National College of Ireland

Optimizing Customer Churn Prediction in Telecom using Machine Learning: A Comparative Study of Sampling Techniques

MSc Research Project
Masters in Data Analytics

Pooja Angale
Student ID: x22239782

School of Computing
National College of Ireland

Supervisor: John Kelly

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Pooja Narendra Angale		
Student ID:	x22239782		
Programme:	Masters in Data Analytics	**Year:**	2023-24
Module:	MSc Research Project		
Supervisor:	John Kelly		
Submission Due Date:	12/08/2024		
Project Title:	Optimizing Customer Churn Prediction in Telecom using Machine Learning: A Comparative Study of Sampling Techniques		
Word Count:	8208 **Page Count:** 19		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Pooja Narendra Angale
Date:	12/08/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Optimizing Customer Churn Prediction in Telecom using Machine Learning: A Comparative Study of Sampling Techniques

Pooja Angale
x22239782

Abstract

Customer churn is one of the most prominent challenges to revenue for any telecommunication company. Precise prediction of customer churn is therefore very critical, but class imbalance in the dataset, with very few customers leaving compared to the large number of those remaining loyal, definitely puts it at risk. Standard machine learning models usually perform poorly on the minority class due to this class imbalance. This paper presents solutions to this problem by examining advanced sampling techniques: MSMOTE, MWMOTE, and IMWMOTE. This is a process where the minority samples will be sorted and borderline and noisy cases are handled with extra care. In this paper, comparative analysis is performed to prove that, against baseline models, these advanced sampling techniques significantly improve performance in churn prediction tasks, offering different advantages conditioned on dataset complexity. Applying these methods will make it easy for a telecommunication company to improve its accuracy in predicting the churning customers for better retention efforts.

1. Introduction

The Oxford dictionary defines “churn rate” as “the number of people who stop using a product and change to another or who leave the company they work for and go to another” (*churn rate noun - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner’s Dictionary at OxfordLearnersDictionaries.com*, no date).
'Churn' is a word derived from change and turn. It means the discontinuation of a contract. There exist three forms of churn:

- active/ deliberate – the customer decides to terminate his contract and switch to another provider.
- rotational/ incidental – the customer resigns under the contract, without the aim of switching to a competitor.
- passive/ non-voluntary: The company itself discontinues the contract (Lazarov and Capota, 2007).

Customer churn is a major issue and an important concern for large organizations and businesses alike. The various industries are making attempts to improve their methods for the prediction of possible customer churn, since it has an immediate effect on revenues. Customer churn prediction is basically a term in machine learning in which multiple models are implemented to predict which customers might discontinue a service or subscription (Lazarov and Capota, 2007). This terminology is particularly important for several industries like telecommunications, IT industry that sells subscription-based products, software as a service (SaaS) organization, and all the businesses where membership plays an important role in their long-term success.

In the real-world churn analysis, there is normally high class imbalance, with the minority class including customers who have churned; meaning their discontinuation of the service; and the majority class composed of loyal customers who continue to use the service. This places a challenge on standard machine learning algorithms, which tend to focus on the majority class because of its larger representation in a dataset. Therefore, it might miss very important patterns related to the minority class. This leads to poor prediction performance for customers who have churned, too. In order to handle this issue and improve the model's predictive power with respect to churning, different sampling techniques have been developed. These techniques either oversample the minority class or undersample the majority class to balance out the dataset, improving the model's focus toward both classes.

Conventional methods to balance an imbalanced dataset are Random Over-Sampling and Random Under-Sampling (Mohammed, Rawashdeh and Abdullah, 2020), where the former is realized through multiple copying of the minority class and the latter by randomly removing instances of the majority class. Both may lead to overfitting and the loss of important information. Another very famous technique is SMOTE, which creates new synthetic samples by mixing existing instances of the minority class. Other variants introduce some randomness, as in Random SMOTE, while others focus on the generation of samples near the decision boundary for better classification, like Borderline SMOTE (Feng, 2022). These methods have been useful, although they continue to show problems dealing with complex imbalances.

While some traditional methods of resampling have been profoundly and thoroughly studied and applied, such as SMOTE (Synthetic Minority Over-sampling Technique) (Rajendran, Devarajan and Elangovan, 2023) less research has been performed on the techniques like MSMOTE, MWMOTE, and even IMWMOTE. These advanced techniques are the potential improvements on basic SMOTE, focusing on the generation of more informative synthetic samples from instances that are hard to classify and, therefore potentially leading to better model performance. Very few studies exist previously that compare the effectiveness of MSMOTE, MWMOTE, and IMWMOTE (Hu *et al.*, 2009)(Barua *et al.*, 2014)(Wang *et al.*, 2024) on the subject matter of customer churn prediction.

In telecommunications, the problem of datasets being highly imbalanced is usually attributed to having more customers who stay compared to those who leave, referred to as churn. Class imbalance makes it hard for predictive models to identify the churned customers effectively since predictive models focus on the majority class, non-churned. Advanced sampling techniques like MSMOTE and its variants such as MWMOTE and IMWMOTE help create synthetic samples of the churned customer classes, hence balancing the dataset. Synthetic samples are generated to rebalance the dataset, allowing the creation of artificial data points for the minority class and increasing the model's capacity in learning and predicting outcomes of underrepresented groups accurately (Rani and Masood, 2023). It improves modeling for accurate churn-prediction capabilities, and hence, companies can take proactive measures towards customer retention.

In this study, techniques will be compared to evaluate which one is more effective in handling class imbalance in churn prediction.

A significant gap and opportunity to research and document comparative performances for these techniques exists. The chosen techniques are explained below:

MSMOTE (Modified Synthetic Minority Oversampling Technique) (Hu et al., 2009):
To overcome the disadvantage of SMOTE, a modified method called MSMOTE was proposed. This algorithm modifies the samples of the minority class into three categories: security samples, border samples, and latent noise samples (Buckland and Gey, 1994) by calculating all the distances of samples. In generating synthetic examples with MSMOTE, different strategies will be used to select near neighbors. This technique is a variant of SMOTE that enhances the original method by better borderline and noisy instances handling. This makes it an up-and-coming candidate for data sets wherein standard SMOTE does not represent the instance of the minority class very well (Hu *et al.*, 2009).

MWMOTE (Majority Weighted Minority Oversampling Technique) (Barua et al., 2014):
Sukarna Barua, Md. Monirul Islam et al., in a study, have proposed a novel synthetic oversampling method that reduce the problems associated with imbalanced learning, generating useful synthetic minority class samples, as the Majority Weighted Minority Oversampling Technique, MWMOTE.
The basic essences behind the proposed method would thus be:

1) selection of the appropriate subset of original minority class samples.
2) assigning weights to the selected samples basing on their importance in the data.
3) using a clustering approach for generation of useful synthetic minority class samples.

This technique identifies instances that are hard to learn in the minority class and gives them a significant weight so that, when generating synthetic samples, it creates more representative examples from the challenging cases, therefore helping in improving a model's ability to generalize such difficult examples.

IMWMOTE (Improved Minority Weighted Minority Oversampling Technique) (Wang et al., 2024):
The traditional sampling algorithms can hardly handle challenging imbalanced classification problems with noise, extreme imbalances, and multi-class situations. In most cases, they also ignore different model interpretability optimization issues. This paper presents a fault diagnosis algorithm based on varied and imbalanced data, especially for high-end equipment like turbine rotors—the Improved Majority Weighted Minority Oversampling Technique (IMWMOTE). To handle complex heterogeneous data, it improves both model interpretability and fault diagnosis performance. This technique enhances MWMOTE by further refining the identification process for critical minority class instances. After that, it generates far more realistic and valuable synthetic samples, significantly improving model performance on the minority class.

The choice for this set of techniques shows a progression from basic oversampling methods to more sophisticated ones that are designed to improve some weaknesses of SMOTE. This study tries to provide insights into their relative efficacy and practical applicability through assessment and comparison of their performance in scenarios of churn prediction.

Research Question:
How well do different advanced sampling methods perform in predicting customer churn in datasets with class imbalance?

2. Literature Review

To set the stage for the exploration of advanced sampling techniques with a rather detailed review of the literature on handling imbalanced datasets, covering methods proposed and tested within a wide range of recent strategies.

Hu et al. paper "MSMOTE: Improving Classification Performance When Training Data is Imbalanced" (Hu *et al.*, 2009) proposes MSMOTE for imbalanced datasets, classifying the minority class samples, based on their distances, into security, border, and latent noise samples. Unlike SMOTE, MSMOTE does not consider the distribution of the minority class samples and latent noise. For each group of samples, it implements different strategies to generate synthetic samples. The combination of this method with AdaBoost in the MSMOTEBoost algorithm significantly improves the classification performance of imbalanced datasets, achieving higher recall, precision, and F-values for the minority class compared to SMOTE and SMOTEBoost. All these enhancements make MSMOTE a robust solution for problems related to imbalanced data classification, showing experimental results on several diverse datasets. Although these developments were necessary, there is still a need for further research on the differential importance of features in the oversampling process, which our research aims to fill.

In a study, the authors Sukarna Barua et al. (Barua *et al.*, 2014) presented MWMOTE sampling technique which focuses on issues related to imbalanced datasets where instances of the minority class are underrepresented. They criticized current oversampling approaches like SMOTE for not actually characterizing the minority class. The authors presented MWMOTE, focusing on the generation of synthetic samples of important examples from the minority class. More weights were put on those instances that are near the majority class examples. It will provide better representation to the minority class in an attempt to improve model performance. The results obtained with MWMOTE are promising, though the challenges still exist, which leaves a space for future research, especially on issues resulting from extremely imbalanced data and overlapping classes.

"IMWMOTE: A Novel Oversampling Technique for Fault Diagnosis in Heterogeneous Imbalanced Data" (Wang *et al.*, 2024) focuses on IMWMOTE, an oversampling technique, works in diagnostics of faults in smart manufacturing systems, where data is usually imbalanced and varied. Although quite effective, these available methods have many limitations with complex and noisy data according to techniques like SUNDO and SMOTEBoost. In this paper, the authors present a new improved oversampling method, IMWMOTE, to improve fault diagnosis accuracy by solving the above issues. They prove that this technique can suit different types of faults in parts such as turbine rotors and bearings. However, it has also been noticed in the paper that many techniques still have difficulty with diverse feature representations and optimization of sampling, hence leaving a gap that is supposed to be filled by IMWMOTE. While IMWMOTE optimizes fault diagnosis with respect to imbalanced and diverse data, it still faces challenges in the accurate characterization of the full spectrum of fault patterns and the optimization of feature representation, especially for highly heterogeneous data.

A study "NI-MWMOTE: An Improving Noise-Immunity Majority Weighted Minority Oversampling Technique for Imbalanced Classification Problems" (Wei *et al.*, 2020) presents NI-MWMOTE sampling technique, which is an improvement on the MWMOTE algorithm. The method handles class imbalance by focusing on the minority classes, which are normally

underrepresented and noisy. NI-MWMOTE discovers adaptive handling of noise through a combination of Euclidean distance and neighbour density measures in identifying and reducing noise, quite useful in making sure that quality synthetic samples can be generated. This will allow for concentrating on critical boundary instances through aggregative hierarchical clustering and significantly increase classification accuracy without much extra, added computational complexity. One of the possible limitations of the NI-MWMOTE method is that it relies on correctly identifying and filtering noise, which might not always be able to handle extremely noisy data sets or complex data distributions.

The paper "A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining" (Wongvorachan, He and Bulut, 2023) compares different techniques to deal with the class imbalance problem in educational datasets: Random Oversampling, Random Undersampling, and a hybrid method combining SMOTE-NC with RUS. They used the data from the 2009 High School Longitudinal Study and proved that Random Oversampling works well for moderately imbalanced data, while hybrid approach works in highly imbalanced situations. These techniques assume actual importance for the accurate prediction of rare, very important events like student dropouts. The study is predominantly a comparison of resampling methods and does not go deep into how they can be used practically in various educational settings or find out their possible impact related to educational policies.

The authors Weijie Yu and Weinan Weng (Yu and Weng, 2022) proposed a research by using an approach for predicting customer churn in telecommunications area by using various machine learning models. They implemented logistic regression, SVM, random forest, AdaBoost, gradient boosting decision trees (GBDT), XGBoost, Light GBM, and CatBoost algorithms to study customer data and predict churn. These models were implemented with proper steps of data preprocessing, cleaning, and transformation, they were also further evaluated with various metrics like accuracy, precision, recall, etc. The results of the study shown that Light GBM and CatBoost outperform other ML models in predicting potential customer churn. As there are high chances of dataset imbalance in churn predicting tasks as the number of potential churn customers is less than the non-churn customers. To deal with this imbalance, the data preprocessing of this study should have involved usage of sampling techniques.

While implementing multiple models to a dataset, the evaluation metrics make it easy to be compared. It can be understood that which model among the others fits the best to that dataset. In a research "Customer Churn Prediction by Classification Models in Machine Learning" (Zhao, Zuo and Xie, 2022) incorporates the random forest and decision tree machine learning models to a dataset. The most effective factor impacting the customer churn was identified to be low-priced count (LC). There is a possibility of more efficient results to be obtained if more factors are considered while the model implementation. Upon evaluation, random forest outperformed the decision tree model, as the predictiveness of customer churn was better with random forest. This study proposes the use of all the sampling techniques with the random forest ML model, this should have better efficiency in predicting the customer churn than the implementation of just the machine learning model without including the sampling in the data preprocessing stage. As this implementation will equalize the major and minor class of the dataset to effectively implement the model.

Many studies focus on predicting the customer churn by one way or other, some of them also focus on the pattern of the churn. The focus should be on the people who are leaving the service, there should be a recognizable pattern to consider. Further, several machine learning models

can be implemented to study this pattern and predict the churn accordingly. In another study, the researchers Peddarapu Rama Krishna et al. (Krishna *et al.*, 2022) followed the same approach and used multiple ML models to predict customer churn based on the pattern of customers who left the service. The random forest outperformed the other models, as it has proven to be best in all the above papers discussed. This is another reason of considering this ML model to be implemented for the chosen research question for this study. The sampling techniques along with random forest are expected to give out better results.

"To Predict Customer Churn By Using Different Algorithms" (Rahman, Alam and Hosen, 2022) discusses that, the feature pattern of the customers who left is important for the efficient churn prediction. The study has implemented and compared ML, deep learning, and impact learning algorithm. This study involves three stages of handling the data which are data collection, handling null values, and data preprocessing. While implementing these, the Impact Learning algorithm has outperformed the Logistic Regression and Artificial Neural Network with more accuracy due to its ability to analyse large amount of data. Also, unlike the previous one focuses on the subscriber data who left the service, the dependent and independent variable of the customer and their values are studied. Depending on this, the present customers are assessed for similar patterns. If any potential churns are detected, strategies are made to keep them interested in the service. Thus, by incorporating such algorithms, it becomes easy for the industries to find potential churn and make innovative strategic plans to maintain the interest of their customers.

A study "Investigating Customer Churn in Banking: A Machine Learning Approach and Visualization App for Data Science and Management" (Singh *et al.*, 2024) suggests that, while working on the financial data like banking to predict the churn, it becomes crucial to have a broad aspect of the dataset with all the features. It is a bank data and has sensitive data, the model accuracy and sensitivity is impacted and generalized. On this financial data, XGBoost and Random Forest ML models has been implemented, sampling has been performed during preprocessing and XGBoost gave the best evaluation metrics. There cannot be a model that can work good with any given dataset, instead depending on the dataset and the problem statement, the approach should be chosen.

The authors of paper "Causal Analysis of Customer Churn Using Deep Learning" (Rudd, Huo and Xu, 2021) suggests that, as an organization, as it is important to predict the customer churn, it is also important to have customer retention strategies. It is always better to invest in customer churn prediction models beforehand to eliminate the costs of customer retention tactics. This research study involves the application of deep feedforward neural networks with sequential pattern mining on the sparse data, casual Bayesian networks are also used to predict the customer churn. In this work, the unbalanced churned and non-churned classed were balanced and then the accuracy was tested. The evaluation metrics of this study on the test data shown better results than any of the previous works. The deep learning models can be effective but complex at the same time, they are called blackbox models as their mechanism is not easy to interpret and understand. Also, the casual analysis can prove to be complex and might need to make some assumptions or considerations.

A study by Chamak Saha, Somak Saha et al. (Saha *et al.*, 2024) encompasses the ChurnNet architecture which comprises of 1D convolution layer with residual block, squeeze and excitation block, and a module to make the performance better. A set of three publicly available datasets are chosen for this study, and the imbalance has been removed during the preprocessing, this elevates the accuracy of the model. The use of deep learning with sampling techniques might give improved results to construct appropriate customer retention strategies.

The proposed study in this paper includes the random forest ML model with the sampling technique, which will be easy to understand and interpret unlike deep learning models.

The churn prediction models are used in a specific period, for instance the monthly customer behavioural data is supplied to the model once and the potential churn is achieved. The paper "Dynamic Behavior-Based Churn Prediction in Mobile Telecom" (Alboukaey, Joukhadar and Ghneim, 2020) discusses the implementation of deep learning models for daily churn prediction like LSTM and CNN, which has proved to be efficient due to continuous supply of data. The models were then evaluated based on daily and monthly churn prediction; the results indicated the daily churn prediction is more accurate. The drawback of this study is that it is extremely time-consuming to train the model as huge data is being supplied to it. The use of deep learning models along with daily prediction pattern makes it more rigorous and lingering. The organizations will have to spend more money for complex systems like these, which is not something that every business can afford. The churn prediction models are frequently used by the professionals so it will be better if its cost and time effective.

“Applying Bayesian Belief Network Approach to Customer Churn Analysis: A Case Study on the Telecom Industry of Turkey” (Kisioglu and Topcu, 2011) highlights the use of Bayesian Belief Network for the prediction of customers who can discontinue the service with the Turkish telecom service provider. While constructing the model, several features of the dataset were taken under consideration like the call frequency, call period, bill amount, calls to different service provider, etc. After finding out these factors, it resulted in three scenarios of promotions to avoid the churn. This study highlights the capability of BBN to understand casual relationships and gives insights for future studies. The study comes with certain limitations as it can work well with small datasets only, and some more variables can be included to improve the prediction.

Research Gap:
While a lot of work (Azhar *et al.*, 2023) has been done with imbalanced datasets and many newer oversampling techniques, most research has focused on the conventional methods, such as SMOTE, Borderline SMOTE, or Random SMOTE. Advanced techniques of sampling like MSMOTE, MWMOTE, and IMWMOTE are not compared between them, despite offering new ways of generating synthetic samples. While such approaches are perfectly designed to handle problems such as noise and the distribution of minority-class samples, their implementation and comparative efficiencies have been less explored in the existing literature.

In this research, the focus will be on the advanced sampling methods, to evaluate how they would be able to balance the imbalanced datasets and then boost up the prediction accuracy. In the same vein, this work is set, critically bridging a gap by providing exhaustive analysis of such advanced sampling techniques and setting it as an output to provide clarity on possible benefits and best usage on real-life problems.

3. Methodology

In the implementation of sampling techniques and analysis of Customer Churn, there is a well-structured methodology based on the Knowledge Discovery in Databases (KDD) framework (Mittal, no date). This eased the process by systematically preparing the data, applying the sampling techniques, building a predictive model, and finally evaluating the results. The steps are explained in detail as below:

3.1 Data Collection

The dataset was obtained from Kaggle (*Telecom Churn Prediction*, no date), which encompasses the data of a telecommunications company, with a wide array of independent variables for studying customer churn. It has columns like below:

Table 1. Dataset Features

Column Name	Data Type	Description	Size
CustomerID	object	Unique identifier for each customer.	10 characters
Gender	object	Gender of the customer.	6 characters (Male/Female)
SeniorCitizen	int64	Indicates if the customer is a senior citizen (1) or not (0).	1 digit (0 or 1)
Partner	object	Indicates if the customer has a partner (Yes/No).	3 characters (Yes/No)
Dependents	object	Indicates if the customer has dependents (Yes/No).	3 characters (Yes/No)
Tenure	int64	Number of months the customer has stayed with the company.	Up to 2 digits
PhoneService	object	Indicates if the customer has phone service.	3 characters (Yes/No)
MultipleLines	object	Indicates if the customer has multiple lines.	3 characters (Yes/No)
InternetService	object	Type of internet service.	Up to 8 characters (DSL/Fiber optic/No)
OnlineSecurity	object	Indicates if the customer has online security.	3 characters (Yes/No)
OnlineBackup	object	Indicates if the customer has online backup.	3 characters (Yes/No)
DeviceProtection	object	Indicates if the customer has device protection.	3 characters (Yes/No)
TechSupport	object	Indicates if the customer has tech support.	3 characters (Yes/No)
StreamingTV	object	Indicates if the customer has streaming TV.	3 characters (Yes/No)
StreamingMovies	object	Indicates if the customer has streaming movies.	3 characters (Yes/No)
Contract	object	Type of contract.	Up to 12 characters (Month-to-month/One year/Two year)
PaperlessBilling	object	Indicates if the customer has paperless billing (Yes/No).	3 characters (Yes/No)
PaymentMethod	object	Payment method.	Up to 20 characters
MonthlyCharges	float64	The amount charged to the customer monthly.	Up to 5 digits (including decimals)
TotalCharges	float64	The total amount charged to the customer.	Up to 7 digits (including decimals)

Churn	object	Indicates if the customer churned (Yes/No).	3 characters (Yes/No)

The rich variety of features within this dataset makes for an extremely detailed view of the relationship of customers with the company, hence leading to the determination of those patterns and factors that may result in customer loss.

3.2 Data Preprocessing

Initial Cleaning:
The preprocessing is initialized by loading the dataset and its detailed description, checking the basic data integrity. This dataset has customer data for a telecommunications company, and the features contain all the knowledge about demographics, services, and account details. Further, the check was performed through ‘df.info()’, ‘df.describe()’, ‘df.head()’ to get insights about data types, summary statistics, and even a peek into the first few rows of the dataset.

Handling Missing and Duplicate Values:
Initially, the quality of data was checked for any missing values and duplicates using the functions ‘isnull()’ and ‘sum()’ across the dataset. In this process, it was found that some records in column 'TotalCharges' were non-numeric as mentioned in Table 1. This could be an entry error or some formatting mistakes. The function ‘to_numeric()’ was applied, which converted this column into numeric data type and marked the non-numeric values as NaN (Not a Number). This has been done to make the dataset numerically consistent for it to be subjected to meaningful analysis and modeling.

After, the rows with missing values were removed using ‘dropna()’ function to avoid possible biases or inaccuracies in the analysis since missing data may cause ambiguous conclusions training the model. The reason behind this was the assumption that the number of missing values were not big enough to impact the representativeness of the dataset.

The duplicate rows were also identified and removed using ‘drop_duplicates()’ function. Duplicates may occur because of errors in data collection or its processing. If not handled, they will inflate the importance of an observation since their occurrence becomes artificially large, leading to biased outputs from the model. Eliminating these redundant entries maintained the integrity and uniqueness of each record in the dataset. This makes sure that the implemented machine learning model would not be skewed because of repeated data points.

Categorical Variable Encoding and Distribution Analysis:
As mentioned in Table 1, the dataset has many categorical features like 'Gender', 'SeniorCitizen', 'Partner', 'Dependents', 'PhoneService', 'MultipleLines', and others like 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling', and 'PaymentMethod'. The target variable was 'Churn’. Distribution counting plots were created to understand the distribution of these categorical variables, showing the frequency of each category within these features. This visualization was a critical factor in understanding class balances, locating probable biases within the dataset.

Since these variables were categorical in nature and they need to be in numeric format to feed into any machine learning algorithm, it was important to encode these categorical variables. This could be done using techniques such as one-hot encoding, where each value of a category

is transformed into its own binary column, making sure that the dataset is optimally prepared to be run by the model of choice. One of the important steps is encoding categorical variables, as this step ensured that the model was able to interpret and make use of information in those features, hence improving on model accuracy and power of prediction.

Cleaning and Analysis of Numerical Features:
Histograms and Density plots for variables such as 'Tenure', 'MonthlyCharges', and 'TotalCharges' were created to understand the distribution of numerical features. These plots provided insights into the dispersion, central tendency, and skewness of the attributes. The visual analysis supported general knowledge of the distribution patterns within numerical data and was critical making informed decisions at later stages of data preparation and modeling.

Bivariate Analysis: Churn vs. Other Features:
The exploration of how the target variable, Churn, is related to other features in the dataset was performed in the bivariate analysis. For categorical features, count plots were used with different colors representing Churn for easy view of how often each category has resulted in churning or not. For numerical features such as Tenure, MonthlyCharges, and TotalCharges, box plots were used to visualize how these values vary for those customers who left versus those that remained. This helped in understanding the differences in customer behaviors that may relate to the churning of customers.

After data preprocessing and cleaning, the resulting dataset was consistently used for all and below performed experiments. Then, one can simply use the same prepared dataset in different modeling techniques to have reliability and comparability of results. By this, one does not have to redo or recreate the dataset for each different predictive modeling technique, thus easing the analysis process and saving time as well as efforts.

3.3 Experimentation with Advanced Sampling Techniques

This dataset maintained a very high level of imbalance, where the number of non-churned customers was higher (majority class) than the number of the churned customers (minority class). Possible outcomes from this imbalance are poor model performance and bias being directed toward the prediction of the majority class. To escape from such potential problems, we used some advanced sampling techniques just to be able to balance the dataset and therefore enhance the model in accurate prediction of customer churn.

Experiment 1: Application of SMOTE
Step 1: In this dataset, the minority class is a churned customer, and the majority class is a non-churned customer. This distinction is critical because SMOTE should balance this data set by creating only new synthetic samples of the minority class. It will also make use of facilities such as 'Counter' to calculate the counts for each class within the target variable 'y', which holds labels for Churn (Yes) and Non-Churn (No). The minority class is the class that has fewer occurrences. This can be seen by comparing the count of each class. The one with a small count will be the minority class.
Step 2: An instance of the SMOTE class was used, oversampling with parameters guiding this process. The 'sampling_strategy' was set at 0.5; the percentage of the minority class was expected to be 50% of the total dataset after resampling. 'k_neighbors' was set at 5, specifying that 5 nearest neighbors should be considered while generating synthetic samples. It does this great balancing act of generating quite diverse samples while keeping them sort of realistic.

Step 3: The SMOTE algorithm uses a k-nearest neighbors approach with ‘k_neighbors = 5’ in this case to find relationships between minority class examples and their nearest neighbors. This was chosen because, when applied as stated here, it provides a balance between underfitting and overfitting to have the synthetic samples diverse yet realistic.

Step 4: In this step, synthetic samples are generated by first determining the ‘k-nearest’ neighbors for each instance in the minority class. After getting the nearest neighbors, a random neighbor is selected and then a synthetic sample from the line segment between the original instance and the chosen neighbor is given. This method ensures that the synthetic sample created would not be only exact duplicates but new, credible instances reflecting natural variability within the minority class itself.

Step 5: Verify the new distribution of the target variable, ‘y’, now that SMOTE has been applied. This is a check to ensure if the class rebalancing due to resampling was successful. The class will verify this distribution using the ‘Counter’ class, which will count the occurrences of each class in ‘y’ post resampling, thus proving that the minority class is now adequately represented.

Experiment 2: Application of MSMOTE

Step 1: Identify the minority samples that are hard to classify using MSMOTE; contrary to standard SMOTE, MSMOTE is targeting instances most challenging for a model to learn. It computes the distance of every minority sample to its nearest majority class neighbors using class ‘NearestNeighbors’. The hardest-to-learn minority samples are defined as those samples of the minority class that are closest to the majority class and are resampled more frequently.

Step 2: It generates these synthetic samples through introducing, where how many neighbors are considered in helping to determine how far spread out the generated samples are set by 'k_neighbors=5'. The reason for this value is to have some reasonable balance in generating a wide variety of synthetic samples since MSMOTE has a focus on hard-to-learn instances.

Step 3: Check the target variable, ‘y’, using MSMOTE. This step would provide some assurance that resampling has been effective and the class with a small population is much better represented in the dataset now. The number of classes is counted in y using the ‘Counter’ class to verify that the dataset is balanced now with special focus on hard cases for learning.

Experiment 3: Initial Implementation of MWMOTE

Step 1: Identify hard-to-classify minority samples. This is done with the ‘NearestNeighbors’ algorithm combined to calculate each minority class instance's distances concerning its closest majority class neighbors and picking those samples that have the maximum distance between them as the most difficult to classify.

Step 2: The weights for these minority examples were therefore set at the inverse of the distance from their nearest neighbors, which was found by setting the ‘k_neighbors=5’ parameter. The reason for 5 neighbors is that the proportion of surrounding points enables a balance in the effect, leading to synthetic samples that are realistic in pointing out variations within the minority class.

Step 3: Generate artificial samples based on the computed weights. Add new samples between minority samples and their nearest neighbors within the minority class, with all those synthetically created being according to their weights; with more synthetic samples generated for those that are harder to classify.

Step 4: Use ‘Counter’ class for the new distribution of ‘y’ target variable after resampling to check if MWMOTE balanced the dataset by over-representing the minority class.

Experiment 4: MWMOTE with Hyperparameter Tuning

Step 1: ‘GridSearchCV’ was used to tune several parameters like ‘k_neighbors’, ‘distance_metric’, and ‘threshold’. On the other hand, values tested for ‘k_neighbors’ in the grid search included those from 3 to 7, while for the ‘distance_metric’, options tried are 'Euclidean' and 'Manhattan'. Different settings were tried on the model in finding the best that would ensure meaningful synthetic samples generated by this model.

Step 2: Using the best parameters identified as ‘k_neighbors=5’ and distance_metric='Euclidean', the MWMOTE algorithm was run, generating synthetic examples based on these optimized settings. The choice of these parameters was validated by their ability to produce a balanced dataset with better model performance.

Step 3: Use the ‘Counter’ class again to check on the new distribution of the target variable, ‘y’. This last check confirms that the resampled dataset, using the adjusted MWMOTE algorithm, now has a balanced data set appropriate for further modeling.

Experiment 5: Initial Implementation of IMWMOTE

Step 1: Run ‘NearestNeighbors’ on difficult-to-learn minority samples. This will return distances between the instances of the minority class; in this case, churning customers, and their nearest neighbors. The farthest samples from their neighbors will be tagged as most challenging to classify and hence will be prioritized by the synthetic sample generation process.

Step 2: Weights were set based on the class difficulty calculated using ‘k-neighbors=5’. This value has been chosen so that the hardest minority samples have enough influence in the generation of synthetic samples to obtain a more balanced dataset.

Step 3: Generate synthetic samples according to the weights calculated by adding between minority samples and their nearest neighbors. The number of synthetic samples generated is proportional to the weight assigned to each minor sample, so that those instances which are most difficult to learn are better represented in the dataset.

Step 4: Recheck the distribution of the target variable ‘y’ by using the ‘Counter’ class after resampling to ensure that this step in the IMWMOTE algorithm overrepresents the proportion of the minority class in the dataset.

Experiment 6: Improved Version of IMWMOTE Algorithm

Step 1: Identification of the hard-to-classify minority samples.
Identify the minority class, which is the class variable with the fewest number of instances in the target variable ‘y’.
Use the ‘NearestNeighbors’ algorithm to find the k-nearest neighbors for each minority sample. This becomes critical in calculating the distances of each of the minority samples from their neighbors, applicable in the following step.

Step 2: Add weights by difficulty.
Weights were computed using the inverse distances to ‘k_neighbors=5’, so the samples far from their nearest neighbors are harder to classify and obtain a higher weight. Also, ‘k’ was put equal to ‘5’ to conserve the balance of influences from surrounding samples while avoiding overfitting in classification.

Step 3: Calculations of synthetic samples.
Include between each minority sample to find its closest neighbors and generate synthetic samples. The number of synthetic samples generated for each of these examples depends on the weight allocated to it. This process inverts to a random selection of a point on the line segment focused on a minority instance and one of its neighbors.

Step 4: Distribution of the target variable 'y' verification.
The next step will be to again check the distribution of the target variable, 'y', using 'Counter' class. This will ensure that by resampling, we are making the dataset appropriately balanced with representations that are much fairer on the minority class.

3.4 Model Implementation

After the dataset was balanced, random forest ML model implementation was done for predicting customer churn. The Random Forest algorithm was used because of its robustness and the fact that it handles complex datasets pretty well. It works by being an ensemble method that constructs multiple decision trees during training and then outputs the class that is the mode of the classes predicted by individual trees (Schonlau and Zou, 2020). This method reduces overfitting, which lets the model generalize better on unseen data.

In this case, the instantiated 'RandomForestClassifier' used key parameters such as 'n_estimators=200', which defines the number of trees in the forest, and 'max_depth=20', which restricts maximum depth of each tree. These were chosen by initial experimentation where it was found that 'n_estimators=200' returned a good balance of computational efficiency against model accuracy and 'max_depth=20' prevented overfitting but with the model able to learn complex patterns.

This was then followed by performing hyperparameter tuning using 'GridSearchCV'. The combinations of the following parameters as 'n_estimators' (100, 200, 300), 'max_depth' (10, 20, 30, None), 'min_samples_split' (2, 5, 10), and 'min_samples_leaf' (1, 2, 4) were tried, so the best combination turned out to be when 'n_estimators=200', 'max_depth=20', 'min_samples_split'=5, and 'min_samples_leaf'=2' since this gave the highest cross-validation score, hence making such a setup generalize to unseen data well.

Finally, a random forest model was trained again under these hyperparameters over the training split in the resampled dataset. The model would train to learn the complex patterns underlying the features differentiating customers who churned from those who did not against the target labels of being churned or not.

The model's performance was then tested after training on another test set, aiming to be able to forecast customer churn accurately. Key metrics used to know how well the model was able to perform are precision, recall, F1-score, and ROC-AUC. Another computed measure is a confusion matrix, giving a view of the classification results of this model: true positives, true negatives, false positives, and false negatives.

3.5 Model Evaluation

Initially, SMOTE was used to achieve an overall accuracy of 86% in the model. From the classification report, there is a good balance of precision and recall for each of the class, churned and the non-churned. The F1 scores are closest, 0.86 for the churned class and 0.85

for the non-churned class, thus, indicating that SMOTE did well with class balancing, although some misclassifications occurred at their respective costs reflected in the confusion matrix. The model correctly classified 1365 non-churned and 1288 churned customers, while misclassifying 198 non-churned and 247 churned customers.

Of all the techniques, MSMOTE performed with an accuracy of 91%. This method was much better at classifying "Yes" for the churned customers, with a precision of 0.95 and a recall of 0.92 for the "Yes" class, thus giving an F1-score of 0.93. This is supported by the confusion matrix, where it is shown that MSMOTE had correctly classified many more of the churned customers compared to SMOTE with less misclassification. This model correctly classified 1377 non-churned and 1673 churned customers, which establishes that MSMOTE works well with an imbalanced dataset.

For the MWMOTE algorithm, the model's performance was slightly below MSMOTE's, still quite robust with accuracy of 87%. Both F1-scores were relatively high, along with a good balance between precision and recall for both classes. This proves that MWMOTE notably improves the performance of the model in differentiating between customers who have churned and those who have not. From the confusion matrix, MWMOTE calculated 1377 non-churned and 1673 churned customers, hence actually classifying them correctly.

While it was designed to further refine MWMOTE, the last one, IMWMOTE (Improved MWMOTE), had slightly lower performance metrics, with an overall accuracy of 83%. The precision for the 'No' class was high, while that for the 'Yes' class was only about 0.73; hence, the F1-score for the "Yes" class turned a bit low at 0.78. It turned out in the confusion matrix that IMWMOTE has classified the number of non-churned customers accurately to be 1399 and that of the churned as 783, but at the same time, it misclassified a higher number of instances in comparison with other techniques, hence requiring more tuning for better performance.

Table 2: Performance Metrics of Random Forest Models Across Different Sampling Techniques

Sampling Technique	Accuracy	Precision (No)	Precision (Yes)	Recall (No)	Recall (Yes)	F1-Score (No)	F1-Score (Yes)
SMOTE	86%	0.85	0.87	0.87	0.84	0.86	0.85
MSMOTE	91%	0.83	0.95	0.9	0.92	0.86	0.93
MWMOTE	87%	0.84	0.9	0.88	0.86	0.86	0.88
IMWMOTE	83%	0.83	0.84	0.9	0.73	0.86	0.78

4. Results

After applying these techniques of sampling on the imbalanced dataset, it changed into one that had a fair share of instances between both the classes, churned and non-churned. The following visualizations depict class balancing for each of the resampling methods applied to the original dataset, hence proving their efficiency in rectifying the imbalance. These resampled balanced datasets make up the base for an improved model performance in the ensuing evaluations.

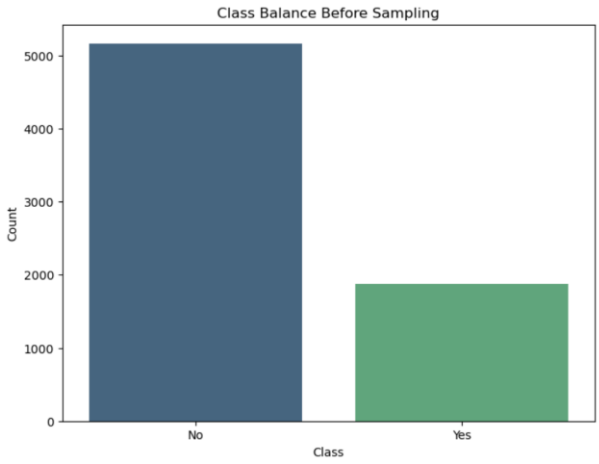

Figure 1. Class balance before Sampling

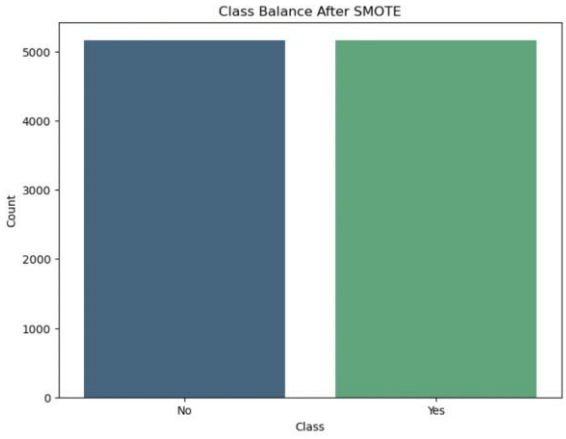

Figure 2. Class balance after SMOTE

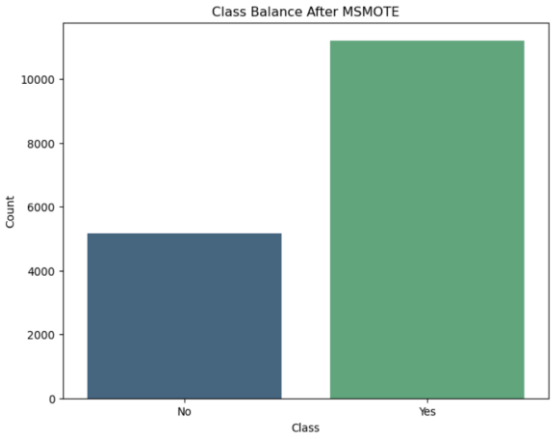

Figure 3. Class balance after MSMOTE

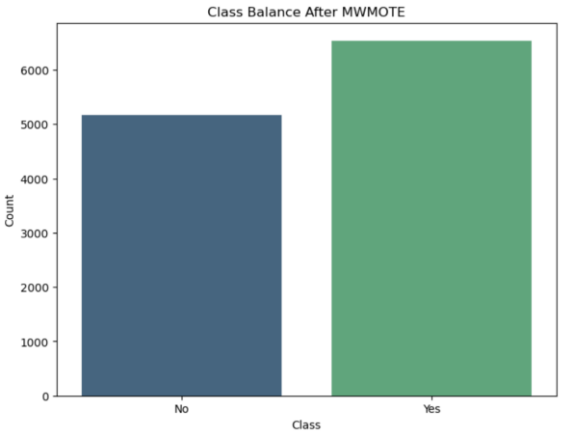

Figure 4. Class balance after MWMOTE

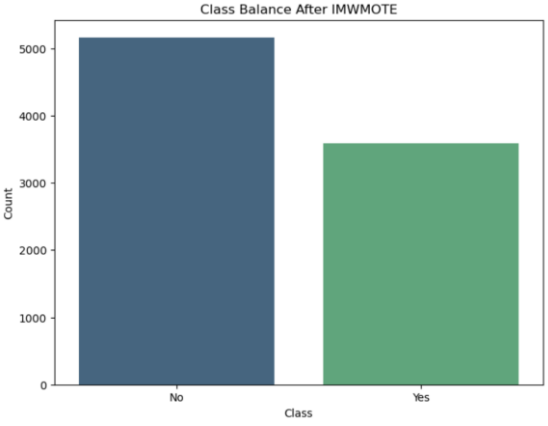

Figure 5. Class balance after IMWMOTE

The below resulting ROC curves plot the true positive rate against the false positive rate for each Random Forest model implemented, so the plots show how each method affects this trade-off. The Area Under Curve (AUC) acts like a summary measure such that the higher the AUC, the better the overall performance is.

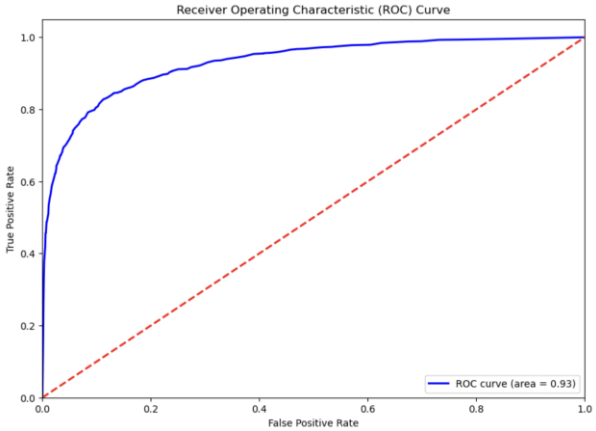

Figure 6. ROC curve for SMOTE

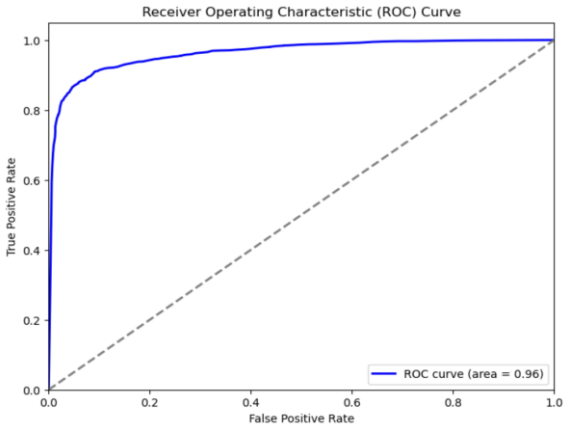

Figure 7. ROC curve for MSMOTE

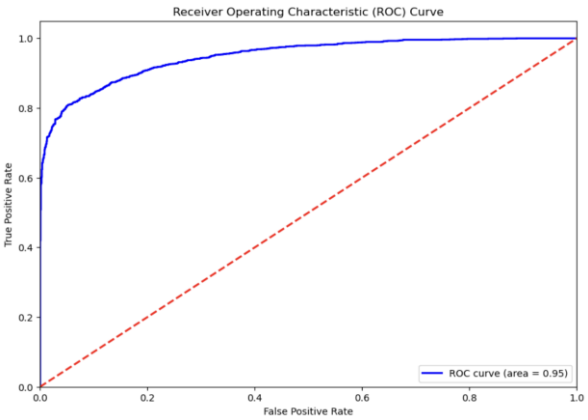

Figure 8. ROC curve for MWMOTE

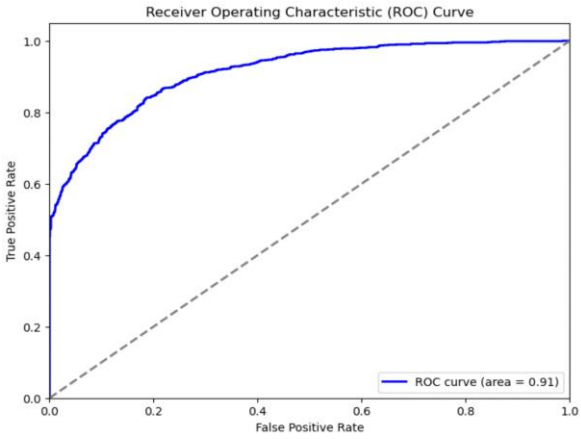

Figure 9. ROC curve for IMWMOTE

5. Future Scope

While some techniques like SMOTE, MSMOTE, MWMOTE, and IMWMOTE have been shown to handle class imbalance effectively for the prediction of customer churn, quite good avenues remain to be further fine-tuned. A promising direction is to develop sampling techniques more sophisticated in capturing the distribution of minority classes without introducing any noise or synthetic data points that might not generalize well. Another approach that could be investigated to enhance further the stability and performance of models is hybrid sampling, combining the benefits of oversampling and undersampling.

In particular, there is a good scope for the improvement in the IMWMOTE algorithm. Future research might focus on optimizing its weighting mechanism in order to rightly identify the most challenging instances of a minority class. Besides, fine-tuning the parameters and finding some adaptive strategies may result in more efficient and dynamic versions of resampling methods. This can further be achieved by either, on one hand,

improving the efficiency and interpretability of IMWMOTE by reducing the computational overhead or by integration with other data augmentation techniques. These improvements will bring about more robust predictive models related to customer churn analysis and, generally, issues involving class imbalance.

6. Conclusion

This research examined various sampling techniques like SMOTE, MSMOTE, MWMOTE, and IMWMOTE in class imbalance handling for customer churn prediction. Systematic experimentation showed that each of these techniques had various benefits to improve the predictive accuracy of the random forest model. SMOTE and MSMOTE established a reasonable baseline by creating synthetic samples of the minority class to balance the dataset for improved model performance.

MWMOTE indispensably improved this method to focus on instances that were relatively harder to classify, thus achieving improved performance in capturing the nuances in churn behavior. The domains of IMWMOTE algorithms are fairly promising; they indicate areas that require further enhancement, particularly in the weighting mechanism and parameter tuning to better represent the most challenging minority samples.

Overall results indicated that the choice of sampling technique significantly impacts how well the churn prediction model works. It also showed that class imbalance handling was one of the important factors, but there is a need for further improvement in methods of sampling for optimal predictive performance. Further research in this regard will be important to remain at the frontier of what can be achieved with machine learning models applied to imbalanced datasets.

BIBLIOGRAPHY

Alboukaey, N., Joukhadar, A. and Ghneim, N. (2020) 'Dynamic behavior based churn prediction in mobile telecom', *Expert Systems with Applications*, 162, p. 113779. Available at: https://doi.org/10.1016/j.eswa.2020.113779.

Azhar, N.A. *et al.* (2023) 'An Investigation of SMOTE Based Methods for Imbalanced Datasets With Data Complexity Analysis', *IEEE Transactions on Knowledge and Data Engineering*, 35(7), pp. 6651–6672. Available at: https://doi.org/10.1109/TKDE.2022.3179381.

Barua, S. *et al.* (2014) 'MWMOTE--Majority Weighted Minority Oversampling Technique for Imbalanced Data Set Learning', *Knowledge and Data Engineering, IEEE Transactions on*, 26, pp. 405–425. Available at: https://doi.org/10.1109/TKDE.2012.232.

Buckland, M. and Gey, F. (1994) 'The relationship between Recall and Precision', *Journal of the American Society for Information Science*, 45(1), pp. 12–19. Available at: https://doi.org/10.1002/(SICI)1097-4571(199401)45:1<12::AID-ASI2>3.0.CO;2-L.

churn rate noun - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner's Dictionary at OxfordLearnersDictionaries.com (no date). Available at: https://www.oxfordlearnersdictionaries.com/definition/english/churn-rate (Accessed: 9 August 2024).

Feng, L. (2022) 'Research on Customer Churn Intelligent Prediction Model based on Borderline-SMOTE and Random Forest', *2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pp. 803–807. Available at: https://doi.org/10.1109/ICPICS55264.2022.9873702.

Hu, S. *et al.* (2009) 'MSMOTE: Improving Classification Performance When Training Data is Imbalanced', in *2009 Second International Workshop on Computer Science and Engineering*. *2009 Second International Workshop on Computer Science and Engineering*, Qingdao, China: IEEE, pp. 13–17. Available at: https://doi.org/10.1109/WCSE.2009.756.

Kisioglu, P. and Topcu, Y.I. (2011) 'Applying Bayesian Belief Network approach to customer churn analysis: A case study on the telecom industry of Turkey', *Expert Systems with Applications*, 38(6), pp. 7151–7157. Available at: https://doi.org/10.1016/j.eswa.2010.12.045.

Krishna, P. *et al.* (2022) *Customer Churn Prediction using Machine Learning*, p. 1040. Available at: https://doi.org/10.1109/ICECA55336.2022.10009093.

Lazarov, V. and Capota, M. (2007) 'Churn Prediction', in. Available at: https://www.semanticscholar.org/paper/Churn-Prediction-Lazarov-Capota/dbf15b7c5f766ef9f84ba83127c626d79b2087b2 (Accessed: 9 August 2024).

Mittal, M.K. (no date) 'Customer Churn Analysis in Telecom Using Machine Learning Techniques'.

Mohammed, R., Rawashdeh, J. and Abdullah, M. (2020) *Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results*, p. 248. Available at: https://doi.org/10.1109/ICICS49469.2020.239556.

Rahman, M., Alam, M. and Hosen, M. (2022) *To Predict Customer Churn By Using Different Algorithms*. Available at: https://doi.org/10.1109/DASA54658.2022.9765155.

Rajendran, S., Devarajan, R. and Elangovan, G. (2023) *Customer Churn Prediction Using Machine Learning Approaches*, p. 6. Available at: https://doi.org/10.1109/ICECONF57129.2023.10083813.

Rani, S. and Masood, S. (2023) *Handling Class Imbalance Problem using Oversampling Techniques for Breast Cancer Prediction*, p. 698. Available at: https://doi.org/10.1109/REEDCON57544.2023.10150702.

Rudd, D.H., Huo, H. and Xu, G. (2021) 'Causal Analysis of Customer Churn Using Deep Learning', in *2021 International Conference on Digital Society and Intelligent Systems (DSInS)*. *2021 International Conference on Digital Society and Intelligent Systems (DSInS)*, pp. 319–324. Available at: https://doi.org/10.1109/DSInS54396.2021.9670561.

Saha, S. *et al.* (2024) 'ChurnNet: Deep Learning Enhanced Customer Churn Prediction in Telecommunication Industry', *IEEE Access*, 12, pp. 4471–4484. Available at: https://doi.org/10.1109/ACCESS.2024.3349950.

Schonlau, M. and Zou, R. (2020) 'The random forest algorithm for statistical learning', *The Stata Journal: Promoting communications on statistics and Stata*, 20, pp. 3–29. Available at: https://doi.org/10.1177/1536867X20909688.

Singh, P.P. *et al.* (2024) 'Investigating customer churn in banking: a machine learning approach and visualization app for data science and management', *Data Science and Management*, 7(1), pp. 7–16. Available at: https://doi.org/10.1016/j.dsm.2023.09.002.

Telecom Churn Prediction (no date). Available at: https://kaggle.com/code/kanuriviveknag/telecom-churn-prediction (Accessed: 10 August 2024).

Wang, J. *et al.* (2024) 'IMWMOTE: A novel oversampling technique for fault diagnosis in heterogeneous imbalanced data', *Expert Systems with Applications*, 251, p. 123987. Available at: https://doi.org/10.1016/j.eswa.2024.123987.

Wei, J. *et al.* (2020) 'NI-MWMOTE: An improving noise-immunity majority weighted minority oversampling technique for imbalanced classification problems', *Expert Systems with Applications*, 158, p. 113504. Available at: https://doi.org/10.1016/j.eswa.2020.113504.

Wongvorachan, T., He, S. and Bulut, O. (2023) 'A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining', *Information*, 14(1), p. 54. Available at: https://doi.org/10.3390/info14010054.

Yu, W. and Weng, W. (2022) 'Customer Churn Prediction Based on Machine Learning', in *2022 4th International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*. *2022 4th International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, pp. 870–878. Available at: https://doi.org/10.1109/AIAM57466.2022.00176.

Zhao, H., Zuo, X. and Xie, Y. (2022) *Customer Churn Prediction by Classification Models in Machine Learning*, p. 407. Available at: https://doi.org/10.1109/ICEEE55327.2022.9772553.