Customer Churn Prediction Analysis using Machine-learning

Introduction

Customer churn is an issue most businesses face, particularly those that employ a subscription service. Companies need to know the factors that contribute to customer churn and make plans to stop customers from leaving before they do so. Most companies measure their CAC or Cost of Acquisition of new Customers, and in most cases, it is far more profitable to keep an existing customer than to try and acquire a new one. The primary goal of this exercise is to build a machine-learning model that can predict customer churn based on the given information in the dataset.

Methodology

The purpose here is to build machine learning models using the 19 independent variables to predict the target feature of "Churn." I will first use the Scikit-Learn library to create a logistic regression model since we are looking at dichotomous or binary target variables. Then I will use a Random Forest Classifier since it utilizes an ensemble method, which may give better results. After this, I can then use cross-validation to see which model fits better.

Data

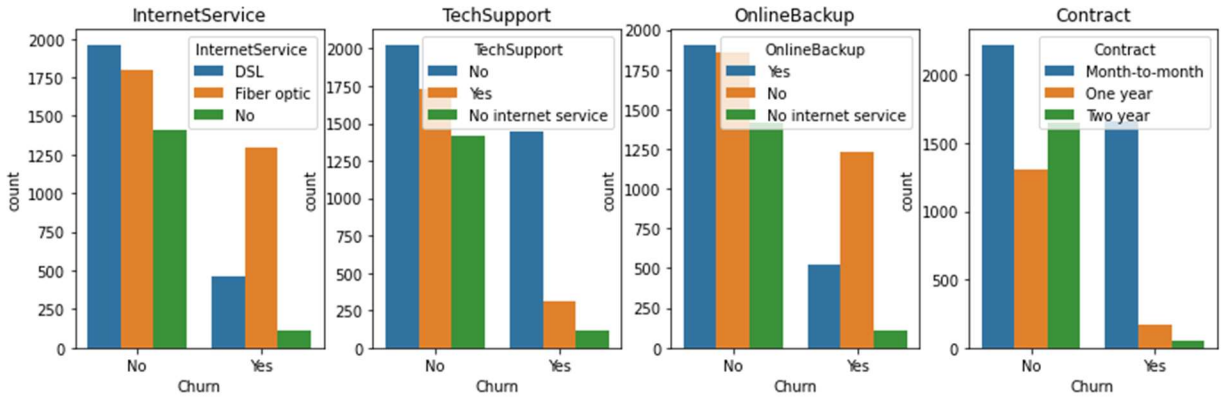
The dataset comes from an IBM development platform for a fictitious phone/internet company called Telco. It can be accessed through IBM and is available on Kaggle as well. The dataset has 21 columns and 7043 rows of data which has information on 5174 current customers and 1869 customers who have churned.

```
df.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No	No
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes	No
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	No	No

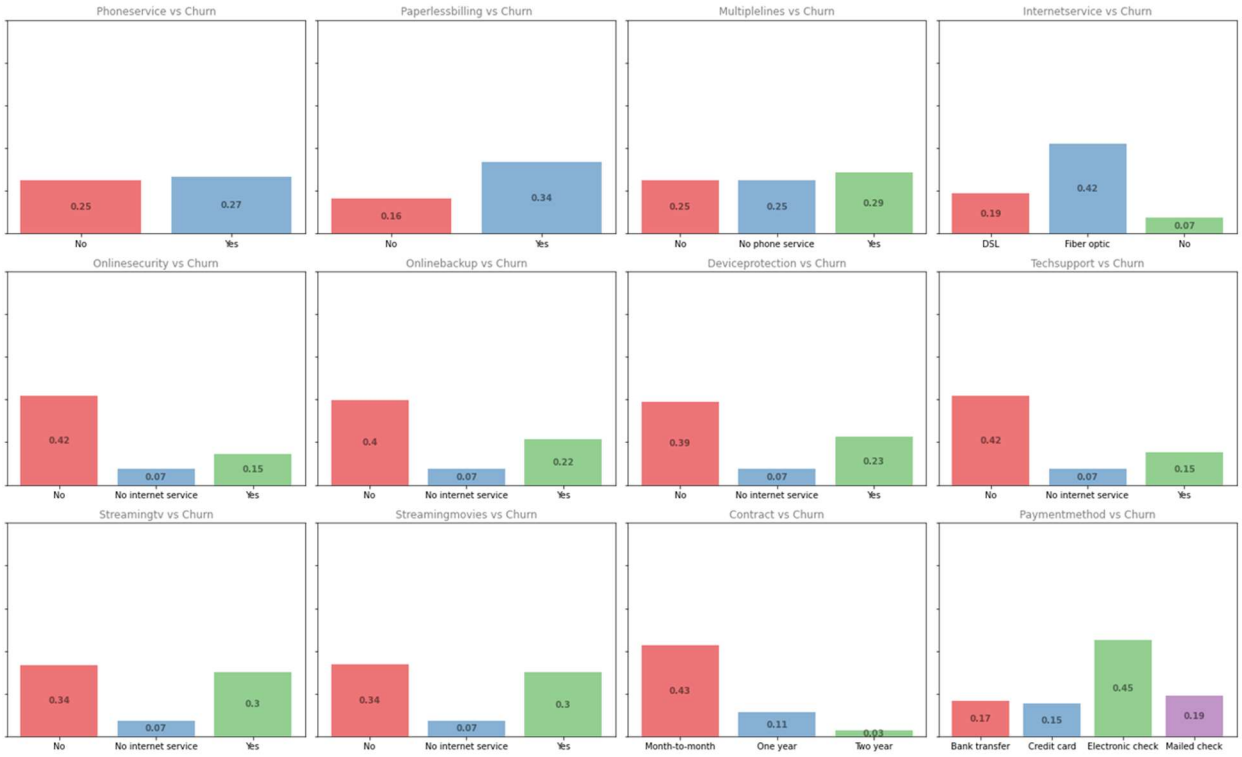
Exploratory Data Analysis

When looking at the data, some of the relationships which can be determined are as follows:

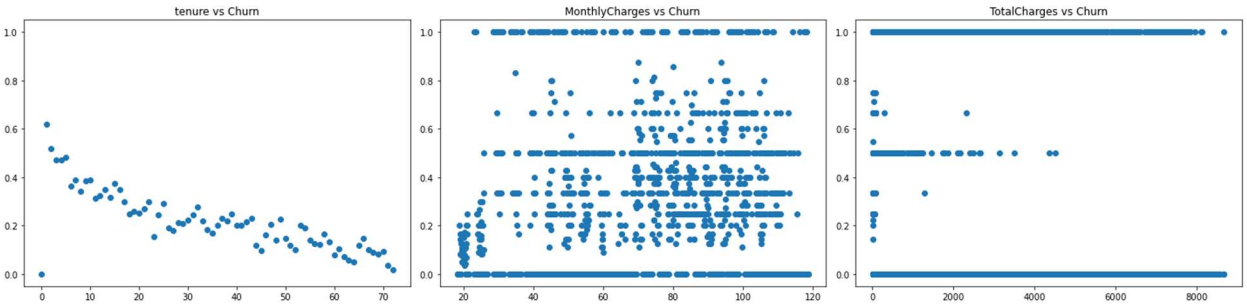


- Customers with no online security or backup, no device protection, and no tech support are from two to three times more likely to churn.
- Customers with no internet service are unlikely to churn.
- Customers with month-to-month contracts are almost four times more likely to churn than customers with yearly contracts. Two-year contractors are very unlikely to churn.

Exploratory Data Analysis Cont….



- Customers without dependents are two times more likely to churn.
- Customers that use paperless billing and optical fiber are more likely to churn.
- Customers that use electronic checks to pay their bills are more likely to churn.



- There is a direct correlation between Tenure and Churn

Preprocessing

I utilized Scikit-Learn's fit-transform feature to turn all categorical features into numeric values.

Out[9]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup
0	0	0	1	0	1	0	1	0	0	2
1	1	0	0	0	34	1	0	0	2	0
2	1	0	0	0	2	1	0	0	2	2
3	1	0	0	0	45	0	1	0	2	0
4	0	0	0	0	2	1	0	1	0	0

Modeling

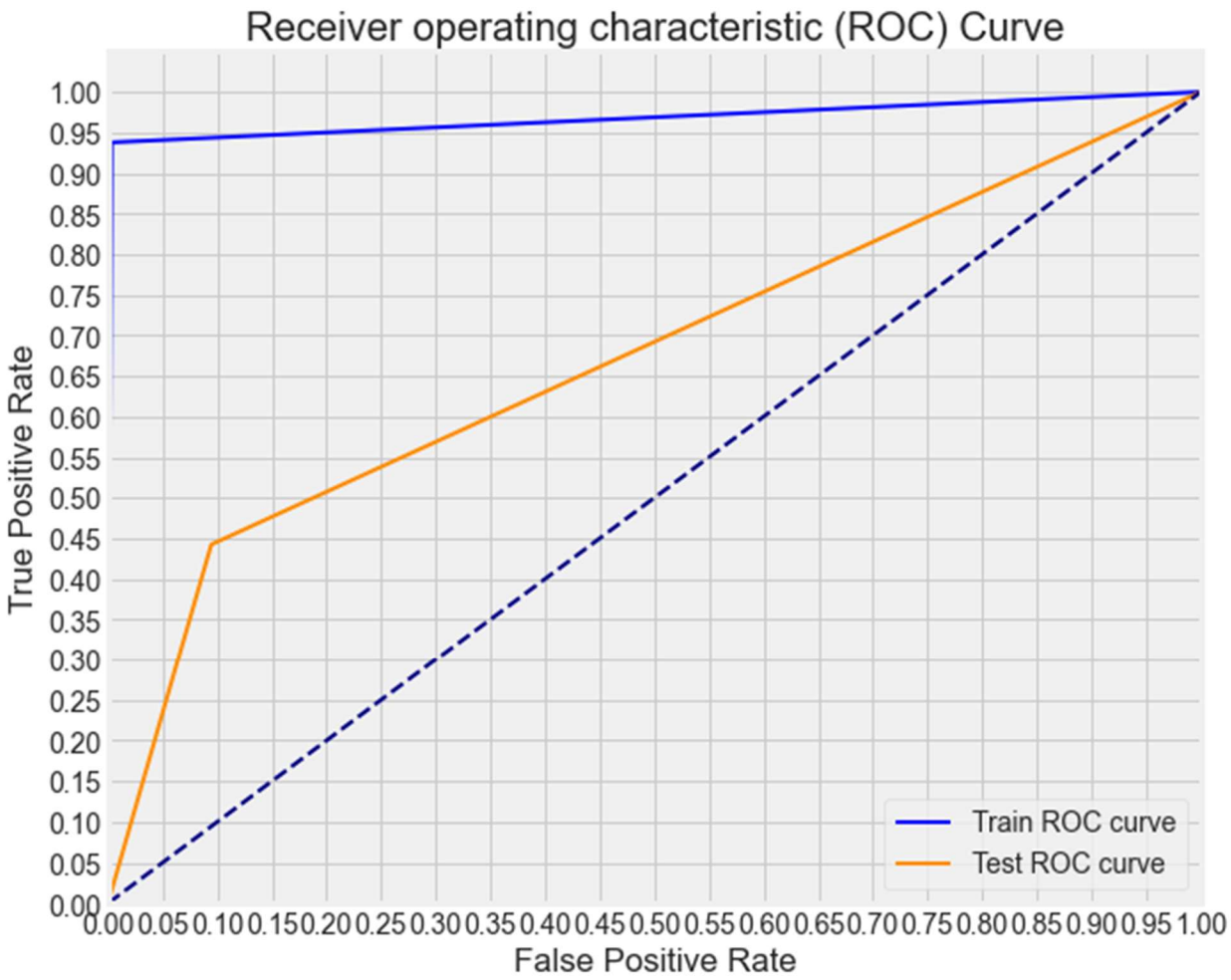
I first used a Logistic Regression model using Scikit-Learn. When the accuracy metrics of this model were calculated, the recall of this model was approximately 90%, meaning the model correctly identified 90% of the retained customers. The precision was about 86%, and the f1 score was 88%. Therefore, the accuracy of this model is about 82%.

In [14]:

```
#Check precision, recall, f1-score
print( classification_report(y_test, predictions) )
```

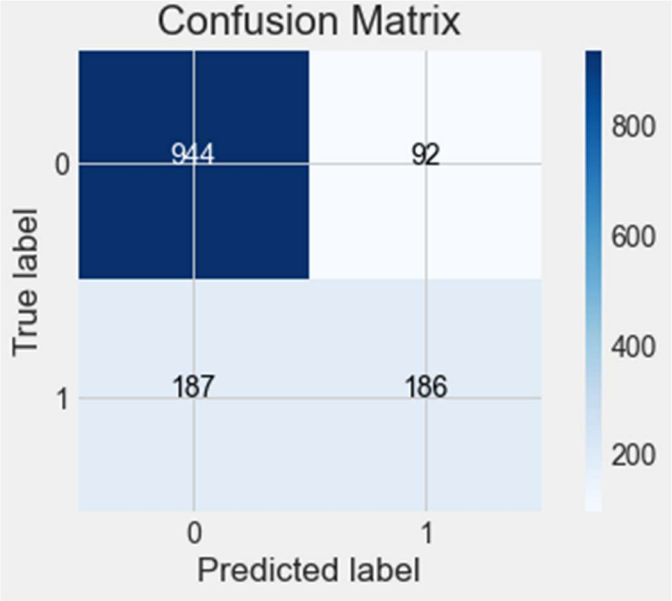
```
              precision    recall  f1-score   support

           0       0.86      0.90      0.88      1036
           1       0.68      0.58      0.62       373

    accuracy                           0.82      1409
   macro avg       0.77      0.74      0.75      1409
weighted avg       0.81      0.82      0.81      1409
```

I then built a Random Forest Classifier model, which had slightly better recall than Logistic regression at 91% but had less precision and less accuracy than the Logistic Regression model.



Hyperparameter tuning

I used a Grid search CV to find the optimal parameters for the Random Forest Model. After putting in the new parameters I was able to increase the accuracy of the Random Forest Model.




```
test accuracy: 0.8019872249822569
train accuracy: 0.8168264110756124
```

test report:

	precision	recall	f1-score	support
0	0.83	0.91	0.87	1036
1	0.67	0.50	0.57	373
accuracy			0.80	1409
macro avg	0.75	0.70	0.72	1409
weighted avg	0.79	0.80	0.79	1409

~~~~~~

Analysis

Even though hyper tuning increased the Random Forest Model slightly, the logistic regression model still performed better and is the model I would use.

Future Improvements

- In the future I would like to add an XGBoost model to see how it would correlate the data
- I would also like to utilize techniques to combat the fact that this is an unbalanced data set.
~~~~~~