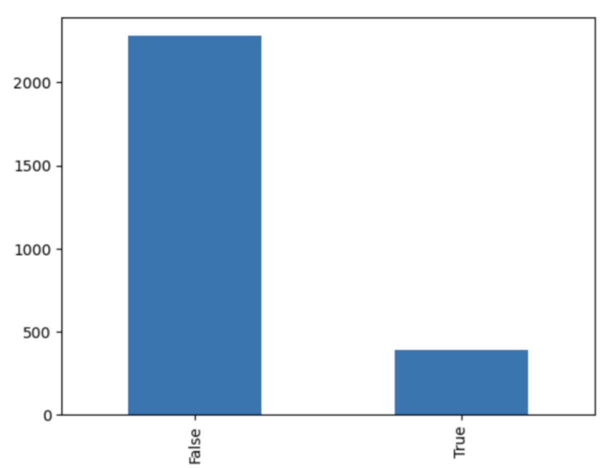
Exploratory Data Analysis:

From the output below we can see that there are no null values in our dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2666 entries, 0 to 2665
Data columns (total 20 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Region                          2666 non-null   object
 1   Tenure                          2666 non-null   int64
 2   Neighborhood                    2666 non-null   int64
 3   Trunk Calling Facility          2666 non-null   object
 4   Voice Messaging                 2666 non-null   object
 5   Number voice messages           2666 non-null   int64
 6   Minutes Peak Hrs                2666 non-null   float64
 7   Calls Peak Hrs                  2666 non-null   int64
 8   Bill Peak Hrs                   2666 non-null   float64
 9   Minutes Off Peak                2666 non-null   float64
 10  Calls Off Peak                  2666 non-null   int64
 11  Bill Off Peak                   2666 non-null   float64
 12  Minutes Night                   2666 non-null   float64
 13  Calls Night                     2666 non-null   int64
 14  Bill Night                      2666 non-null   float64
 15  Trunk Call Minutes              2666 non-null   float64
 16  Trunk Calls                     2666 non-null   int64
 17  Trunk Call Bill                 2666 non-null   float64
 18  Contact for Grievances/Changes  2666 non-null   int64
 19  Acct Closed?                    2666 non-null   bool
dtypes: bool(1), float64(8), int64(8), object(3)
```

From the graph below, we can see that there is a data imbalance in our dataset between our False and True values for the column ‘Acct Closed?’



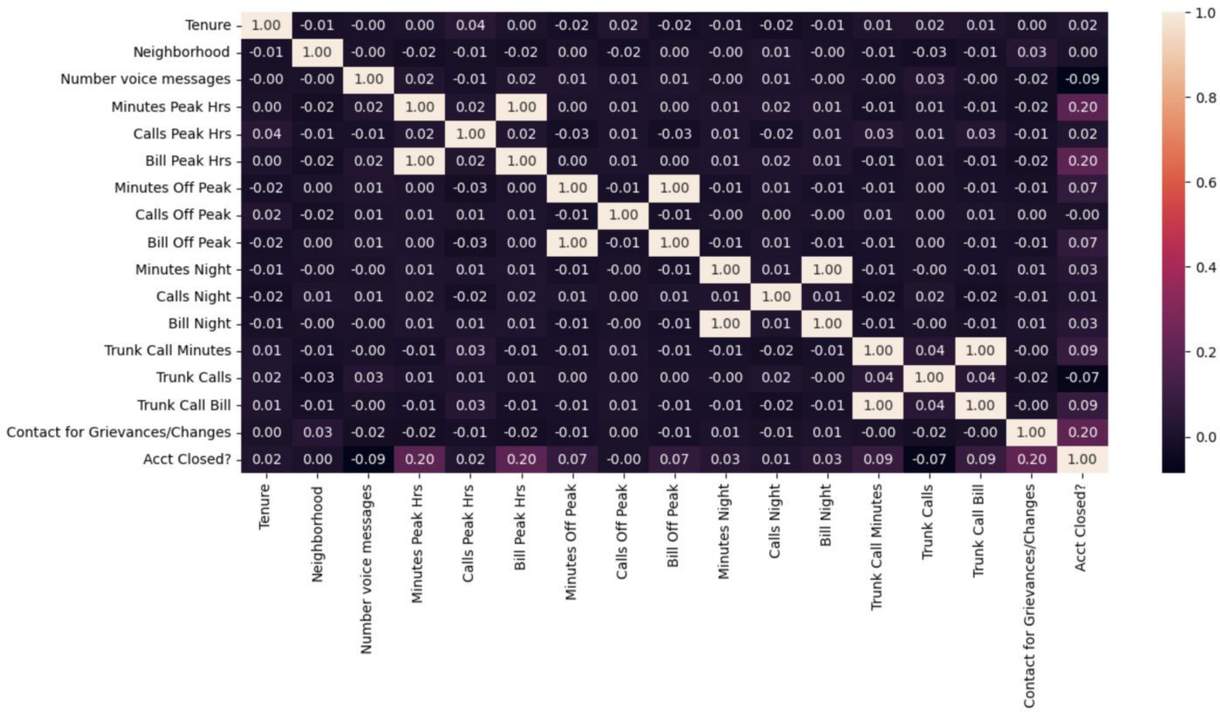
From the correlation heatmap below, we can see that the columns below are auto correlated:

1. "Minutes Peak Hrs" & "Bill Peak Hrs"

2."Minutes Off Peak" & "Bill Off Peak"

3. "Minutes Night" & "Bill Night"

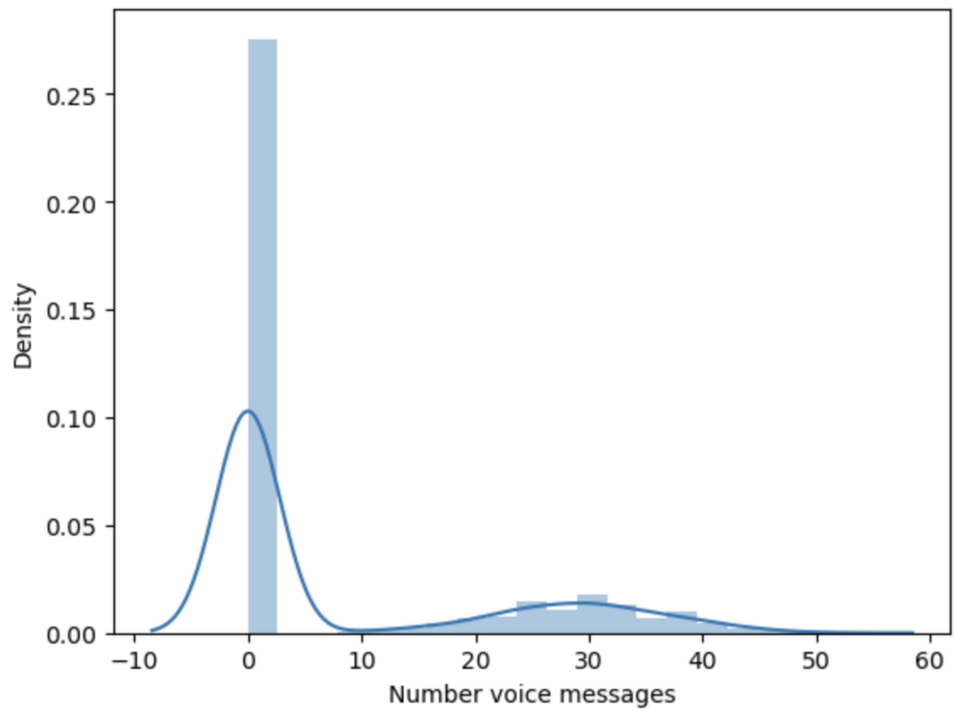
4. “Trunk Call Bill” & “Trunk Call Minutes”



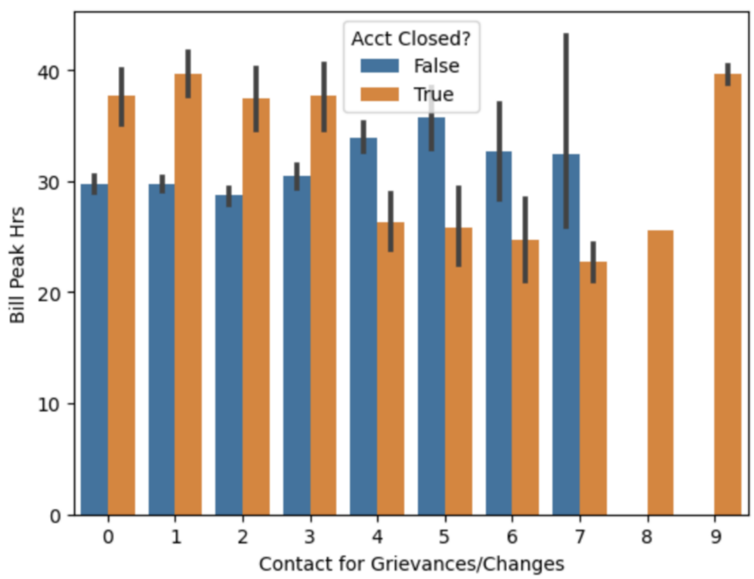
From the output below, We can see that there are 51 unique values for the column “Region”.

```
['KS' 'OH' 'NJ' 'OK' 'AL' 'MA' 'MO' 'WV' 'RI' 'IA' 'MT' 'ID' 'VT' 'VA'
 'TX' 'FL' 'CO' 'AZ' 'NE' 'WY' 'IL' 'NH' 'LA' 'GA' 'AK' 'MD' 'AR' 'WI'
 'OR' 'DE' 'IN' 'UT' 'CA' 'SD' 'NC' 'WA' 'MN' 'NM' 'NV' 'DC' 'NY' 'KY'
 'ME' 'MS' 'MI' 'SC' 'TN' 'PA' 'HI' 'ND' 'CT']
51
```

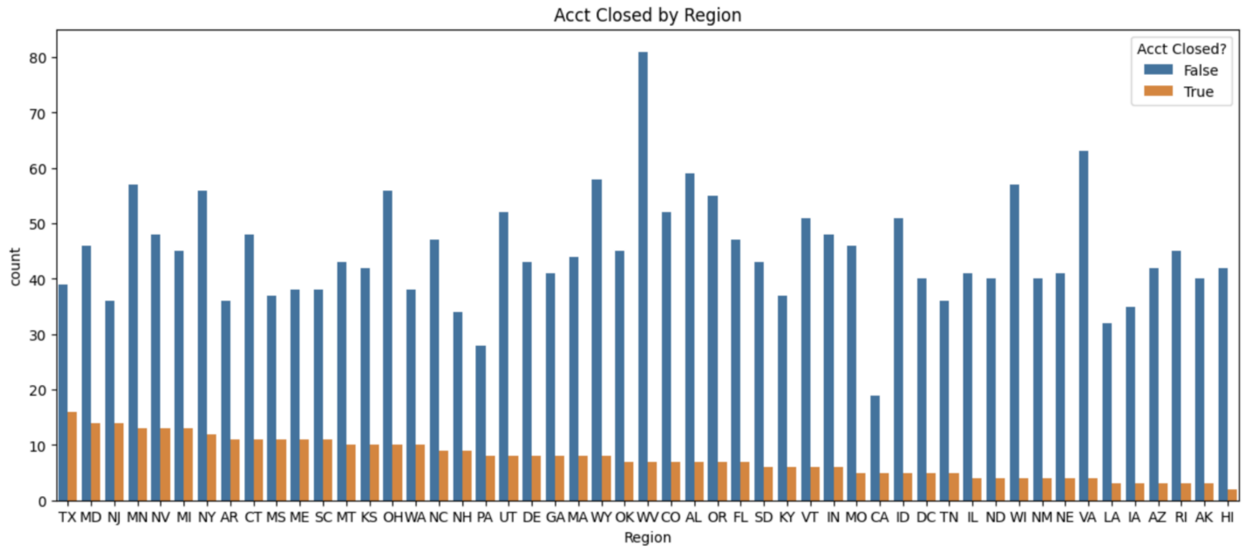
The graph below shows that most of the values for “Number of Voice Messages” are between 0-10.



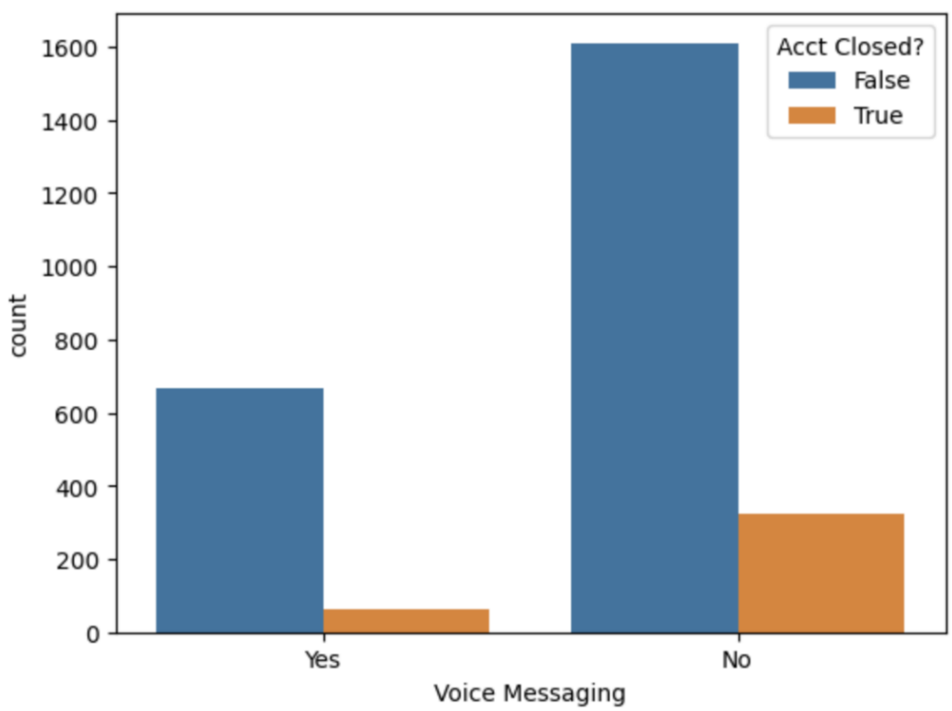
The graph below shows that if the value of “Contact for Grievances/Changes” is 8 or 9, The account is always closed



The graph below shows that the most number of Accounts closed fall under the “TX” region.

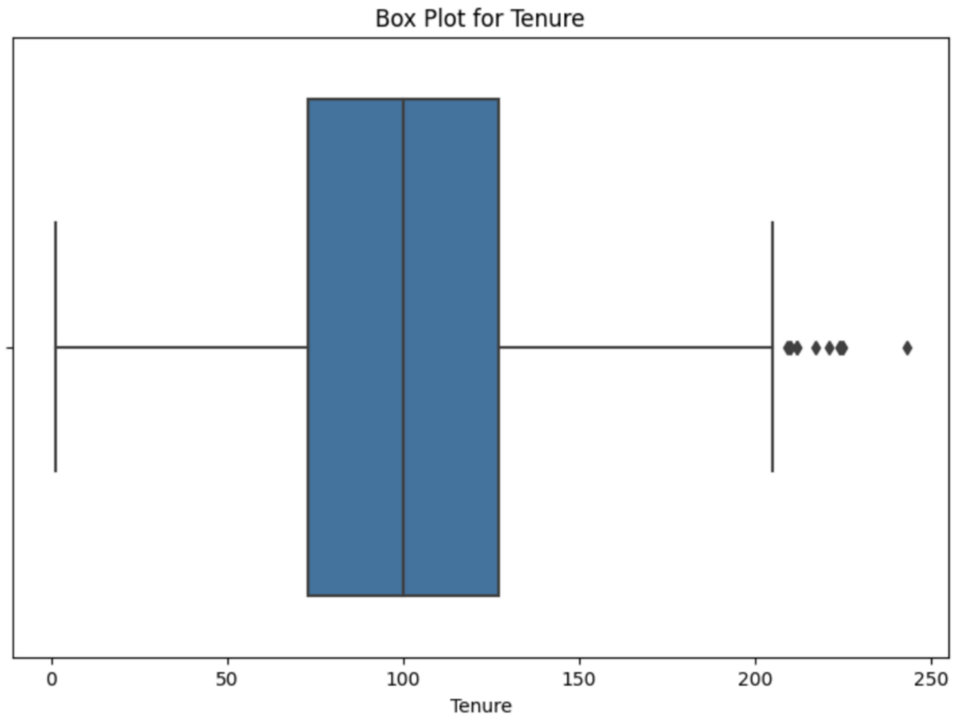


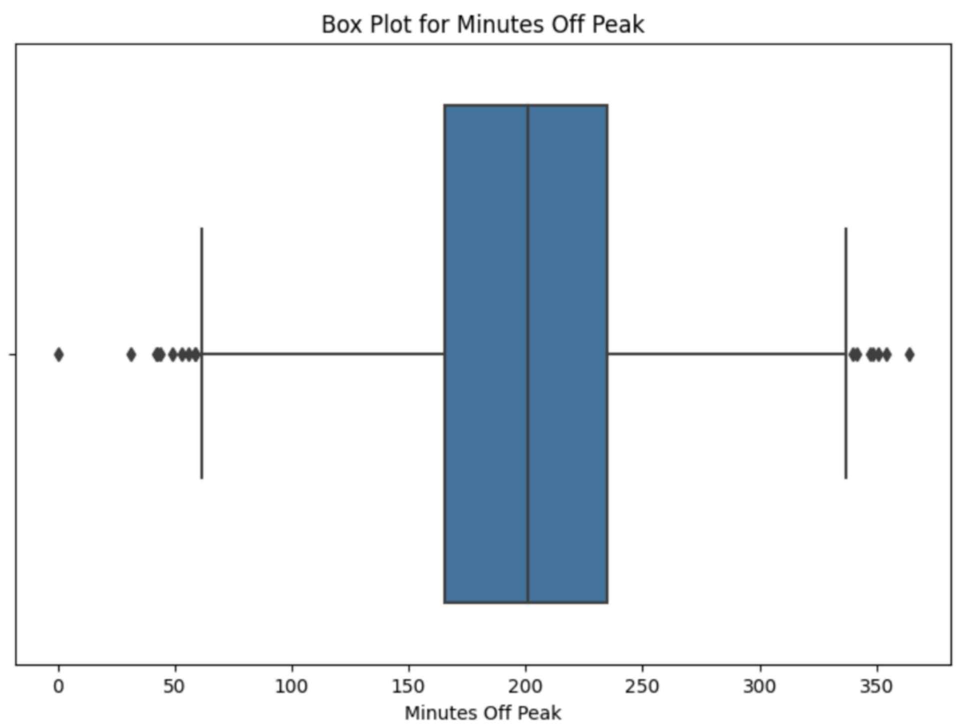
From the following graph we can infer that customers that do not have voice messaging enabled have a higher chance of closing their account.

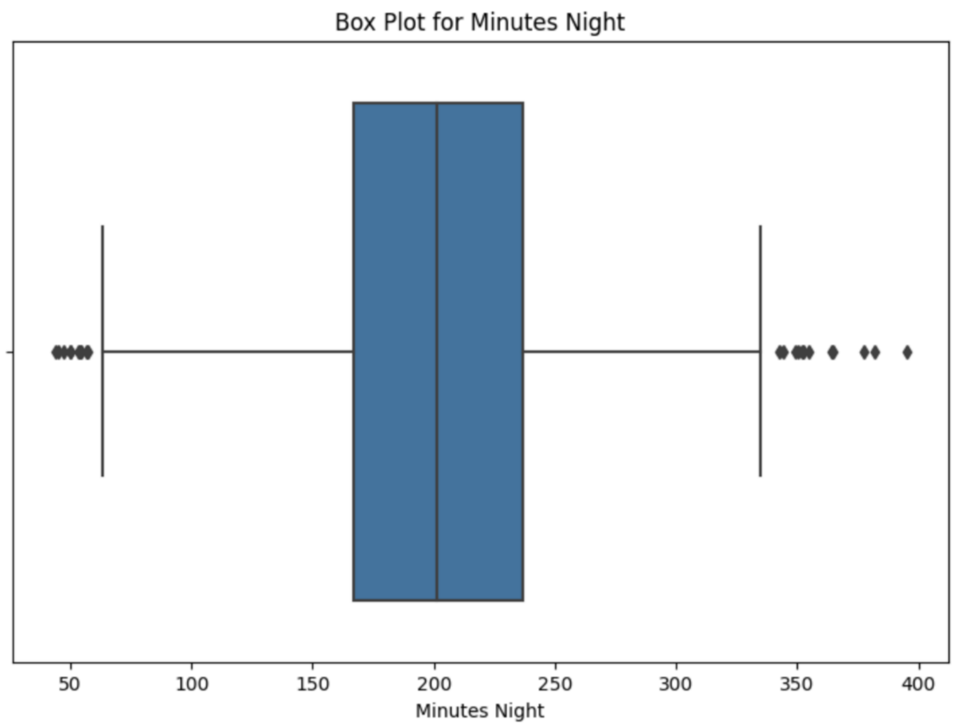


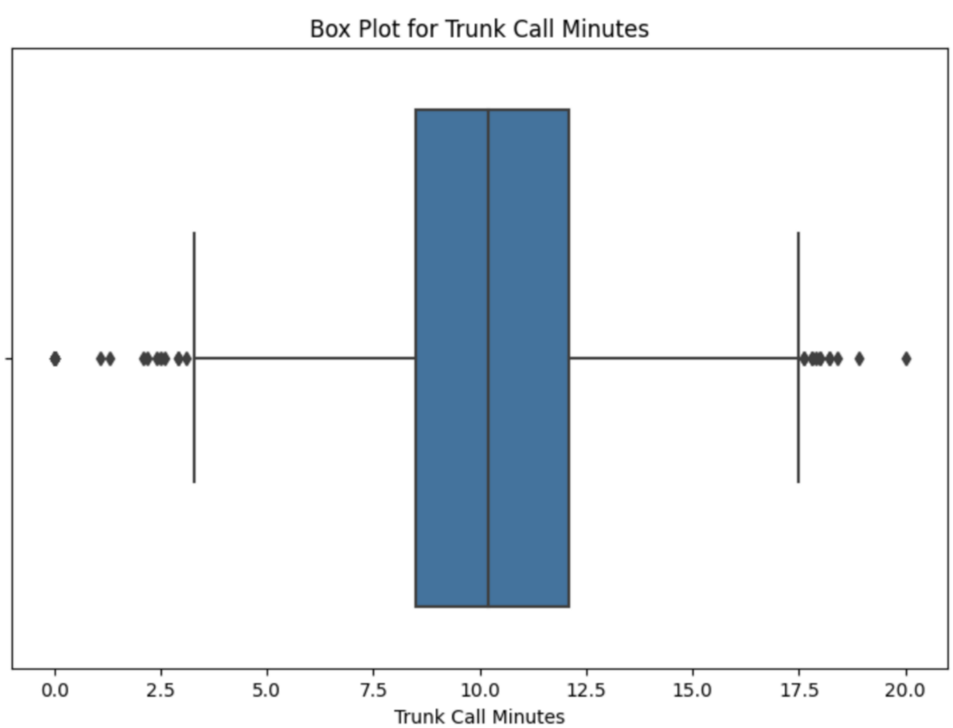
Outlier Analysis:

The boxplots below show the outliers for the numerical columns


Box Plot for Tenure
0
50
100
150
200
250
Tenure


Box Plot for Minutes Off Peak
0
50
100
150
200
250
300
350
Minutes Off Peak


Box Plot for Minutes Night
50
100
150
200
250
300
350
400
Minutes Night


Box Plot for Trunk Call Minutes
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Trunk Call Minutes

Initial Inferences:

1. Data is imbalanced.
2. We can merge "Region" and "Neighborhood" to reduce the number of features.
3. There are outliers in numerical columns.
4. We need to perform some feature engineering to deal with the columns that have auto correlation.

1st iteration - Data Preprocessing and Feature Engineering ():

1. Split the data into training and testing set and performed
2. Split the dataframe into dependent and independent features
3. Removing "Voice Messaging column" as its information is already available in "Number voice messages" column. e.g. 0 in Number voice messages column indicates No voice messaging facilities.
4. Outliers in our numerical columns that are quantative in nature by defining an IQR and removing the values above and below our threshold from our dataset. The total number of outlier values removed were 142.
5. Merge columns of Minutes data and Bill data as it provides similar information.
6. Merge Bill Columns to create new feature "Total Bill"
7. Drop the columns which was used to create merged feature, as it will be highly correated with each other.
8. Performed One Hot Encoding on "Region" and "Trunk Calling Facility".

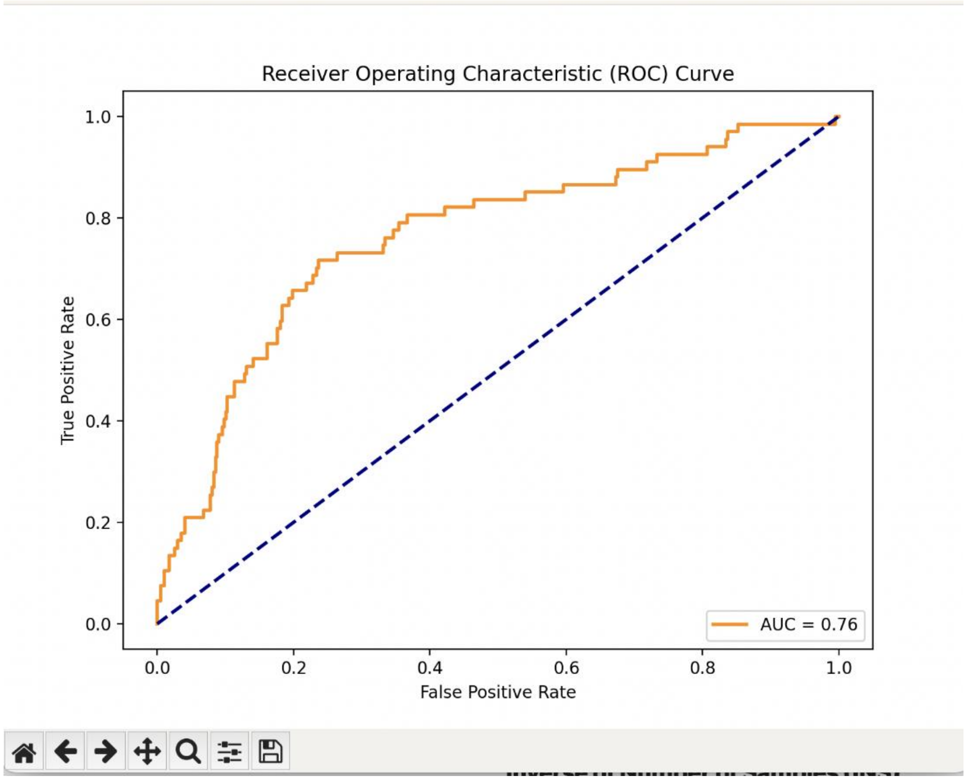
```
 Accuracy: 0.84

 Classification metrics
              precision    recall  f1-score   support

           0       0.88      0.95      0.91       398
           1       0.40      0.21      0.27        67

    accuracy                           0.84       465
   macro avg       0.64      0.58      0.59       465
weighted avg       0.81      0.84      0.82       465


 Confusion Matrix:
[[377  21]
 [ 53  14]]
```



2nd iteration – Under Sampling

- Used under sampling on the training data set to deal with imbalanced data.

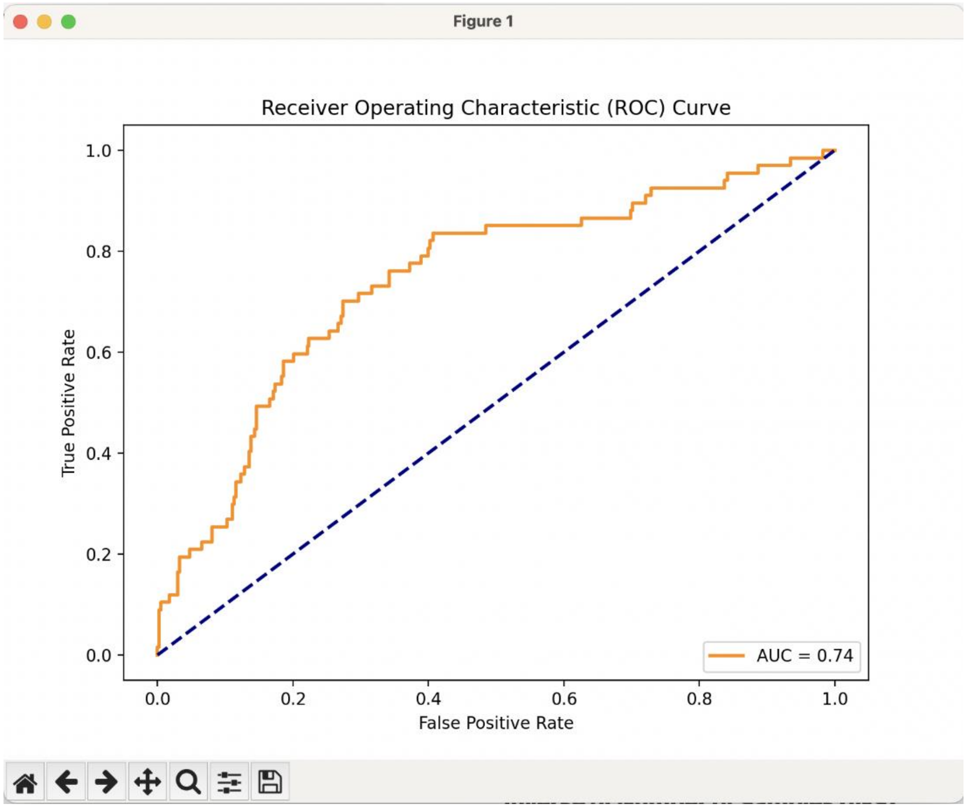
```
 Accuracy: 0.74

 Classification metrics
              precision    recall  f1-score   support

           0       0.92      0.76      0.84       398
           1       0.31      0.63      0.41        67

    accuracy                           0.74       465
   macro avg       0.62      0.70      0.63       465
weighted avg       0.84      0.74      0.78       465


 Confusion Matrix:
[[304  94]
 [ 25  42]]
```



3rd iteration – Over Sampling

- Used over sampling on the training data set as undersampling did not show a significant increase in our metrics.

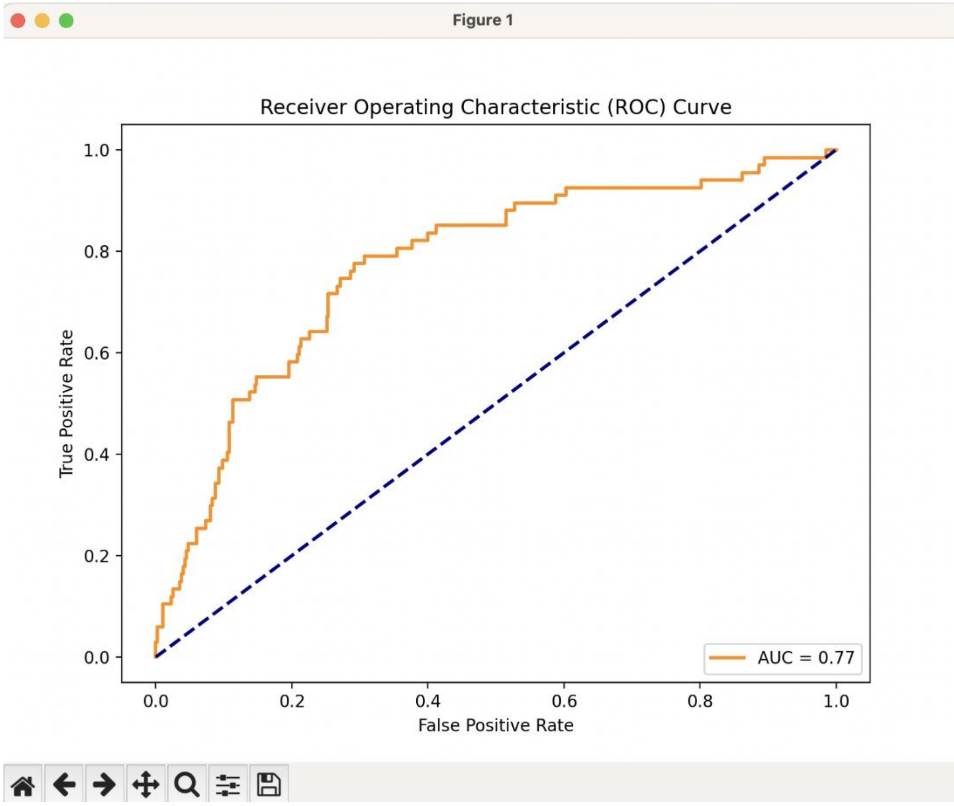
```
 Accuracy: 0.75

 Classification metrics
              precision    recall  f1-score   support

           0       0.93      0.76      0.84       398
           1       0.31      0.64      0.42        67

    accuracy                           0.75       465
   macro avg       0.62      0.70      0.63       465
weighted avg       0.84      0.75      0.78       465


 Confusion Matrix:
[[304  94]
 [ 24  43]]
```



4rd iteration – SVMSMOTE Over Sampling

- Used SVMSMOTE over sampling on the training data set, because we did see an improvement in our metrics using oversampling. After trying out different oversampling techniques including SMOTE, we found that SVMSMOTE works best with our dataset to deal with imbalanced data.

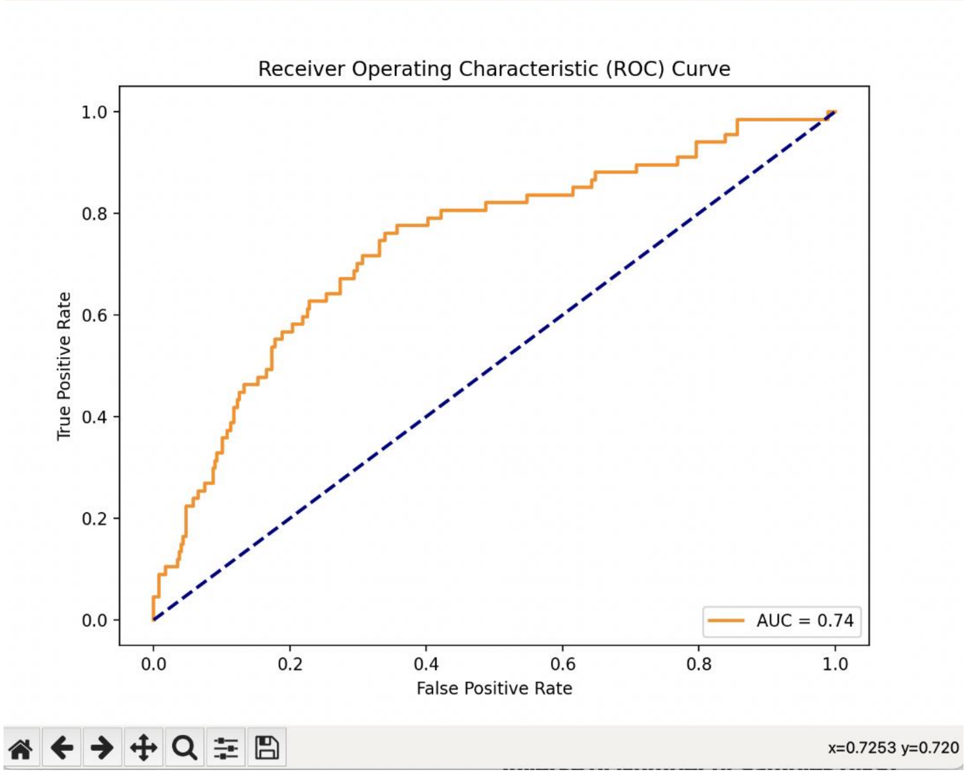
```
 Accuracy: 0.79

 Classification metrics
              precision    recall  f1-score   support

           0       0.91      0.84      0.87       398
           1       0.34      0.48      0.40        67

    accuracy                           0.79       465
   macro avg       0.62      0.66      0.64       465
weighted avg       0.82      0.79      0.81       465


 Confusion Matrix:
[[336  62]
 [ 35  32]]
```



5th iteration

- Used “lbfgs” solver in Logistic Regression
- Added “L2” regularization to deal with overfitting in our training set
- Tweaked max iterations so Logistic Regression has an easy time converin

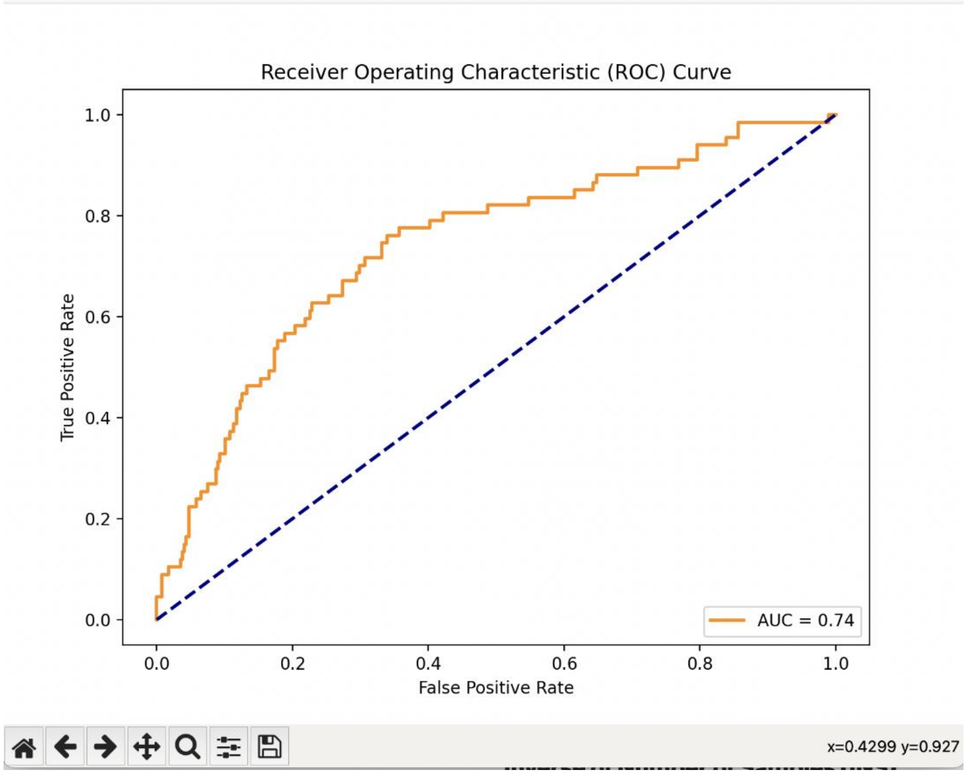
```
 Accuracy: 0.79

 Classification metrics
              precision    recall  f1-score   support

           0       0.91      0.84      0.87       398
           1       0.34      0.48      0.40        67

    accuracy                           0.79       465
   macro avg       0.62      0.66      0.64       465
weighted avg       0.82      0.79      0.81       465


 Confusion Matrix:
[[336  62]
 [ 35  32]]
```



Cross Validation

Grid Search Cross validation

```
In [ ]: best_params = grid_search.best_params_
```

```
In [ ]: best_params
Out[60]: {'C': 0.01, 'penalty': 'l2'}
```

K-fold Cross validation with

```
mean accuracy: 0.86
Standard deviation: 0.00
```

Final findings

- The data has high imbalance, and the best metrics are achieved by using SMVSMOTE oversampling of the minority class with 'lbfgs' logistic regression solver.
- Choosing RandomUnderSampling or RandomOverSampling techniques are not useful in finding a balance between recall and precision for the minority class. When the model is built using one or a combination of these methods the metrics report a high increase in either Precision or Recall, hence an imbalance.
- Cannot group Region and Neighborhood column to reduce the number of features because the data is flawed in this context. Running the code to group Region features based on the 3 categories in Neighborhood column resulted in all the 3 new grouped categories to have the same Region column elements, stating that every unique Region category is in all 3 unique Neighborhood Categories.
- Model performs better by scaling the test and training datasets using StandardScaler. Using MinMaxScaler results in worse metrics.
- Combining the columns that have auto correlation results in a reduction of features and deals with Collinearity, resulting in better metrics.