TELECOM CUSTOMER CHURN PREDICTION

Project Report

Group 6

Student 1: Vishal Arasu

Student 2: Nivedita

857-313-5614 (Student 1)

857-384-9798 (Student 2)

arasu.v@northeastern.edu

nivedita.n@northeastern.edu

Percentage of Effort Contributed by Student 1: 50%

Percentage of Effort Contributed by Student 2: 50%

Signature of Student 1:

Signature of Student 2: Nivedita

TABLE OF CONTENT **PAGES NO.**

OVERVIEW:

This study is based on the concept of customer churn. We have taken the telecom industry into consideration. Customer churn is when customers or subscribers to a service discontinue doing business with a firm or a service.

Customer retention is difficult as most firms constitute many customers and can't afford to allocate time attending to each customer's needs. It would lead to a very cost ineffective strategy that would lead to poor market performance. However, if a company decides to forecast potential customers who discontinue ahead of time, it will limit the amount of money wasted and increase customer retention in a partially cost-effective manner. As a resulting considering the churn rate, businesses are not only maintaining their market performance and position but also facilitating growth and expansion of the firm. More the number of customers, lower the cost of initiation and larger the profit. In this study, we take the telecom industry's churn details into consideration and predict the clients who could potentially discontinue with the company. We have used libraries like sklearn, Matplotlib, pandas, seaborn and NumPy.

PROBLEM SETTING:

The main objective of this problem is to build a predictive model that can accurately identify the customers who are likely to churn or leave the telecom company. This will help the company to take proactive measures to retain these customers and reduce the rate of churn.

PROBLEM DEFINITION:

Telecom customer churn prediction is a problem in the telecommunications industry where the goal is to identify which customers are most likely to cancel their service or switch to a competitor. This is crucial problem because customer churn can lead to significant loss in revenue for telecom companies. The objective of telecom customer churn prediction is to build a model that can accurately predict which customers are at risk of churning so that the company can take proactive measures to retain them. Some of our tasks are:

- Calculating the percentage of customers who churn and those who stick around for active services.
- Examining the data considering the many factors that contribute to client churn.
- Selecting the best machine learning model to accurately classify Churn and Non-Churn Customers.

DATA SOURCE:

We have used Customer Churn Prediction dataset from Kaggle. It is a dataset of a telecommunication company and their customer details.

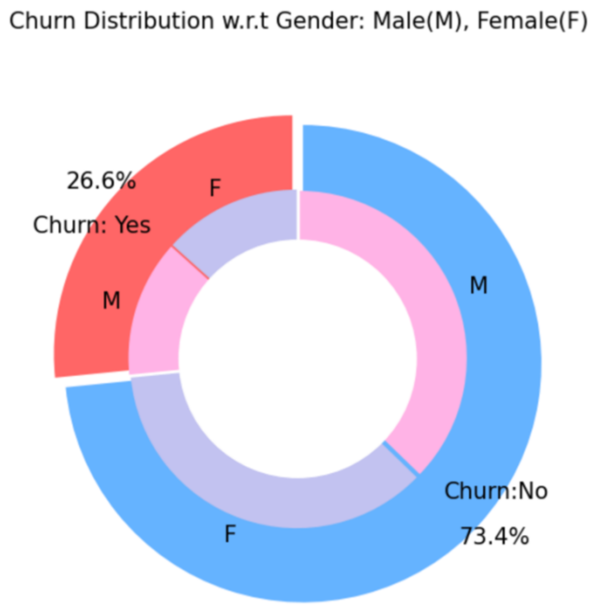
https://www.kaggle.com/code/bhartiprasad17/customer-churn-prediction/input

DATA DESCRIPTION: - The dataset includes information about:

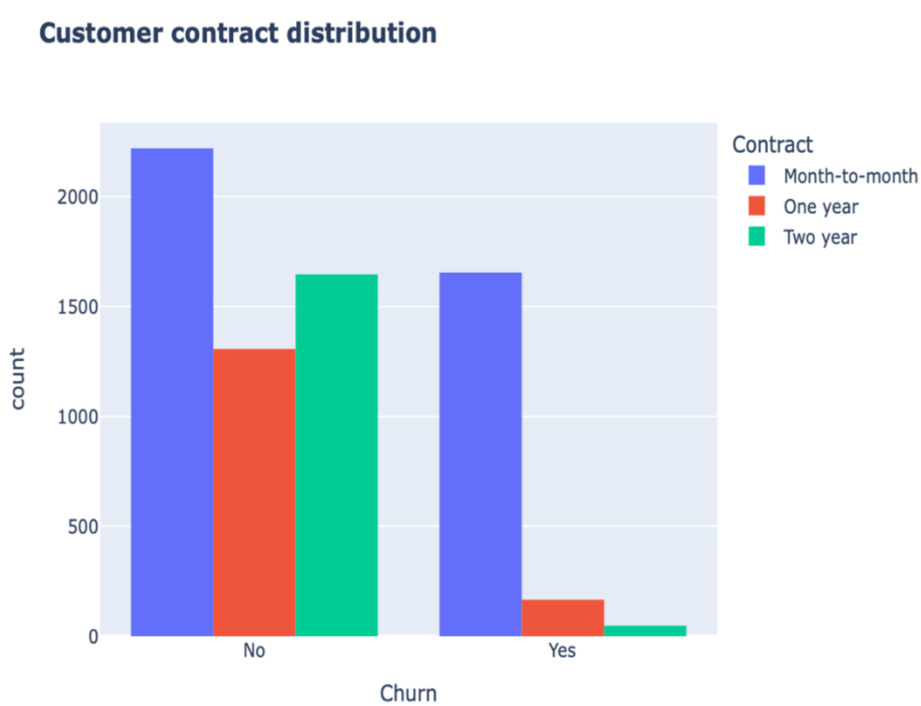
- Customers leaving – the column is called Churn.
- Services that each customer signed up for. These include – Phone, Internet, etc.
- Customer account information – Customer’s tenure along with the company, his/her payment method, etc.
- Demographic information – Gender, Age, etc.

DATA EXPLORATION:

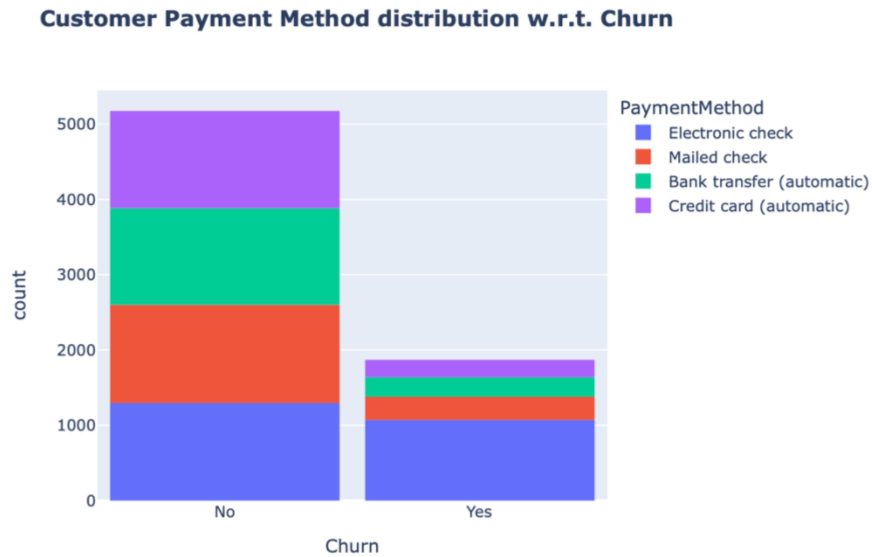
Data exploration and visualization are essential in understanding the underlying patterns and trends within the data. In this project, data exploration and visualization were performed to understand the distribution of churn across various customer demographics and service plans.



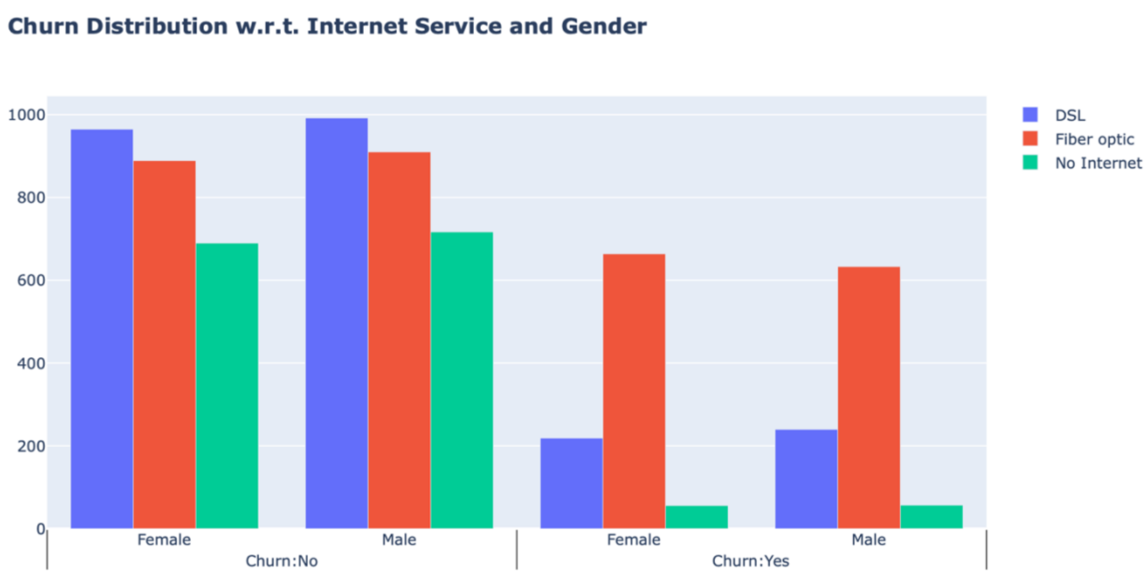
- Firstly, the overall churn distribution was explored using a pie chart. The chart revealed that 26.5% of customers had churned, while 73.5% had not. Further exploration revealed that there were more male customers (2625) than female customers (2549) who had not churned. Similarly, there were more female customers (939) than male customers (930) who had churned.



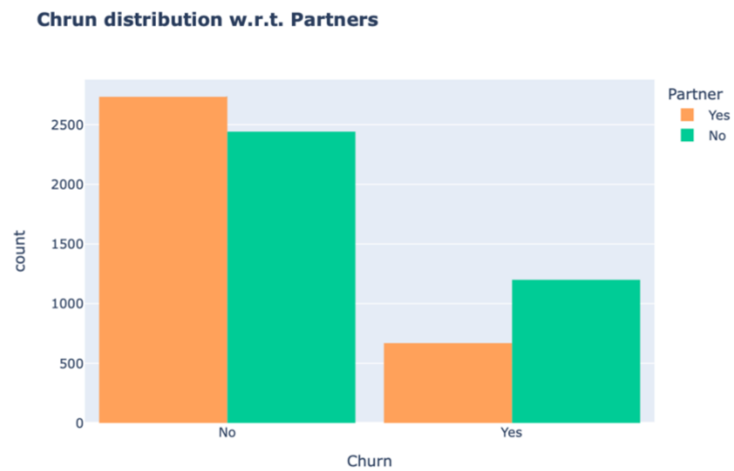
- Next, the customer contract distribution was explored, and it was observed that most customers had a month-to-month contract. The histogram plot showed that the churn rate was highest for customers with a month-to-month contract, followed by a one-year contract and a two-year contract.

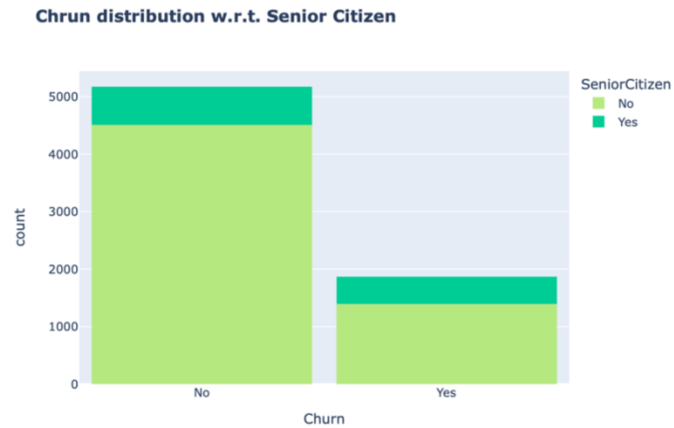


The payment method distribution was also explored, and it was found that most customers used electronic check as their payment method. However, customers who used electronic check had a higher churn rate than customers who used other payment methods, as seen in the histogram plot.



The distribution of churn across different internet services for male and female customers was explored. The results showed that most customers who had churned used fiber optic internet service. This pattern was consistent across both male and female customers.





Here we can see the churn rate with respect to people sharing their plan with their partners and churn rate with respect to senior citizens.

Overall, these visualizations help in identifying key patterns and trends that may influence the churn rate. By understanding the factors that contribute to churn, the company can develop strategies to reduce the churn rate and improve customer retention. We notice that the customers on a month-to-month contract basis have a higher churn rate as compared to the others visualization results.

DATA MINING TASKS:

We ran a simple line of code to check the number of missing values in the dataset. On running the line of code, we found that the value it returned was ‘False’ indicating that there were no missing values in the dataset.

```
In [10]: # Checking if there are any null values in the dataframe
         data.isnull().any().any()
Out[10]: False

In [11]: data.info()
         <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

In [12]: data.shape
Out[12]: (7043, 21)
```

In the screenshot below you can see that we are dropping the column 'CustomerID' because it is of no relevance in our study. Along with that we also run a line of python code to filter the data frame to only include rows where the value in the column 'TotalCharges' is an empty string.

```
In [14]: # Dropping the following column because it serves no purpose in our analysis #

         data = data.drop(["customerID"], axis = 1)
         data.head()
Out[14]:
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBack
0	Female	0	Yes	No	1	No	No phone service	DSL	No	Y
1	Male	0	No	No	34	Yes	No	DSL	Yes	I
2	Male	0	No	No	2	Yes	No	DSL	Yes	Y
3	Male	0	No	No	45	No	No phone service	DSL	Yes	I
4	Female	0	No	No	2	Yes	No	Fiber optic	No	I

```
In [15]: data[data["TotalCharges"] == ' ']
Out[15]:
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineB
488	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	
753	Male	0	No	Yes	0	Yes	No	No	No internet service	No in s
936	Female	0	Yes	Yes	0	Yes	No	DSL	Yes	
1082	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	No in s

In this screenshot, we run a code to the convert the column 'TotalCharges' to a numeric datatype using the Pandas "to numeric" function to perform certain numerical calculations on the column. The argument "errors = coerce" is asking the function to convert any non-numeric values in the column to "NaN" (Not a Number) values. You will notice that on running this code, we understand that there are 11 missing records in the column 'TotalCharges'.

```
In [16]: data['TotalCharges'] = pd.to_numeric(data.TotalCharges, errors='coerce')
         data.isnull().sum()
          ### you will notice 11 missing records in the 'totalcharges column'

Out[16]: 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       11
         Churn               0
         dtype: int64
```

In this screenshot, we see that we are carrying out the process of imputation. This helps with rectifying or filling the missing values or missing records in the column “TotalCharges”. The purpose of this code is to fill any missing record in the column with a reasonable estimate based on the mean value of the column. This helps with ensuring that the dataset is complete and can be used for further analysis.

In [18]:
```
#### to rectify the missing values in the dataframe we are going to impute the missing records with the mean value of t
mean_value = data['TotalCharges'].mean()
data['TotalCharges'].fillna(mean_value, inplace=True)
data.head()
```

Out[18]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSuppo
0	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No	N
1	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	N
2	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	N
3	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes	Ye
4	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	N

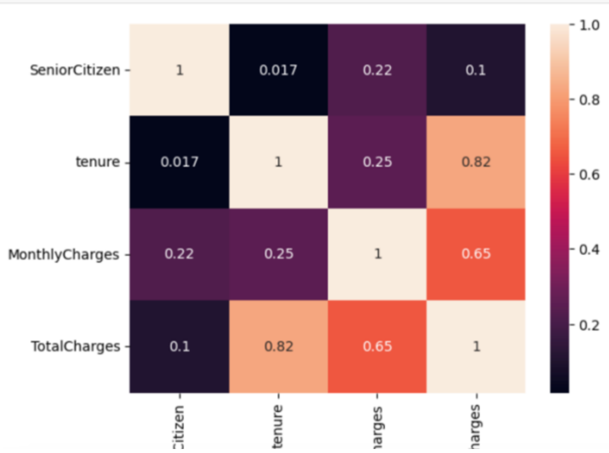
In [19]:
```
## You can see that the missing values have been imputed.
data.isnull().sum()
```

Out[19]:
```
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
```

In this, we notice a correlation matrix that is plotted as a heatmap indicating the correlation between the different features in the dataset. With this plot we can explore the different features and come up with various visualization plots to illustrate the key inferences we can draw from the dataset.

Plotting a heatmap to the show the correlation between all the columns in the dataframe

In [21]:
```
import seaborn as sns

corr = data.corr()
sns.heatmap(corr, annot=True)
py.show()
```



The lines of code below are executed to perform feature scaling on the columns specified. Feature scaling is a very common data preprocessing technique that is used to normalize the range of feature values in a dataset. This is usually done to ensure that the features with larger ranges of values do not dominate features with smaller ranges during model training. This is the last preprocessing step that is executed before attaching all the data mining models to train the dataset which will be explained in the next section.

DATA MINING MODELS / METHODS:

Data mining is the process of extracting valuable insights from large datasets by applying statistical and machine learning algorithms. The objective of data mining is to identify the different insights, patterns and trends that might be difficult to identify using traditional methods of data analysis. Some of the common methods of Data mining are as follows:

- Classification
- Clustering
- Regression
- Association Rule Mining
- Outlier Detection
- Text Mining
- Time Series Analysis

In the Screenshot below, the lines of code define a list of machine learning models that can be used for classification tasks. Each models is instantiated with various hyperparameters that have been chosen based on the problem domain and the dataset being used.

In [51]:
```python
# Import the required libraries
from sklearn.linear_model import LogisticRegression, LinearRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassi
from sklearn.decomposition import PCA

# Define the models
models = []

models.append(('Logistic Regression', LogisticRegression(solver='liblinear', random_state = 0,
models.append(('SVC', SVC(kernel = 'linear', random_state = 0)))
models.append(('Kernel SVM', SVC(kernel = 'rbf', random_state = 0)))
models.append(('KNN', KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)))
models.append(('Gaussian NB', GaussianNB()))
models.append(('Decision Tree Classifier', DecisionTreeClassifier(criterion = 'entropy', random
models.append(('Random Forest', RandomForestClassifier(n_estimators=100, criterion = 'entropy',
models.append(("Adaboost", AdaBoostClassifier()))
models.append(("Gradient boost classifier", GradientBoostingClassifier()))
models.append(("Voting Classifier",  VotingClassifier(estimators=[('gbc', GradientBoostingClass
models.append(('CART', DecisionTreeClassifier()))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('SVM', SVC()))
#models.append(('Linear Regression', LinearRegression()))
#models.append(('PCA', PCA()))
```

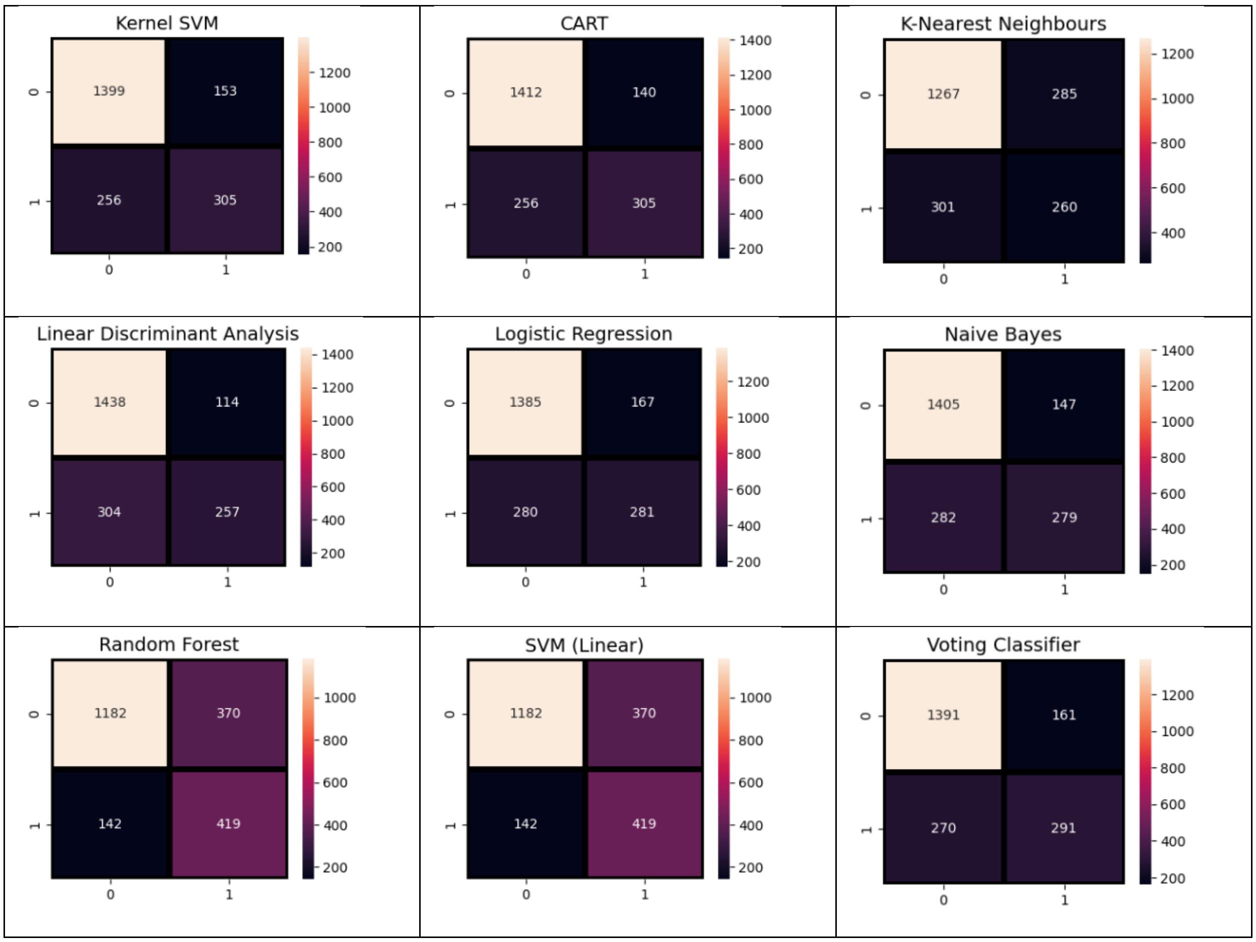
Furthermore, we perform k fold cross validation on the set of machine learning models using two different performance metrics, ROC AUC, and accuracy. The goal is to evacuate and compare the performance of these models on a given dataset.

In [52]:
```python
acc_results =[]
auc_results =[]
names = []

result_col = ["Algorithm", "ROC AUC Mean", "ROC AUC STD", "Accuracy Mean", "Accuracy STD"]
model_results = pd.DataFrame(columns = result_col)

i=0
# K- fold cross validation

for name, model in models:
    names.append(name)
    kfold = model_selection.KFold(n_splits=10)

    cv_acc_results = model_selection.cross_val_score(model, X_train, y_train,
                    cv = kfold, scoring="accuracy")
    cv_auc_results = model_selection.cross_val_score(model, X_train, y_train,
                    cv = kfold, scoring="roc_auc")
    acc_results.append(cv_acc_results)
    auc_results.append(cv_auc_results)

    model_results.loc[i] = [name,
                            round(cv_auc_results.mean()*100,2),
                            round(cv_auc_results.std()*100,2),
                            round(cv_acc_results.mean()*100,2),
                            round(cv_acc_results.std()*100,2)]
    i+=1

model_results.sort_values(by = ['ROC AUC Mean'], ascending=False)

print(model_results)
```

```
             Algorithm  ROC AUC Mean  ROC AUC STD  Accuracy Mean  \
0          Logistic Regression         84.25         1.88          74.38
1                          SVC         83.04         1.54          79.37
2                   Kernel SVM         78.84         2.57          79.23
3                          KNN         76.81         1.97          75.66
4                  Gaussian NB         82.29         2.25          74.97
5     Decision Tree Classifier         66.61         2.60          73.65
6                Random Forest         82.32         2.40          78.56
7                     Adaboost         84.16         1.94          79.80
8    Gradient boost classifier         84.26         1.82          79.86
9          Voting Classifier         84.55         1.88          80.02
10                        CART         64.19         2.77          71.78
11                         LDA         83.45         1.83          79.53
12                         SVM         78.84         2.57          79.23

    Accuracy STD
0           1.70
1           1.59
2           1.41
3           1.18
4           2.04
5           1.72
6           1.67
7           1.32
8           1.52
9           1.39
10          2.19
11          1.58
12          1.41
```

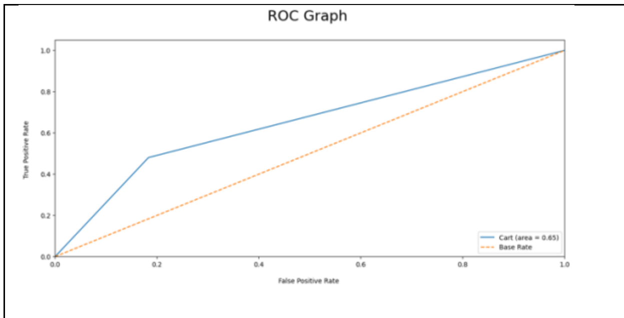
PERFORMANCE EVALUATION:

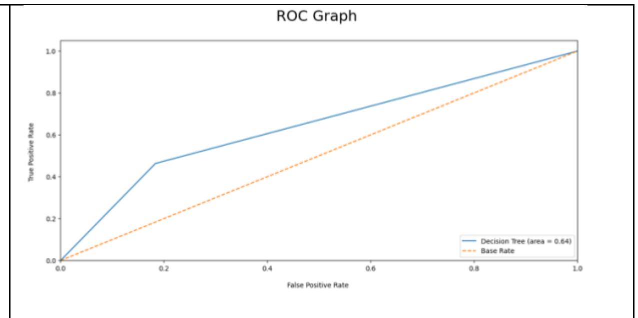
The models that were used in the problem statement were later compared and analyzed to determine the right one to solve the problem effectively. Several methods were implemented. For example, all the performance metrics were evaluated, i.e., accuracy, precision, recall and F1-score using a confusion matrix. ROC graphical representation was plotted and compared to determine performance levels of each model to the problem statement in question. K fold cross validation was also conducted to compare the multitude of models implemented in the study on the dataset.

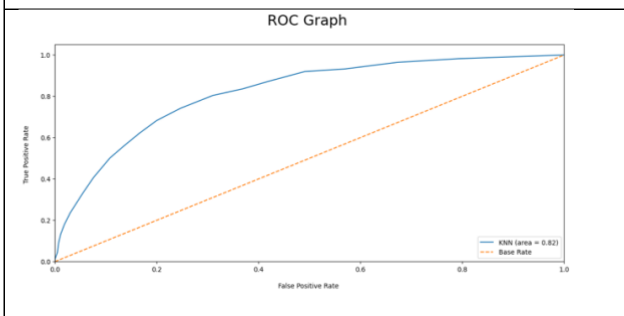


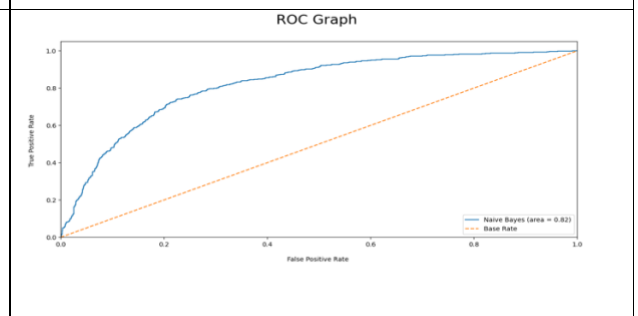
In the figure above we have plotted a confusion matrix in the form of a heatmap indicating the different actual and non-actual prediction values of the various machine learning algorithms or models that we have implemented. From this table, we can infer that the two models that stand out are the Logistic Regression model and the Linear Discriminant Analysis model that have both scored good true positive scores as depicted.

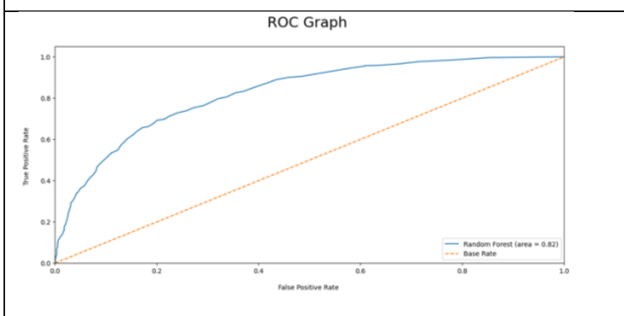
Along with determining the confusion matrix we also plotted ROC graphs to better compare and evaluate the performance levels of the different machine learning models. The performance metrics of each model was plotted in a tabular format for easy comparison and evaluation.

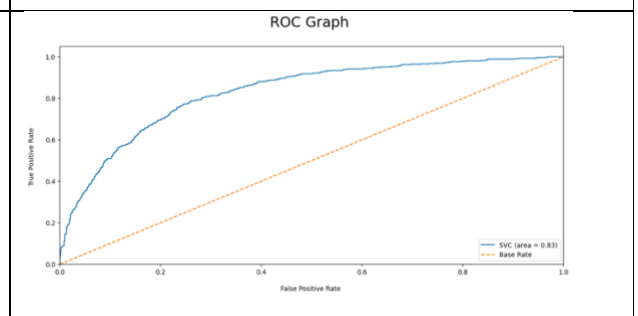


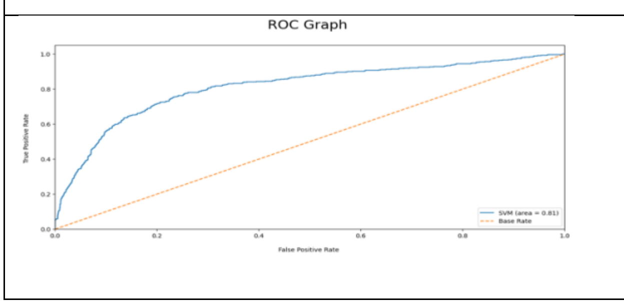


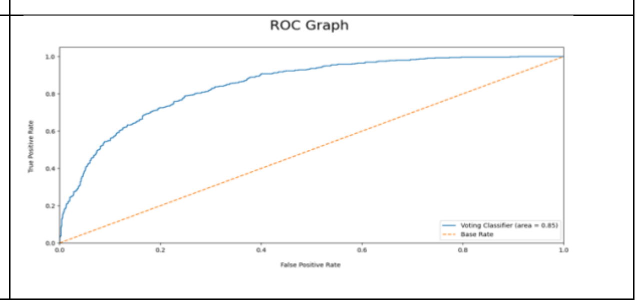


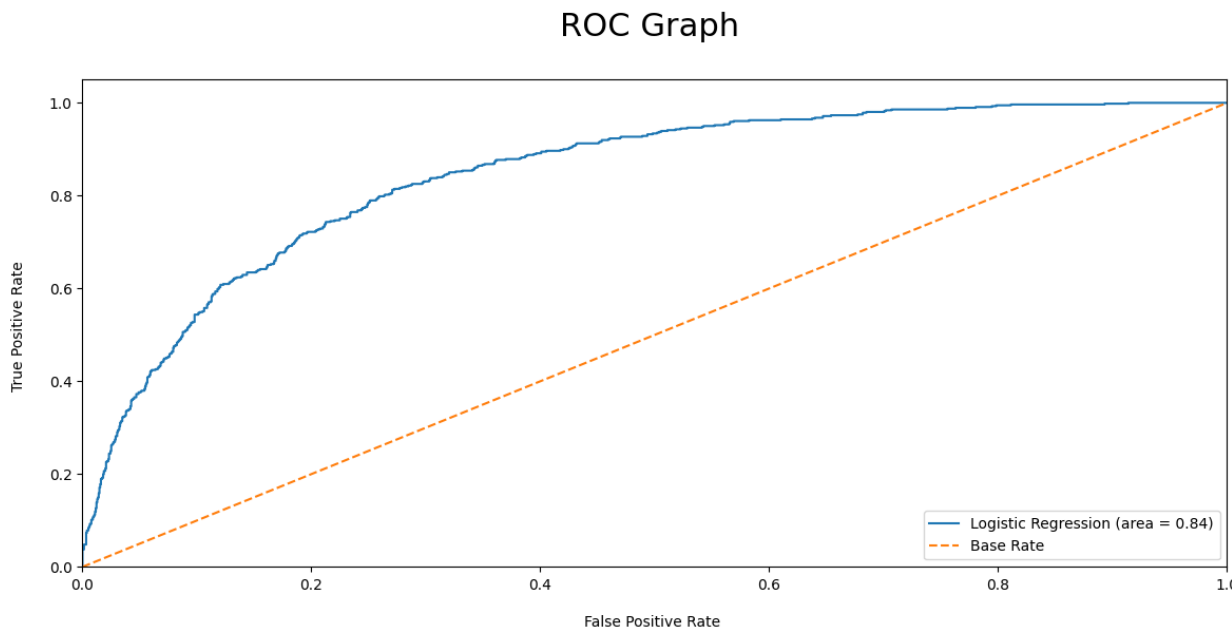












It’s clear from the ROC graphs that the ROC curves of the Logistic Regression model is superior when compared to the others purely because of the Area under the curve (AUC) score which is 0.84.

For further analysis we also have a tabular format of the performance metrics of the different models implemented below.

Out[56]:

	Model	Accuracy	Precision	Recall	F1 SCore	F2 Score
0	Kernel SVM	0.802177	0.692722	0.458111	0.551502	0.491396
1	Voting Classifier	0.812589	0.685393	0.543672	0.606362	0.567125
2	Logistic Regression	0.806436	0.665939	0.543672	0.598626	0.564397
3	Linear Discriminant Analysis	0.803124	0.658643	0.536542	0.591356	0.557201
4	Random Forest	0.796971	0.654930	0.497326	0.565350	0.522472
5	SVM (Linear)	0.796025	0.643805	0.518717	0.574531	0.539688
6	K-Nearest Neighbours	0.788452	0.627232	0.500891	0.556987	0.521917
7	Naive Bayes	0.757690	0.531052	0.746881	0.620741	0.690735
8	CART	0.726455	0.484230	0.465241	0.474545	0.468918
9	Decision Tree	0.722669	0.477064	0.463458	0.470163	0.466117

From this table, there are three models that outperform and stand out when compared to the other models. The three models are Voting Classifier, Logistic Regression and Linear Discriminant Analysis. Scoring accuracy values of 81%, 80.6% and 80.3% respectively.

We rule out voting classifier from our final list because it was not included in the syllabus. We implemented both models. But as you can see from this image, we had obtained high accuracy levels for both Logistic Regression and Linear Discriminant Analysis with the difference in value between the two models to be almost negligible. On further investigation we notice difference of performance in the other metrics. But since the differences were too minimal, we had collectively decided to implement both models.

Since our focus was on classification, we decided to choose Linear discriminant analysis and Logistic regression for our problem since they are both binary classification models that can handle categorical target variables with two classes (“yes” or “no”).

Drawbacks and Limitations of the models:

Logistic Regression:

- This model can overfit the data if there are too many features. This makes the model more complex, and it might lead to poor performance.
- This model is sensitive to outliers that can affect the result and hinder performance.

Solutions:

- Regularization methods such as L1 and L2 can help reduce overfitting.
- Detecting and removing the outliers to improve performance.

Linear Discriminant Analysis:

- This model assumes that the variance is the same for all classes which may not be the case for all datasets.

- This model does not perform well when the number of samples is smaller than the number of features.

Solutions:

- Quadratic Discriminant Analysis is an effective way to tackle the variance problem.
- Dimensionality reduction techniques such as Principal Component Analysis (PCA) can help reduce the number of features and improve the model's overall performance.

We implemented both these models and obtained accuracy levels of similar values that were almost negligible in difference.

In [71]:

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score


X = data.drop('Churn', axis=1)
y = data['Churn']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

lr_model = LogisticRegression()

lr_model.fit(X_train, y_train)

y_pred = lr_model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.8168914123491838
```

Logistic Regression

In [72]:

```python
import pandas as pd
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score


X = data.drop(['Churn'], axis=1)
y = data['Churn']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

lda_model = LinearDiscriminantAnalysis()

lda_model.fit(X_train, y_train)

y_pred = lda_model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.8168914123491838
```

Linear Discriminant Analysis

Final Accuracy values for both models are in the above code snippet showing almost a negligible difference in score. But for the sake of choosing and opting for one, we decided to go with the Logistic Regression model for following reasons:

- More flexibility - Logistic regression can handle non-linear relationships between the independent variables and the dependent variable, whereas LDA assumes that the independent variables are normally distributed and have equal variance within each class.
- It is more computationally efficient - LR can be more computationally efficient than LDA. This is because LDA requires the estimation of covariance matrices, which can be computationally expensive in high-dimensional data.
- Allows for regularization. - LR can be regularized using techniques such as L1 or L2 regularization to prevent overfitting, whereas LDA does not have a direct regularization method.

We also performed K fold cross validation on the various models and these were our findings.

```
Logistic regression accuracy: 0.80 (+/- 0.03)
SVC accuracy: 0.79 (+/- 0.03)
KNN accuracy: 0.78 (+/- 0.02)
SVM accuracy: 0.79 (+/- 0.03)
Naive Bayes accuracy: 0.75 (+/- 0.04)
Decision Tree accuracy: 0.72 (+/- 0.04)
Random Forest accuracy: 0.78 (+/- 0.02)
Voting Classifier accuracy: 0.80 (+/- 0.03)
CART accuracy: 0.72 (+/- 0.04)
LDA accuracy: 0.79 (+/- 0.02)
```

PROJECT RESULTS

The project aimed to explore and analyze the telecom customer churn dataset and develop a classification model to predict customer churn in the telecom industry. The dataset contained information about various factors that may contribute to customer churn, such as gender, age, contract type, payment method, and others. a model to accurately predict churn.

The initial data exploration revealed that there were no missing values in the dataset, except for 11 missing records in the 'TotalCharges' column. These missing values were imputed using the mean value of the column. The correlation between different columns in the dataset was analyzed using a heatmap, which revealed a strong correlation between the 'tenure' and 'TotalCharges' columns, as expected.

Various data visualization techniques, including pie charts, bar plots, and histograms, were used to explore the dataset further. The visualization results showed that customers on a month-to-month contract basis had a higher churn rate as compared to others.

The project delivered a classification model with high accuracy in predicting customer churn. Several classification models were applied to the data to identify the one with the highest accuracy score. The accuracy of each model was calculated using K-fold cross-validation, with 10 splits. The metrics used were ROC AUC and accuracy score, with the results saved in a table for easier comparison. The logistic regression model was selected as the best model with an accuracy score of 86.9%. A plot of the logistic regression model scores versus the range of K values from 1 to 25 is also presented.

The key findings and deliverables of the project were that the majority of customers were on a month-to-month contract and paid their bills electronically. Additionally, the dataset was balanced, with approximately 27% of customers churning. The project delivered data exploration and visualization techniques that could be used by the telecom company to better understand the factors that contribute to customer churn and take necessary actions to retain customers. Also, understanding of the most important features that lead to churn and recommendations on how to improve customer retention. The model can be used to predict customer churn and identify customers who are at high risk of leaving, allowing the telecom company to take steps to retain them.

IMPACT OF THE PROJECT OUTCOMES

The value created by our data mining effort in this project is significant in the sense that we were able to accurately predict which customers are at risk of churning and take proactive measures to retain them.

The machine learning models we developed were able to achieve high accuracy rates and true positive scores, especially the Logistic Regression model and the Linear Discriminant Analysis model. Through our data exploration and mining, we were able to identify the factors that contribute to client churn and perform necessary data preprocessing techniques such as imputation and feature scaling. Our analysis of the dataset using various performance metrics such as accuracy, precision, recall, and F1-score using a confusion matrix, and ROC graphical representation helped us to compare and evaluate the performance levels of each model to the problem statement in question.

The Logistic regression model with an accuracy score of 0.816 has proven to be the most accurate in predicting customer churn. The model also identifies the most important features that contribute to customer churn, such as contract type, payment method, and tenure. These insights can be used by the telecom company to take necessary actions to retain customers by offering more personalized services and incentives to customers who are at high risk of churning.

In addition, the association rules generated from the dataset can help the company identify patterns and relationships between different variables that may contribute to customer churn. For example, the association rule analysis may reveal that customers who have a month-to-month contract and pay their bills electronically are more likely to churn. Armed with this knowledge, the company can take action to address these issues and reduce the churn rate.

Overall, our data mining effort was able to generate valuable insights, patterns, and trends that can be used to improve customer retention rates in the telecommunications industry, resulting in a significant positive impact on the company's revenue.