Customer Churn Prediction and Retention Strategies for a Telecommunications Company—ML Classification Project

Project Description

The project analysis aim was to identify the key indicators of customer churn for a telecommunications company and develop a machine learning model to predict which customers are likely to churn. The project provided insights into effective retention strategies that the company can implement to reduce customer churn. The data was processed and analyzed using various techniques such as data cleaning, bivariate and multivariate analysis, and exploratory data analysis. The best-performing model was selected and evaluated, and suggestions for model improvement were provided. The ultimate goal of this project was to help the telecommunications company reduce customer churn and improve customer retention.

Introduction

Telecom companies operate in a highly competitive industry, and customer churn is one of their biggest challenges. Customer churn refers to the rate at which customers stop using a company's services. This is a significant problem for telecom companies because acquiring new customers is more expensive than retaining existing ones. Therefore, companies need to identify customers who are likely to churn and take appropriate measures to retain them. In this article, I will discuss a customer churn prediction model and retention strategies for telecom companies. I will

use a classification model to predict customer churn and suggest some retention strategies that can be implemented to reduce churn rates.

Objective

The goal of this classification was to analyze the data and utilize machine learning algorithms to predict customer churn and retention strategies for the telco company.

Hypothesis and Questions

The analysis was guided by three(3) null hypothesis and their corresponding alternate hypothesis respectively. also, six(6) questions were asked.

Hypothesis

ONE (1)

H0*: There is no significant difference in churn rates between male and female customers.*

H1*: There is a significant difference in churn rates between male and female customers.*

Two (2)

H0: There is no significant relationship between the customer's internet service provider and their likelihood to churn.

H1: There is a significant relationship between the customer's internet service provider and their likelihood to churn.

Three(3)

H0: There is no significant difference in churn rates between customers on different types of payment methods.

H1: There is a significant difference in churn rates between customers on different types of payment methods.

Questions

Here are five questions that guided the project:

1. *What percentage of customers have churned?*
2. *Is there a correlation between a customer's length of tenure with the company and their likelihood of churning?*

3. *Are there any specific groups of customers based on demographic that are more likely to churn than others?*
4. *Can customer retention be improved by offering longer contract terms?*
5. *How much money could the company save by reducing customer churn?*
6. *What is the relationship between Internet Services and churn rate?*

Data Understanding

The dataset used in this classification project is a Telco customer churn dataset. The data contains 7043 records of customers with 21 attributes that describe customer demographics, services used, and customer account information. The objective of the analysis is to predict customer churn and develop effective retention strategies to reduce churn rates.

The dataset has 21 columns, which are described as follows:

- ***CustomerID****: A unique identifier for each customer.*
- ***Gender****: The customer's gender (Male/Female).*
- ***SeniorCitizen****: A binary variable indicating if the customer is a senior citizen or not (1, 0).*
- ***Partner****: A binary variable indicating if the customer has a partner or not (Yes, No).*
- ***Dependents****: A binary variable indicating if the customer has dependents or not (Yes, No).*
- ***Tenure****: The number of months the customer has been with the company.*
- ***PhoneService****: A binary variable indicating if the customer has a phone service or not (Yes, No).*
- ***MultipleLines****: A binary variable indicating if the customer has multiple lines or not (Yes, No, No phone service).*
- ***InternetService****: The type of internet service the customer has (DSL, Fiber optic, No).*
- ***OnlineSecurity****: A binary variable indicating if the customer has online security or not (Yes, No, No internet service).*
- ***OnlineSecurity****: A binary variable indicating if the customer has online backup or not (Yes, No, No internet service).*
- ***DeviceProtection****: A binary variable indicating if the customer has device protection or not (Yes, No, No internet service).*
- ***TechSupport****: A binary variable indicating if the customer has tech support or not (Yes, No, No internet service).*
- ***StreamingTV****: A binary variable indicating if the customer has streaming TV or not (Yes, No, No internet service).*
- ***StreamingMovies****: A binary variable indicating if the customer has streaming movies or not (Yes, No, No internet service).*
- ***Contract****: The type of contract the customer has (Month-to-month, One year, Two years).*
- ***Paperless billing:*** *A binary variable indicating if the customer has paperless billing or not (Yes, No).*
- ***PaymentMethod****: The payment method the customer uses (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic)).*
- ***MonthlyCharges****: The amount charged to the customer monthly.*

- ***Total charges****: The total amount charged to the customer over the entire tenure.*
- ***Churn***: *This variable indicates whether a customer has churned or not. It was the target variable for the project (Yes, No)*

The dataset contained no missing values, and all the attributes are in the correct data type. The next step is to perform exploratory data analysis and feature engineering to prepare the data for modeling. Also, the project aims to develop a predictive model that can identify customers who are at risk of churning and implement retention strategies to reduce churn rates.

Packages Used

The following packages were used for the project

```
# Data handling
import pandas as pd # used for data manipulation and analysis, such as
loading data into data frames and performing various data transformations.
import numpy as np  # used for numerical operations and computations, such as
handling missing values and performing array operations.

# Vizualisation (Matplotlib, Plotly, Seaborn, etc. )
import seaborn as sns #used for advanced data visualization, such as creating
heatmaps and categorical plots.
import matplotlib.pyplot as plt  # used for creating basic plots and charts.
%matplotlib inline # used to create easier and view plots quickly and
efficiently

# Feature Processing (Scikit-learn processing, etc. )
from sklearn.impute import SimpleImputer # used for imputing missing values
in the data.
from sklearn.model_selection import train_test_split  # used for splitting
the data into training and testing sets.
from sklearn.preprocessing import OrdinalEncoder   # used for encoding
categorical features as integer values.
from sklearn.preprocessing import LabelEncoder, OneHotEncoder  # used for
encoding categorical features as integer labels.
from sklearn.preprocessing import StandardScaler # used for standardizing the
data.
from sklearn.preprocessing import MinMaxScaler   # used for scaling the data
to a specific range
from collections import Counter  # used for counting the number of
occurrences of each element in a list.
from imblearn.over_sampling import RandomOverSampler # used for oversampling
the minority class to balance the dataset.
import scipy.stats as stats   # used for performing statistical tests and
calculations.
from scipy.stats import chi2_contingency   #

# Machine Learning (Scikit-learn Estimators, Catboost, LightGBM, etc. )
from sklearn.datasets import make_classification
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier,
RandomForestClassifier
from sklearn.linear_model import LogisticRegression, SGDClassifier # used for
building a logistic regression model
```

```
from sklearn.neighbors import KNeighborsClassifier   # used for building a K-
Nearest Neighbors model.
from sklearn.svm import SVC # used for building a Support Vector Machines
model.
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score, roc_curve, auc, fbeta_score
from sklearn.metrics import confusion_matrix #  used for evaluating the
performance of the machine learning models.

# Hyperparameters Fine-tuning (Scikit-learn hp search, cross-validation, etc.
)
from sklearn.model_selection import KFold, cross_val_score  # used for
performing K-fold cross-validation
from sklearn.model_selection import GridSearchCV # used for performing
hyperparameter tuning through grid search.
from sklearn.ensemble import GradientBoostingRegressor # sed for building a
gradient boosting regression model.

# Other packages
from tabulate import tabulate  # used for creating tables to display the
results of the machine learning models.
import os, pickle # used for saving and loading the trained machine learning
modelsimport warnings # used for filtering warning messages from the output.
warnings.filterwarnings('ignore')
```

Dataframe and Datatypes Understanding

The dataset was loaded into a Pandas DataFrame using the pd.read_csv function. Then, the `DataFrame.head()` method was used to display the first 5 rows of the dataset:

```
customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService
MultipleLines InternetService  ...     DeviceProtection          TechSupport
StreamingTV      StreamingMovies        Contract PaperlessBilling
PaymentMethod MonthlyCharges  TotalCharges Churn
0  7590-VHVEG  Female              0     Yes         No       1           No
No phone service             DSL  ...                   No
No                    No                    No  Month-to-month              Yes
Electronic check          29.85         29.85    No
1  5575-GNVDE    Male              0      No         No      34          Yes
No             DSL  ...                  Yes                   No
No                    No        One year               No              Mailed
check          56.95       1889.50    No
2  3668-QPYBK    Male              0      No         No       2          Yes
No             DSL  ...                   No                   No
No                    No  Month-to-month              Yes              Mailed
check          53.85        108.15   Yes
3  7795-CFOCW    Male              0      No         No      45           No
No phone service             DSL  ...                  Yes
Yes                   No                    No        One year
No  Bank transfer (automatic)          42.30       1840.75    No
4  9237-HQITU  Female              0      No         No       2          Yes
No     Fiber optic  ...                   No                   No
No                    No  Month-to-month              Yes           Electronic
check          70.70        151.65   Yes
```

```
[5 rows x 21 columns]
```

From the output, we can see that the dataset has 21 columns/features and 7043 rows/observations. Each row represents a customer and each column represents a feature of that customer, such as gender, senior citizen status, tenure, phone service, and whether or not the customer churned.

The `DataFrame.info()` method was used to display information about the the data types of each column, the number of non-null values, and memory usage.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   customerID       7043 non-null   object
 1   gender           7043 non-null   object
 2   SeniorCitizen    7043 non-null   int64
 3   Partner          7043 non-null   object
 4   Dependents       7043 non-null   object
 5   tenure           7043 non-null   int64
 6   PhoneService     7043 non-null   object
 7   MultipleLines    7043 non-null   object
 8   InternetService  7043 non-null   object
 9   OnlineSecurity   7043 non-null   object
 10  OnlineBackup     7043 non-null   object
 11  DeviceProtection 7043 non-null   object
 12  TechSupport      7043 non-null   object
 13  StreamingTV      7043 non-null   object
 14  StreamingMovies  7043 non-null   object
 15  Contract         7043 non-null   object
 16  PaperlessBilling 7043 non-null   object
 17  PaymentMethod    7043 non-null   object
 18  MonthlyCharges   7043 non-null   float64
 19  TotalCharges     7043 non-null   object
 20  Churn            7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

This output shows that there are 21 columns in the DataFrame, with column names and their corresponding data types. It also shows us that there are no missing values (since all columns have 7043 non-null values), but the TotalCharges column is in object instead of float64. This suggests that there may be some non-numeric values in this column that need to be cleaned.

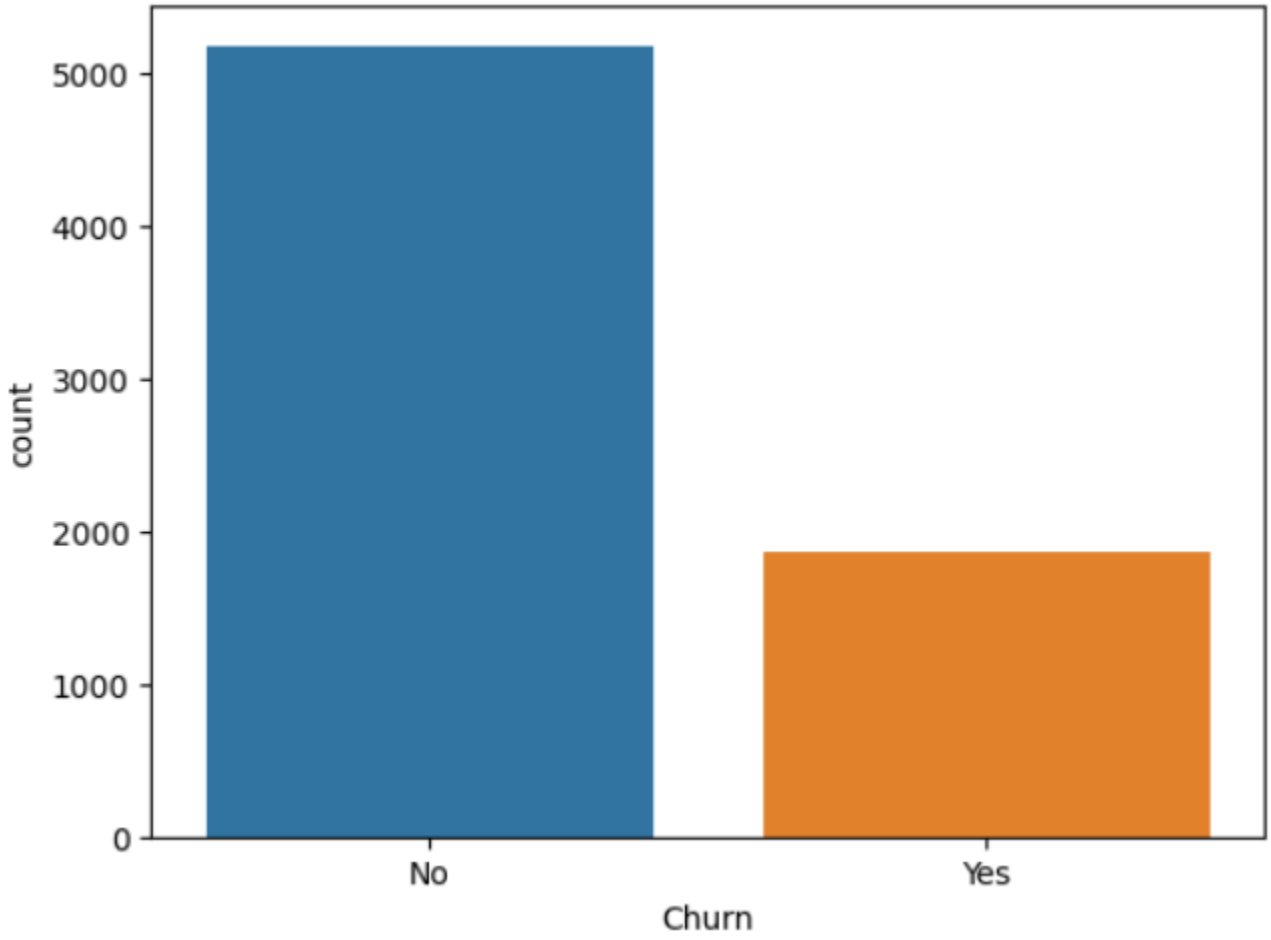
Findings from the Univariate, Bivariate & Multivariate Analysis

Based on the univariate, bivariate, and multivariate analysis performed here are some findings:

Univariate Analysis

Histogram

A simple histogram was created with the 'Churn' variable as a target, which is a binary indicator of whether a customer has left the service or not. The histogram revealed that the distribution of this variable among the customers in the DataFrame. However, the target variable (Churn) was imbalanced, statistically with about 26.5% of customers churning. This was taken into account when building our models.

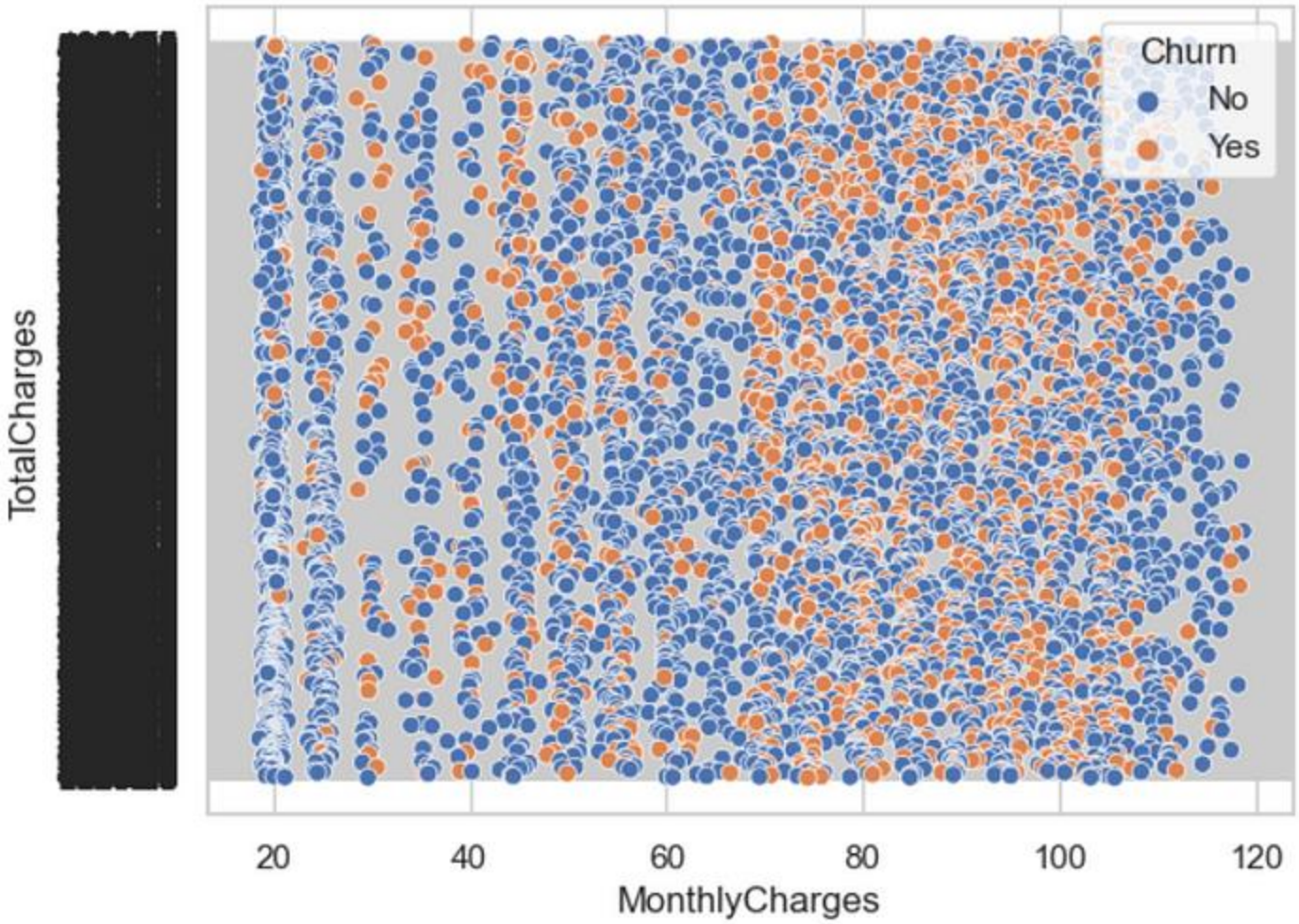


The Histogram indicated that the NO responses were about 5000 whiles the YES was less than 2000

Bivariate & Multivariate Analysis

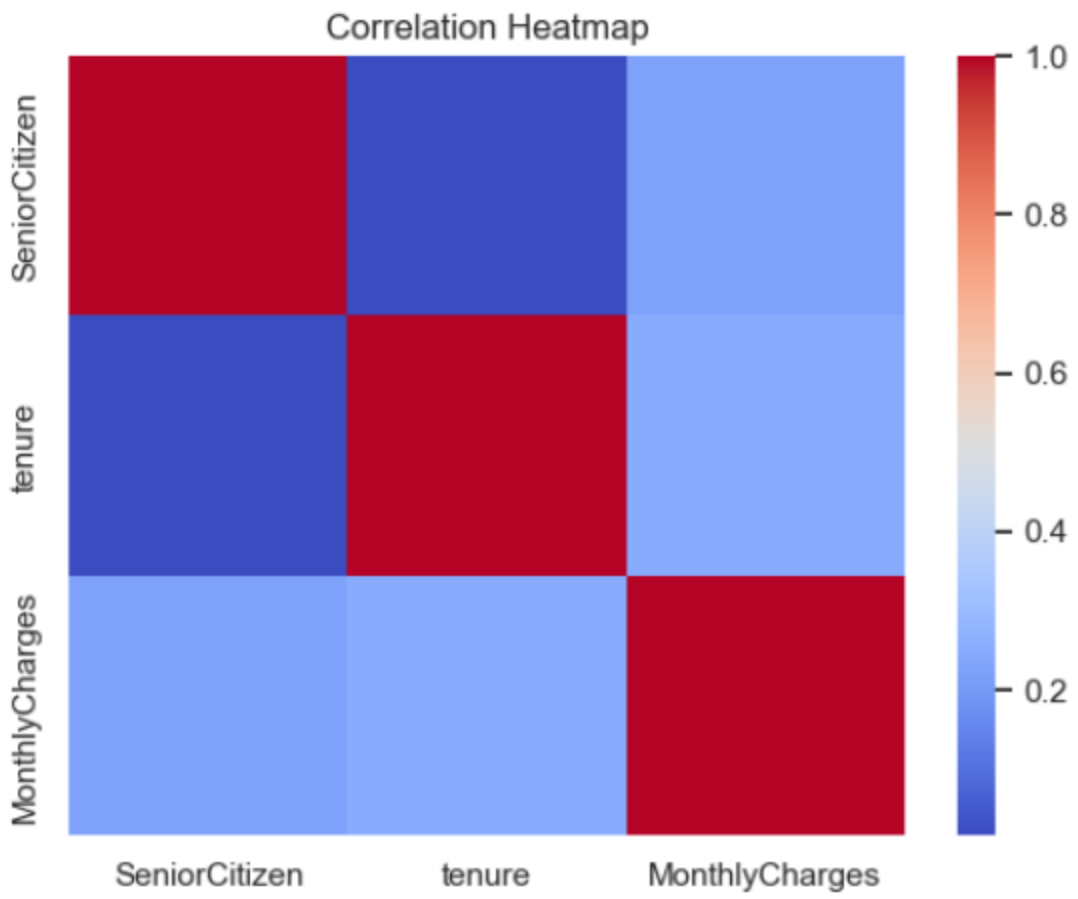
Scatter Plot

A scatter plot was generated with Seaborn to visualize the relationship between two numerical variables 'MonthlyCharges' and 'TotalCharges' with the color encoding for a categorical variable. The MonthlyCharges and TotalCharges variables were skewed to the right, indicating that there are some customers with high charges.



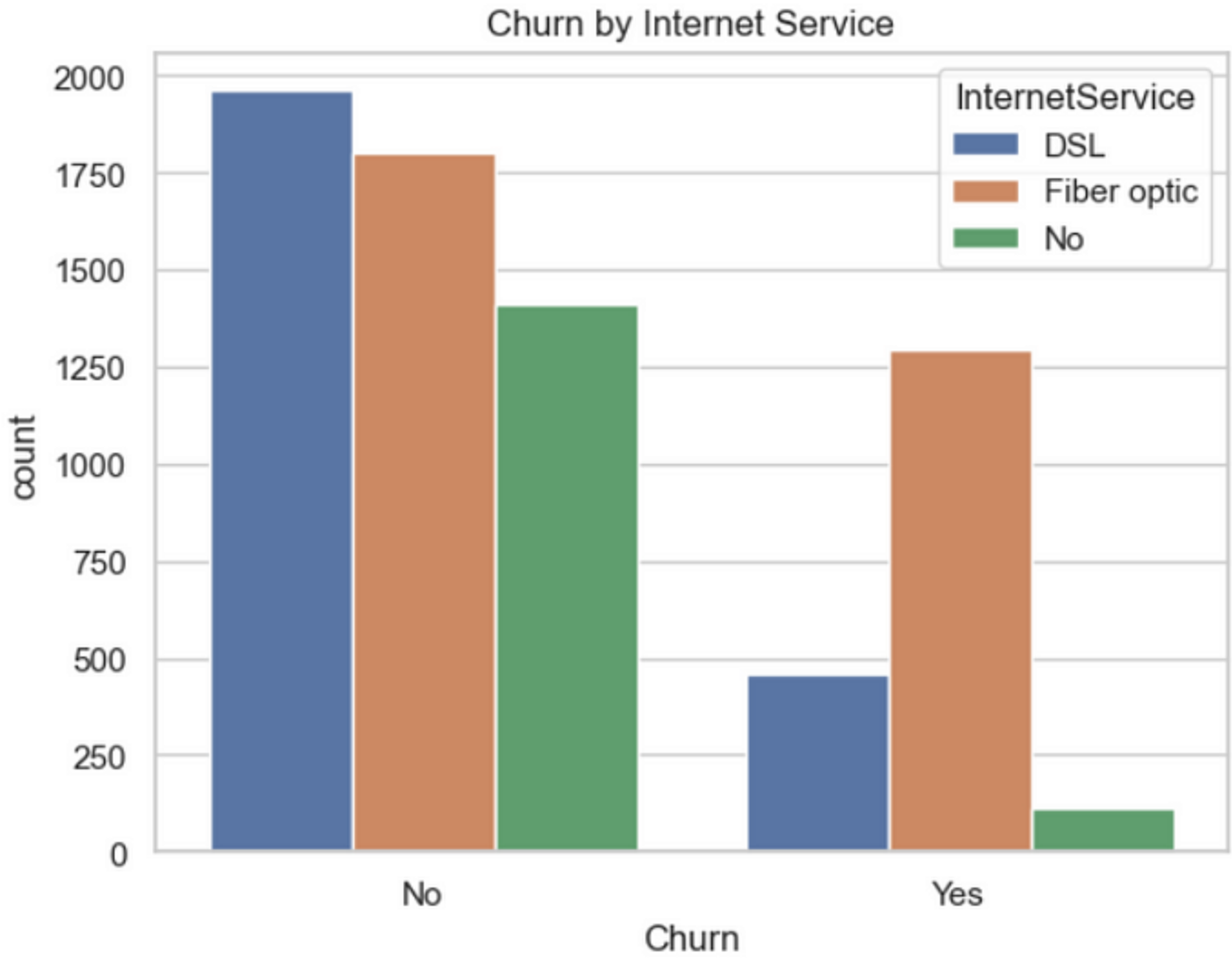
The findings presented good visualization of the relationship between ‘MonthlyCharges’ and ‘TotalCharges’ with the color encoding for ‘Churn’. However, further analysis and context were required to draw significant conclusions from this plot. It was considered to perform additional exploratory analysis, such as correlation analysis, to assess the strength and direction of the relationship between the variables.

Heatmap



Countplot

A count plot was generated to visualize the relationship between customer churn and gender on who churn most by Internet Service. This so important to check for statistical significance of the differences in churn rates between genders, and to control for potentially confounding variables such as age, income, and subscription plan.



Customers who have Fiber optic internet service have a higher median MonthlyCharges compared to those who have DSL internet service.

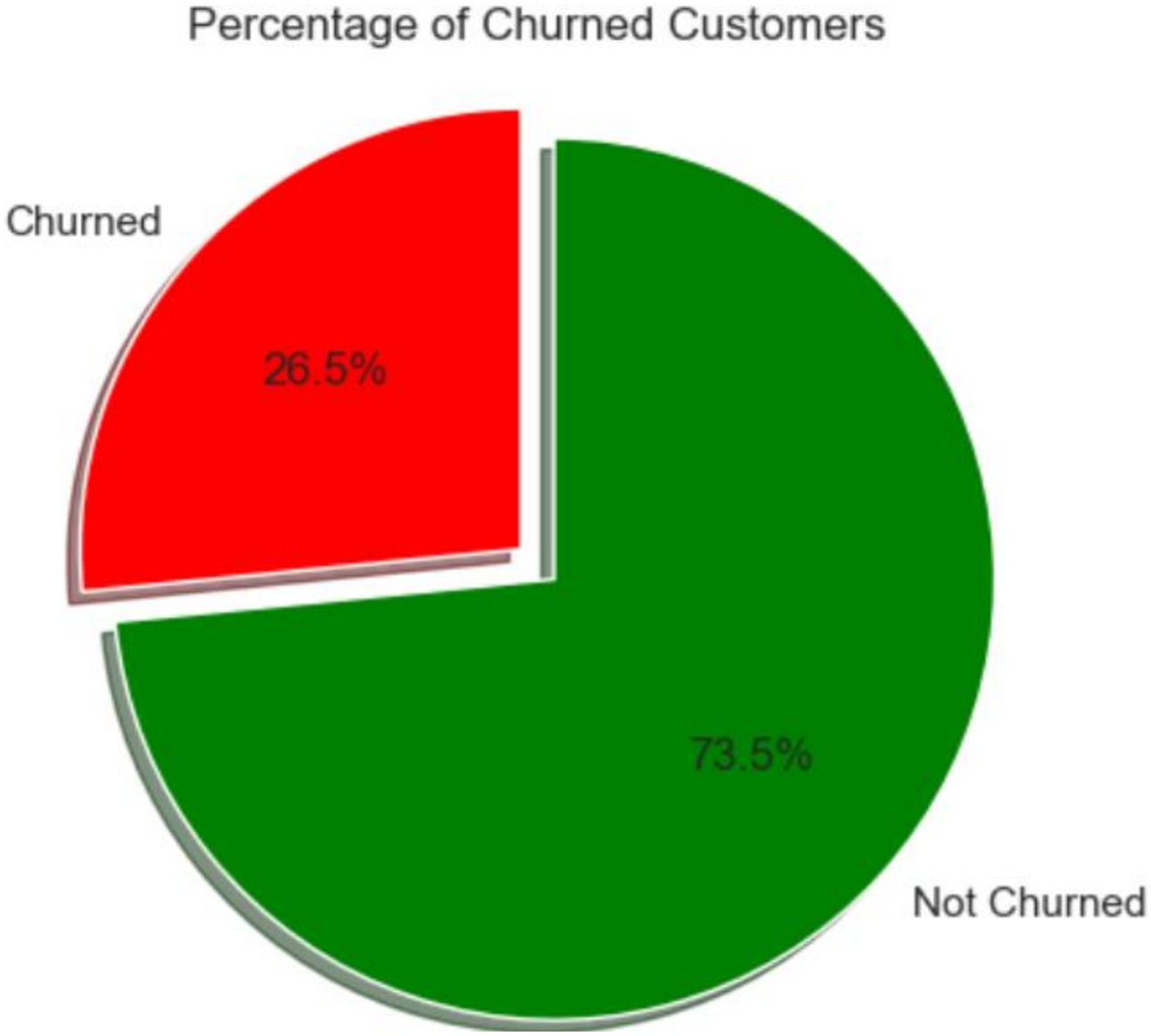
Answering Questions

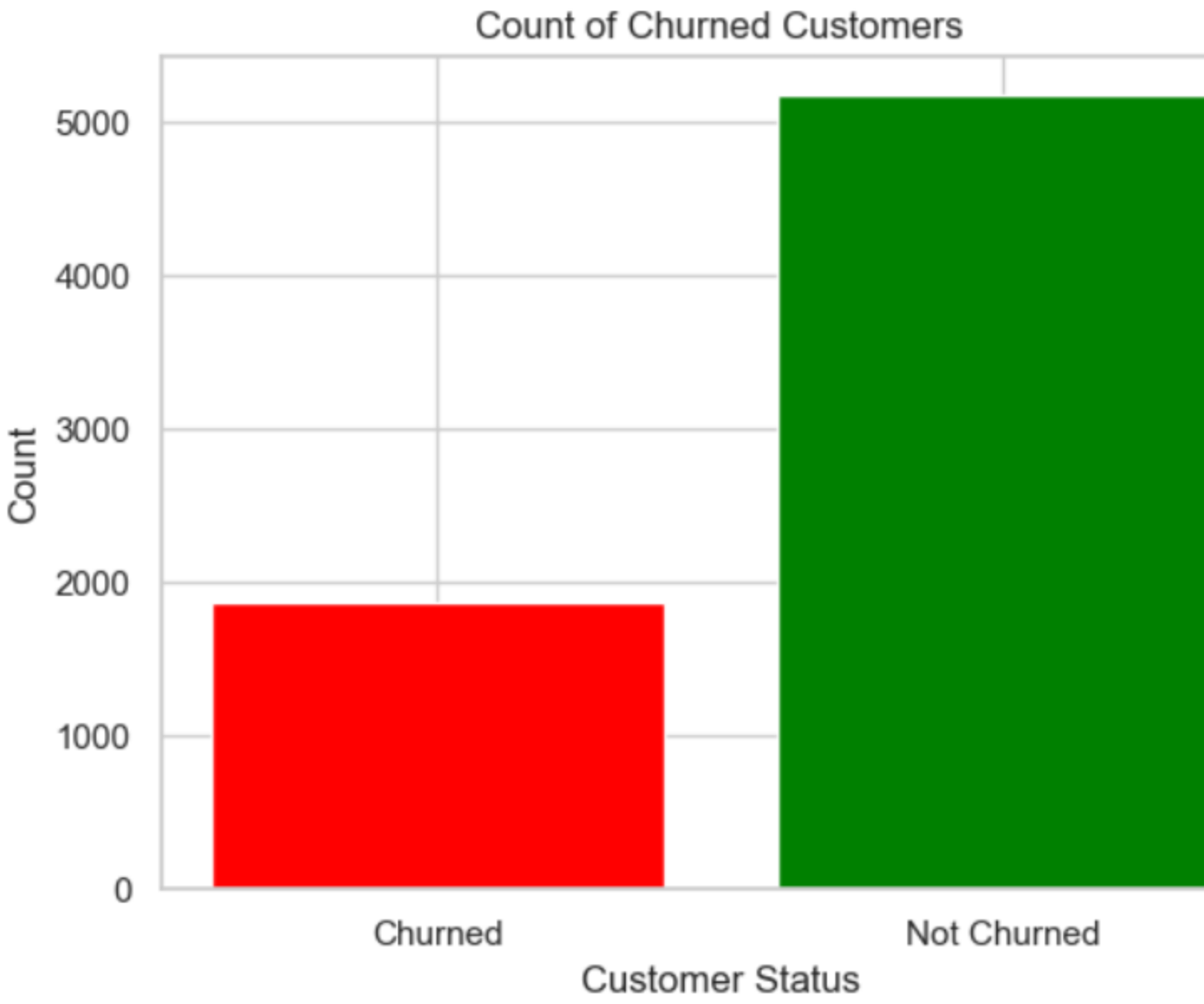
At this part of the article, I will combine the Analysis and Share stages of the data analysis process through the code and visualizations.

1. What percentage of customers have churned?

A pie chart and a bar plot were generated to provide insight to understand the percentage of customers who have churned.

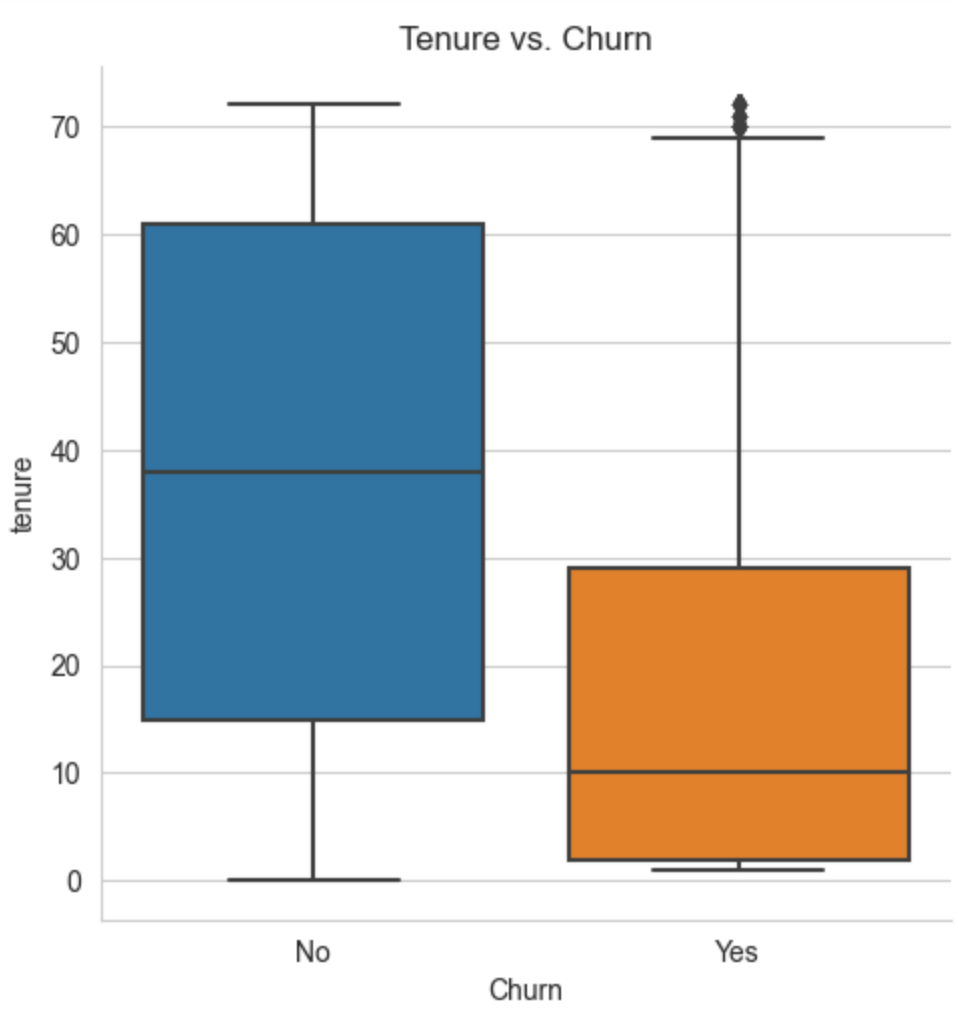
The analysis provided an insight that 26.5% were churned whiles 73.5% were not. This means 1,869 ustomers churned whiles 5,174 customers did not churn.¶


Percentage of Churned Customers
Churned
26.5%
73.5%
Not Churned



2. Is there a correlation between a customer's length of tenure with the company and their likelihood of churning?

A box plot was generated to find the correlation between **tenure** and **churn**.



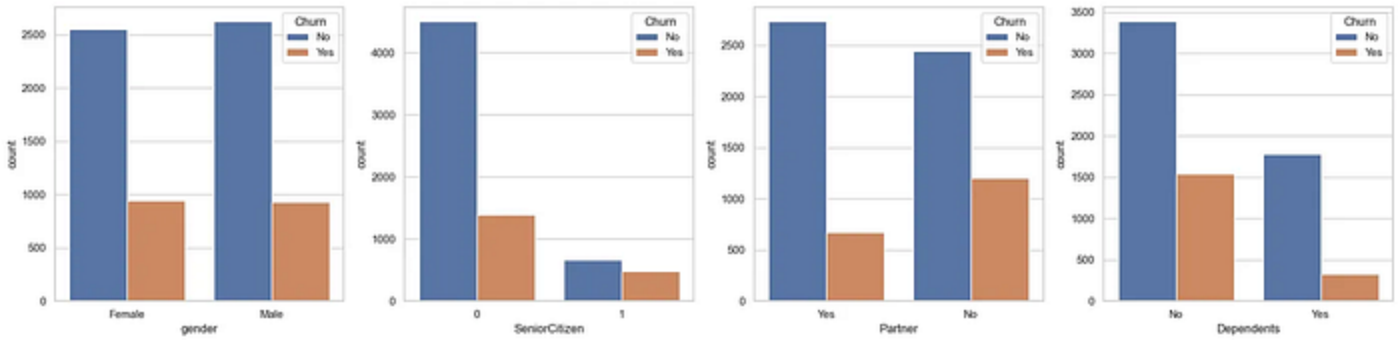
Customers who did not churn have a normal distribution, whiles customers who churned are positively skewed.

Insight from the box plot was analyzed from as the following

1. *The average tenure of customers that did not churn is higher at 38, whiles that of those that churned is lower at 10*
2. *An outlier was identified within churned customers*

3. Are there any specific groups of customers based on demographic that are more likely to churn than others?

To better understand customers by gender, type of citizen, partner, and dependant status, a **count plot** was generated to analyse their churn statuses in ***generated subplots***



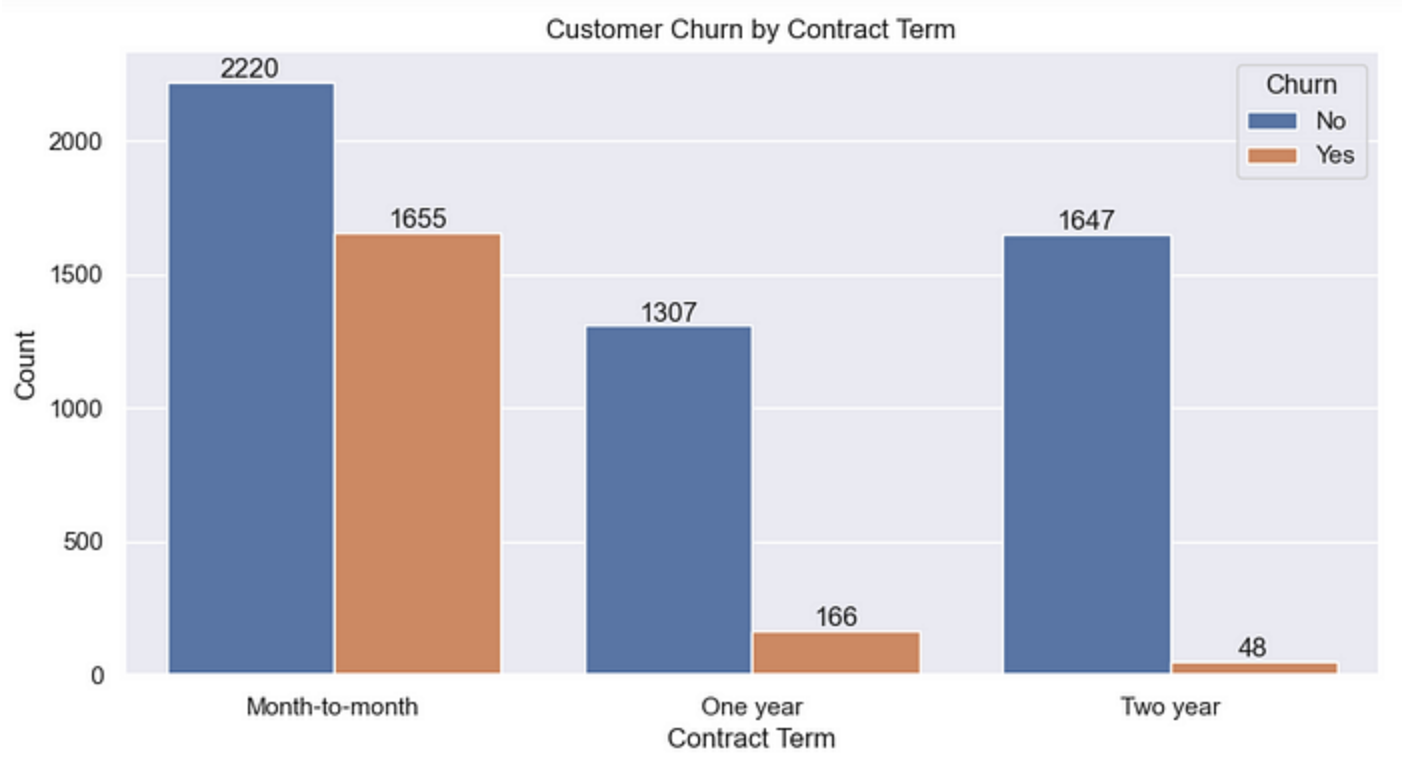
Subplot displaying customers by gender, type of citizen, partner, and dependent status

Insight and findings from the above analysis indicated that

1. *There was marginally high churn for females than males*
2. *Senior citizens are less likely to churn compared to non-senior citizens*
3. *Customers without partners are more likely to churn compared to customers with partners*
4. *There is a high rate of churn for customers without dependants as against customers with dependants*

4. Can customer retention be improved by offering longer contract terms?

To extract relevant information to understand customer retention to be either improved by offering longer contract terms, a count plot was generated.

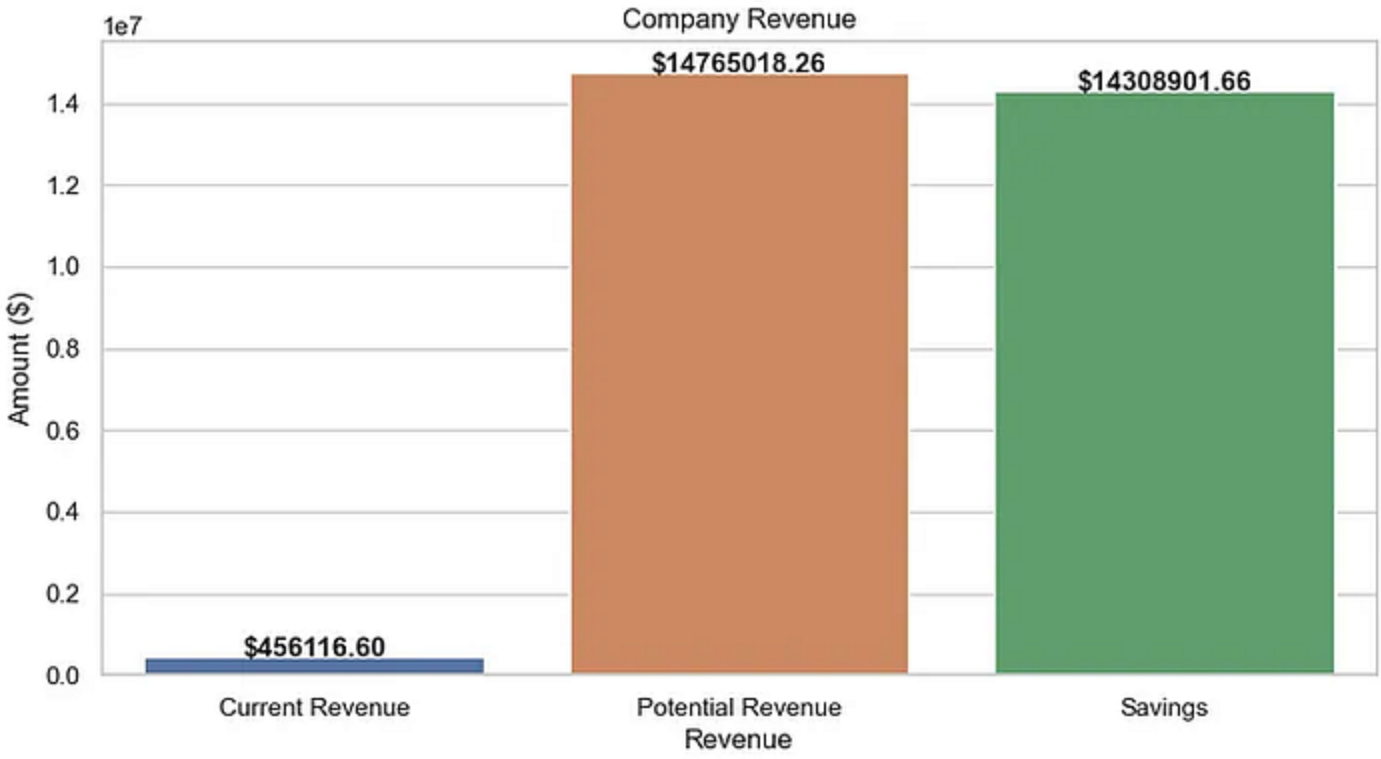


A count plot showing correlation customer churn by contract term

Insight and Findings

1. *There was a positive correlation between higher contract terms and retention rate*
2. *From the analysis, customers who undergo two-year contract terms have a higher retention rate than the one-year and month-to-month contract terms.*
3. *Also, Customers that undergo one-year contract term have a higher retention rate than the month-to-month customers*

5. How much money could the company save by implementing effective retention strategies and reducing customer churn?

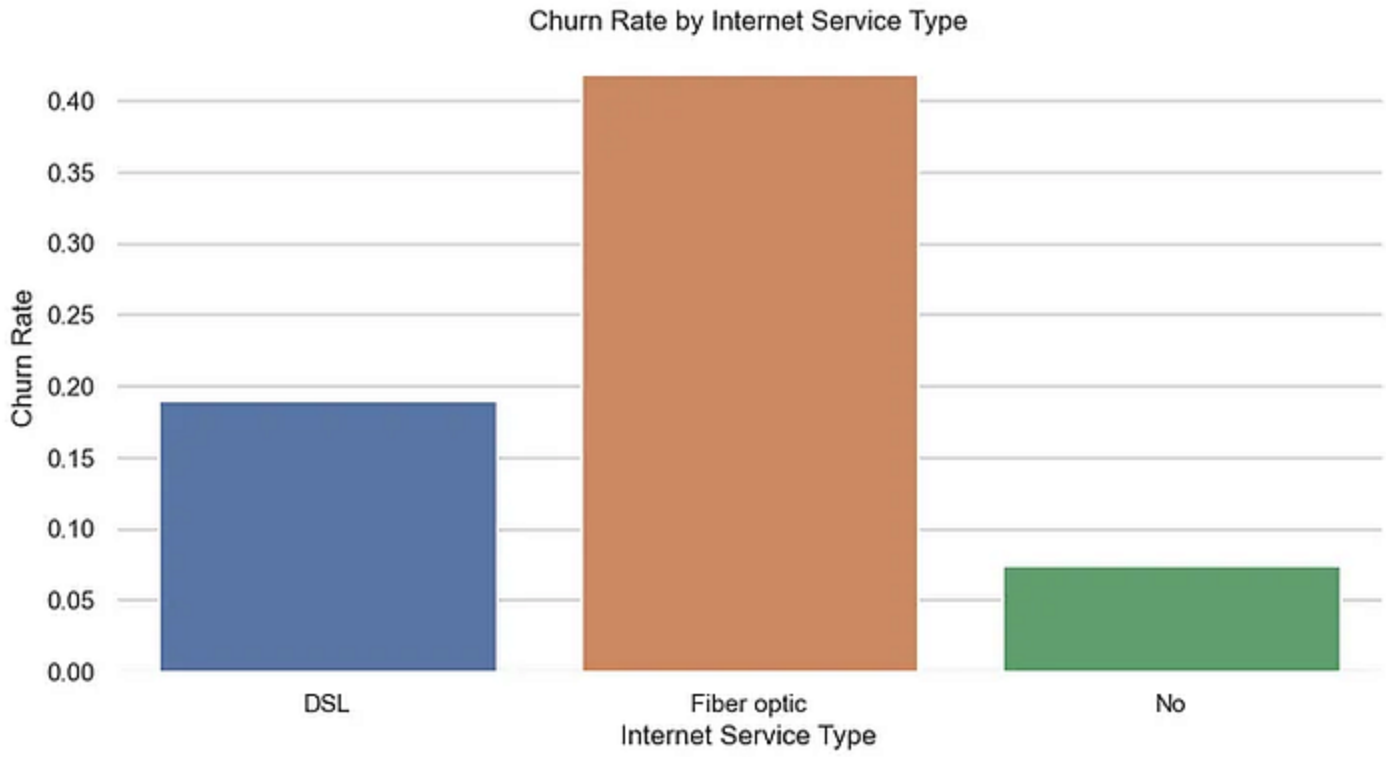


Graph showing the Company's total revenue and saving help to implement effective retention strategy (ies)

Result/Finding

1. *After conducting an analysis, it was discovered that the company has the potential to save more than $14 million in revenue by implementing efficient strategies and decreasing churn rates.*

6. What is the relationship between Internet Services and churn rate?



Fibre Optic internet service users experience higher churn rate than both providers.

Findings

1. *The findings indicate that fiber optic internet service users experience the highest churn rates, exceeding 40%. In contrast, DSL internet service type users exhibit a churn rate of 19%, while customers with no internet service have a churn rate of 7.5%.*

Answering Hypotheses

Hypothesis (1).

H0: There is no significant difference in churn rates between male and female customers.

H1: There is a significant difference in churn rates between male and female customers.

From the analysis, it was revealed that there was no significant difference in churn rate between male and female customers as produced.

```
# Calculate the total number of male and female customers
total_male = df['gender'].value_counts()['Male']
total_female = df['gender'].value_counts()['Female']
```

```
# Calculate the number of male and female customers who churned
churned_male = df[(df['gender'] == 'Male') & (df['Churn'] == 'Yes')].shape[0]
churned_female = df[(df['gender'] == 'Female') & (df['Churn'] ==
'Yes')].shape[0]

# Calculate the churn rates for male and female customers
churn_rate_male = (churned_male / total_male).round(2)
churn_rate_female = (churned_female / total_female).round(2)

print("Churn rate for male customers:", churn_rate_male)
print("Churn rate for female customers:", churn_rate_female)

# Results/Finding(s)
Churn rate for male customers: 0.26
Churn rate for female customers: 0.27
t-satistic: nan
p-value: nan
Fail to reject null hypothesis. There is no significant difference in churn
rates between male and female customers.
```

From the above findings, There is no significant difference in churn rates between male and female customers which makes my **Null hypothesis true**. Also, the p-value and t-statistic were *nan.*

Hypothesis (2)

H0: There is no significant relationship between the customer's internet service provider and their likelihood to churn.

H1: There is a significant relationship between the customer's internet service provider and their likelihood to churn.

churn rates were calculated by internet service provider and the findings below were presented

```
InternetService
DSL            0.190
Fiber optic    0.419
No             0.074
Name: Churn, dtype: float64
```

A contingency table of internet service provider and churn was created to perform chi-squared test and printed the results

```
Chi-squared test results:
Chi2 = 732.31, p-value = 0.0000, degrees of freedom = 2
Reject null hypothesis: There is a significant relationship between the
customer's internet service provider and their likelihood to churn.
```

From the above finding presented, the analysis failed to reject the null hypothesis that indicated that ***there is a significant relationship between the customer's internet service provider and their likelihood to churn***

Hypothesis (3).

H0: There is no significant difference in churn rates between customers on different types of payment methods.

H1: There is a significant difference in churn rates between customers on different types of payment methods.

To prove the authenticity of my analysis, my 3rd null hypothesis was tested. The finding below was generated

```
chi-square statistic = 648.14
p-value = 0.0000
degrees of freedom = 3
Expected counts:
[[1134.26891949  409.73108051]
 [1118.10705665  403.89294335]
 [1737.40025557  627.59974443]
 [1184.22376828  427.77623172]]
Reject null hypothesis: there is a significant difference in churn rates
between customers on different types of payment methods.
```

The null hypothesis that there is no significant difference in churn rates between customers on different types of payment methods was rejected. This means the alternate hypothesis that **There is a significant difference in churn rates between customers on different types of payment methods guided the analysis.**

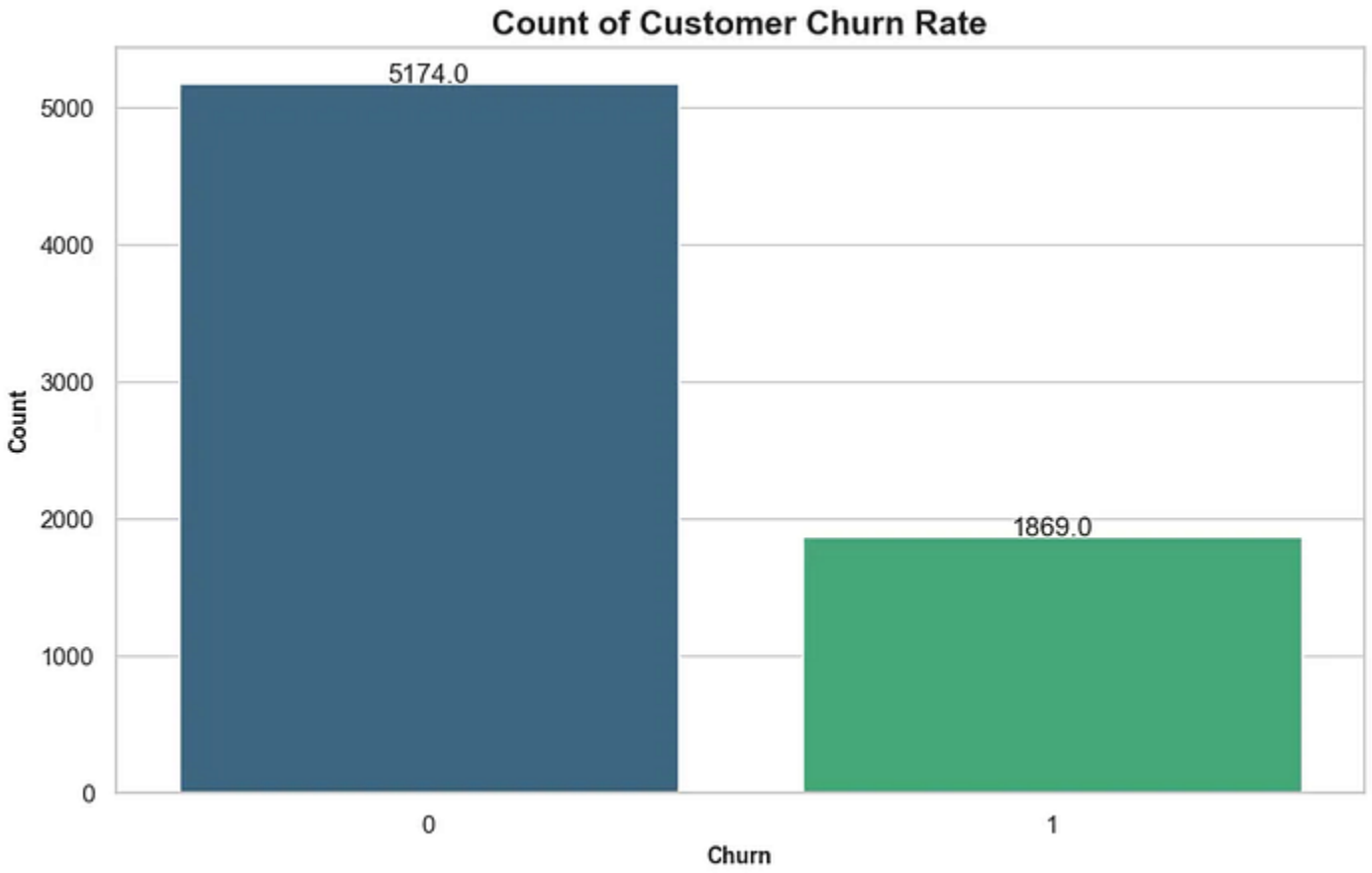
Summary of Findings for the three(3) Hypothesis

1. *Findings from the analysis revealed that the first hypothesis was true which indicates* ***that there is no significant difference in churn rates between male and female customers.***
2. *Also, the second null hypothesis was true. This means* ***there is a significant relationship between the customer's internet service provider and their likelihood to churn***
3. The third hypothesis was rejected which made a way for the alternate hypothesis to influence the analysis. This indicated that ***there is a significant difference in churn rates between customers on different types of payment methods guided the analysis.***

Data Imbalance and Balancing Data

When the number of samples in different classes of a dataset is significantly different, it results in data imbalance. This can pose a challenge for machine learning models as they tend to perform better when the classes are equally represented in the data.

Here is a chart displaying the degree of data imbalance for the target variable



Imbalance Dataset

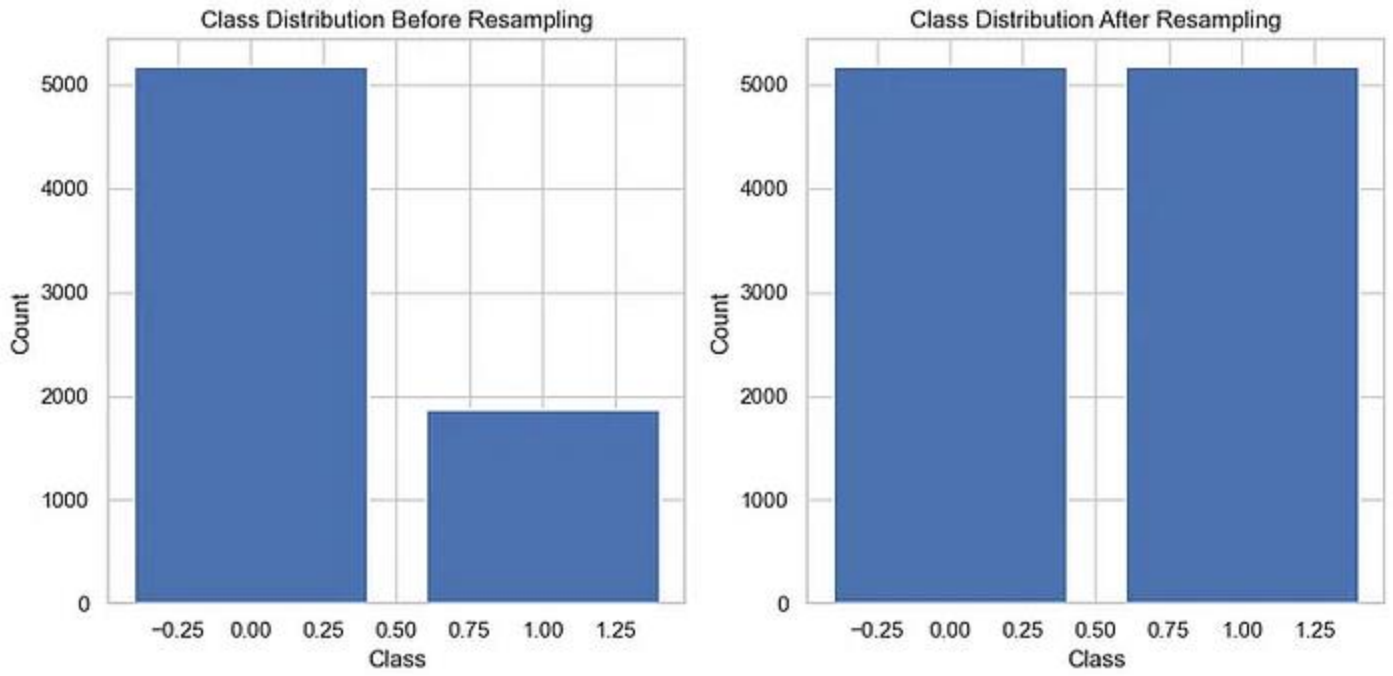
Froom the imbalance dataset,

1. The ratio or proportion of customer who churn is 0.265
2. The ratio of customers who churn is 0.735

The data is imbalanced, as fewer people tend to churn (27%) compared to the number that does not churn (73%).

When the data was imbalanced, it was important to balance the data when doing this classification modeling. For this purpose, the oversampling method was used to balance the dataset.

Below is a chart depicting a balanced dataset after oversampling.



Balance dataset for the distribution after resampling

Based on the updated chart, it appears that the data is now balanced

In a nutshell, balancing data imbalances is a crucial step in creating robust machine learning models that can effectively generalize to new data.

Feature Processing and Engineering

Feature processing is the initial step in the data preprocessing pipeline, where raw data is transformed into a numerical format that can be used as input to machine learning models. This conversion process involves various techniques, including normalization, scaling, and encoding, depending on the type and nature of the data.

However, feature engineering, on the other hand, involves the creation of new features or the selection of relevant features from the dataset to improve the model's performance.

Feature Encoding

In order to utilize categorical data as input to machine learning models, it must be converted into a numerical format, a process known as feature encoding. Categorical data refers to information that describes qualitative attributes, such as gender, color, or country of origin, which are typically represented as text labels. Machine learning models, however, require numerical data for processing and analysis. Therefore, feature encoding is a crucial step in the data preprocessing pipeline, allowing categorical data to be transformed into a usable format for machine learning algorithms.

The following code was utilized for data encoding:

Features Encoding

In [57]:
```
# encode categorical variable using ordinal encoding
encoder = OrdinalEncoder()
cat_cols = X.select_dtypes(include='object').columns
X_encoded = pd.DataFrame(encoder.fit_transform(X[cat_cols]))
X_encoded.columns = cat_cols

# combine encoded categorical variables with numerical variables
num_cols = X.select_dtypes(include='number').columns
X_processed = pd.concat([X_encoded, X[num_cols]], axis=1)
```

In [58]:
```
X_processed.head()
```

Out[58]:

	gender	SeniorCitizen	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	Strea
0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	2.0	0.0	0.0	
1	1.0	0.0	0.0	0.0	1.0	0.0	0.0	2.0	0.0	2.0	0.0	
2	1.0	0.0	0.0	0.0	1.0	0.0	0.0	2.0	2.0	0.0	0.0	
3	1.0	0.0	0.0	0.0	0.0	1.0	0.0	2.0	0.0	2.0	2.0	
4	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	

The Encoded dataset (Categorical and Numerical)

Feature Scaling

To ensure equal contribution of all numerical features in the training process and to prevent features with higher values from overshadowing those with lower values, feature scaling is employed. Feature scaling is a data preprocessing technique that involves transforming numerical features into a standardized range. This ensures that all features are weighted equally in the model and can lead to improved performance and accuracy of machine learning algorithms.

The following code was used for data scaling, and the resulting scaled data is displayed in the results table.

Features Scaling

In [60]:
```
# initialize MinMaxScaler object with specified parameters
scaler = MinMaxScaler(feature_range=(0, 1), copy=True)

# select numerical columns to scale
num_cols = X_processed.select_dtypes(include='number').columns

# scale numerical columns using MinMaxScaler
X_processed[num_cols] = scaler.fit_transform(X_processed[num_cols])
```

In [61]:
```
X_processed.head()
```

Out[61]:

	gender	SeniorCitizen	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	Stre
0	0.0	0.0	1.0	0.0	0.0	0.5	0.0	0.0	1.0	0.0	0.0	
1	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	
2	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0	
3	1.0	0.0	0.0	0.0	0.0	0.5	0.0	1.0	0.0	1.0	1.0	
4	0.0	0.0	0.0	0.0	1.0	0.0	0.5	0.0	0.0	0.0	0.0	

Scaled Dataset

Machine Learning Modelling

Classification modeling is a type of machine learning technique that involves predicting the categorical class labels of a target variable using one or more input features. The primary objective of classification modeling is to develop a model capable of accurately predicting the class labels of new and unseen data. By analyzing the patterns and relationships between the input features and target variable, the model can learn to make predictions that classify the data into predefined categories or classes.

To build, train, evaluate, and test models on the dataset for this project, the following machine learning algorithms were utilized.

1. *Random Forest*
2. *Decision Tree*
3. *Gradient Boosting*
4. *Support Vector Machines*
5. *Logistic Regression*
6. *Stochastic Gradient Descent*
7. *K-Nearest Neighbours*

NOTE:

The main focus of this article in relation to the analysis in the jupyter notebook will be on the best-performing model, which was selected based on the evaluation results.

1. Random Forest

The selected model is an extension of decision trees known as random forests. This algorithm involves creating multiple decision trees and combining their predictions to improve the model's accuracy. Random forests are particularly useful in handling high-dimensional datasets and are less likely to overfit than decision trees.

Create Model

```
# Create a Random Forest model
rf_model = RandomForestClassifier()
```

Train Model

```
# Train the model on the training data
rf_model.fit(X_train, y_train)
# Use the trained model to predict on the test data
rf_pred = rf_model.predict(X_test)
```

Model Evaluation

To evaluate the performance of the selected random forest model, various metrics were employed. These metrics include accuracy, precision, recall, F1 score, and F2 score. These evaluation measures are commonly used in machine learning to assess the model's ability to correctly classify data.

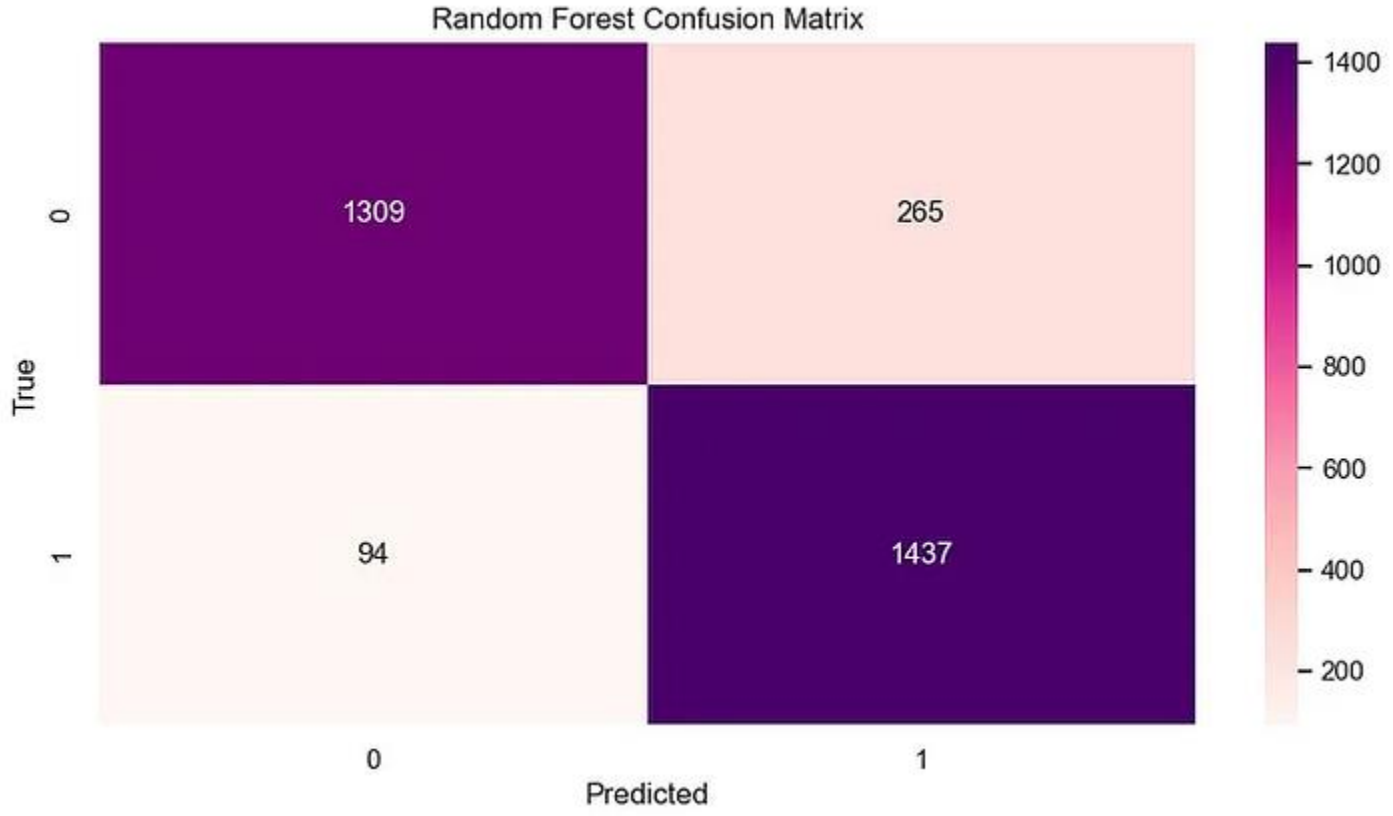
1. *Accuracy represents the ratio of correctly classified samples to the total number of samples in the dataset*
2. *The F1 score is the harmonic mean of precision and recall and is useful when the dataset is imbalanced*
3. *The F2 score is a similar metric that emphasizes recall more than precision and is used when recall is of greater importance than precision*
4. *Precision measures the proportion of correctly predicted positive samples to the total number of samples predicted as positive*
5. *Recall, on the other hand, measures the proportion of correctly predicted positive samples to the total number of actual positive samples.*

```
# Calculate accuracy, precision, recall, f1 score, and f2 score
rf_acc = round(accuracy_score(y_test, rf_pred), 3)
rf_prec = round(precision_score(y_test, rf_pred), 3)
rf_rec = round(recall_score(y_test, rf_pred), 3)
rf_f1 = round(f1_score(y_test, rf_pred), 3)
rf_f2 = round(fbeta_score(y_test, rf_pred, beta=2), 3)

# Calculate the false positive rate, true positive rate, and AUC for the ROC
curve
rf_fpr, rf_tpr, _ = roc_curve(y_test, rf_pred)
rf_auc = round(auc(rf_fpr, rf_tpr), 3)

# compute the confusion matrix using true label values and predicted label
values
rf_cm = confusion_matrix(y_test, rf_pred)
```

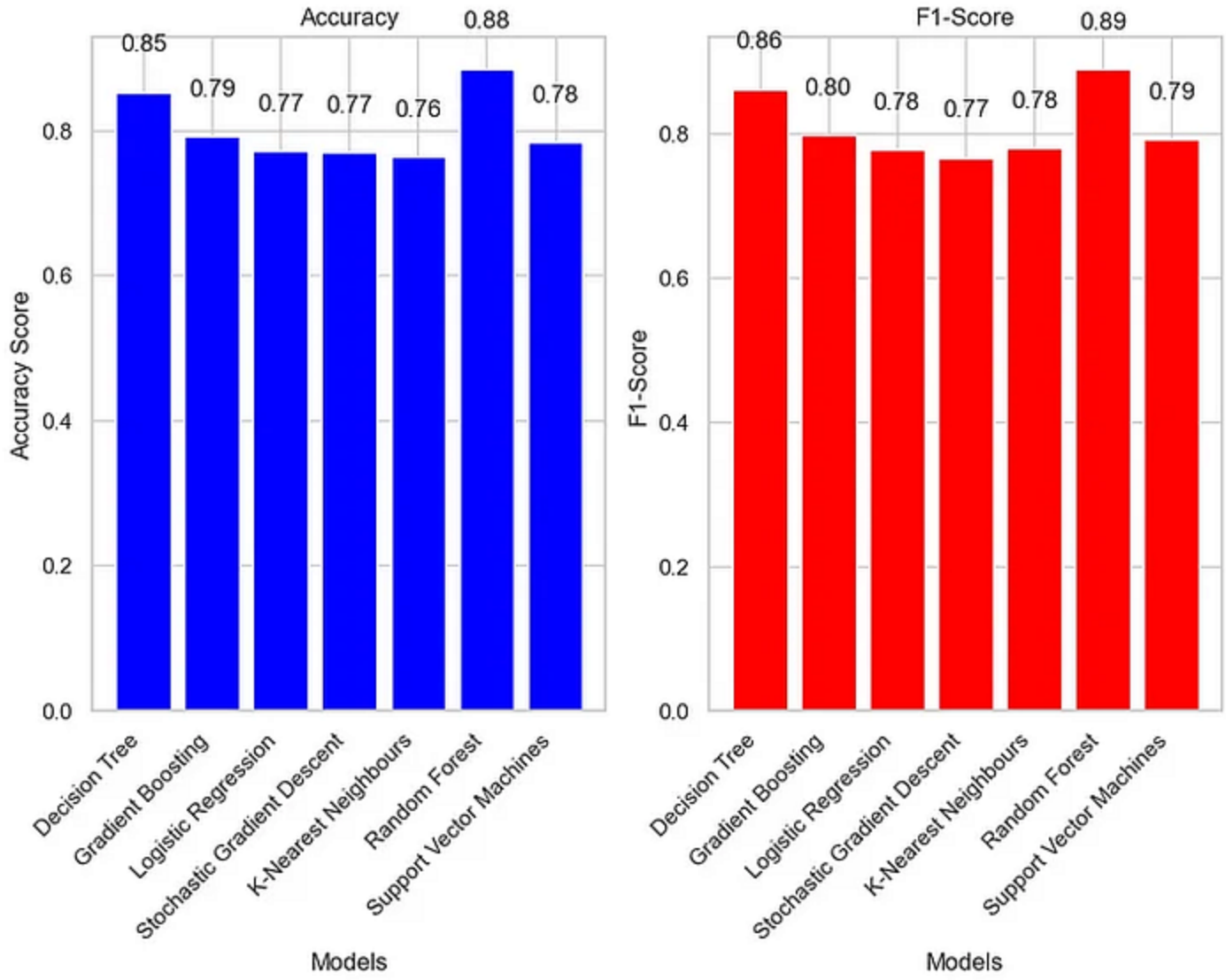
```
# plot the confusion matrix using seaborn heatmap with annotations and color
map
sns.heatmap(rf_cm, annot=True, fmt=".0f", cmap='RdPu')
plt.title("Random Forest Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("True")
plt.show()
```



Comparison of Findings/Results from all models

	Model	Accuracy	Precision	Recall	F1 Score	F2 Score	AUC
5	Random Forest	0.884	0.844	0.939	0.889	0.918	0.885
0	Decision Tree	0.852	0.805	0.926	0.861	0.899	0.853
1	Gradient Boosting	0.790	0.760	0.841	0.798	0.899	0.791
6	Support Vector Machines	0.782	0.752	0.833	0.791	0.816	0.783
2	Logistic Regression	0.770	0.746	0.810	0.777	0.796	0.771
3	Stochastic Gradient Descent	0.768	0.763	0.770	0.766	0.769	0.768
4	K-Nearest Neighbours	0.762	0.718	0.851	0.779	0.821	0.763

Findings from the above table present the results of all models used including their F1 and F2 scores



Results indicating statistical accuracy for the models

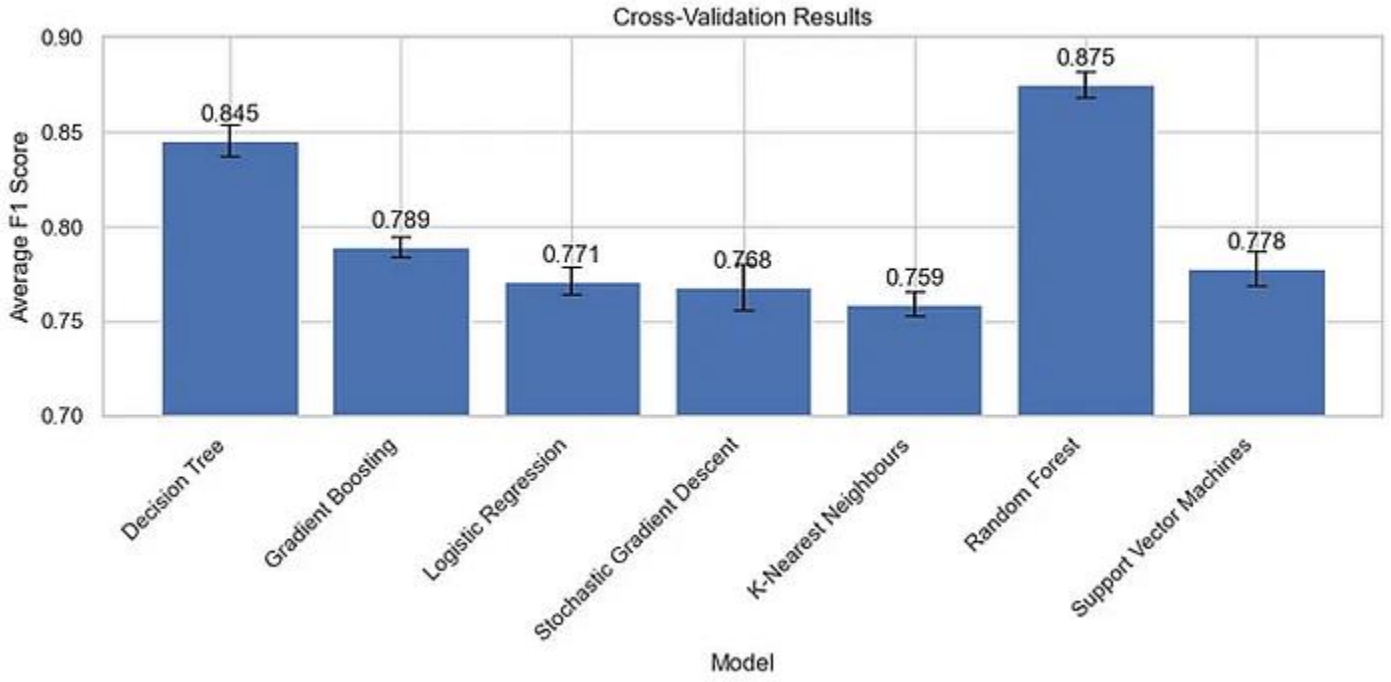
Findings

1. *It is evident from Accuracy and the F1 score that the **Random Forest** had the highest Score. So the **Random Forest was selected to** tune the hyperparameters and retrained it with the best parameters for the best results.*

K-Fold Cross Validation

```
# Set the number of folds for cross-validation
rf_k = 5
# Initialize a k-fold cross-validation object
kf = KFold(n_splits=rf_k, shuffle=True, random_state=42)
# Perform k-fold cross-validation
score = cross_val_score(rf_model, X_train, y_train, cv=kf, scoring='f1')
```

```
# Calculate the mean and standard deviation of the cross-validation scores
rf_cv_score_mean = np.mean(score)
rf_cv_score_std = np.std(score)
# Print the results
print('Cross-validation f1 scores: {}'.format(score))
print('Average f1 score for all folds: {:.3f}'.format(rf_cv_score_mean))
print('Standard deviation of f1 scores: {:.3f}'.format(rf_cv_score_std))
Cross-validation f1 scores: [0.88030888 0.87128713 0.88617363 0.87193099
0.86611147]
Average f1 score for all folds: 0.875
Standard deviation of f1 scores: 0.007
```



Findings from the K-fold validated analysis

Findings for the K-Fold

1. *After the K-fold cross-validation scores, the average cross-validation F1 scores revealed that the **Random Forest Model** as the best model as indicated earlier.*
2. *Also, cross Validation was done to assess the models by training several models on various subsets of the input data. It is also used as a technique to identify overfitting in the models.*

Hyperparameters tuning

Hyperparameters are user-defined settings that specify how a machine learning algorithm should behave during training. These settings include variables such as the learning rate, regularization strength, and number of hidden layers, among others. The process of hyperparameter tuning involves selecting the best values for these settings to optimize the performance of the model.

However, hyperparameter tuning was a critical step in machine learning model development for the prediction that significantly improved the model's performance.

The following section outlines the hyperparameter tuning process that was performed on the Random Forest model.

```
# Define the model
rfc = RandomForestClassifier(random_state=42)
# Define the parameter grid to search over
param_grid = {
    'n_estimators': [100, 500, 1000],
    'max_depth': [3, 5, 7],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
}
# Define the search strategy and perform the search
rfc_p = GridSearchCV(estimator=rfc, param_grid=param_grid, cv=5, n_jobs=-1)
rfc_p.fit(X_train, y_train)
# Print the best hyperparameters and their corresponding score
best_params = rfc_p.best_params_
best_score = round(rfc_p.best_score_, 3)
print(f"Best parameters: {best_params}")
print(f"Best score: {best_score}")
```

Finding/Result for Hyperameter tuning

Following the successful hyperparameter tuning of the Random Forest model,the analysis revealed several insights into the performance and behavior of the model

1. *Best parameters: **{'max_depth': 7, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 500}***
2. *Best score: **0.791***

Conclusion

In conclusion, the telecommunications industry is highly competitive and customer churn is a major challenge. Accurately predicting customer churn can assist businesses in taking proactive measures to retain customers and improve their overall customer experience. In the case of the modern era.

This project provides a detailed analysis of customer churn prediction and retention strategies for a telecommunications company. Among the models analyzed, it can be inferred that the Random Forest Model accurately predicts the churn outcome of customers. Specifically, the model gives an average cross-validation of approximately 87% from the K-fold cross-validation done.

Key Insights

- Feature engineering and selection are critical steps in machine learning projects and can greatly affect model performance.

- Choosing the right machine-learning algorithm and tuning hyperparameters can significantly improve model accuracy.
- Preprocessing steps like scaling and encoding categorical variables are essential for many machine learning models.
- Handling class imbalance is an important consideration for classification tasks, as it can affect model performance and bias the results.
- Interpreting model results and evaluating performance metrics are key steps in assessing the effectiveness of a model and its suitability for real-world applications.
- There can be challenges in working with real-world datasets, such as missing or incomplete data, outliers, and other issues that require careful consideration and handling.
- Machine learning can have significant real-world applications, such as predicting customer behavior or fraud detection, and can provide valuable insights for decision-making.

Challenges

- **Dealing with imbalanced data**: The dataset had more samples for one class than the other, which can make it difficult for the model to learn the minority class. Various techniques were used to handle this issue, such as upsampling, downsampling, and using class weights.
- **Feature engineering**: Feature engineering is an important aspect of any machine learning project, and it can be challenging to determine which features to use and how to transform them to improve model performance.
- **Hyperparameter tuning**: Finding the optimal set of hyperparameters for a machine learning model can be a time-consuming and iterative process. Grid search and cross-validation were used in this project to find the best hyperparameters.
- **Model Evaluation**: The analysis used a single evaluation metric (accuracy) to evaluate the performance of the model. It is important to evaluate the model using multiple metrics (such as precision, recall, and F1-score) to get a more comprehensive view of its performance. It was challenging to establish the fact the evaluation metric is the only best method for our analysis prediction.
- **Retention Strategies**: Because the analysis or insight was drawn for a single dataset, it was challenging to provide detailed recommendations for retention strategies to reduce customer churn. However, it would be beneficial to explore and recommend specific retention strategies that the teleco company to implement to reduce churn.

Way Forward

- **Collect more data**: The dataset used in this project is relatively small, and collecting more data could improve the model's performance and make it more robust.
- **Hyperparameter tuning**: There is still room for improvement in the model's performance by further optimizing the hyperparameters.
- **Deploy the model**: After the model is finalized, it can be deployed in a production environment, for example, as a web application or integrated into a larger software system.

- **Monitor and update the model**: Once the model is deployed, it is important to monitor its performance and update it regularly with new data to ensure it remains accurate and effective.

GITHUB LINK

https://github.com/kagajugrace/LP2_Classification