naïve Bayes. It is a straightforward learning technique that relies on the Bayes rule and a strong presumption, in which the characteristics are uncorrelated, given the classifier. Even though this independence requirement is frequently broken in practice, nave Bayes classification accuracy is generally competitive. Because of its computing effectiveness and several other appealing characteristics, Naïve Bayes is commonly used in practice [23].

K-Nearest Neighbor is one of the most valuable and applicable non-parametric algorithms for neighboring studies. K-Nearby is also called the lazy algorithm, which uses all the training data in the test phase. There is no training phase, and total data points are applied directly to the test phase, so it should be used when the test needs to be passed. K-Uses the distance between entries for use in the nearest adjacent classification. To estimate the distance between points, the K-Near neighbor assumes that these points are multidimensional or scalar vectors in the property space. All data points are vectors of the property field, and the label points to their classes. The most straightforward case is if these class labels are binary, but it is helpful in arbitrary class numbers. K-A single parameter must be set in the nearest neighbor. K The number of neighbor's examples is calculated to touch some

class. A very similar class among the neighbors defines the model. The calculated distances are used to identify the set of study samples closest to the new point and to separate the label from them. Despite its simplicity, the K-nearest neighbor has been used for various applications [24].

Logistic Regression. Statistical analysis (also known as a logit model) is commonly used for predictive analytics and modeling and machine learning applications. A categorical dependent variable's output is predicted using logistic regression. As a result, the result must be a discrete or absolute value. It can be Yes or No, 0 or 1, true or false, and so on, but instead of giving precise values like 0 and 1, it delivers probabilistic values between 0 and 1. Logistic regression may be used to categorize observations based on many forms of data and can quickly identify the most valuable factors for classification [25].

The multilayer perceptron is one of the most common forms of neural networks. It comes from the artificial neural network (ANN), a data processing model composed of a collection of basic processing units known as neurons. Neurons interact by transmitting signals via a vast amount of weighted nodes. In the business and industrial spheres, MLP neural networks offer a variety of applications for classification and prediction tasks [26].

Support Vector Machines (SVM). Known as one of the most robust prediction techniques, the SVM training algorithm creates a model that assigns new examples to one of two categories, giving it a non-probabilistic binary linear classifier, given a series of training instances, individually labeled as belonging to one of two categories. SVM translates training sets to points in space to widen the distance

between the two categories as much as possible. New instances are then mapped into the same area and classified accordingly on which side of the divide they land on [27].

Random Forest. It is a classification, regression, and other tasks ensemble learning approach that works by creating a large number of decision trees during training. Regarding classification problems, Random Forest's output, do most trees choose the class? The mean or average forecast of the individual trees is supplied for regression tasks. Random decision forests address the problem of decision trees overfitting their training set. Random forests outperform decision trees in most cases, but they are less accurate than gradient-enhanced trees. However, data features can influence how well they function [28].

XGBoost. It stands for Extreme Gradient Boosting, which produces a prediction model by combining weak prediction models. In most cases, decision trees are the weak learner. When that happens, the resultant approach is known as gradient-boosted trees. It is based on the assumption that when the best total estimation error is minimized, the best potential future model is coupled with prior models trees algorithm is constructed in the same stage-wise manner as other boosting approaches, but that it allows optimization of any differentiable loss function [29].

2.3 Resampling techniques

One way to deal with the class imbalance problem is to resample the training data set randomly. The two main approaches to random resampling of an unbalanced data set are to eliminate examples from the majority class, called undersampling, and to repeat instances from the minority class, called oversampling.

Random oversampling involves selecting random examples from a minority class and populating the training data with multiple copies of that example so that a single sample can be chosen more than once (Fig. 1).

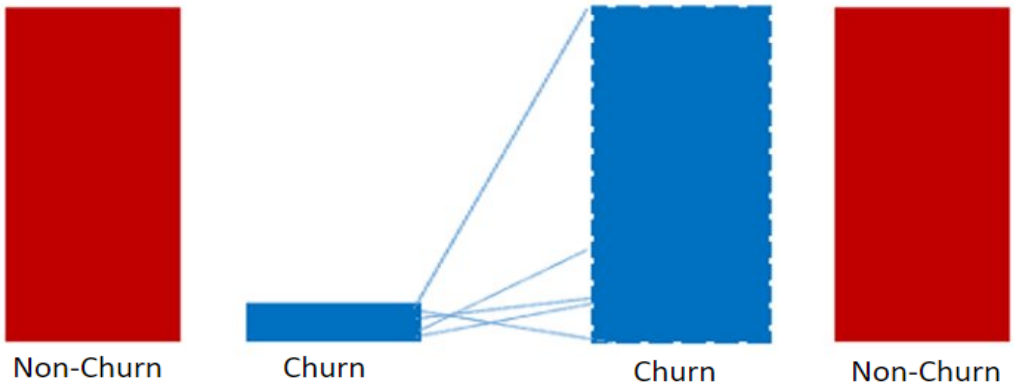

Figure 1. Random oversampling example

There are different methods of oversampling. Below we look at a few popular ones:

- Simple random oversampling: the primary approach of random sampling with replacement from the minority class.
- It was oversampling with shrinkage: based on random sampling, adding some noise/shrinkage to disperse the new samples.
- Oversampling using SMOTE: synthesize new samples based on the minority class.

Random undersampling is the opposite of random oversampling. This method seeks to select and remove samples from the majority class randomly, thereby reducing the number of examples in the majority class in the transformed data (Fig. 2).

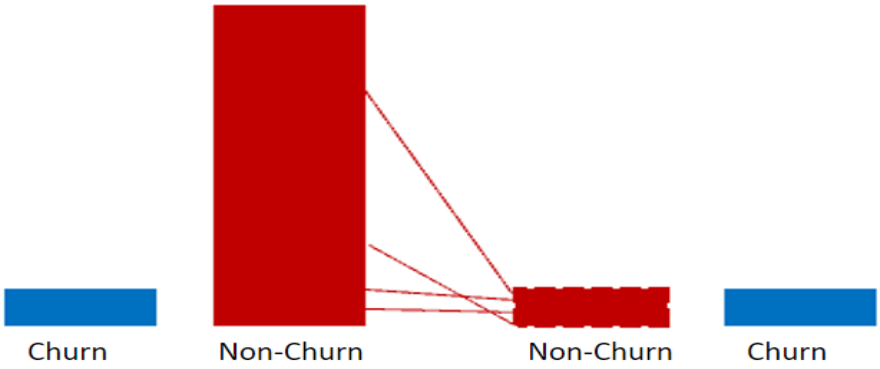

Figure 2. Random undersampling example

There are also many methods of undersampling. We'll cover the below popular ones:

- Simple random undersampling: the primary approach of random sampling from the majority class.
- Undersampling using K-Means: synthesize based on the cluster centroids.
- Undersampling using Tomek links: detects and removes samples from Tomek links.

As a resampling technique, this study used SMOTE. It is the most popular oversampling method and is effective in unbalanced classifications. SMOTE is the simplest method of balancing an unbalanced data set, a random oversampling replicating samples of the existing minority class. However, it does not provide additional information to the classification model and may lead to overfitting. SMOTE was proposed to overcome the weakness of random oversampling [30]. It increases the minority class in the following steps. First, select a case that belongs to a random minority class. Second, identify the k nearest neighbors of the case. Next, choose one of the random neighbors. Then, create a synthetic sample at a randomly selected point between the two states. The equation below represents the

synthesized sample C and a general view of SMOTE technique and Python code (Fig. 3 and 4).

$$C=A+\text{rand}(0, 1)*|A-B|$$

A represents a minority class sample, and B is one of its K nearest neighbors. Rand (0, 1) illustrates a random number between 0 and 1 and |A-B| Euclidean distance between A and B.

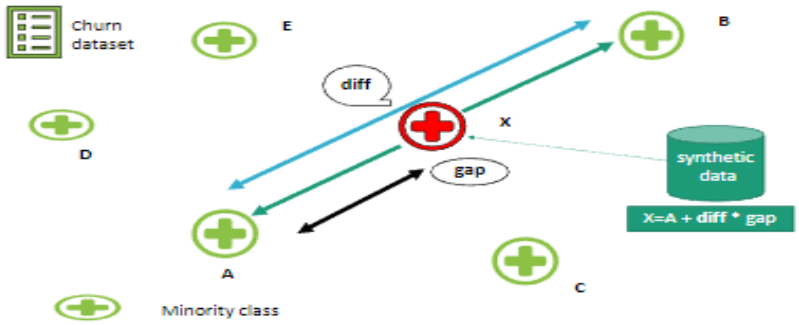

Figure 3. A general view of SMOTE technique

```
from imblearn.over_sampling import SMOTE

counter = Counter(y_train)
print('Before',counter)
# oversampling the train dataset using SMOTE
smt = SMOTE()
#X_train, y_train = smt.fit_resample(X_train, y_train)
X_train_sm, y_train_sm = smt.fit_resample(X_train, y_train)

counter = Counter(y_train_sm)
print('After',counter)
```

```
Before Counter({0: 18497, 1: 4208})
After Counter({0: 18497, 1: 18497})
```

Figure 4. Python code for smote

2.4 Ensemble learning

One of the main tasks of machine learning algorithms is to create a fair model from a set of data. Creating models from data is called learning or training, and the learned model can be called a hypothesis or learner. Learning algorithm Ensemble methods are learning algorithms that build a set of classifiers and then classify new data points by selecting their predictions more accurately than the individual classifiers that make them up. Ensemble methods, also known as committee-based learning or multiple classification system learning, train multiple hypotheses to solve the same problem. One of the most common examples of ensemble modeling is random forest trees (Fig. 5), where a series of decision trees are used to predict outcomes.

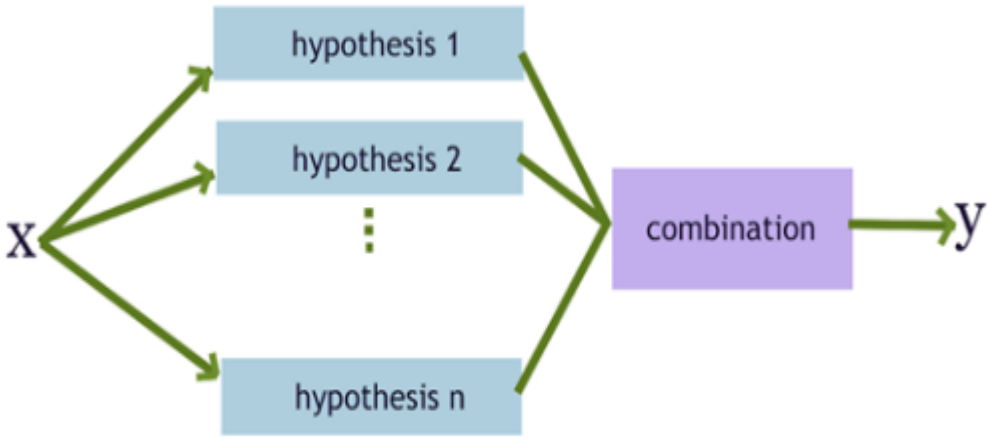

Figure 5. A general ensemble architecture

Some of the common types of ensembles are discussed below:

Bagging or Bootstrap Aggregation is a powerful, efficient and straightforward ensemble method. The method uses multiple versions of the training set using the bootstrap, which means sampling with replacement and can be used with any model for classification or regression. Bagging is only effective when using unstable non-

linear models (a slight change in the training set can cause a significant difference in the model).

Boosting is a meta-algorithm that can be considered a model-averaging method. This is the most common ensemble method and one of the most potent learning ideas. This method was initially designed for classification but can be usefully extended to regression. The original boosting algorithm combined three weak learners to create a muscular leaner.

Table 1. Difference between bagging and boosting

Bagging or Bootstrap Aggregation	Boosting
Various training data subsets are randomly drawn with replacements from the training dataset.	Each new subset contains the components that previous models misclassified.
Bagging attempts to tackle the over-fitting issue.	Boosting tries to reduce bias.
If the classifier is unstable (high variance), we must apply it to the bag.	If the classifier is steady and straightforward (high bias), we need to apply to boost.
Every model receives an equal weight.	Models are weighted by their performance.
The objective is to decrease variance, not bias.	The objective is to decrease bias, not variance.
It is the easiest way to connect predictions of the same type.	It is a way of connecting predictions that belong to the different types.
Every model is constructed independently.	New models are affected by the performance of the previously developed model.

Stacking is related to combining multiple classifiers created using different learning algorithms into a single dataset consisting of pairs of feature vectors and their classifications. This method mainly consists of two steps, in the first step, a set of

base-level classifiers is created, and in the second step, a meta-level classifier is learned that combines the results of the base-level classifiers (Fig 6). Stacking algorithms are another way of combining multiple clasifiers. In contrast to bagging and boosting stacking is used to combine individual models built by different algorithms.

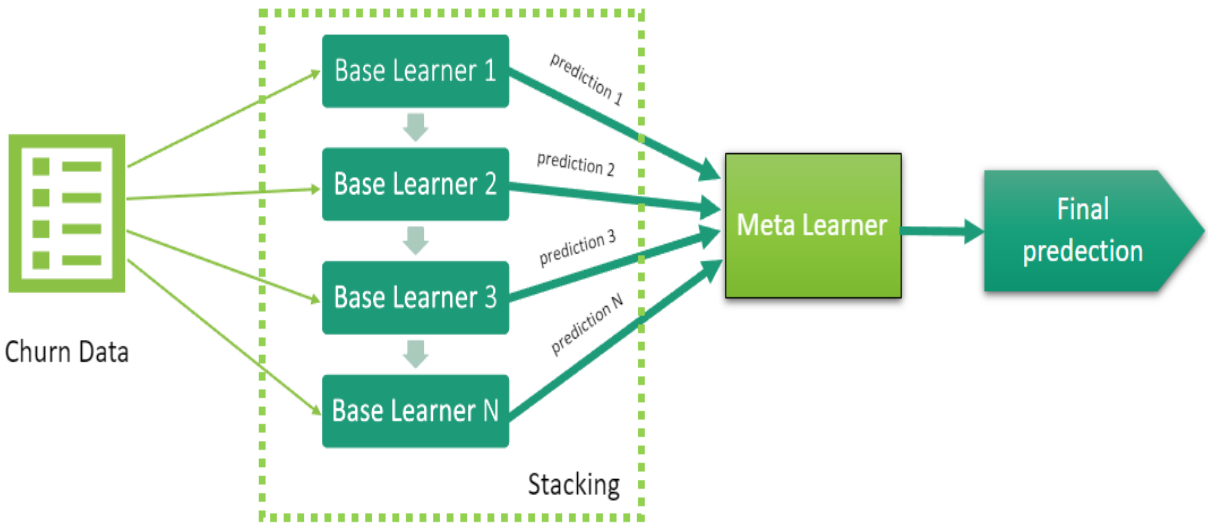

Figure 6. Example of stacking algorithm

Stacking algorithms are an ensemble learning method that combines the decision of different regression or classification algorithms. The component models are trained on the entire training dataset. After these component models are trained, a meta-model is assembled from the different models, and then it's trained on the outputs of the component models. This approach creates a heterogeneous ensemble because the component models are usually different algorithms.

2.5 Voting ensemble

Each model has its strengths and weaknesses, even optimal parameters. To achieve better results, it is better to have teamwork that can combine all the advantages of each model; this technique is called ensemble learning. There are two popular methods of combining individual model predictions to make a final prediction (Fig. 7 and 8): hard voting and soft voting. The first is by the majority, that is, the final prediction – for what more than half of the voters voted.

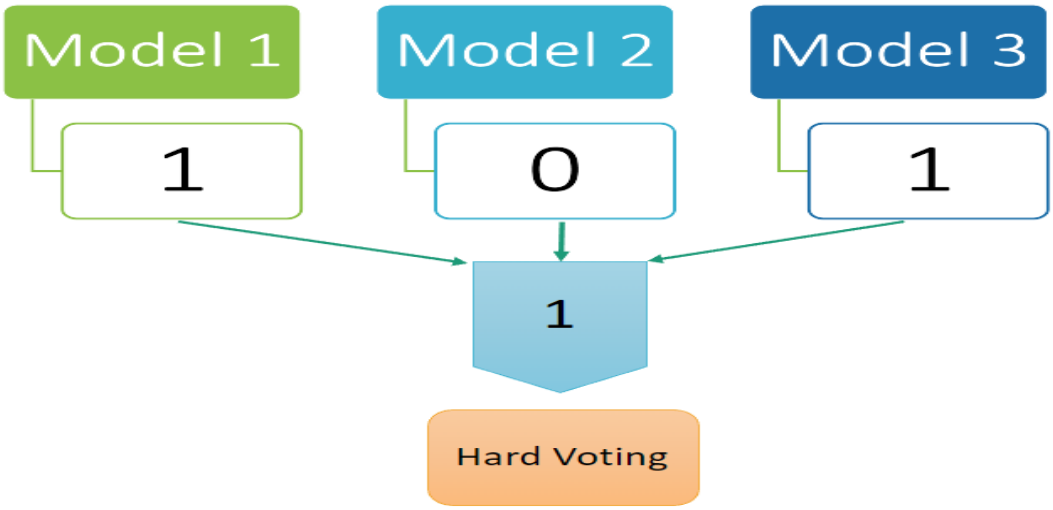

Figure 7. Hard voting classifier

On the other hand, soft voting calculates the average of the class probabilities predicted by the individual models and then makes a final prediction based on the average probability. Although the Soft voting method only works with classifiers that can calculate the probability of outcomes, it allows each classifier to assign weights, meaning that the classifier that performs better gets more votes in the final prediction. In this study, we use a weighted soft voting ensemble model.

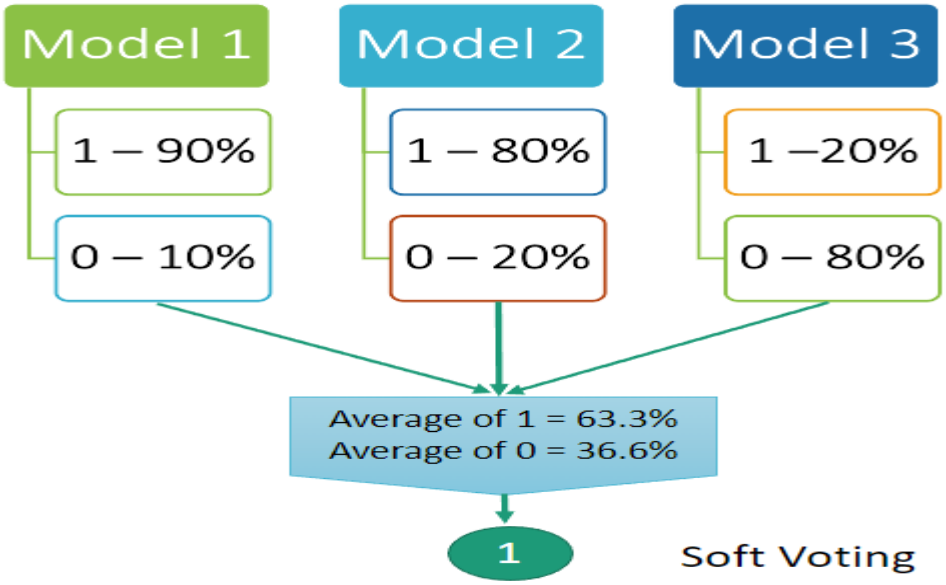

Figure 8. Soft voting classifier

III. Applied methods

First, we processed our dataset and then went for candidates of model selection to create an ensemble model classifier.

3.1. Introduce to dataset

We used the Telco customer churn dataset, which contains information about a telco company that provided home phone and internet services to 7043 customers and 21 variables in California in Q3. There are 21 features in the dataset – "Churn" is the target/dependent variable, and rest 30 are independent variables which we need to explore further. Only 3 features are numeric: "SeniorCitizen" (categorical), and "tenure" and "MonthlyCharges" (continuous). Datatype of the rest of the features is object, looking at the sample data they look like to be of type string. Some of these features are categorical, which we will map into numerical values. We split the data into the training set, validation set, and test set in the ratio of 6:2:2. It indicates which customers have left, stayed, or signed up for their service. Multiple important demographics are included for each customer, as well as a Satisfaction Score, Churn, and Customer Lifetime Value (CLTV) index [32]. The general view of the dataset is shown in (Fig. 9).

#	Variable Name	Definition
1	CustomerID	CustomerID
2	Gender	Whether the customer is a male or a female
3	senior citizen	Whether the customer is a senior citizen or not (1, 0)
4	Partner	Whether the customer has a partner or note (Yes, No)
5	Dependents	Whether the customer has dependents or not (Yes, No)
6	Tenure	Number of months the customer has stayed with the company
7	phone service	.Whether the customer has a phone service or note (Yes, No)
8	Multiple lines	Whether the customer has multiple lines or not (Yes, No, No phone service)

9	Internet service	Customer's internet service provider (DSL, Fiber optic, No.
10	online security	Whether the customer has online security or not (Yes, No, No internet service)
11	online backup	Whether the customer has online backup or not (Yes, No, No internet service)
12	DeviceProtection	Whether the customer has device protection or not (Yes, No, No internet service)
13	TechSupport	Whether the customer has tech support or not (Yes, No, No internet service)
14	StreamingTV	Whether the customer has streaming TV or not (Yes, No, No internet service)
15	StreamingMovies	Whether the customer has streaming movies or not (Yes, No, No internet service)
16	Contract	The contract term of the customer (Month-to-month, One year, Two years)
17	paperless billing	Whether the customer has paperless billing or not (Yes, No)
18	payment method	The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
19	MonthlyCharges	The amount charged to the customer monthly
20	total charges	The total amount charged to the customer
21	Churn	Whether the customer churned or not (yes or no)

Figure 9. Samples from the telco churn prediction dataset

The features of the dataset are the following:

- Customers who left within the last month – the column is called churn;
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies;
- Customer account information – how long they have been a customer, contract, payment method, paperless billing, monthly charges, and total charges;
- Demographic info about customers – gender, age range, and if they have partners and dependents.

3.1.1 Data processing

In the real world, churn datasets typically contain irrelevant information, which includes exterior features, imbalanced datasets, disparate feature scales, non-numeric features, and missing values. We present five procedures in this research to process data before submitting it to training.

3.1.2 Missing values

We implemented several missing value solutions depending on the amount of missing data in the feature. The dataset excludes attributes with more than 90% missing values. For the empty data, column means and modes for categorical and numeric features, accordingly, are substituted in place of incomplete data in the dataset.

3.1.3 Categorical variables

Although many machine-learning algorithms work only with numerical values, many critical real-world features are categorical rather than numerical. As categorical properties, they take degrees or values. Since it cannot directly use these categorical features in most machine learning algorithms, it must convert them into numerical features. Although there are many techniques for changing these properties, the most common method is one hot encoding.

3.1.4 Class imbalance

The classes of churn data in the real world are typically unbalanced; specifically, the ratio of churners to non-churners is frequently substantially smaller. To have the same amount of data for different labels, we applied Synthetic Minority Oversampling Technique (SMOTE) to generate the synthetic data [26], which increased the size of the minority class by random sampling in Shown (Fig. 10 and 11).

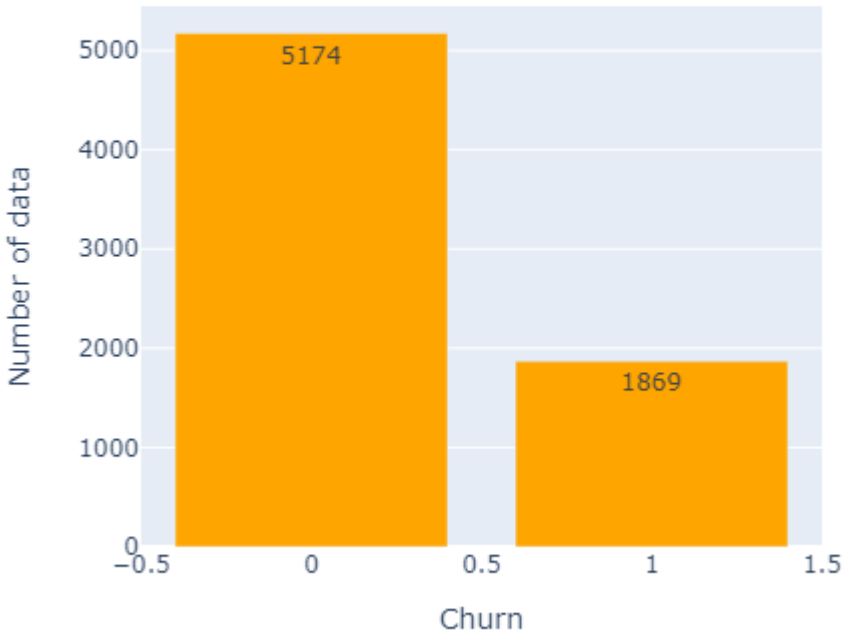

Figure 10. Statistics of churn and non-churn customers

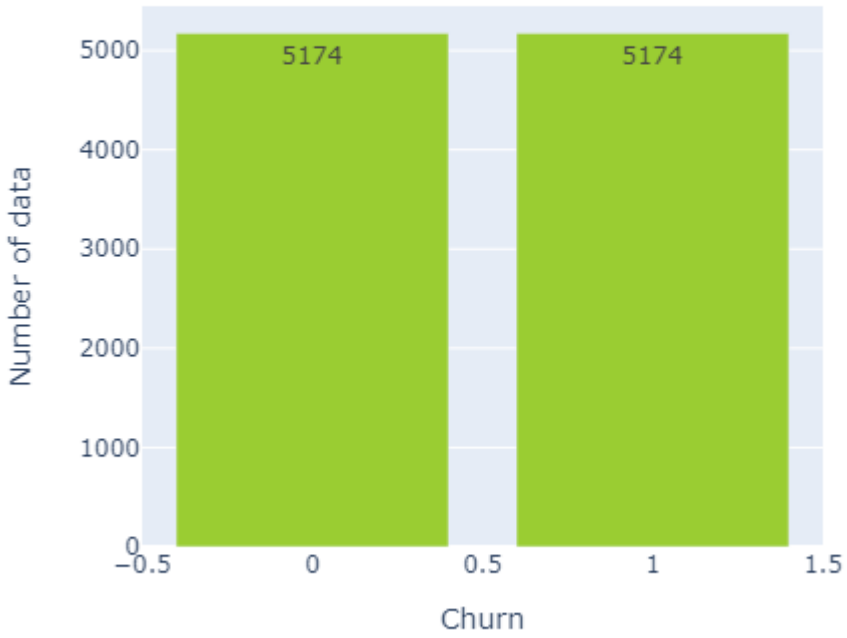

Figure 11. Statistics of churn and non-churn customers after performing smote technique

3.1.4 Feature selection

We select the most favorable features as churn indicators. First, new features are created based on the feature selection algorithm. Secondly, highly correlated features are eliminated since they often increase computing costs without improving model prediction capability. If we had 7043 data with 21 features at the beginning, now we possess 31 characteristics after the feature selection process.

3.1.5 Normalization

Lastly, each feature is scaled through standardization, involving rescaling the parts with the properties of a standard normal distribution with a mean of zero and a standard deviation of one. It helps the network training to converge better and faster, accelerating the model process speed. Min-Max normalization is the process of taking data measured in its engineering units and transforming it to a value between 0.0 and 1.0. Where by the lowest (min) value is set to 0.0 and the highest (max) value is set to 1.0. This provides an easy way to compare values that are measured using different scales or different units of measure. The normalized value is defined as: Equation 1.

$$\text{min\_max\_norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

3.2 Hyperparameter optimization

We used the default hyper-parameters which have been preset by the package in the previous step. However, these hyper-parameters are not guaranteed to be optimal. Therefore, we need a technique to tune the machine learning models and the optimal hyper-parameters for it, which is called hyper-parameter tuning.

Table 2: Hyper-parameters for top models

Model	Hyper-parameter	Description
Catboost	depth=9	Depth of the tree
	iterations=30	The maximum number of trees that can be built when solving machine learning problems.
	learning_rate=0.01	Used for reducing the gradient step
	logging_level= Silent	The logging level to output to stdout. Silent:Do not output any logging information to stdout.
XGBoost	objective=binary: logistic	logistic regression for binary classification, output probability
	eta=0.035	Step size shrinkage used in update to prevents overfitting.
	max_depth=2	Maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit. 0 indicates no limit on depth. Beware that XGBoost aggressively consumes memory when training a deep tree.
	subsample=0.8	Upsample ratio of the training instances. Setting it to 0.5 means that XGBoost would randomly sample half of the training data prior to growing trees. and this will prevent overfitting. Subsampling will occur once in every boosting iteration.
	colsam_bytree=0.9	This is a family of parameters for subsampling of columns.
Neural Network	max_iter=1000	Indicates the number of epochs.
	learning_rate=1	Size of step.
Random Forest	max_depth=2	The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
	n_estimators=50	The number of trees in the forest.
	oob_score=True	Whether to use out-of-bag samples to estimate the generalization score. Only available if bootstrap=True.

Gridsearch is essentially an optimization algorithm which lets you select the best parameters for your optimization problem from a list of parameter options that you provide, hence automating the 'trial-and-error' method.

Although it can be applied to many optimization problems, but it is most popularly known for its use in machine learning to obtain the parameters at which the model gives the best accuracy. All of these four classifiers have been optimized further with the hyper-parameters listed in Table 2.

3.3 Evaluation metrics

There are a few measures that are commonly used within machine learning. Some steps have more than one name, which can lead to confusion, and the user names often depend on the technical area in which they are used. The essential measures used in this study are precision, recall, and F1-score, which in turn rely on the concepts of true positive rate and false positive rate [33]. Equation 2 calculates the accuracy metric. It identifiesa number of instances that were correctly classified.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{2}$$

Here:

- True Positives (TP): predicted positive, true value positive
- False Positives (FP): predicted positive, true value negative
- False Negatives (FN): predicted negative, true value positive
- True Negatives (TN): predicted negative, true value negative

These tells us what portion of the data is correctly classified as positive.

For any classifier, the TP rate must be high. TP rate is calculated by using Equation 3.

$$\text{TP rate} = \frac{True\ Positives}{Actual\ Positives} \tag{3}$$

FP Rate tells us which part of the data are incorrectly classified as positive. The result of the FP rate must be low for anyclassifier. It is calculated by using Equation 4.

$$\text{FP rate} = \frac{False\ Positives}{Actual\ Negatives} \tag{4}$$

Precision, also known as Positive Predictive Value (PPV), indicates which part of the prediction data is positive. It is calculated by using Equation 5.

$$\text{Precision} = \frac{True\ Positive}{(True\ Positive + False\ Positive)} \tag{5}$$

Recall in Machine Learning is defined as the ratio of Positive samples that were properly categorized as Positive to the total number of Positive samples. It is the probability that all the relevant instances are selected by the system. The low value of recall means many false negatives. It is calculated by using Equation 6.

$$\text{Recall} = \frac{True\ Positive}{(True\ Positive + False\ Negative)} \tag{6}$$

The F1-score is a trade-off between correctly classifying all the data points and ensuring that each class contains points of only one class. It is calculated by using Equation 7.

$$\text{F1-score} = 2 * \left(\frac{Precision * Recall}{Precision + Recall} \right) \quad (7)$$

A confusion matrix (CM) is a popular evaluation metric for classification problems. To illustrate the idea, we can think of the classification problem as a binary problem where the instance is classified correctly or not (Fig. 12).

	Predicted	
	Not-churn	Churn
Not-churn	True Positive	False Positive
Churn	False Negative	True Negative
	Not-churn	Churn

Figure 12. Confusion matrix for customer churn prediction

3.4 Receiver operating characteristic curve

A standard method for evaluating performance in churn prediction is to use Receiver Operating Characteristic (ROC) curves. To extract a measure from ROC curves, it is common to use the area under the curve (AUC). This metric can, for instance, be used to compare different types of classifiers to each other or the same kind of classifier but with varying parameter values [34].

Unlike accuracy, AUC is applicable when there is a class imbalance and evaluates the ability of a model to distinguish between classes based on the class membership probabilities. Previous research has also found that AUC is generally a better evaluation metric than accuracy regarding statistical consistency and discrimination.

This makes it a suitable evaluation metric also when the data is balanced. The ROC AUC metric was used in this study to evaluate the final performance of the model.

Area Under the Receiver Operating Characteristics is an experiment analysis for the classification problem for given varying thresholds. AUC (Area Under The Curve) indicates the measurement or degree of distinction, whereas ROC (Receiver Operating Characteristics) is a likelihood curve. It shows how well the model can discriminate between classes. The higher AUC score indicates how well the model at predicting between classes. In our case, The higher the curve, the better the model distinguishes between churn and non-churn cases.

IV. Experimental results

4.1 Introduce to proposed model

The procedure used for churn predection in this study is shown in (Fig. 13). The figure illustrates the techniques and algorithms used in this work. The main goal of our study is to compare two approaches to ensemble learning such as stacking and voting clasifier, to investigate how they improve the perpormance of models to assist with customer churn prediction. Several models can be used as base learners for ensemble learning purpose. To investigate the performance of ensemble methods in customer churn prediction context the following steps were performed. First, individual models are trained with the training data set and then evaluated against the test set. Then, we select the best models with high accuracy and low log loss to create the ensemble model selection. In addition, we give a voting classifier technique as ensemble learning. A soft voting classifier is chosen to search for optimal parameters. When soft voting is used, the final prediction is made based on the average of the class probabilities predicted by each model. This allows assigning weights to each classifier, meaning that a more robust classifier gets more votes in predicting the results. This study proposes an efficient customer-churn prediction model consisting of model selection, a stacking model, and a soft voting approach.

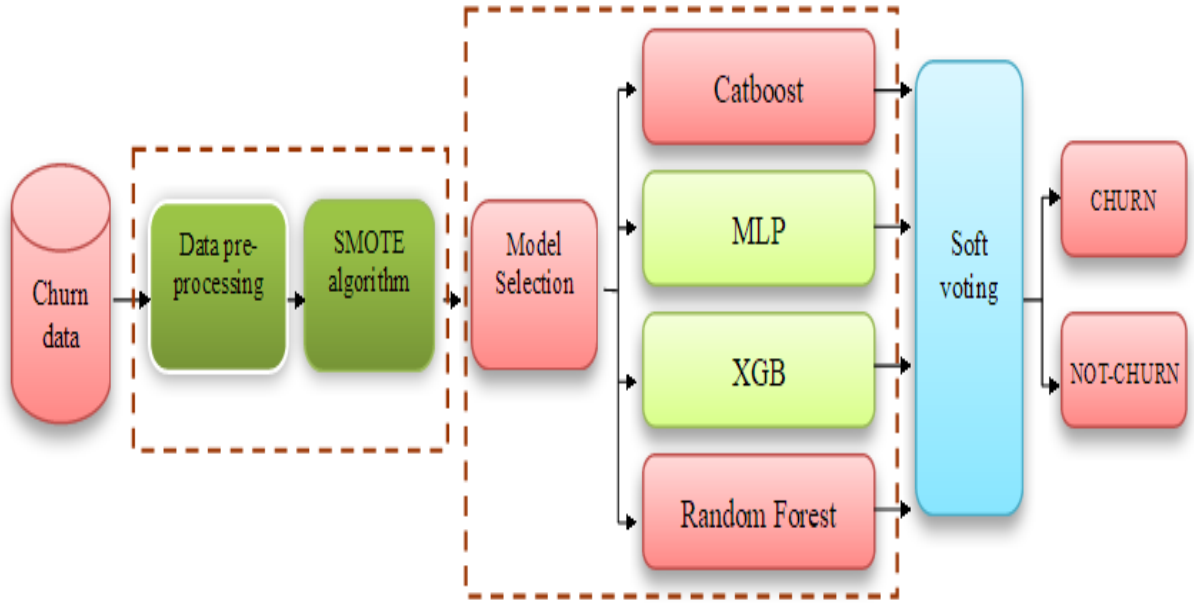

Figure 13. Proposed an efficient stacking ensemble method for customer-churn prediction model

4.2 Results with imbalanced data

Accuracy and AUC scores for all individual models showed in Table 3. The research study indicates that Random Forest, XGBoost, Multilayer Perceptrons (MLPs), and CatBoost are the top four classifiers in terms of performance with

Table 3. Results of individual models

Model	Accuracy	ROC-AUC
CatBoost	0.8452	0.7135
MLP	0.8286	0.7014
XGBoost	0.7397	0.7162
Random Forest	0.7288	0.6994
SVM	0.7161	0.7090
KNN	0.7024	0.6895
Naïve Bayes	0.6753	0.7257

accuracy scores of 0.7288, 0.7397, 0, 8286, and 0.8452, respectively. In summary, preliminary model exploration findings revealed that both Boosting classifiers and Multilayer Perceptrons (MLPs) perform well on this dataset. Compared to these four classifiers, other models showed less performance and efficiency.

4.3 Results with balanced data

In the previous section, we mentioned that the churn prediction dataset differs significantly between churn and non-churn classes, with one class having a higher value than the other one. It showed that after SMOTE data balancing algorithm, minority class “churn” is increased and balanced with “non-churn” classes. The new points added here are synthetically generated points, not exact replications of existing minority class instances. Thus, the overfitting problem caused by random oversampling was solved by SMOTE algorithm. Therefore, we compared model performance before applying SMOTE technique to the training data.

Table 4. Results of smote algorithm before and after balancing the dataset

Model	Accuracy	
	Imbalance dataset	Balanced dataset
CatBoost	0.7936	0.8452
MLP	0.7813	0.8286
XGB	0.7993	0.7397
Random Forest	0.7893	0.7288

The data in Table 4 shows that after SMOTE algorithm, the results of our balanced dataset are much better than the unbalance dataset's results, except for XGBoost and Random Forest.

4.4 Model selection

Furthermore, using those high-performed models, we defined soft voting ensemble models. To choose which models for the ensemble, we compared seven machine-learning models with accuracy and log loss and then got the four top models that showed the highest precision and lowest log loss. As we see from (Fig. 14 and 15), CatBoost, MLP, XGB, and Random Forest achieved the best accuracy and log loss results.

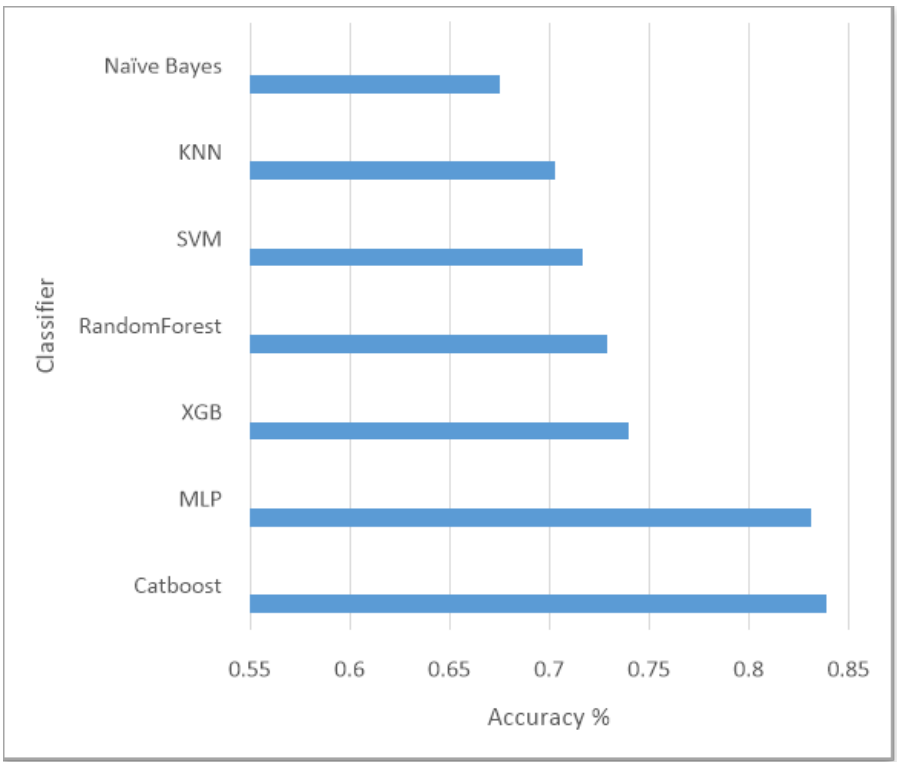

Figure 14 Classifier models in terms of accuracy

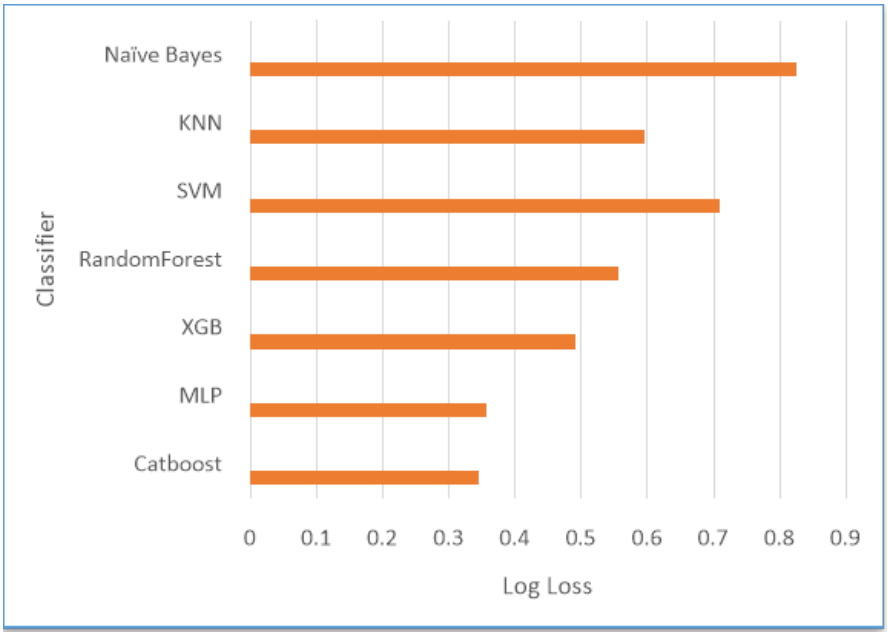

Figure 15. Classifier models in terms of log loss

4.4.1 Soft voting results for combination of the models

Another way of constructing an ensemble classifier is by voting among classifiers. Each independent classifier assigns a class label for each instance. Then using a voting scheme the class label of each instance is determined in this work.

Table 5. Soft voting results for combination of the models

Model selection	Models	Accuracy%	Precision	Recall	F1_score
1	1, 2, 3, 4	0.7804	0.5514	0.8940	0.7926
2	2, 3, 4, 5	0.8457	0.6471	0.9120	0.8527
3	3, 4, 5, 6	0.8622	0.6852	0.8833	0.8672
Proposed Model	4, 5, 6, 7	0.8820	0.7088	0.8958	0.8797
1. Naïve Bayes 2. KNN 3. SVM 4. Random Forest 5. XGB 6. MLP 7. CatBoost					

Additionally, models with high accuracy, precision, recall, and f1-score are chosen in the soft voting process to achieve better results. Seven individual model combinations are calculated to implement the soft voting method. In Table 5, we can see that the particular model results are compared. Our proposed ensemble model achieved better results than other individual models.

4.4.2 Comparison with other works

Table 6. Compares the proposed an efficient stacking ensemble method with other works recently. The proposed model shows the best accuracy.

Table 6. Comparison with other works

Works	Model	Accuracy %
Afifah Ratna Safitri [35]	Using smote and genetic algorithms	78.46%
Takuma Kimura [36]	Hybrid resampling and ensemble learning	77.10%
M. Imron [37]	Z-score normalization and particle swarm optimization	82.50%
Our work	An efficient stacking ensemble method	88.20%

4.6 Further analysis

It is only sometimes helpful to calculate the accuracy of the metric score, especially for unbalanced data. Therefore, to compare the different algorithms more clearly, we present a graphical type of confusion matrix and ROC-AUC curve analysis in (Fig. 16 and 17), respectively. As can be seen from these two graphs, our proposed ensemble model showed the best performance and balance in false positive (FP) and false negative (FN) rates, respectively.

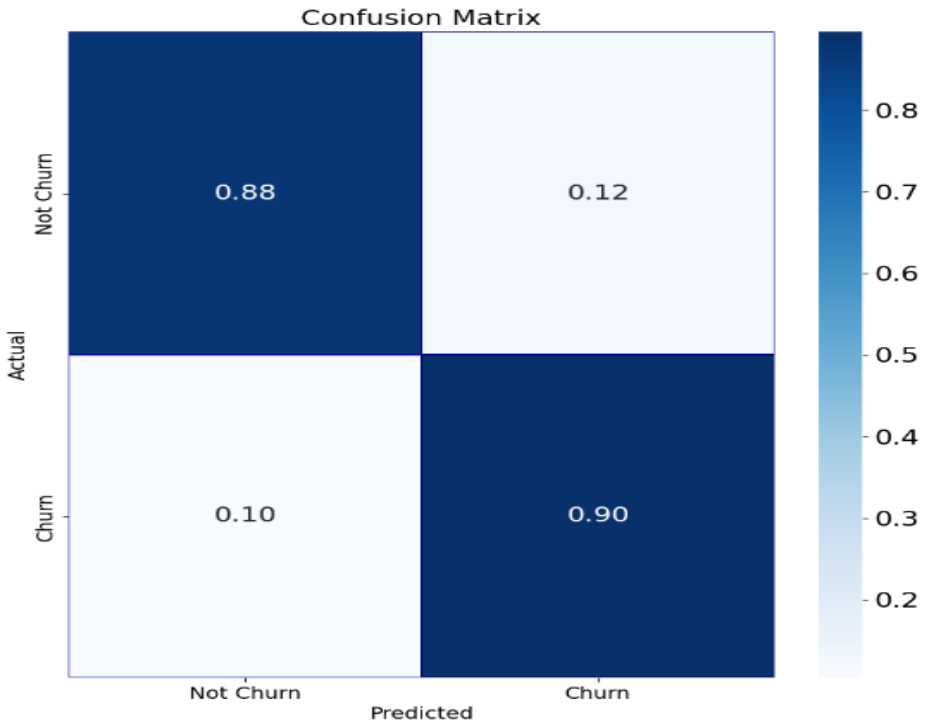

Figure 16. Confusion matrix for proposed ensemble model

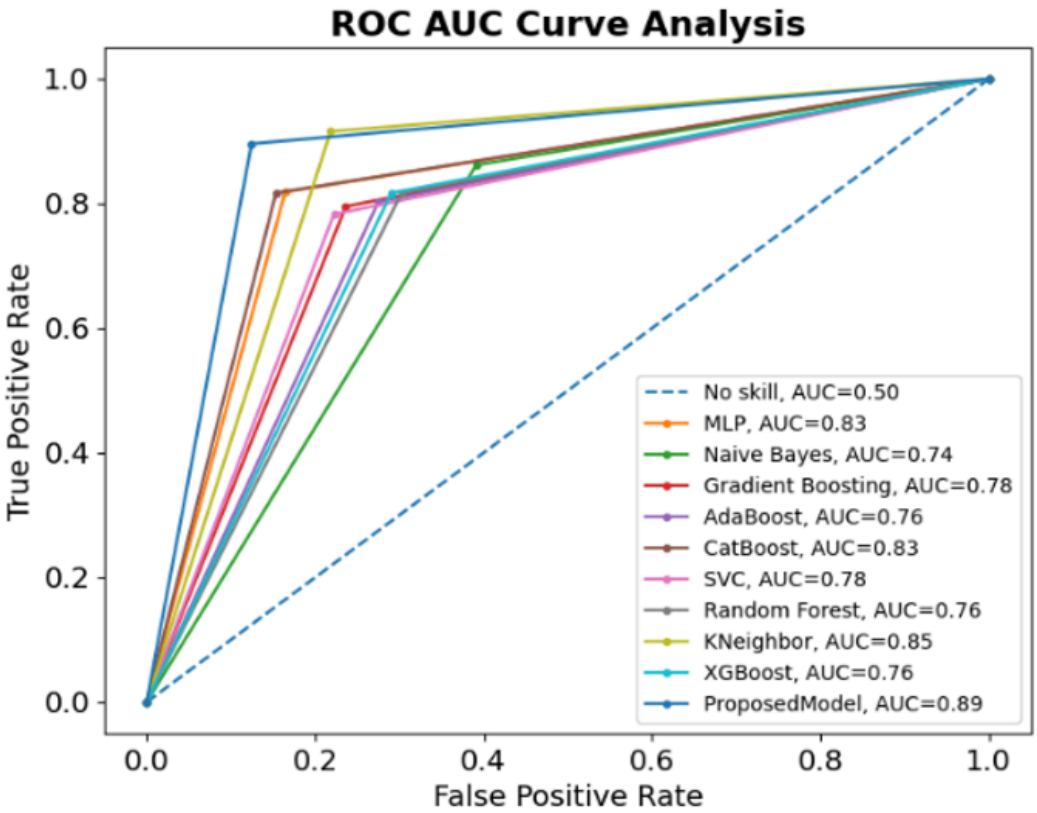

Figure 17. Comparison of roc auc curves for our work and other ml models

V. Conclusions

Nowadays, various machine-learning techniques have been used in telecommunications for customer churn. Our research comprehensively measures the most popular state-of-the-art machine learning methods. Quality measurements of all candidate models were evaluated for the public data set in the telecom industry. We have chosen the top four models-MLP, Random Forest, CatBoost, and XGBoost-based on an efficient stacking churn prediction model. We eventually created an ensemble model utilizing the soft voting approach in addition to these four separate models with the best hyper-parameters, which had the best AUC score. The result of the proposed model showed the best accuracy of 78.79% and 88.20% for the original imbalance dataset and the balanced dataset, respectively, compared to other prediction models. This proposed model can provide early detection of customer churn in the telecom industry.

References

[1] O. Adwan, H. Faris, K. Jaradat, O. Harfoushi and N. Ghatasheh, ”Predicting customer churn in telecom industry using multilayer perceptron neural networks: modeling and analysis," Igarss. 2014

[2] X. Liang, S. Chen, C. Chen, & T. Zhang, "Research on Telecom Customer Churn Prediction Method Based on Data Mining ."CCF Conference on Computer Supported Cooperative Work and Social Computing, 2019.

[3] A. Mishra, and U. Reddy, "A comparative study of customer churn prediction in telecom industry using ensemble based classifiers."2017 International Conference on Inventive Computing and Informatics (ICICI). IEEE, 2017.

[4] M. Singh, S. Singh, N. Seen, S. Kaushal, & H. Kumar, "Comparison of learning techniques predictingn of customer churn in telecommunication", IEEE. 2018.

[5] S. Raeisi, and H. Sajedi, "E-Commerce Customer Churn Prediction By Gradient Boosted Trees," IEEE, 2020.

[6] T. Chen & C. Guestrin, Xgboost: "A scalable tree boosting system," Proceedings of the 22nd acm signed international conference on knowledge discovery and data mining, 2016.

[7] J. Hadden, A. Tiwari, R. Roy, and D. Ruta, "Churn prediction using complaints data," in Proceedings Of World Academy Of Science, Engineering and Technology, 2006.

[8] I. Brandusoiu, G. Toderean, B. Ha "Methods for churn prediction in the prepaid mobile telecommunications industry," In International conference on communication, 2016.

[9] A. K. Ahmad, A. Jafar, and Kadan Aljoumaa," Customer churn prediction in telecom using machine learning in big data platform," J Big Data, 2019.

[10] Y. Sun, A.K. Wong, M.S. Kamel, Classification of imbalanced data: A review, International Journal of Pattern Recognition and Artificial Intelligence 23(4), 2009.

[11] J. Wang, M. Xu, H. Wang, and J. Zhang, "Classification of imbalanced data by using the smote algorithm and locally linear embedding," Proc. 8th Int. Conf. Signal Process, 2006.

[12] C. S. Ertekin, "Adaptive oversampling for imbalanced data classification," Proc. 28th Int. Symp. Comput. Inf. Sci., vol. 264, pp. 261-269, Sep. 2013.

[13] D. Burez and V. den Poel, "Handling class imbalance in customer churn prediction," Expert Syst Appl, 2009.

[14] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, A. Hussain, "Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study," IEEE Access, 2016.

[15] T. Zhang, S. Moro, R. F. Ramos, "A Data-Driven Approach to Improve Customer Churn Prediction Based on Telecom Customer Segmentation," Future Internet, 2022.

[16] V. Dorogush, V. Ershov "CatBoost: a gradient boosting with support for categorical features," 2018.

[17] J. T. Hancock & T. M. Khoshgofar, "CatBoost for Big Data: An Interdisciplinary Review ."Journal of Big Data, 2020.

[18] Sami Ben Jabeura, Cheima Gharib, Salma Mefteh-Wali, Visual Ben Arfi "CatBoost Model and Artificial Intelligence Techniques for Corporate Failure

Prediction," Technological Forecasting and Social Change, 2021.

[19] S. B. Kotsiantis, “Supervised Machine Learning: A Review of Classification Techniques,” 2007.

[20] R. Saravanan and P. Sujatha, “A State of Art Techniques on Machine Learning Algorithms: APerspective of Supervised Learning Approaches in Data Classification,” in 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018.

[21] P. Flach, Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge: Cambridge University Press, 2012.

[22] Alekhya Kanneganti, “Using Ensemble Machine Learning Methods in Estimating Software Development Effort," 2020.

[23] Webb G.I. Naïve Bayes. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA, 2011.

[24] A. Keramatia, R. Jafari-Maranda & I. Ahmadianc, “Addressing churn prediction problem with Meta-heuristic, Machine learning, Neural Network, and data mining techniques: a case study of a telecommunication company. Metaheuristics and Engineering”, 2014.

[25] Zhang K, Wei Z, Nie Y, “Comprehensive analysis of clinical logistic and machine learning-based models for the evaluation of pulmonary nodules ."JTO Clin Res Rep. 2022.

[26] L. Landryová, J. Sikora, R. Wagnerová, “The Learning Path to Neural Network Industrial Application in Distributed Environments ." Proceedings, 2021.

[27] C. Jair, G. Farid, R. Lisbeth, L. Asdrubal, “A comprehensive survey on Support Vector Machine classification: Applications, challenges, and trends, Neurocomputing, Volume 408”, 2020.

[28] L. Breiman, Random Forests. Machine Learning 45, 5–32, 2001.

[29] T. Chen, C. Guestrin, XGBoost: ”A Scalable Tree Boosting System ."In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[30] N. Chawla, W. Bowyer, & W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique ." Journal of Artificial Intelligence Research, 2020.

[31] Z. Zhou, Ensemble Methods: Foundation Sand Algorithms, CRC Press: Boca Raton, FL, USA, 2012

[32] IBM. "Predict behavior to retain customers. You can analyze all relevant customer data and develop focused customer retention programs", 2019.

[33] Max Bramer, “Principles of Data Mining," Springer London, 2 edition, 2013.

[34] Koen W, De Bock, and Dirk Van den Poel, “Reconciling performance and interpretability in customer churn prediction using ensemble learning based on generalized additive models”, Expert Systems with Applications, 2012.

[35] Afifah Ratna Safitri and M. A. Muslim, “Improved accuracy of Naive Bayes classifier for determination of customer churn uses SMOTE and genetic algorithms”, Journal of Soft Computing Exploration, 2020.

[36] T. Kimura, “Customer churn prediction with hybrid resampling and ensemble learning”. Journal of Management Information and Decision Sciences, 2022.

[37] M. Imron , Budi Prasetyo, "Improving algorithm accuracy k-nearest neighbor using Z-score normalization and particle swarm optimization to predict customer Churn", Journal of Soft Computing Exploration, 2020.

Korean Abstract

고객 이탈 예측을 위한 효율적인 스태킹 앙상블 학습 방법

Mukhammadiev Komiljon Jahongir ugli

전남대학교대학원 컴퓨터공학과

(지도교수 : 임창균)

최근 몇 년 동안 고객 이탈은 통신 회사의 중요한 문제이자 가장 중요한 관심사 중 하나였습니다. 신규 고객을 확보하기보다는 기존 고객을 유지하는 데 집중하는 것은 통신 산업에서 비용을 절감하고 수익을 늘리는 데 중요한 전략입니다. 이 연구에서는 불균형 데이터, 누적 모델 및 소프트 투표를 위한 SMOTE(합성 소수 과잉 샘플링) 기술로 구성된 앙상블 학습 기술을 사용하는 효율적인 고객 이탈 예측 모델을 제안합니다. 랜덤 포레스트, 익스트림 그래디언트 부스팅(XGBoost), Catboost 및 MLP(MultiLayer Perceptron) 머신 러닝 알고리즘을 선택하여 스태킹 앙상블 모델을 구축하고 네 가지 알고리즘의 결과를 소프트 투표에 사용합니다.다른 예측 모델과 비교하여 제안된 모델은 원래 불균형 데이터 세트와 새로운 균형 데이터 세트에 대해 각각 78.79% 및 88.20%의 최고의 정확도를 보였습니다. 우리가 제안한 모델은 통신 산업에서 고객 이탈을 조기에 감지할 수 있습니다.

ACKNOWLEDGEMENT

I want to express my deep gratitude to my supervisor, Professor Chang Gyoon Lim, during the study time. The supervisor professor has provided valuable suggestions and plans for my thesis. The professor not only patiently answers the technical questions in the thesis but also guides the revision of the thesis.

Secondly, I sincerely thank our department's professors, teaching assistants, and Ph.D. student Senfeng Cen. They provided constructive suggestions in the process of correcting the dissertation is complete at this stage.

Finally, I sincerely thank my parents for their support and encouragement. With your hard work, I could complete my dissertation.