Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7778561

PREDICTING CUSTOMER CHURN IN TELECOMMUNICATIONS SECTOR USING MACHINE LEARNING METHODSWITH MAJORITY OF VOTING CLASSIFIERS

Subhash Chandra Jat
Computer Science and Engineering, Rajasthan College of Engineering for Women, Jaipur Rajasthan, India, subhashccjat@yahoo.com

Dazy Arya
Computer Science and Engineering, Rajasthan College of Engineering for Women, Jaipur Rajasthan, India, dazyarya@gmail.com

Abstract—The loss of customers is an issue for businesses of all kinds. When a client leaves, it is a big deal for the business. The phrase "churn prediction (CP)" is used to describe the process of figuring out which customers are most likely to terminate a service subscription. For many businesses, this forecast is especially important because acquiring new clients can be more costly than maintaining current ones. There is a current problem with customer churn (CC) research and prediction in the telecommunications industry since it is crucial for these businesses to know which consumers are likely to cancel their service. Machine learning (ML) strategies and models play a big role for businesses in the present financial environment. This is because the cost of obtaining a new customer is higher than the expense of maintaining an existing one. This project focuses on several ML algorithms for forecasting customer turnover using the telecom CC dataset from Kaggle. Our proposed approach has six facets. The first two sides are dedicated to preliminary data processing and exploratory data analysis (EDA). After considering data balance with SMOTE in the third stage, the data is split into a trainset and a testset with a 70%:30% split. On the train set, we employed the most popular prediction models, including Extra tree, Decision Tree (DT), XGBoost, and Voting, and we compared their accuracy using ensemble methods. A final analysis of the test set results in python simulator was performed using accuracy, precision, recall, & the f1-score curve. It was found that Proposed Voting had the highest accuracy, at 85.63 percent. Finally, the results of the experiments illustrate how the suggested system performs in comparison to other systems.

Keywords- Customer Churn Prediction, Telecom Industry, Machine Learning, Extra Tree, Decision Tree, Xgboost, And Voting Classifiers.

• Introduction

The telecommunications sector is a vital part of today's advanced economies. Competition has enhanced as a result of new technologies and a rise in the number of businesses offering similar services[1]. Companies are employing a wide range of techniques in an effort to succeed in this market. There are three basic approaches offered to increase earnings [2]: One must get new customers, two must upsell current ones, and three must lengthen the time customers stay around. Although the first two techniques have higher initial costs, the third option offers the highest return on investment (RoI), demonstrating that keeping an existing client costs significantly less than finding a new one [3], Additionally, it is simpler than the upselling technique [4]. clients are the most valuable resource for every business, as they are the primary generator of revenue. These days, businesses are aware that they need to work hard to keep their current consumers happy in addition to winning over new ones. People who "churn" from one firm to another do so for a variety of reasons. Customers are less likely to leave a business that can accurately forecast their actions and identify causal elements in CC. Predicting who will churn and who will not is a simple case of binary classification [5]. In order for businesses to successfully implement the third approach, "the clients movement from one provider to another" (also known as "churn") must be reduced [6][7].

In service industries that are highly competitive with one another, the loss of customers is a significant source of anxiety. On the other hand, if done in the early part of the process, anticipating the consumers who are likely to quit the organisation would offer a potentially big extra source of revenue for the business [3].The process of people switching from one financial institution to another is what is known as CC. Common factors that lead to CC are dissatisfaction with the level of service received, heavy fees, limited plan alternatives, and astronomical prices. The cost of acquiring new clients is five to six times greater than the cost of keeping existing ones, making this an expensive problem for businesses of all sizes. [8]. Predicting which customers are most likely to depart your organization and why is an enormous chance for growth. Losing a client means not only losing their future business but also the possibility that the customer's spending has not yet recouped the expenditures of recruiting them. Predicting which customers may churn from your business is the primary goal of CC analysis. The banking business has to understand the causes of churn in order to keep their current clientele. This understanding may be gained via analysis of collected data [5].Numerous studies demonstrated that the use of ML technologies is particularly effective in forecasting events like this one. This method is implemented by drawing conclusions from previously collected data[9][10].A client that churns might be labelled as such without considering the issues that led to their departure. Not all churn patterns are created equal, and they should not be handled in the same way. Some consumers may not churn readily than others. There is a pressing demand in the market today for a more accurate model of forecasting that can anticipate client attrition. The ability to target specific groups of customers with effective retention measures, such as tailored incentives based on their unique churn drivers, is a huge competitive advantage. Motivated by the data and findings shown above, this research proposes a model for CP using multiple ML techniques.

- Research motivation and contribution

The mainaim of this approach is to conduct an investigation into the numerous ML techniques that are necessary for the development of CC prediction models. Additionally, it seeks to determine the factors that contribute to CC in order to provide these factors with retention plans and strategies. The proposed system collects telco customers' data by applying ML classification techniques like Voting, extra tree, XGBoost, and decision tree Classifier. The fundamental objective for CP in the telecom business is to decrease churns while keeping current customers. The study presents a useful business approach for preventing customer churn and financial loss by analysing CC data and providing precise estimates of CC. Because of this research, churn prediction (CP) will be improved, more company industries will be investigated, and prediction models to keep their present consumers will be developed. The thorough study of several machine learning methods to develop a CP model and solve the problem of CP in the telecom industry aids in improving model accuracy and scope to forecast churn in other business sectors. Customers' data and activity patterns will be analyzed using machine learning techniques to help in establishment of retention strategies, improve customer service, and avoid churn. The research contribution that are given below:

- The telco customer churn dataset from the Kaggle repository was utilised to conduct the studies presented in this work. Each client is represented by a row, and each attribute is listed in the corresponding column. There are details on 7043 customers included. The "Churn" spreadsheet is missing 11 numbers in the Total Charges column despite the fact that every client has access to all 21 functions.
- Classifying and preprocessing data eliminates clutter and streamlines its use. In order to get clean, meaningful data from the actual world, a data pre-processing phase is necessary. Data cleanliness and usability are greatly improved by classification or pre-processing.
- The collected telco dataset is highly imabalnced, so solve the problem of class imbalance used SMOTE oversampling techniques.
- Our recommended method split the data in half, creating a training set and a test set. 70% of the data set is used to train the algorithms in what is called the training group. Algorithms are put through their paces on the 30% of the dataset that makes up the "test group."
- In this work used four machine learning classifiatcion techniques such as extra tree classifier, decision tree classifier, XGBoost classifier and Voting classifiers.
- To evaluate the machine learning model efficiency in terms of accuracy, precision, recall, f1-socre and ROC. Also shows the proposed model results in confusion metrix and classification report.
- The comparative analysis shows the proposed models outperformance in term of base models with all measures.

The remainder of the paper is structured as follows: Sections II and III explore related research and approaches, with a focus on the reliability of the most advanced techniques. Section IV details the investigation's findings to emphasise its originality. The inquiry is concluded in the fifth portion.

• Literature Review

Predicting CC in the telecom sector has been the subject of several techniques. In most cases, ML and data mining have been applied. Most previous efforts in this area have centred on employing a single data mining approach to draw conclusions, while others have compared multiple methods to churn prediction.

In order to increase profits and market share, several different telecommunications businesses are now focusing on differentiating between highly valued customers and prospective churners. As part of this study[11], They use several data mining techniques to foresee the CC behaviour. In the long run, it will aid in dissecting customer habits to determine if they are likely to churn. They use data sets hosted in the Kaggle repository, and we used a variety of methods, such as bagging methods, to predict customer behaviour with an accuracy of 99.8 percent.

This paper proposes [12], CC forecasting in the telecoms sector using a combination of classification and clustering approaches to detect and understand the root reasons of CC. Features are selected using a ranking filter based on data acquisition and correlation attributes. CC data is first classified using the Random Forest (RF) and other classification approaches in the proposed model. method did particularly well in this task, successfully classifying occurrences in 88.63% of all tests. Utilising the RF method and k-means clustering, our findings show that our suggested CP model improved churn categorization.

The proposed work presents [13], efficient ML strategy that makes use of both boosting methods and a Gaussian mixture model to classify churned consumers into distinct groups. The suggested research also includes the use of the Light Gradient Boosting (LightGB) model, which is 15 times more effective than Adaboost and XGBoost in CP. In addition, the clustering option in the Gaussian Mixture Model (GMM) produced a silhouette score of 0.36.

Methods from the field of ML, such as those used to construct the model shown here, were [14], These include the NB method, the SVM, the MLP, as well as the RF. By combining the Information Gain and Ranker methods, this research proposes a fresh strategy for feature selection. With feature selection, performance improved to 95.02% from 92.92%. In terms of precision and F-measure, their algorithms' results were on level with those of preexisting methods.

This paper presents[15],An approach to CP in the telecommunications sector that seeks to improve the accuracy with which potentially churning consumers are identified. We use three freely available datasets to train 7 models. The experimental findings of this research demonstrate remarkable achievements compared to current works in terms of recall, F1-score, & AUC-score. Then, an explainable AI method called SHapley Additive exPlanations (SHAP) is used to provide an explanation for the top models.

The existing literature on churn prediction in the telecommunications industry demonstrates a substantial focus on leveraging diverse data mining and ML techniques to identify and characterize customer churn. While several studies have achieved high accuracy levels using algorithms such as Bagging, RF, and boosting algorithms like LGBM, there is a notable gap in the comprehensive exploration and comparison of these methods across different datasets. Moreover, there is a need for more in-depth exploration of feature selection methods, as highlighted in some studies using Information Gain, Ranker, and innovative combinations, to enhance the predictive performance and generalization of the models. Furthermore, the literature suggests that the incorporation of explainable AI techniques, like SHapley Additive exPlanations (SHAP), is crucial for providing insights into the identified churn predictors and enhancing the model's interpretability. Overall, the existing research provides a foundation for

CP in the telecom industry but leaves room for more comprehensive comparisons, interpretability considerations, and advanced data balancing, scalling, and machine learning techniques.

- **Problem Discription**

Given the significant financial consequences of losing customers, organisations in the telecommunications industry have a key difficulty in expecting customer churn. The complex nature of consumer behaviour research is a major problem because identifying the causes of churn necessitates a sophisticated comprehension of many datasets. Previous research has mostly concentrated on the use of machine learning methods; nevertheless, organisations have encountered difficulties in deriving actionable insights from forecasts due to issues with these models' interpretability. The unbalanced character of churn datasets—where a lower percentage of customers leave than do—presents another difficulty, necessitating careful consideration of sampling strategies in order to achieve efficient model training. Additional problems include the scalability of churn prediction algorithms to big customer bases and their generalizability across various telecom datasets. The difficulty with controlling CC in the telecommunications industry is exacerbated by the necessity for strong methods for choosing features to improve model performance and the absence of consistent assessment measures across studies. Furthermore, the dynamic nature of the telecom sector makes it difficult to modify predictive models according to changing market dynamics and the customer behaviour. It is essential to tackle these obstacles in order to devise pragmatic and efficient measures aimed at reducing customer loss and the consequent financial consequences.

- **Research Methodology**

The customer churn prediction task's suggested technique includes a thorough set of stages that make use of a range of machine learning classifiers within a Voting Classifier ensemble. First, the Telco Customer Churn dataset is imported and goes through a thorough preparation step where missing values are resolved, categorical characteristics are converted, and class imbalance is handled using methods like SMOTE. A thorough exploratory data analysis is carried out in order to learn more about the properties of the dataset. Then, using RandomizedSearchCV, the hyperparameters of three different classifiers—XGBoost, Extra Tree, and Decision Tree—are individually adjusted. To improve forecast accuracy, the ensemble model—which was built with the help of a voting classifier—combines the advantages of these fine-tuned classifiers. The suggested technique makes use of metrics including confusion matrices, classification reports, and ROC curves to provide comprehensive assessments of each classifier's performance on training and testing datasets. The goal of this multi-step process is to develop a reliable and accurate model for predicting CC that can efficiently use the unique characteristics of each classifier within of an individual ensemble context. The methodlogy process shows in figure 1, and their description provide in subsections and results section.

Figure 1: Flowchart of stages of proposed work

Proposed System

- Step1->Loading Data

- Step2->Data preprocessing, Feature scalling, labelling, and Data Visualization
- Step3->Splitting train and test Data.
- Step4->Subsequently, employ the train-test split procedure to assess the predictive ability of ML techniques (Extra Tress, DT, XGBoost, and Voting) on telco data.
- Step5->Representing the results using Bar plot, confusion metrix, classification report and ROC.

- **Data Pre-Processing**

The pre-processing of data is the method for making the raw data ready for deployment in a ML approach. The pre-processing is the initial and most significant step in establishing any type of ML approach. Pre-processing pertains to the modifications made to our data before giving it to an algorithm for processing. Data The raw data is cleaned up via a procedure called pre-processing.

ML is used to forecast CCC, and the Telco CC dataset is loaded from Kaggle during the preparation phase. Following this, an initial analysis is carried out in order to comprehend the dataset's hierarchical organisation. The missing values in the dataset, in particular in the "TotalCharges" column, are discovered and handled using approaches that are appropriate for the situation. In order to simplify the dataset, unnecessary columns have been removed. For the purpose of gaining new insights, exploratory data analysis is performed, and label encoding is used to transform category columns into numeric ones. In order to ensure that the scales of the numerical characteristics are consistent, standard scaling is employed. The dataset is then dissected into X-dimensional features and Y-dimensional observations. The problem of social stratification may be tackled with the use of the SMOTE. Finally, the dataset is distritributed into a training set and a testing set to lay the framework for using ML methods to predict CC.

- **Label Encoding**

Several machine learning algorithms need numeric data, and Label Encoding is a technique for converting category columns into numbers. It is a vital part of any ML effort that comes before the actual learning itself.

- **Feature Scaling**

Feature scaling is a technique used in data preparation to normalise the dataset's independent variables to a specified range. Values are standardised around the mean with a standard deviation of one in order to facilitate comparisons. The characteristics will be rescaled so that the mean and standard deviation are both zero. Mean StanderScaler was the standardisation technique we utilised to scale our data. This is the equation used for standardization:

In this context, μ represents the mean and σ signifies the σ standard deviation of the feature values.

- **SMOTE solve Class Imbalance Problem**

These prior probabilities are taken for granted by the majority of ML techniques. This assumption is, however, broken in a wide variety of practical contexts. When processing data sets with a class imbalance, the ML classifier is more likely to incorrectly categorise members of the minority class because of its bias towards the majority class. In such situations, the majority of the instances have one label, while the minority have the other label[16][17].

Synthetic Minority Oversampling Technique (SMOTE): One common approach to rebalancing sample sizes in unbalanced datasets is the SMOTE[18].Under-sampling involves eliminating or reducing data from the majority class while over-sampling involves adding data from the minority class to achieve a more equitable class distribution. Combining or hybrid sampling strategies of under-sampling and over sampling may create a more even data distribution. Since under-sampling will exclude data in the majority class, leading to a loss of valuable data, over-sampling is typically preferred[19]. The SMOTE method was applied to deal with the imbalanced data, which equalised the quantity of samples across classes and enhanced the classification algorithm's performance. Figure 2 depicts the steps taken throughout the process of execution.

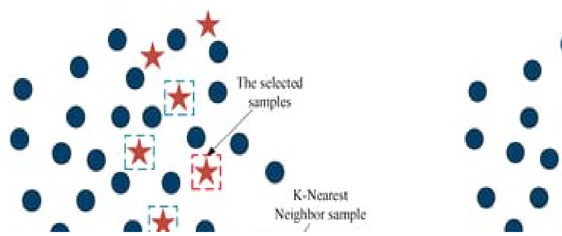


Figure 2: (a) Calculating the proximity of the selected sample K; (b) Synthesizing new samples.

First, a sample xi was drawn at random from a subset of samples. Second, using sampling multiplicity N, K neighbours of sample xi were randomly selected to get N samples xi. Finally, the following synthesising equation was used to generate samples with a random order between $\tilde{x}$ and xi:

where xnewshows the "synthetic" sample, $\tilde{x}$ stands for the minority sample, xi represents the sample from the selected K-neighborhood, and rand(0,1) generates a random number between 0 & 1.

- **Data splitting**

The test-train split methodology is done for examining the fulfillment of the ML methods.It is used in the categorical techniques of supervised learning. In this approach, we begin with a dataset and divide it in half. For initial project fitting, the "training dataset" is used. The following section is not used to train the task, but rather serves as input; the project uses this information to make predictions and derive differentiations depending on how well those predictions match the target values. This sort of dataset is referred to as the "testing dataset". Our proposed technique involves splitting the data into a training set and a test set. 70% of the data set is employed to train the techniques in what is called the training group. Techniques are evaluated using a subset (30%) of the full dataset.

- **Training Dataset:** It is utilized to fit our ML project. Here we used 70% training dataset.

- **Testing Dataset:** It is utilized to examine our fitted ML project. Here we used 30% testing dataset.

- **Classification Technique (Machine learning)**

The Classification type of methodology is a Supervised Learning method that utilizes the training data for determining the categories of new observation. Classification is the method of acquiring the software knowledge from numerous observations or a dataset and then we execute the classification of the new observations into one among the several groupings or categories. Those groupings include 0 or 1and many more.

In order to improve predicted accuracy and resilience, a Voting Classifier is used in the categorization of churn. This particular classifier combines the advantages of the XGBoost Classifier, the ET Classifier, and the DT Classifier.

- **XGBoost Classifier**

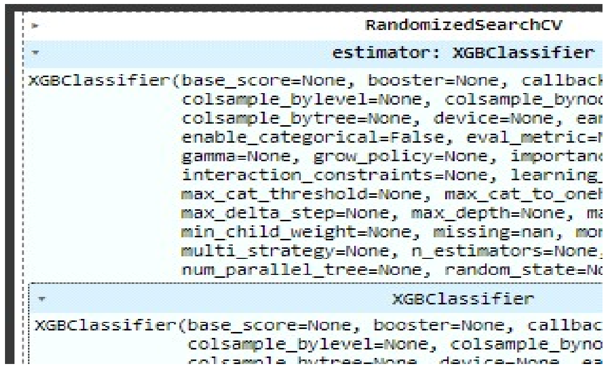
XGBoost[20],is an acronym that stands for "Extreme Gradient Boosting" and refers to a method that is based on the idea of "gradient boosting trees."[21]. The objective function, which is expressed as an equation with two sections labelled training loss and regularisation term, is the primary distinction between this method and other gradient boosting-based approaches.

Where, $\Omega(f)$=

The prediction ability of the model is determine in terms of the training data via the training loss. To enhance model generalisation, the regularisation term regulates model complexity. The mean squared error (MSE) is a popular option for the training loss. In the case of XGBoost, the polynomials loss function is expanded using the Taylor expansion up to the second order.

Implimentaion process:

- In this we performing hyperparameter tuning for an XGBoost Classifier.
- First, a dictionary par_am_gr_id is defined with possible values for three parameters: n_estimators, max_depth, and learning_rate. These parameters control the number of gradient boosted trees, maximum depth of each tree, and the learning rate for the model respectively.
- Next, an instance of the XGBClassifier class is created and stored in xgb. This is the base model that will be tuned.
- Then, an instance of the RandomizedSearchCV class is created with xgb and par_am_gr_id as arguments. This will perform a randomized search over the parameter grid to find the best parameters for the model. The argument cv=10 specifies that 10-fold cross-validation should be used during this process.
- Finally, Training data (X_tra_in, Y_tra_in) is sent into the fit method of the RandomizedSearchCV instance. This begins the process of training and tuning the model. The best parameters found during this process will be used in the final model.



```
RandomizedSearchCV
estimator: XGBClassifier
XGBClassifier(base_score=None, booster=None, callback
              colsample_bylevel=None, colsample_bynod
              colsample_bytree=None, device=None, ear
              enable_categorical=False, eval_metric=N
              gamma=None, grow_policy=None, importanc
              interaction_constraints=None, learning_
              max_cat_threshold=None, max_cat_to_oneh
              max_delta_step=None, max_depth=None, ma
              min_child_weight=None, missing=nan, mon
              multi_strategy=None, n_estimators=None,
              num_parallel_tree=None, random_state=No
XGBClassifier
XGBClassifier(base_score=None, booster=None, callbac
              colsample_bylevel=None, colsample_byno
```

- **Extra Tree Classifier**

It seems that Extra Trees (ET) Classifier is a tree-based ensemble method. In order to prevent overlearning and over-fitting, both ET Classifier and RF randomise subsets of the data and individual classifications. Unlike RF, ET does not bootstrap observations and separates nodes utilising random splits rather than the optimal splits, despite the fact that both methods include the construction of numerous trees and the use of random feature subsets. For beginnings, it constructs multiple DT with the bootstrap parameter set to False, which implies that it does not sample any replacement, and secondly, it distributes these nodes randomly over some subset of the attributes that are set for each node. ET, which are an alternative to bootstrapping, generates randomization by splitting all observations at random. As part of the ET method, several unpruned DTs are generated based on the training dataset. Classifier estimates are created by majority voting, whereas regression predictions are formed by averaging the DT forecasts. Like other ensembles of decision tree methods, such as bootstrapping aggregation (bagging) and RF, it is a collection of trees that have been used to make a decision.

Implimentaion process:

- In this we are performing hyperparameter tuning for an Extra Trees Classifier.
- First, several lists of potential values for different hyperparameters are defined. Certain parameters affect the estimation process. These include the number of trees applied (n_estimators), the maximum number of characteristics to consider (max_features), the tree's depth (max_depth), the smallest amount of samples needed to separate an internal node (min_samples_split), the minimum number of samples needed to be at a leaf node (min_samples_leaf), and the use of samples from bootstrap (bootstrap).
- Next, a dictionary param_grid is created to hold these lists.
- An instance of the ExtraTreesClassifier class is then created and stored in et. This is the base model that will be tuned.
- Then, an instance of the RandomizedSearchCV class is created with et and param_grid as arguments. This will perform a randomized search over the parameter grid to find the best parameters for the model. The argument cv=10 specifies that 10-fold cross-validation should be used during this process.
- The X_tra_in and Y_tra_in training data are then passed to the fit procedure of the RandomizedSearchCV instance. This begins the process of training and tuning the model. The best parameters found during this process will be used in the final model.

```
RandomizedSearchCV
RandomizedSearchCV(cv=10, estimator=ExtraTreesClassifi
                   param_distributions={'bootstrap': [
                                        'max_depth': [


                                        'max_features'
                                        'min_samples_l
                                        'min_samples_s
                                        'n_estimators'
```

- **Decision Tree Classifier**

Not just in machine learning, but also in other domains like statistics and DT, data mining, are used as models for prediction. DT is a non-parametric supervised learning approach. A tree structure, a predictive model, is constructed by inferring rules from the data points consisting of feature vectors. The following formula produces the entropy of a DT: We choose a K-node and assign labels to J-classes. j can take on any value between 1 and J. The mathematical derivation is as follows:

Implimentaion process:

- In this we are performing hyperparameter tuning for a Decision Tree Classifier.
- First, a dictionary param_dist is defined with potential values for four parameters: The parameters min_samples_leaf, max_depth, max_features, and criteria determine the maximum depth of the tree, the number of features to examine, and the minimum number of samples needed at a leaf node, respectively.
- Next, an instance of the DecisionTreeClassifier class is created and stored in tree. This is the base model that will be tuned.
- Then, an instance of the RandomizedSearchCV class is created with tree and param_dist as arguments. This will perform a randomized search over the parameter grid to find the best parameters for the model. The argument cv=5 specifies that 5-fold cross-validation should be used during this process.
- Finally, the fit method is called on the RandomizedSearchCV instance, passing in the training data (X_tra_in, Y_tra_in). This begins the process of training and tuning the model. The best parameters found during this process will be used in the final model.

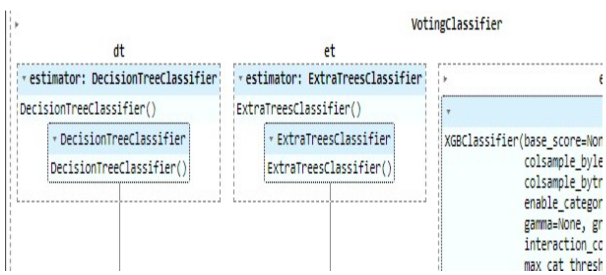
```
RandomizedSearchCV
RandomizedSearchCV(cv=5, estimator=DecisionTreeClassi
                   param_distributions={'criterion': [
                                        'max_depth': [
                                        'max_features'

                                        'min_samples_l

                   estimator: DecisionTreeClassi
```

- **Voting Classifier**

Voting regressor [22],uses the idea of averaging the results of many ML methods to get a single set of predictions. Voting regressors are a type of base regressor approach where many regressors are fit to the whole dataset. A group of similarly performing estimators can benefit from using such a regressor to compensate for their differences in performance. When the predictors can be separated as much as feasible, ensemble approaches perform better. The standard approach to this problem is to train each regressor independently using a separate method. This raises the probability that they will make a wide variety of mistakes, which is good for the ensemble as a whole. Xgboost, ET, and a DT classifier are the three we employ in this model.

Implimentaion process:

- In this we are creating and training an ensemble model that combines three different classifiers: a Decision Tree Classifier (dt), an Extra Trees Classifier (et), and an XGBoost Classifier (xgbc). These classifiers have been previously tuned using RandomizedSearchCV.
- The VotingClassifier class is used to create the ensemble model. The estimators argument requires a sequence of pairs, each of which will identify a classifier by its name and instance. The voting argument specifies how the ensemble model makes predictions. In this case, voting='hard' means that the ensemble model will use majority voting, i.e., it will choose the class that has the most votes from the individual classifiers.
- The fit technique is employed to train the ensemble model on the training data (X_tra_in, Y_tra_in). This will fit each classifier in the ensemble using the best parameters found by RandomizedSearchCV.
- The output of this is an ensemble model that has been trained on the input data and can be employed to make predictions on new data. It is anticipated that the efficiency of the ensemble model will be superior to that of any one classifier included in the ensemble due to the ensemble system's ability to decrease the variance and bias of individual classifiers.



• RESULTS & DISCUSSIONS

The research will be carried out on an HP workstation that has a total of 32 gigabytes of random access memory (RAM), a one terabyte (TB) hard drive, the Windows 10 operating system, a 24 gigabyte (GB) Nvidia graphics processing unit (GPU), as well as an Intel Core i7 central processing unit (CPU). This work implimenton Python programming language to accomplish a variety of activities by using the Jupyter notebook. The recommended work was created using Python 3.7 and the necessary tools matplotlib, pandas, as well as numpy. The Kaggle platform was used to collect the Telco Customer Churn dataset that split into train and test. In the next section, provide the results of proposed voting classifiers with each models and do a comparative analysis of the machine learning system's f1-score, recall, precision, and accuracy. Include a comparison between the proposed model and the base model. Numerous statistical measures were used to evaluate the research's findings, and they will be further discussed in the parts that follow.

- **Dataset Discription**

In this work used Telco Customer Churn that collected from the Kaggle. Each client is represented by a row, and their characteristics are listed in the columns. This data collection contains details about:

- A column labelled "Churn" shows recent customer defections from the last month.
- Each subscriber's set of services, including phone lines, additional internet connections, online backups, device protection, tech support, streaming TV and movies, and more.
- Details about the client account, such as their length of service, contract details, payment options, whether or not they choose for paperless billing, as well as their monthly & yearly costs.
- Customers' basic demographic information, includedage range, gender, & family status.

- **Exploratory Data Analysis**

In the process of analysing data, exploratory data analysis (EDA) is an essential stage that plays a significant part in discovering patterns, correlations, with outliers in the data, testing hypotheses, and confirming hypotheses via the utilisation of summary statistics and visualisations. In order to get insights into the essential properties of the dataset's distinct entities, EDA is a crucial step in doing a different type of analysis on the supplied dataset. It does a great job at depicting the traits and how they relate to one another. Graphs of the correlation matrix, histograms of the input dataset, a pie chart with the SMOTE balanced dataset, and graphs of the taget variables and all services analysis from the input telco dataset are shown in Figures 3–7.

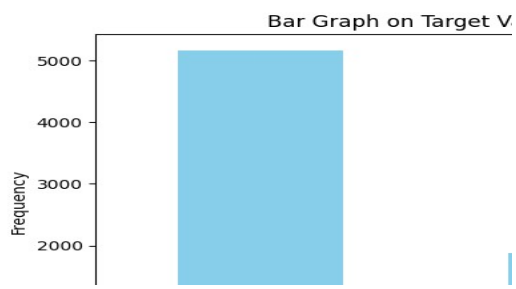


Figure 3: Bar graph of the target variable

The above figure 3 shows the bar graph of the target veriables. In figure x and y- axis shows the categoes with yes and no and their frequency. No class contain 5000 approx. variables and yes class contain onlye 2000 approx. variables.

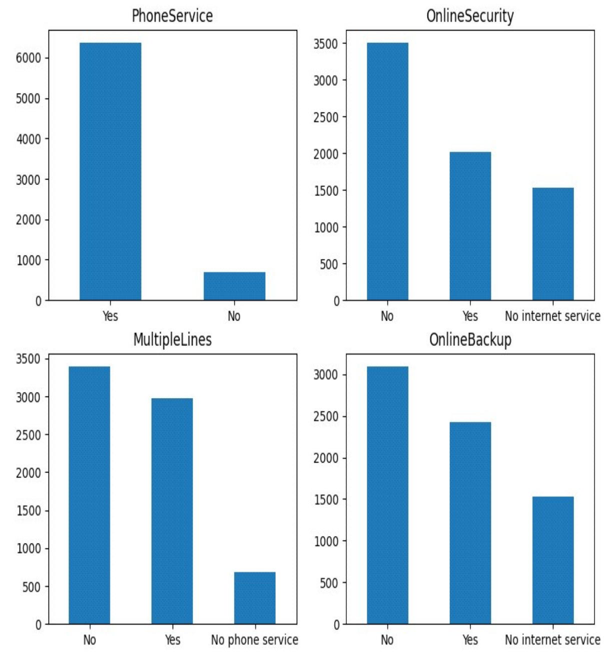


Figure 4: Bar grph of dataset with all the services

Figure 4 depicts a dataset's bar graph containing all the services it offers, including multiple lines, phone service, streaming TV, online security, internet access, tech support, cloud-based backup, device protection, and streaming movies. In figure x-axisshows the three classes of this dataset with No, Yes and No internet services and y- axis shows the class values.

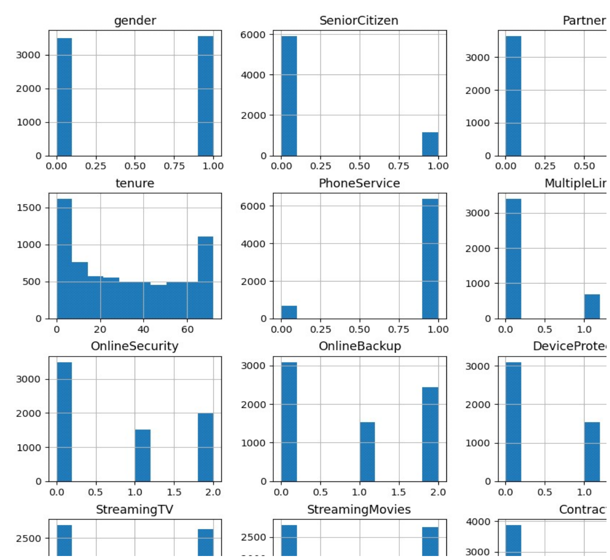


Figure 5: Plotted Histogram of Dataset

The above figure 5 shows the Plotted Histogram of telco dataset. A histogram is a bar chart showing the range of values for a given numerical variable. The first stage in making a histogram is to establish a bin of the ranges, into which the entire range of values may be broken down.

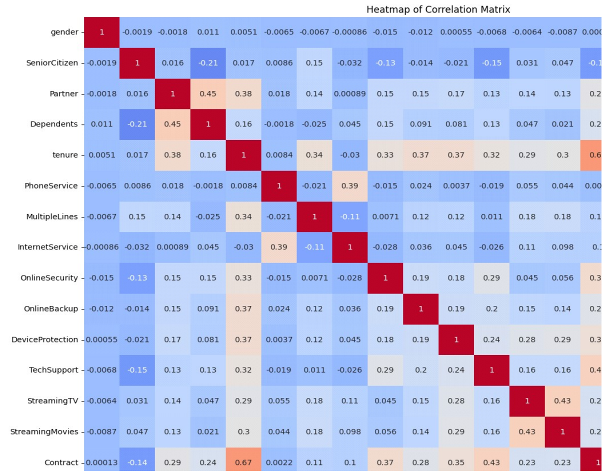


Figure 6: Plotted heatmap correlation matrix of telco dataset

Figure 6 depicts the telecom dataset correlation matrix as a plotted heatmap. The correlation between many variables can be visualised as a color-coded matrix using a correlation heatmap.

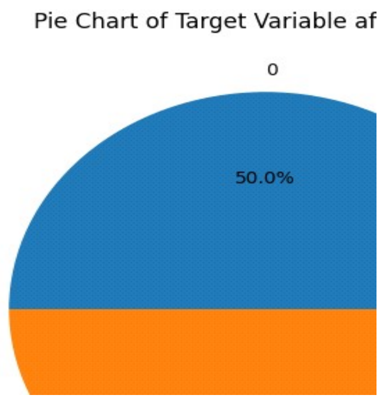


Figure 7: Pie chart of tagetveriable after SMOTE with balanced dataset

The above figure 7 shows the pie chart of tagetveriable after SMOTE with balanced dataset. After used SMOTE data balancing techniques telco dataset quely distributed with 50%-50% ratio.

- Evolution Parameters

It is necessary to assess the effectiveness of ML approaches since they do not always produce the best results when applied to the data that is available. Commonly employed measures of a binary classifier's effectiveness comprise precision, accuracy, recall, and F1-score. A confusion matrix (CM), which may be utilized in the process of determining these parameters, is depicted in Figure 8.

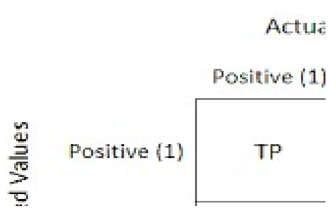


Figure 8: Confusion Matrix for binary classification.

The efficacy of a classifier may be define with the help of a CM a term from ML that takes into account both the actual and anticipated classifications. For each classification, there are four possible outcomes: Negatives (TN), True Positives (TP), or False Positives (FP), True False Negatives (FN)[23].

Accuracy: The accuracy of a classifier is indicative of how well it performs in general. It is a measure of the proportion of data that was correctly labelled.

Precision: Precision is the percentage of positive cases that were accurately anticipated. In our case, the measure reveals the model's accuracy in making predictions about the churning class.

RecallThe ability of the classifier to identify samples with a positive label is measured by a metric called recall. This demonstrates the binary classifier's capability of distinguishing between examples of different classes. This is how we compute recall:

F1-Score: Classifier performance is often measured employing the F-score. The F-score is a statistic that takes into consideration both precision and recall, and it is sometimes called to as the "harmonic mean of these two measures of accuracy." Improved model recall and accuracy correspond to decreasing F-scores.

AUC: The AUC measure must first use a ROC curve to characterise the model's efficacy. For a given set of cutoff values, this curve depicts the relationship between the True Positive Rate and the False Positive Rate. AUC, is the corresponding statistic. Therefore, it offers a comprehensive performance metric across all feasible cutoffs for rankings. This metric may be viewed as the likelihood of a positive occurrence being accurately labelled as such by the model.

- **Experimented Proposed Results**

This section represent the outcome of proposed voting classifiers with extra tree, DT and XGBoost for churn prediction evaluations with Telco Kaggle dataset. To examine the model performance in terms of confusion metrix, classification report and ROC-AUC figures, also table provides the each model performance with respect of accuracy, precision, recall and f1-score.

- **XGBoost results**

The first proposed XGBoost classifier performance shows figure 9 that obtain 85.21% accuracy, recall and f1-score but 85.22% precision, respectively, for the prediction of telco churn prediction.

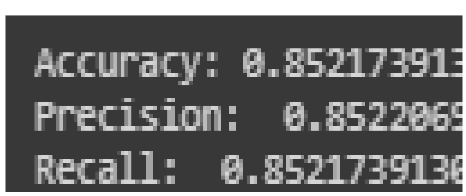


Figure 9: Parameter performance of XGBoost classifier

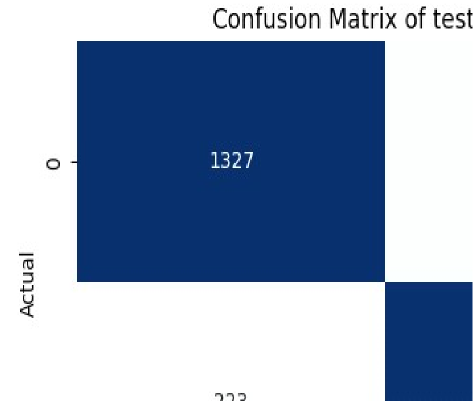
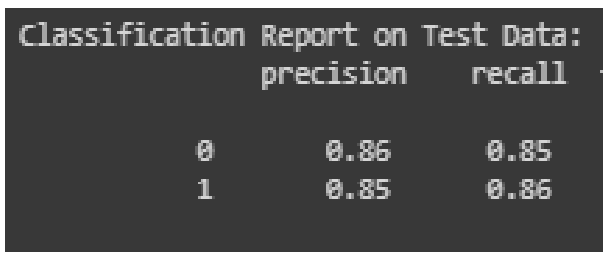


Figure 10: Confusion metrix of XGBoost classifier

From the figure 10 shows the confusion metrix of proposed XGBoost classifier that predicted TP instances of 1327, and TN instances of 1319 while FN instances of 223 and false positive instances 236, respecteevly.



Classification Report on Test Data:

	precision	recall
0	0.86	0.85
1	0.85	0.86

Figure 11: Classification report of XGBoost classifier

In figure 11 shows the classification report of XGBoost classifier that shows the telco dataset performance with both classes 0 and 1 using accuracy, precison, recall, f1-score with support values. XGBoost classifier obtain 86% and 85% parameter performance with suppoer 1563, 1542 and 3105, respectively.

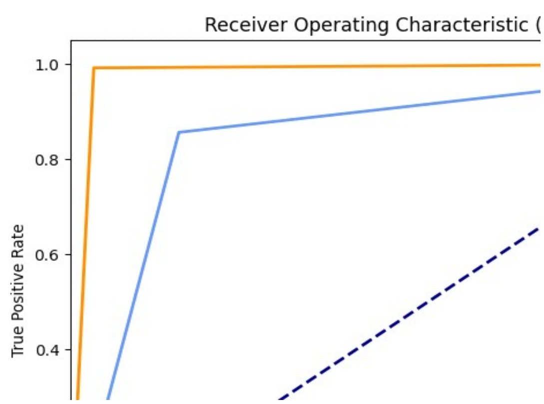


Figure 12: ROC-AUC of XGBoost classifier

Figure 12 depicts the ROC-AUC curve of the suggested XGBoost classifier. The picture depicts the FPR value of the proposed model (x-axis) and the TPR (y-axis). Proposed XGBoost model obtain training ROC-AUC of 98% and testing ROC-AUC of 85%, repectevely.

- **Extra Tree results**

The first proposed Extra Tree classifier performance shows figure 13 that obtain 86.05% accuracy, recall and f1-score but 86.11% precision, respectively, for the prediction of telco churn prediction.

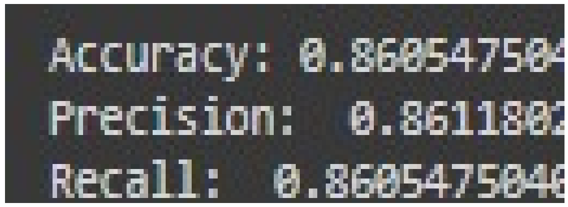


Figure 13: Parameter performance of Extra Tree classifier

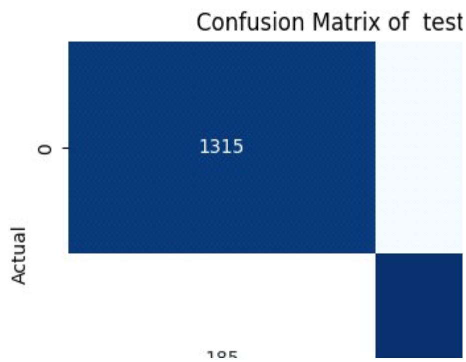
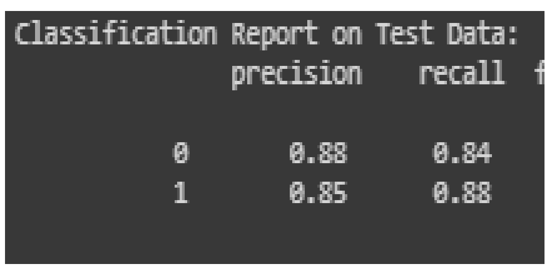


Figure 14: Confusion metrix of Extra Tree classifier

Figure 14 displays the CM for the suggested ET classifier, which correctly identified 1315 positive and 1357 negative examples while making 185 and 248 incorrect predictions, respectively.



	precision	recall
0	0.88	0.84
1	0.85	0.88

Figure 15: Classification report of Extra Tree classifier

In figure 15 shows the classification report of Extra Tree classifier. Extra Tree classifier obtain 84%, 85%, 86% and 88% parameter performance with suppoer 1563, 1542 and 3105, respectively.

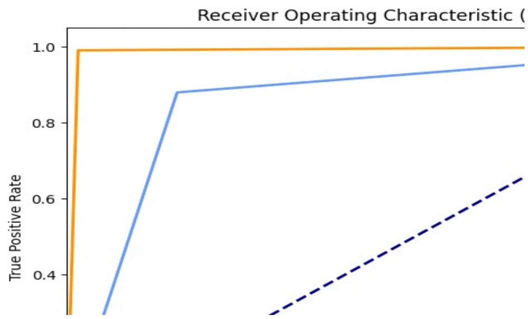
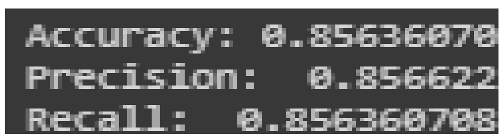


Figure 16: ROC-AUC of Extra Tree classifier

The ROC-AUC curve of proposed ET classifier represent in figure 16. Proposed Extra Tree model obtain training ROC-AUC of 99% and testing ROC-AUC of 86%, repectevely.

- **Voting results**

The first proposed Voting classifier performance shows figure 17 that obtain 85.36% accuracy, recall and f1-score but 85.66% precision, respectively, for the prediction of telco churn prediction.



```
Accuracy: 0.85636070
Precision:  0.856622
Recall:  0.856360708
```

Figure 17: Parameter performance of Voting classifier

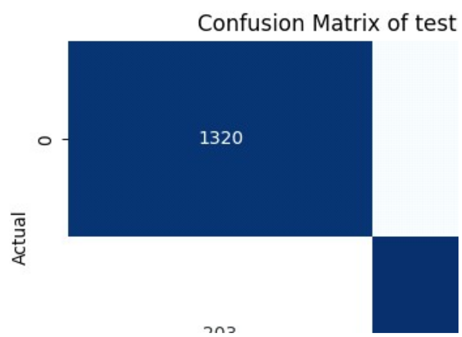
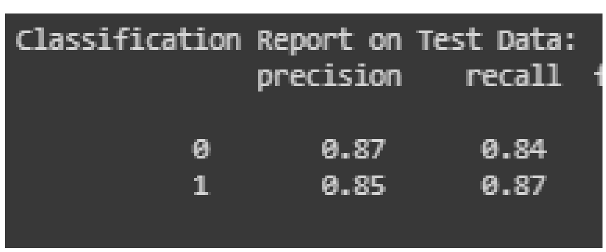


Figure 18: Confusion metrix of Voting classifier

From the figure 18 shows the confusion metrix of proposed Extra Tree classifier that predicted TP instances of 1320, and TN instances of 1339 while FN instances of 203 and false positive instances 243, respecteevly.



```
Classification Report on Test Data:
              precision    recall

           0       0.87      0.84
           1       0.85      0.87
```

Figure 19: Classification report of Voting classifier

In figure 19 represent the classification report of Voting classifier. Voting classifier obtain 84%, 85%, 86% and 87% parameter performance with suppoer 1563, 1542 and 3105, respectively.

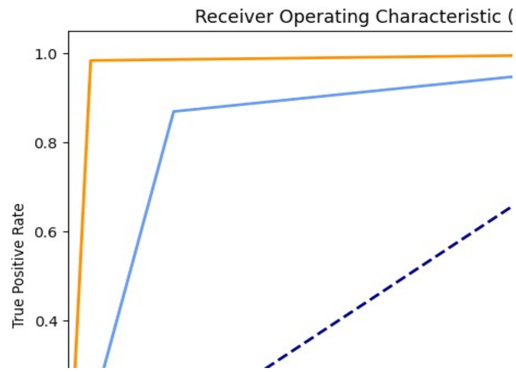


Figure 20: ROC-AUC of Voting classifier

The ROC-AUC curve of proposed Voting classifier shows in figure 20. Proposed Voting model obtain training ROC-AUC of 97% and testing ROC-AUC of 86%, repectevely.

- Comparative Results and Discussion

In this portion provide the comparative analysis of proposed Voting classifier(Extra tree, decision tree and XGboost) and base models (k-Nearest Neighbor (kNN), Linear Regression (LR), stochastic gradient booster (SGB), and Random Forest (RF)) respect of precision, recall, accuracy & f1-score measures. The following table 1 shows the comparison between base and proposed models for CP.

Table I: Machine learning models comparison for churn

Perameters performance	**Proposed Model**	**Base Models**			
	Voting	**KNN**	**LR**	**SGB**	**RF**
Accuracy	85.63	79	80	80	80
Precision	85.66	78	79	79	79
Recall	85.63	79	80	80	80
F1-Score	85.63	78	79	79	79
AUC	86	83	84	83	84

Figure 21: Bar graph of accuracy comparison between base & proposed approach

Figure 22: Bar graph of precision comparison between base & proposed approach

Figure 23: Bar graph of recall comparison between base and proposed techniques

Figure 24: Bar graph of F1-Score comparison between base & proposed techniques

Figure 25: Bar graph of AUC comparison between base & proposed approach

For the prediction of telecom customer churn, the suggested model—which is built on a voting ensemble of extra trees, decision trees, and XGBoost—performs better than the individual base models (KNN, LR, SGB, and RF) across a range of assessment measures. The suggested model's accuracy shows in figure 21, which stands out at 85.63%, surpasses that of all the base models and is much better than that of KNN, the basic model with the greatest accuracy (79%). This shows that the ensemble technique makes better use of the advantages of each component model to provide an overall forecast that is more accurate.

The recommended model frequently performs superior to the traditional approaches in terms of accuracy, recall, and F1-Score. In order to prevent mistakenly labelling devoted clients as churners, precision—a metric that measures the accuracy of positive predictions—is essential in the churn prediction process. The suggested model significantly outperforms the basic models with a precision of 85.66% shows in figure 22, indicating its capacity to provide favourable predictions.

Recall, which measures the model's accuracy in identifying every relevant incident, reaches its maximum for the suggested model at 85.63%, shows in figure 23. This is crucial in relation to telecom CC prediction because it ensures that a sizeable percentage of genuine churn instances will be identified. The suggested model performs better than individual base models, as shown by the F1-Score shows in figure 24, which strikes a balance between accuracy and recall.

Moreover, the suggested model has the largest area under the curve (AUC) shows in figure 25, a comprehensive measure that takes into account the model's capacity to differentiate between classes, at 86%. This suggests a strong discriminating ability, which is essential in practical applications, particularly in situations such as telecom customer churn prediction, where precisely identifying probable churners is critical.

To sum up, the ensemble model that has been suggested, which integrates Extra Trees, Decision Trees, and XGBoost via a voting mechanism, performs better in terms of prediction than the basic models (KNN, LR, SGB, RF). By leveraging the various capabilities of distinct models, the ensemble technique improves accuracy, precision, recall, F1-Score, and AUC. This shows that the suggested model, which provides a more dependable tool for detecting probable churners and enabling focused retention tactics, is a more successful approach to telecom customer churn prediction.

• CONCLUSION AND FUTURE WORK

Approaches for predicting CC depend primarily on AI classification systems. High dimensionality and unbalanced data sets are obstacles for these classification approaches, which in turn hinders reliable CP. Some data mining methods' performance is significantly affected by factors such as the amount of data available for analysis and whether or not the dataset is balanced or imbalanced. It is also common practice for CP models to employ ensemble approaches. For leveraging complex models with limited computing resources, they are precise and user-friendly. Telecom businesses may benefit greatly from this sort of study since it will increase their profits. In this work employed multiple ML approaches for custmor

CP with a telecom dataset. The suggested voting classifier has a higher f1-score and AUC of 86% and 85% in terms of accuracy, precision, recall, and f1-score, etc. All classifiers and other ML models perform better with the suggested approaches than with the basic models.

In the future, this churn avoidance algorithm can be upgraded to deliver more nuanced and accurate suggestions. In the future, researchers may make use of more complicated techniques like DL, RNNs to assess the chance of surviving. These networks assist in finding nonlinear correlations between data variables.Customers may be retained through a variety of techniques if one understands the causes of CC and the attrition/churn rate forecast based on ML and DL techniques. Also, a mobile or desktop application may be built which may additionally simplify and aid organisations, and startups to anticipate CC or churn rate and focus on crucial variables impacting CC.

References

[1] T. J. Gerpott, W. Rams, and A. Schindler, "Customer retention, loyalty, and satisfaction in the German mobile cellular telecommunications market," Telecomm. Policy, 2001, doi: 10.1016/S0308-5961(00)00097-5.

[2] C. P. Wei and I. T. Chiu, "Turning telecommunications call details to churn prediction: A data mining approach," Expert Syst. Appl., 2002, doi: 10.1016/S0957-4174(02)00030-1.

[3] S. A. Qureshi, A. S. Rehman, A. M. Qamar, A. Kamal, and A. Rehman, "Telecommunication subscribers' churn prediction model using machine learning," in 8th International Conference on Digital Information Management, ICDIM 2013, 2013. doi: 10.1109/ICDIM.2013.6693977.

[4] E. Ascarza, R. Iyengar, and M. Schleicher, "The perils of proactive churn prevention using plan recommendations: Evidence from a field experiment," J. Mark. Res., 2016, doi: 10.1509/jmr.13.0483.

[5] S. Kumar and C. D., "A Survey on Customer Churn Prediction using Machine Learning Techniques," Int. J. Comput. Appl., 2016, doi: 10.5120/ijca2016912237.

[6] O. Adwan, H. Faris, K. Jaradat, O. Harfoushi, and N. Ghatasheh, "Predicting customer churn in telecom industry using multilayer preceptron neural networks: Modeling and analysis," Life Sci. J., 2014.

[7] A. K. Ahmad, A. Jafar, and K. Aljoumaa, "Customer churn prediction in telecom using machine learning in big data platform," J. Big Data, 2019, doi: 10.1186/s40537-019-0191-6.

[8] B. Huang, M. T. Kechadi, and B. Buckley, "Customer churn prediction in telecommunications," Expert Syst. Appl., 2012, doi: 10.1016/j.eswa.2011.08.024.

[9] W. Yu, D. N. Jutla, and S. C. Sivakumar, "A churn-strategy alignment model for managers in mobile telecom," in Proceedings of the 3rd Annual Communication Networks and Services Research Conference, 2005. doi: 10.1109/cnsr.2005.5.

[10] V. Umayaparvathi and K. Iyakutti, "A Survey on Customer Churn Prediction in Telecom Industry: Datasets, Methods and Metrics," Int. Res. J. Eng. Technol., 2016.

[11] M. Ali, A. U. Rehman, and S. Hafeez, "Prediction of Churning Behavior of Customers in Telecom Sector Using Supervised Learning Techniques," in Proceedings on 2018

IEEE 3rd International Conference on Computing, Communication and Security, ICCCS 2018, 2018. doi: 10.1109/CCCS.2018.8586836.

[12] 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, 2019, doi: 10.1109/ACCESS.2019.2914999.

[13] A. Vakeel, N. R. Vantari, S. N. Reddy, R. Muthyapu, and A. Chavan, “Machine Learning Models for Predicting and Clustering Customer Churn Based on Boosting Algorithms and Gaussian Mixture Model,” in 2022 International Conference for Advancement in Technology, ICONAT 2022, 2022. doi: 10.1109/ICONAT53423.2022.9725957.

[14] Y. K. Saheed and M. A. Hambali, “Customer Churn Prediction in Telecom Sector with Machine Learning and Information Gain Filter Feature Selection Algorithms,” in 2021 International Conference on Data Analytics for Business and Industry, ICDABI 2021, 2021. doi: 10.1109/ICDABI53623.2021.9655792.

[15] J. Mohajon and A. S. Mohammad Arif, “Churn Prediction With Explainability for the Customers of Telecom Industry,” in 2023 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), 2023, pp. 179–183. doi: 10.1109/ICICT4SD59951.2023.10303395.

[16] M. M. Rahman and D. N. Davis, “Addressing the Class Imbalance Problem in Medical Datasets,” Int. J. Mach. Learn. Comput., 2013, doi: 10.7763/ijmlc.2013.v3.307.

[17] F. Rodríguez-Torres, J. F. Martínez-Trinidad, and J. A. Carrasco-Ochoa, “An Oversampling Method for Class Imbalance Problems on Large Datasets,” Appl. Sci., 2022, doi: 10.3390/app12073424.

[18] N. A. Azhar, M. S. Mohd Pozi, A. M. Din, and A. Jatowt, “An investigation of SMOTE based methods for imbalanced datasets with data complexity analysis,” IEEE Trans. Knowl. Data Eng., 2023, doi: 10.1109/TKDE.2022.3179381.

[19] B. Santoso, H. Wijayanto, K. A. Notodiputro, and B. Sartono, “Synthetic over Sampling Methods for Handling Class Imbalanced Problems : A Review,” in IOP Conference Series: Earth and Environmental Science, 2017. doi: 10.1088/1755-1315/58/1/012031.

[20] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., vol. 13-17-Augu, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.

[21] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” Ann. Stat., 2001, doi: 10.1214/aos/1013203451.

[22] K. An and J. Meng, “Voting-averaged combination method for regressor ensemble,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2010. doi: 10.1007/978-3-642-14922-1_67.

[23] B. Ghaffari and Y. Osman, “Customer churn prediction using machine learning A study in the B2B subscription based service context,” no. June, 2021, [Online]. Available: www.bth.se