Government of Nepal
Bedkot Municipality

Bedkot, Kanchanpur

2024

A Proposal Report on
"Customer Churn Prediction using Machine Learning Models"

In partial fulfillment of the requirement for AI Training in Bedkot Municipality

Submitted to:
Asst. Professor Yagya Raj Pandey

Submitted By:
Niraj Bahadur Pal

Abstract

The term “churn” occurs in the situation of subscription products and symbolizes that customers are cancelling a service. Churn also applies to services and products that clients are reaching out with over a significant period. Customer churn is a critical concern for the telecommunication industry. Understanding and predicting customer churn can lead to more effective retention strategies and an increase in profitability. Predicting customer churn allows telecommunication companies to identify potentially dissatisfied customers early on and take proactive measures to retain them. Due to a large client base, the telecom industry generates a large volume of data daily. Decision makers and business analysts stressed that acquiring new customers is more expensive than retaining existing ones. Business analysts and customer relationship management (CRM) analysts must understand the reasons for customer churn as well as behavior patterns from existing churn data. The primary goal is to develop and evaluate predictive models that can accurately identify customers at risk of churn. Key processes include data preprocessing, exploratory data analysis (EDA), feature engineering, model training, and evaluation. The results demonstrate the effectiveness of logistic regression, random forest, gradient boosting, and decision tree classifiers, with a focus on metrics such as accuracy, precision, recall, F1 score, and ROC-AUC.

Keywords: Machine learning; supervised learning; churn prediction; CRM; telecom; retention.

Table of Content

Abstract i

Table of Content ii

Chapter 1: Introduction 1

1.1 Introduction 1

1.2 Problem Statement 2

1.3 Objective 3

1.4 Research Questions 3

1.5 Solution Approach 3

Chapter 2: Background and Literature Review 4

2.1 Background 4

2.2 Relate Works 4

Chapter 3: Methodology 6

3.1 Methodology 6

3.1.1 Data Collection and Preprocessing 6

3.1.2 Data Preprocessing 7

3.1.3 Exploratory Data Analysis (EDA) 8

3.1.4 Feature Engineering 9

3.1.5 Model Building 10

3.2 Tools Used 11

3.3 Experimental Environment 12

3.4 Performance Measures 12

Chapter 4: Results and Discussion 14

4.1 Experimental Results 14

4.1.1 Random Forest: 14

4.1.2 Gradient Boosting: 15

4.1.2 Decision Tree Model: 16
4.2 Result Analysis and Interpretation....................17
Chapter 5: Conclusion and Recommendation....................20
5.1 Conclusion 20
5.2 Recommendations....................20
References....................21

Chapter 1: Introduction

1.1 Introduction

Churn occurs when a client cancels a subscription or discontinues using a service. While service companies often focus on customer acquisition, minimizing churn is crucial for achieving long-term success. If churn is not addressed proactively, the service will not reach its full potential. The term "churn" originates from "churn rate," which refers to the ratio of customers leaving within a specific time frame, leading to changes in the customer base over time. Historically, "churn" meant "to move about vigorously," as in churning butter, but in business, it can be used both as a verb ("the customer is churning") and as a noun ("the customer is a churn") [1].

Conversely, retaining customers who continue to use a service and renew their subscriptions, is termed customer retention, the positive counterpart to churn. Reducing churn equates to increasing customer retention. When aiming to retain clients for extended periods, it's essential to prioritize not only saving at-risk customers but also keeping them engaged. Engaged customers are more likely to upgrade to advanced service versions, providing additional revenue. Thus, the main goals for services with continuous customer interactions include boosting engagement, reducing churn, and upselling. These objectives share a common focus on customer attention and satisfaction.

Customer churn refers to the phenomenon where customers stop using a company's products or services, which can significantly impact the profitability and sustainability of a business. In the highly competitive telecommunications industry, retaining existing customers is often more cost-effective than acquiring new ones. Therefore, predicting customer churn has become a crucial aspect of customer relationship management (CRM) strategies. [2]

Significance of Churn Prediction:

- **Financial Impact:** High churn rates directly reduce revenue and can increase costs associated with customer acquisition.
- **Customer Retention:** Identifying customers likely to churn allows companies to implement targeted retention strategies, such as personalized offers or improved customer service, thereby enhancing customer loyalty.

- **Market Competitiveness:** Effective churn prediction models can provide a competitive edge by enabling proactive measures to retain valuable customers.

Challenges in Churn Prediction:

- **Complexity of Factors:** Customer churn is influenced by many factors, including service quality, pricing, customer satisfaction, and external competitive actions. This complexity requires sophisticated analytical methods to accurately predict churn.
- **Data Quality:** High-quality, relevant data is essential for building effective predictive models. Inconsistent or incomplete data can lead to inaccurate predictions and misguided strategies.
- **Model Selection:** Choosing the right machine learning model is crucial for balancing accuracy, interpretability, and computational efficiency. [3]

Machine Learning in Churn Prediction: Machine learning (ML) offers powerful tools for analyzing large datasets to identify patterns and predict outcomes. ML models can handle complex relationships between variables and adapt to new data, making them well-suited for churn prediction. Commonly used models include:

- **Logistic Regression:** A simple, yet effective model for binary classification problems like churn prediction.
- **Decision Trees:** These provide interpretable rules for churn prediction based on feature importance.
- **Random Forest and Gradient Boosting:** Ensemble methods that combine multiple decision trees to improve prediction accuracy and robustness.[4]

1.2 Problem Statement

Telecommunications companies face substantial losses due to customer churn, which involves customers discontinuing their services. Predicting churn is challenging due to the complex interplay of various factors influencing customer decisions. This project aims to develop predictive models to identify at-risk customers, enabling proactive retention efforts.

1.3 Objective

The primary objective of this study is to develop and evaluate machine learning models for predicting customer churn in the telecommunications sector. By leveraging a rich dataset of customer attributes and behaviors, the study aims to:

- **Preprocess and analyze data** to identify key factors influencing churn.
- **Train and compare multiple ML models** to determine the most effective one for churn prediction.
- **Provide actionable insights** that can help telecom companies implement effective retention strategies.

1.4 Research Questions

- What is the role of customer churn prediction in telecom using machine learning in big data platform?
- What are all the significances of the customer churn models in telecom industry?
- What are all the overall impacts of churn rate in telecom industry?
- Which algorithms are all effective in reducing churn rate in telecom companies?

1.5 Solution Approach

The major contribution of the current study is to present a model of churn prediction that helps telecom companies to find users who are more intending to churn. The model formulated in this employs ML algorithms on the platform of Big-Data and develops an innovative way of reducing churn. The significance of this purpose is apparent, given the reality that the expenditure for acquisition of customers is comparatively higher than that of the retention of customers. In consequence, techniques to develop and apply machine learning models are considered necessary and are critical business intelligence applications.

Chapter 2: Background and Literature Review

2.1 Background

It is well-known that retaining clients with an increased churn risk is one of the hardest challenges (Miguéis, et al., 2012) because nowadays there are a large number of service and products providers, and customers have a lot of options to churn. Usually, clients tend to compare their providers with others, and this leads to churning (Balle, et al., 2013). According to P. Kotler in 1994, the price of convincing a client not to churn to the opponent is 16 times less than the price of finding and determining contact with a new client. Also, the cost of convincing new clients is 5 to 6 times more than for maintaining the existing ones.

According to studies, it is approximated that a service supplier can increase their returns by between 25% and 85% by decreasing the customer churn rate by 5% (Reichheld and Sasser, 1990) [5].

Churn affects businesses everywhere around the world and churn rates fluctuate often. Mobile phone companies in Europe have churn rates between 20% and 38%. Wireless business could improve their earnings by almost 10% if they took steps in order to reduce churn [6].

2.2 Relate Works

Several techniques have been proposed in literature for churn prediction. These techniques include data mining, machine learning and hybrid strategies. These techniques help businesses identify, predict, and retain churn customers, as well as aid decision-making and CRM. Decision trees are the most commonly used method for predicting problems related to customer churn [7]. Decision trees have the limitation that they are not suitable for complex nonlinear connections between attributes but perform better for linear data where attributes are interdependent. However, the accuracy of decision trees are improved using pruning. In [8], the authors proposed a forecasting approach that uses a two-phase strategy based on their recency, frequency and monetary value (RFM). The related functions of the RFM are classifieds into four clusters in the first phase. In the second phase, the churn data collected in the first phase are extracted and evaluated using decision trees, Neural Networks and other machine learning algorithms. Experimental results show prediction results are better using hybrid approaches. In [9], the authors proposed a hybrid approach for churn prediction by combining genetic programming with induction algorithm from an existing tree. The proposed algorithm used the behaviour of customers to generate classification rules. The proposed model is used to predict different custom groups based on time of use, location, and underlying social networks, and represents a practical approach to churn models at the human level rather than the mathematical level. The authors in used three models to evaluate the performance of churn models. The models include ANN, classification trees and logistic regression. They selected ten features based on their exploratory data analysis and business experience. AlShourbaji et al.proposed a novel Feature Selection strategy ACO-RSA. In this approach two metaheuristic algorithms algorithms (ant colony optimization (ACO) and reptile search algorithm (RSA)) are integrated for the selection of subset features that are important for churn prediction. The proposed model is evaluated using the state- of-the-art test functions and open-source datasets for churn preductions. This is evaluated alongside Standard ACO, Gray Wolf Optimiser, Multiverse Optimser and the results show that ACO-RSA outperforms the compared approaches. Similarly, the authors in propose a new framework for saturated markets. In, the authors propose a machine learning churning model with six phases. Preprocessing and analysis of features are the first two phases while the third phase is the feature selection phase using gravitational search algorithm. With ratios of 80% and 20%, the data is divided inot tensile and test set respectively. Logistic regression, support vector machines, decision trees, naives bayes and random forest were evaluated and K-fold cross validation was used to optimise the hyperparameters. The results of the experiments show that Adaboost and XGboost outperformed the other approaches compared. Al-Najjar et al. [10] proposed a churn prediction model for the prediction of credit card cancellations. In their proposed approach feature selection was adopted with five machine learning models. Independent selection of variabes was carried out using k-nearest neighbor selection, two-level clustering and feature selection. Five machine learning models were evaluated incuding c5-tress, Bayesian networks, chi-square trees for automatic interaction detection (CHAID) and neural networks. Experimental results showed that the integration of multiple feature variables improved the accuracy of the performance of the prediction model.Similarly, Zhang et al. proposed a churn prediction model for the telecom industry. Three chinese telecommunication companies were used for data collection. Their predicion model was built using logistic regression and Fisher’s discriminant equation. Their results showed that logistic regression outperformed the other model compared with a prediction accuracy of 94%.

Chapter 3: Methodology

3.1 Methodology

In this section, the proposed model is presented in Fig. 1 with a detailed description. The methodology for this project is structured to systematically handle data collection, preprocessing, exploratory analysis, feature engineering, model building, and evaluation. Each step is crucial for ensuring the effectiveness and accuracy of the predictive models.

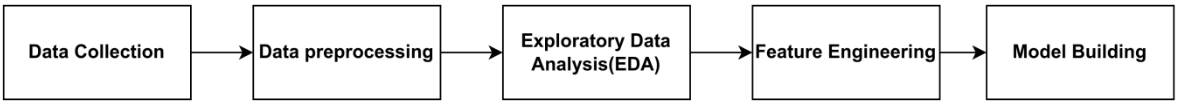
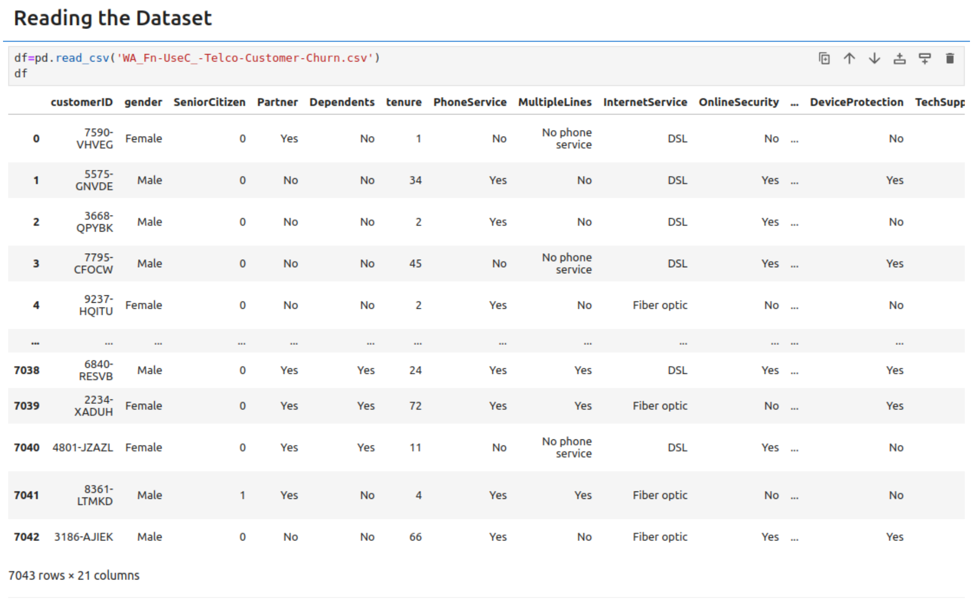


Figure 1: Prediction Model for Customer Churn

3.1.1 Data Collection and Preprocessing

Data Collection:

- The dataset used for this project was sourced from Kaggle, titled "WA_Fn-UseC_-Telco-Customer-Churn.csv". It contains 7043 records and 21 features, including customer demographics, account information, and usage details [11].



Reading the Dataset

```
df=pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
df
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupp
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	Yes	
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	Yes	
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...	No	
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...	No	
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...	Yes	

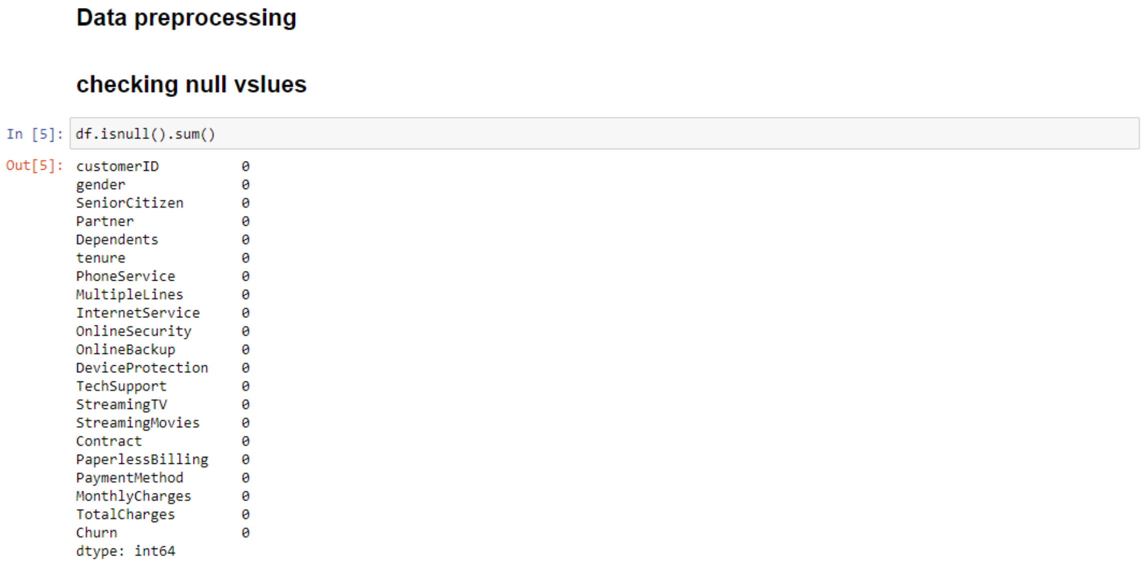
7043 rows × 21 columns

Figure 2: Reading the Dataset of csv

3.1.2 Data Preprocessing

Data Cleaning:

- Initial data inspection revealed that the TotalCharges column had some missing values. These were converted to numeric data type and filled with the median value to handle missing entries.
- Ensured that all other columns had no missing values and were appropriately formatted for analysis.



Data preprocessing

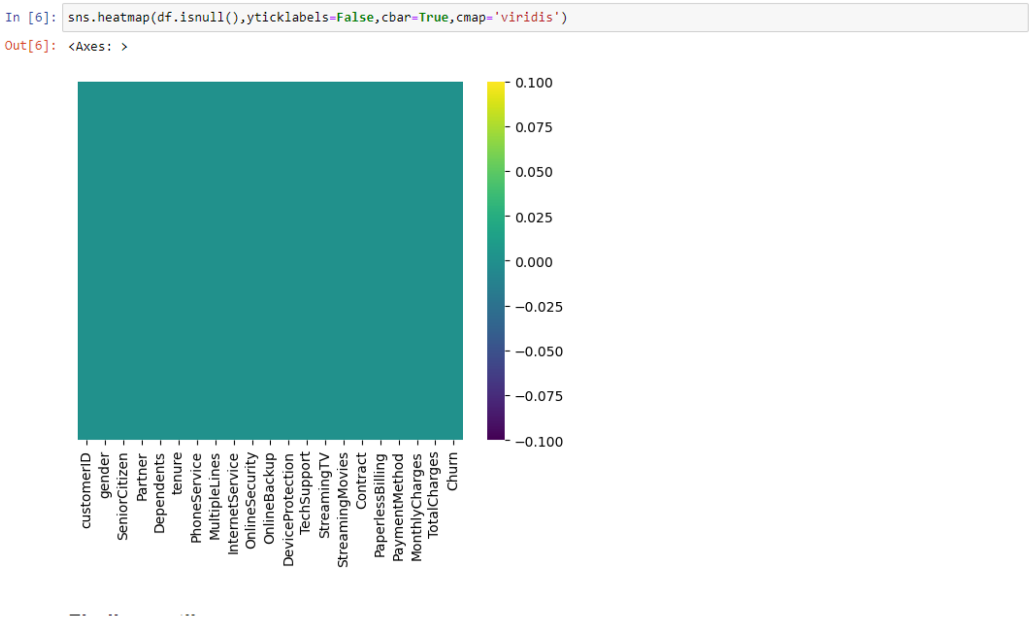
checking null vslues

```
In [5]: df.isnull().sum()

Out[5]: customerID          0
        gender              0
        SeniorCitizen       0
        Partner             0
        Dependents          0
        tenure              0
        PhoneService        0
        MultipleLines       0
        InternetService     0
        OnlineSecurity      0
        OnlineBackup        0
        DeviceProtection    0
        TechSupport         0
        StreamingTV         0
        StreamingMovies     0
        Contract            0
        PaperlessBilling    0
        PaymentMethod       0
        MonthlyCharges      0
        TotalCharges        0
        Churn               0
        dtype: int64
```

Figure 3: Data Cleaning

- **Categorical Variables Conversion:** Categorical variables were converted into numerical representations using label encoding and mapping. For instance, binary categorical variables such as gender, Partner, Dependents, etc., were mapped to 0 and 1.
- Multiclass categorical variables such as InternetService, PaymentMethod, etc., were label encoded.
- **Scaling Numerical Features:** Numerical features such as tenure, MonthlyCharges, and TotalCharges were scaled using StandardScaler to ensure they are on the same scale, which is crucial for models like Logistic Regression.



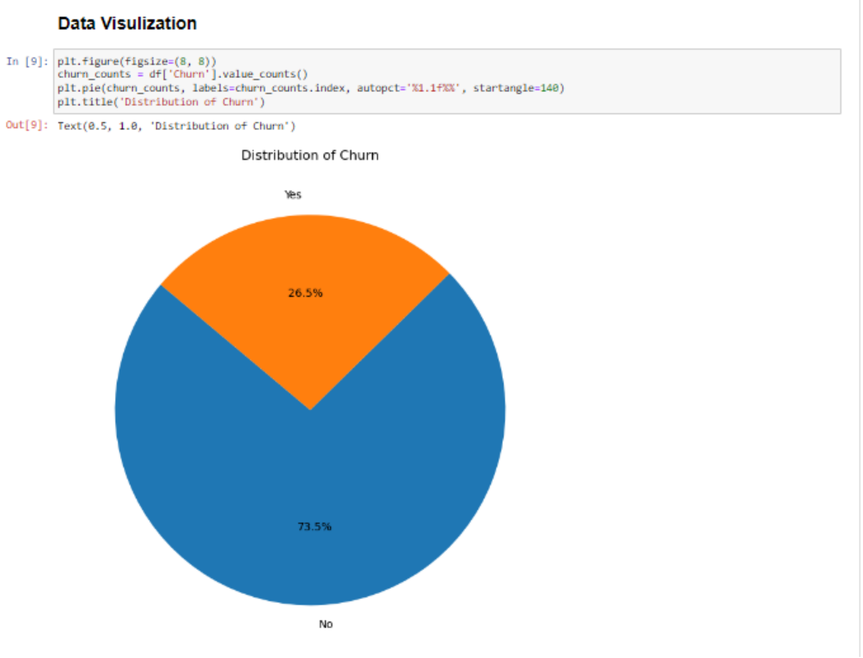
```
In [6]: sns.heatmap(df.isnull(),yticklabels=False,cbar=True,cmap='viridis')
Out[6]: <Axes: >
```

Figure 4: Data preprocessing

3.1.3 Exploratory Data Analysis (EDA)

Distribution Analysis:

- Count plots and pie charts were used to visualize the distribution of categorical variables like Churn, gender, Contract, and PaymentMethod
- Box plots were used to understand the distribution and outliers in numerical features such as tenure and MonthlyCharges.



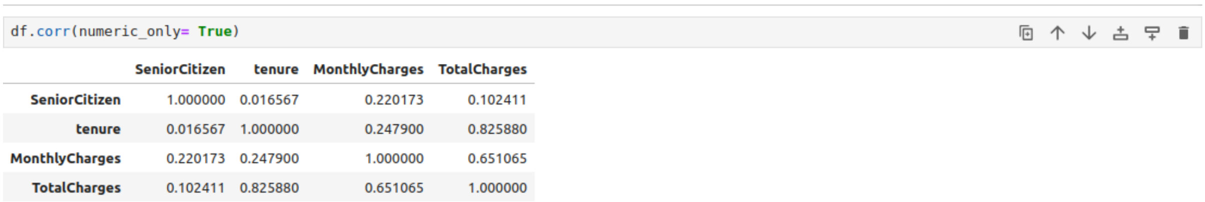
Data Visulization

```
In [9]: plt.figure(figsize=(8, 8))
        churn_counts = df['Churn'].value_counts()
        plt.pie(churn_counts, labels=churn_counts.index, autopct='%1.1f%%', startangle=140)
        plt.title('Distribution of Churn')
Out[9]: Text(0.5, 1.0, 'Distribution of Churn')
```

Figure 5: Distribution Analysis

Correlation Analysis:

- Conducted correlation analysis to understand the relationships between different features and their potential impact on customer churn. Heatmaps were used to visualize the correlation matrix.



```
df.corr(numeric_only= True)
```

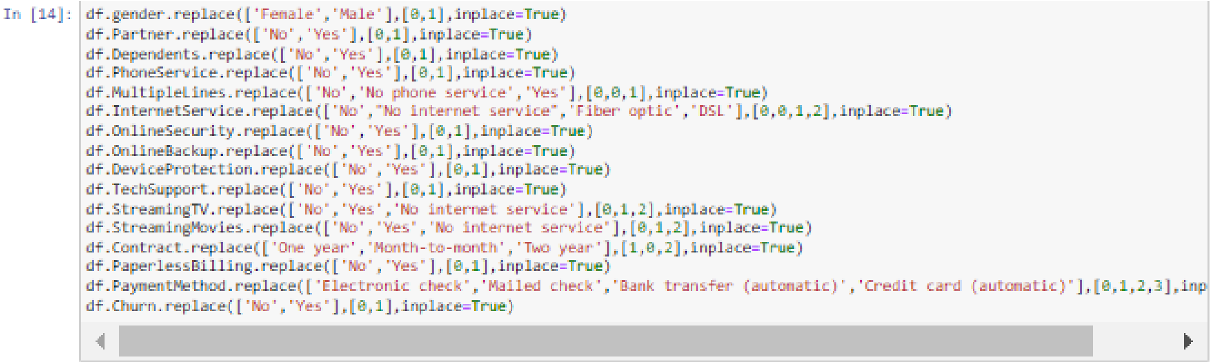
	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
SeniorCitizen	1.000000	0.016567	0.220173	0.102411
tenure	0.016567	1.000000	0.247900	0.825880
MonthlyCharges	0.220173	0.247900	1.000000	0.651065
TotalCharges	0.102411	0.825880	0.651065	1.000000

Figure 6: Correlation Analysis

3.1.4 Feature Engineering

Binary Encoding:

- Binary categorical variables (e.g., gender, Partner, Dependents, PhoneService, PaperlessBilling, Churn) were encoded into 0 and 1.



```
In [14]: df.gender.replace(['Female','Male'],[0,1],inplace=True)
         df.Partner.replace(['No','Yes'],[0,1],inplace=True)
         df.Dependents.replace(['No','Yes'],[0,1],inplace=True)
         df.PhoneService.replace(['No','Yes'],[0,1],inplace=True)
         df.MultipleLines.replace(['No','No phone service','Yes'],[0,0,1],inplace=True)
         df.InternetService.replace(['No',"No internet service",'Fiber optic','DSL'],[0,0,1,2],inplace=True)
         df.OnlineSecurity.replace(['No','Yes'],[0,1],inplace=True)
         df.OnlineBackup.replace(['No','Yes'],[0,1],inplace=True)
         df.DeviceProtection.replace(['No','Yes'],[0,1],inplace=True)
         df.TechSupport.replace(['No','Yes'],[0,1],inplace=True)
         df.StreamingTV.replace(['No','Yes','No internet service'],[0,1,2],inplace=True)
         df.StreamingMovies.replace(['No','Yes','No internet service'],[0,1,2],inplace=True)
         df.Contract.replace(['One year','Month-to-month','Two year'],[1,0,2],inplace=True)
         df.PaperlessBilling.replace(['No','Yes'],[0,1],inplace=True)
         df.PaymentMethod.replace(['Electronic check','Mailed check','Bank transfer (automatic)','Credit card (automatic)'],[0,1,2,3],inp
         df.Churn.replace(['No','Yes'],[0,1],inplace=True)
```

Figure 7: Binary Encoding

Label Encoding:

- Categorical variables with more than two categories (InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, PaymentMethod) were label encoded.

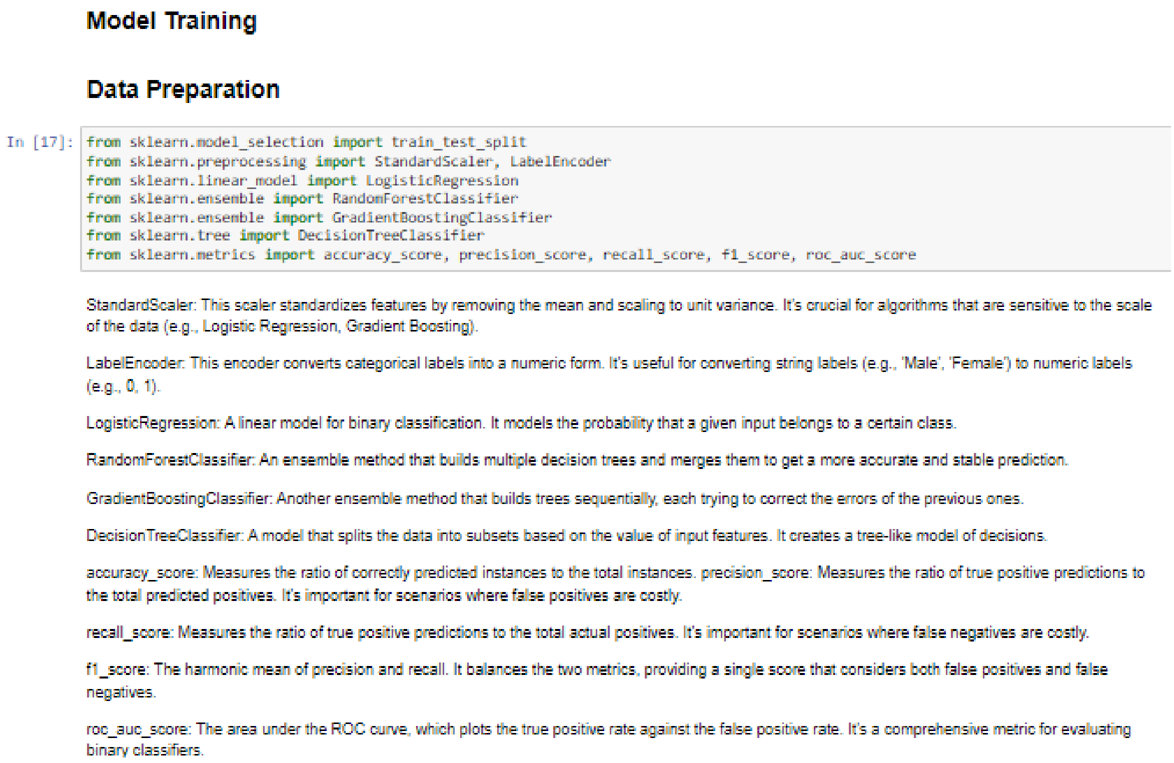
Handling Imbalance:

- The target variable Churn was found to be imbalanced. Techniques such as oversampling the minority class or using stratified sampling were considered to handle this imbalance.

3.1.5 Model Building

Model Selection:

- Various machine learning models were selected to predict customer churn, including Logistic Regression, Random Forest, Gradient Boosting, and Decision Tree.



Model Training

Data Preparation

```
In [17]: from sklearn.model_selection import train_test_split
         from sklearn.preprocessing import StandardScaler, LabelEncoder
         from sklearn.linear_model import LogisticRegression
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.ensemble import GradientBoostingClassifier
         from sklearn.tree import DecisionTreeClassifier
         from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score
```

StandardScaler: This scaler standardizes features by removing the mean and scaling to unit variance. It's crucial for algorithms that are sensitive to the scale of the data (e.g., Logistic Regression, Gradient Boosting).

LabelEncoder: This encoder converts categorical labels into a numeric form. It's useful for converting string labels (e.g., 'Male', 'Female') to numeric labels (e.g., 0, 1).

LogisticRegression: A linear model for binary classification. It models the probability that a given input belongs to a certain class.

RandomForestClassifier: An ensemble method that builds multiple decision trees and merges them to get a more accurate and stable prediction.

GradientBoostingClassifier: Another ensemble method that builds trees sequentially, each trying to correct the errors of the previous ones.

DecisionTreeClassifier: A model that splits the data into subsets based on the value of input features. It creates a tree-like model of decisions.

accuracy_score: Measures the ratio of correctly predicted instances to the total instances. precision_score: Measures the ratio of true positive predictions to the total predicted positives. It's important for scenarios where false positives are costly.

recall_score: Measures the ratio of true positive predictions to the total actual positives. It's important for scenarios where false negatives are costly.

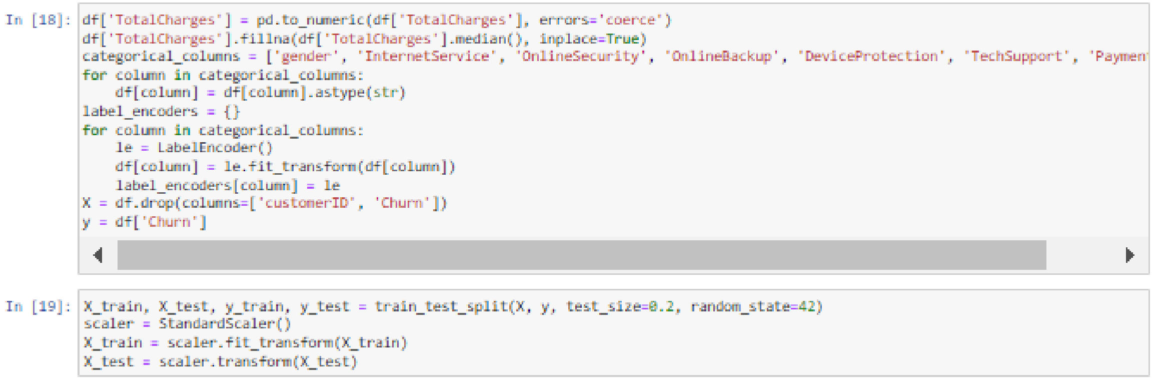
f1_score: The harmonic mean of precision and recall. It balances the two metrics, providing a single score that considers both false positives and false negatives.

roc_auc_score: The area under the ROC curve, which plots the true positive rate against the false positive rate. It's a comprehensive metric for evaluating binary classifiers.

Figure 8: Model Selection

Training and Testing Split:

- The dataset was split into training (80%) and testing (20%) sets using train_test_split to ensure that model evaluation is done on unseen data.



```
In [18]: df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
         df['TotalCharges'].fillna(df['TotalCharges'].median(), inplace=True)
         categorical_columns = ['gender', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Paymen
         for column in categorical_columns:
             df[column] = df[column].astype(str)
         label_encoders = {}
         for column in categorical_columns:
             le = LabelEncoder()
             df[column] = le.fit_transform(df[column])
             label_encoders[column] = le
         X = df.drop(columns=['customerID', 'Churn'])
         y = df['Churn']

In [19]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
         scaler = StandardScaler()
         X_train = scaler.fit_transform(X_train)
         X_test = scaler.transform(X_test)
```

Figure 9: Training and Testing Split

Model Training:

- Each model was trained on the training set, with hyperparameters tuned using cross-validation where necessary. Hyperparameter tuning helps in finding the best parameters for the models to improve their performance.

Model Evaluation:

- Models were evaluated on the test set using key performance metrics: accuracy, precision, recall, F1 score, and ROC-AUC. The **evaluate_model** function was defined to return these metrics for consistent comparison across different models.

3.2 Tools Used

- **Programming Language**: Python
- **Libraries**:
  - **pandas** for data manipulation
  - **Scikit-learn** for machine learning models
  - **Matplotlib and Seaborn** for data visualization
- **Development Environment**: Jupyter Notebook

3.3 Experimental Environment

- Experiments were conducted on a standard laptop equipped with an Intel Core i5 processor, 8GB of RAM, and the Ubuntu operating system.
- The experiments were conducted on a standard computing environment with Python installed, using Jupyter Notebooks for coding and visualization.
- Key libraries used include Pandas for data manipulation, Scikit-learn for machine learning models, and Matplotlib and Seaborn for data visualization.

3.4 Performance Measures

To evaluate the performance of machine learning model, it is very important to utilize the correct metric. When an incorrect metric is used, it may cause the machine learning model to perform poorly when used in real life [12]. Some standard evaluation metrics are:

Accuracy:

- The ratio of correctly predicted instances to the total instances. It gives an overall measure of model performance.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Accuracy\ (Y) = \frac{TP+TN}{Total}$$

Precision:

- Precision is the ratio of true positives and total positives predicted. It indicates the model’s accuracy in predicting positive instances. The formula for Precision is given as:

$$P = \frac{TP}{TP + FP}$$

Recall:

- The ratio of true positive predictions to the actual positives. It measures the model's ability to identify all relevant instances.

$$R = \frac{TP}{TP + FN}$$

F1 Score:

- The harmonic mean of precision and recall, providing a balance between the two metrics, especially useful for imbalanced datasets.

$$R = \frac{Precision}{Precision + Recall}$$

ROC-AUC Score:

- ROC curve is a plot of **true positive rate** (recall) against **false positive rate** (TN/(TN+FP))
- The higher the AUC-ROC score means the better the model performance.

Each of these steps and measures ensures that the predictive models developed are robust, accurate, and reliable for practical implementation in identifying customer churn.

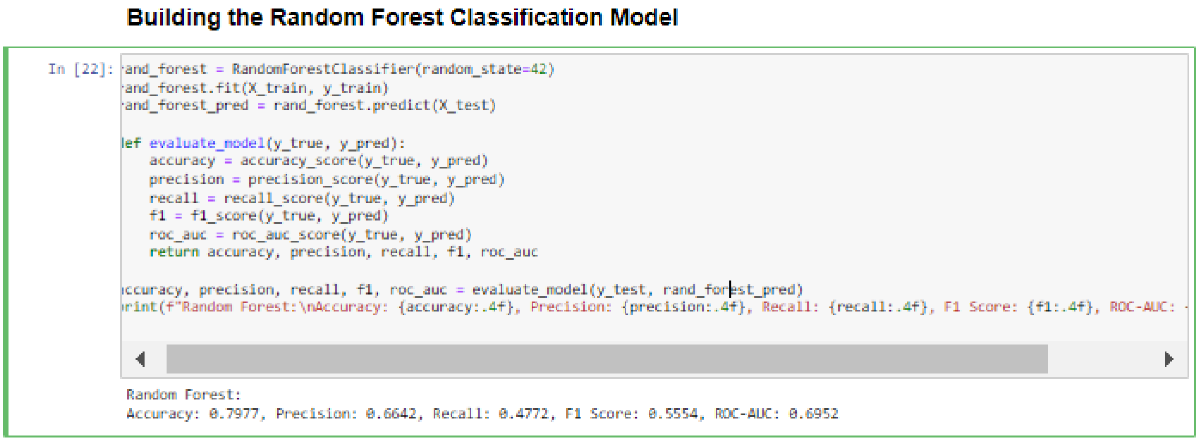
Chapter 4: Results and Discussion

This chapter presents the outcomes of the machine learning models applied to the customer churn prediction problem and provides a detailed analysis and interpretation of these results. The discussion focuses on comparing the performance of different models and understanding the implications of the results for practical applications.

4.1 Experimental Results

4.1.1 Random Forest:

An ensemble method that builds multiple decision trees and merges them to get a more accurate and stable prediction. The model was implemented using the **RandomForestClassifier** class from the scikit-learn library, with a random seed set to ensure reproducibility (**random_state=42**).



Building the Random Forest Classification Model

```
In [22]: and_forest = RandomForestClassifier(random_state=42)
         and_forest.fit(X_train, y_train)
         and_forest_pred = rand_forest.predict(X_test)

         ef evaluate_model(y_true, y_pred):
            accuracy = accuracy_score(y_true, y_pred)
            precision = precision_score(y_true, y_pred)
            recall = recall_score(y_true, y_pred)
            f1 = f1_score(y_true, y_pred)
            roc_auc = roc_auc_score(y_true, y_pred)
            return accuracy, precision, recall, f1, roc_auc

         ccuracy, precision, recall, f1, roc_auc = evaluate_model(y_test, rand_forest_pred)
         rint(f"Random Forest:\nAccuracy: {accuracy:.4f}, Precision: {precision:.4f}, Recall: {recall:.4f}, F1 Score: {f1:.4f}, ROC-AUC:
```

```
Random Forest:
Accuracy: 0.7977, Precision: 0.6642, Recall: 0.4772, F1 Score: 0.5554, ROC-AUC: 0.6952
```

Figure 10: Random Forest

Model Evaluation

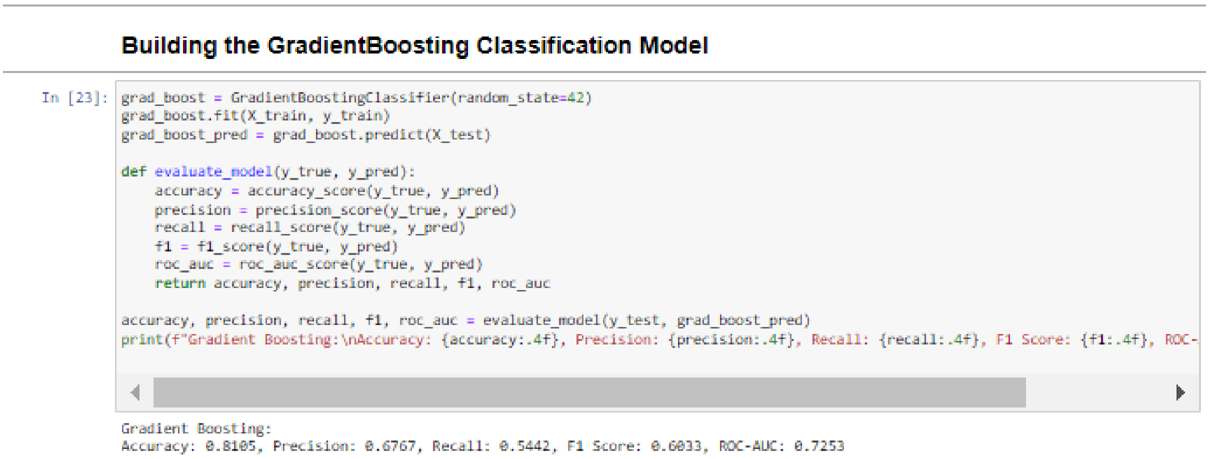
To assess the performance of the Random Forest model, various evaluation metrics were calculated, including accuracy, precision, recall, F1 score, and ROC-AUC score. These metrics provide insights into different aspects of the model's predictive ability. The evaluate_model function was employed to compute these metrics based on the model's predictions on the test set.

Results

- **Accuracy:** 0.7977
- **Precision:** 0.6642
- **Recall:** 0.4772
- **F1 Score:** 0.5554
- **ROC-AUC Score:** 0.6952

4.1.2 Gradient Boosting:

In this section, a Gradient Boosting Classification model was developed to predict customer churn in the telecommunications dataset. The Gradient Boosting algorithm is a machine learning technique that builds multiple weak learners sequentially, with each new model correcting errors made by the previous one. The model was implemented using the **GradientBoostingClassifier** class from the scikit-learn library, with a random seed set to ensure reproducibility (r**andom_state=42**).



Building the GradientBoosting Classification Model

```
In [23]: grad_boost = GradientBoostingClassifier(random_state=42)
         grad_boost.fit(X_train, y_train)
         grad_boost_pred = grad_boost.predict(X_test)

         def evaluate_model(y_true, y_pred):
             accuracy = accuracy_score(y_true, y_pred)
             precision = precision_score(y_true, y_pred)
             recall = recall_score(y_true, y_pred)
             f1 = f1_score(y_true, y_pred)
             roc_auc = roc_auc_score(y_true, y_pred)
             return accuracy, precision, recall, f1, roc_auc

         accuracy, precision, recall, f1, roc_auc = evaluate_model(y_test, grad_boost_pred)
         print(f"Gradient Boosting:\nAccuracy: {accuracy:.4f}, Precision: {precision:.4f}, Recall: {recall:.4f}, F1 Score: {f1:.4f}, ROC-
```

```
Gradient Boosting:
Accuracy: 0.8105, Precision: 0.6767, Recall: 0.5442, F1 Score: 0.6033, ROC-AUC: 0.7253
```

Figure 11: Gradient Boosting

Model Evaluation

To evaluate the performance of the Gradient Boosting model, various evaluation metrics were computed, including accuracy, precision, recall, F1 score, and ROC-AUC score. These metrics provide insights into different aspects of the model's predictive ability. The

evaluate_model function was utilized to calculate these metrics based on the model's predictions on the test set.

Results

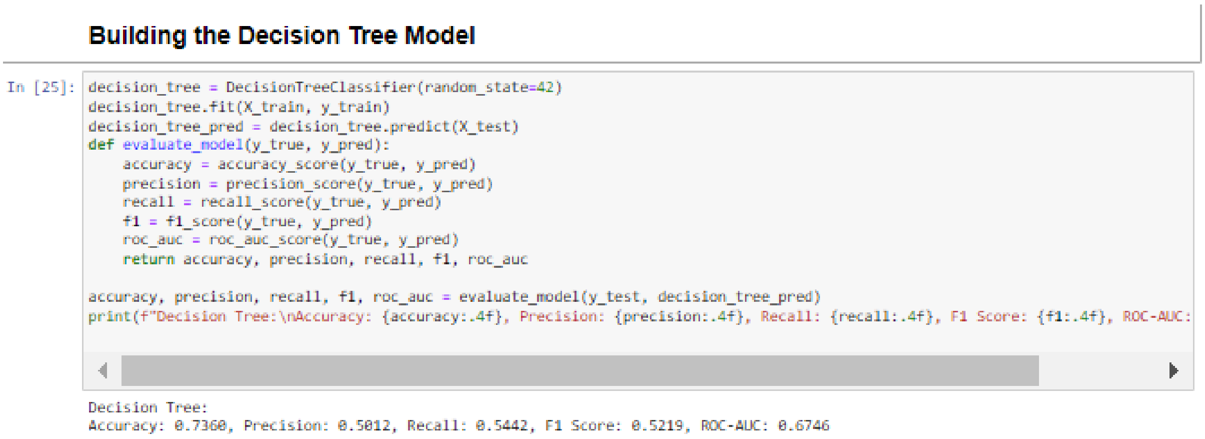
- **Accuracy:** 0.8105
- **Precision:** 0.6767
- **Recall:** 0.5442
- **F1 Score:** 0.6033
- **ROC-AUC Score:** 0.7253

4.1.2 Decision Tree Model:

This section outlines the development and evaluation of a Decision Tree Classifier model for predicting customer churn in the telecommunications dataset. The Decision Tree algorithm is a popular choice for classification tasks due to its simplicity and interp3retability. The model was instantiated using the **DecisionTreeClassifier** class from the scikit-learn library, with a fixed random state for reproducibility (**random_state=42**).

Model Evaluation

To assess the performance of the Decision Tree model, various evaluation metrics were computed, including accuracy, precision, recall, F1 score, and ROC-AUC score. These metrics provide insights into different aspects of the model's predictive ability, such as its overall accuracy and its ability to correctly identify positive cases (churn) and minimize false Positives.



Building the Decision Tree Model

```
In [25]: decision_tree = DecisionTreeClassifier(random_state=42)
         decision_tree.fit(X_train, y_train)
         decision_tree_pred = decision_tree.predict(X_test)
         def evaluate_model(y_true, y_pred):
             accuracy = accuracy_score(y_true, y_pred)
             precision = precision_score(y_true, y_pred)
             recall = recall_score(y_true, y_pred)
             f1 = f1_score(y_true, y_pred)
             roc_auc = roc_auc_score(y_true, y_pred)
             return accuracy, precision, recall, f1, roc_auc

         accuracy, precision, recall, f1, roc_auc = evaluate_model(y_test, decision_tree_pred)
         print(f"Decision Tree:\nAccuracy: {accuracy:.4f}, Precision: {precision:.4f}, Recall: {recall:.4f}, F1 Score: {f1:.4f}, ROC-AUC:
```

```
Decision Tree:
Accuracy: 0.7360, Precision: 0.5012, Recall: 0.5442, F1 Score: 0.5219, ROC-AUC: 0.6746
```

Figure 12: Decision Tree Model

Results

- **Accuracy:** 0.7458
- **Precision:** 0.4969
- **Recall:** 0.5278
- **F1 Score:** 0.5119
- **ROC-AUC Score:** 0.688

Performance Summary:

- The **Gradient Boosting** model showed the highest accuracy (0.8105) and ROC-AUC score (0.7253), indicating its superior ability to distinguish between churn and non-churn customers.
- The **Random Forest** model performed well with a balanced precision (0.6642) and F1 score (0.5554), but had a lower recall (0.4772), suggesting it missed a significant number of actual churn cases.
- The **Decision Tree** model had the lowest accuracy (0.7458) among the three, with a moderate recall (0.5278) but lower precision (0.4969) and F1 score (0.5119).

4.2 Result Analysis and Interpretation

Accuracy:

- Accuracy measures the proportion of correctly predicted instances. Gradient Boosting achieved the highest accuracy, indicating it correctly classified a higher number of churn and non-churn customers compared to the other models.

Precision and Recall:

- Precision is crucial when the cost of false positives is high. In this context, a false positive means predicting a customer will churn when they will not, which could lead to unnecessary retention efforts.
- Recall is essential when the cost of false negatives is high. A false negative means predicting a customer will not churn when they actually will, leading to missed opportunities for retention.
- The Gradient Boosting model achieved a good balance between precision (0.6767) and recall (0.5442), making it more reliable for practical applications where both false positives and false negatives have significant implications.

F1 Score:

- The F1 score balances precision and recall, providing a single metric to evaluate the model's performance. The Gradient Boosting model had the highest F1 score (0.6033), indicating its robustness in handling both precision and recall effectively.

ROC-AUC Score:

- The ROC-AUC score evaluates the model's ability to distinguish between classes across all thresholds. The Gradient Boosting model's ROC-AUC score of 0.7253 suggests it has a strong capability to differentiate between churn and non-churn customers.

Confusion Matrix Analysis:

- Confusion matrices for each model were examined to understand the distribution of true positives, true negatives, false positives, and false negatives. This analysis helps in identifying the strengths and weaknesses of each model in predicting churn.

Model Comparison:

- Gradient Boosting outperformed the other models in almost all metrics, making it the best choice for predicting customer churn in this dataset.

- Random Forest showed competitive performance but with a lower recall, indicating it may need further tuning or additional data to improve its ability to capture all churn cases.
- Decision Tree, while simpler and easier to interpret, had lower overall performance and may benefit from being part of an ensemble method to enhance its predictive power.

Practical Implications:

- Implementing the Gradient Boosting model in a telecommunications company's CRM system could enable more effective identification of at-risk customers.
- The insights gained from feature importance in the models can guide targeted retention strategies, focusing on the most influential factors driving churn.

Chapter 5: Conclusion and Recommendation

5.1 Conclusion

The study successfully demonstrated the use of machine learning models, particularly Gradient Boosting, in effectively predicting customer churn with high accuracy and robust performance metrics. Implementing these models can help telecommunications companies proactively identify and retain at-risk customers, ultimately reducing churn rates and improving profitability.

5.2 Recommendations

Future work should explore additional data sources and advanced modeling techniques, such as deep learning, to further enhance predictive accuracy and model robustness. Continuous monitoring and refinement of the deployed model in a real-world environment are recommended to ensure its effectiveness and adapt to changing customer behavior patterns.

References

[1] Amal, M., Aksoy, M.S. and Alzahran, R., 2014. A Survey on Data Mining Techniques In Customer Churn Analysis For Telecom Industry. Int. Journal of Engineering Research and Applications, 4(5), pp.165-171.

[2] Mehta, N., Pickens, A., and Martinez, M., 2020. The Customer Success Economy. [online] Available at: <https://learning.oreilly.com/library/view/the-customer-success/9781119572763/> [Accessed 14 Apr. 2021].

[3] Davis, J., 2017. Measuring Marketing. [online] Available at: <https://learning.oreilly.com/library/view/measuring-marketing/9781501507229/> [Accessed 14 Apr 2021]

[4] Balle, B., Casas. B., Catarineu, A., Gavalda, R. and Manzano-Macho D., 2013. The Architecture of a Churn Prediction System Based on Stream Mining. In Artificial Intelligence Research and Development: Proceedings of the 16th International Conference of the Catalan Association for Artificial Intelligence, vol. 256, Article number: 157.

[5] Reichheld, F. and Sasser, W.E., 1990. Zero Defections: Quality Comes to Services. Harv. Bus. Rev., 68(5), pp.105–11.

[6] Vaidyanathan, A. and Ranago, R., 2020. The Customer Success Professional's Handbook. [online] Available at: <https://learning.oreilly.com/library/view/the-customer-success/9781119624615/> [Accessed 15 Apr. 2021].

[7] Ballings M, Van den Poel D, Verhagen E. Improving Customer Churn Prediction by Data Augmentation Using Pictorial Stimulus-Choice Data. In: Casillas J, Mart´ınez-Lopez FJ, Corchado Rodriguez JM, eds. Management Intelligent Systems. Springer Berlin Heidelberg; 2012:217-226.

[8] Kim S, Lee H. Customer Churn Prediction in Influencer Commerce: An Application of Decision Trees. Procedia Computer Science. 2022;199:1332-1339. DOI:10.1016/j.procs.2022.01.169

[9] Prabadevi B, Shalini R, Kavitha BR. Customer churning analysis using machine learning algorithms. International Journal of Intelligent Networks. 2023;4:145-154. DOI:10.1016/j.ijin.2023.05.005 '

[10] Khattak A, Mehak Z, Ahmad H, Asghar MU, Asghar MZ, Khan A. Customer churn prediction using composite deep learning technique. Scientific Reports. 2023;13(1):17294. DOI:10.1038/s41598-023-44396-w D. E. Ilea and P. F. Whelan,

“CTex—An Adaptive Unsupervised Segmentation Algorithm Based on Color-Texture Coherence,” *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1926–1939, 2008, doi: 10.1109/TIP.2008.2001047.

[11] Kaggle dataset source available at: <https://www.kaggle.com/datasets/blastchar/telco-customer-churn/>

[12] Hwang, Y.H., 2019. Hands-On Data Science for Marketing. [online] Available at: <https://learning.oreilly.com/library/view/hands-on-data-science/9781789346343/> [Accessed 15 Apr. 2021].