GR5293 Project 2

Classification Methods for Predicting Telco Subscription Customer Churn

Team #7
Daoyang E (de2418)
Jiazhen Lin (jl5990)
Xinru Hao (xh2500)
Wentao Chen (wc2768)

Background & Data Overview

- **Data Set Information:**

  This dataset comes from a customer retention program. The program found out that the cost of retaining an existing customer is far less than acquiring a new one.

- **Goals of Works:**

  Our goal is to build a classifier to predict if a customer will leave or remain with existing data. We will try multiple methods of classification and choose the best classifier.

- **Data Set Characteristics :**

  The raw data contains 7043 rows (customers) and 21 columns (features). Each row represents a customer, each column contains customer's attributes.

- **Response:**

  Customers who left within the last month – the column is called Churn.

- **Features:**

  Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies

  Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges

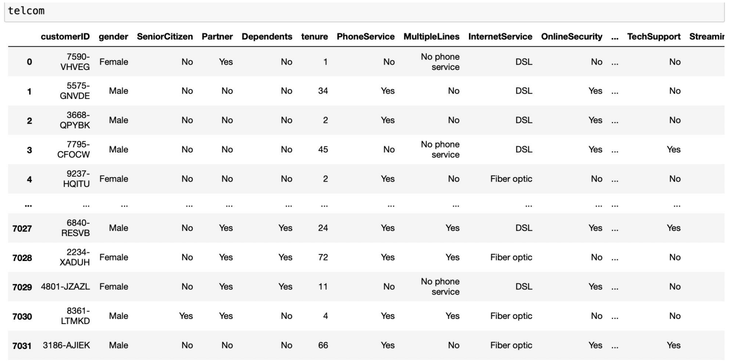
  Demographic info about customers – gender, age

```
Features :
 ['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure', 'PhoneService', 'MultipleLines', 'Int
ernetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies
', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn']
```

Data Manipulation, Processing & Analysis

Data Manipulation

- Replace spaces with null in total charges column
- Drop null values from total charges column
- Reframe the index
- Convert TotalCharges column to float type
- Replace 'No internet service' to No for the following columns
- Replace 1 and 0 in SeniorCitizen column with yes and no
- Change Tenure Column into Categorical Variable
- Separate churn and non churn customers
- Separate categorical and numerical columns

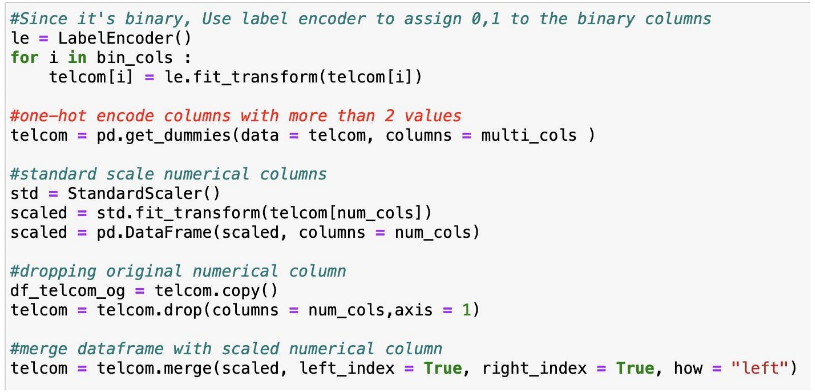


```
telcom
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	TechSupport	Streamir
0	7590-VHVEG	Female	No	Yes	No	1	No	No phone service	DSL	No	...	No	
1	5575-GNVDE	Male	No	No	No	34	Yes	No	DSL	Yes	...	No	
2	3668-QPYBK	Male	No	No	No	2	Yes	No	DSL	Yes	...	No	
3	7795-CFOCW	Male	No	No	No	45	No	No phone service	DSL	Yes	...	Yes	
4	9237-HQITU	Female	No	No	No	2	Yes	No	Fiber optic	No	...	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	
7027	6840-RESVB	Male	No	Yes	Yes	24	Yes	Yes	DSL	Yes	...	Yes	
7028	2234-XADUH	Female	No	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	No	
7029	4801-JZAZL	Female	No	Yes	Yes	11	No	No phone service	DSL	Yes	...	No	
7030	8361-LTMKD	Male	Yes	Yes	No	4	Yes	Yes	Fiber optic	No	...	No	
7031	3186-AJIEK	Male	No	No	No	66	Yes	No	Fiber optic	Yes	...	Yes	

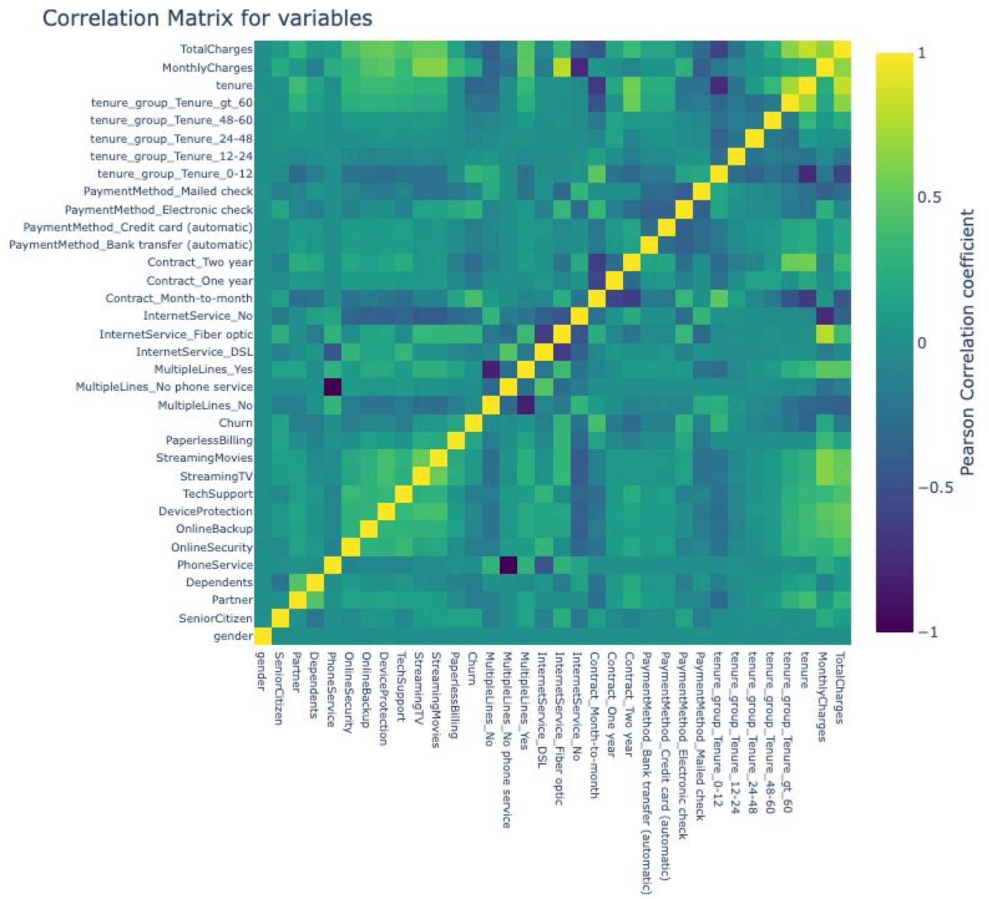
Data Preprocessing

- Use label encoder to assign 0,1 to the binary columns
- Get one-hot encode columns with more than 2 values
- Standard scale numerical columns
- Drop original numerical column
- Merge dataframe with scaled numerical column



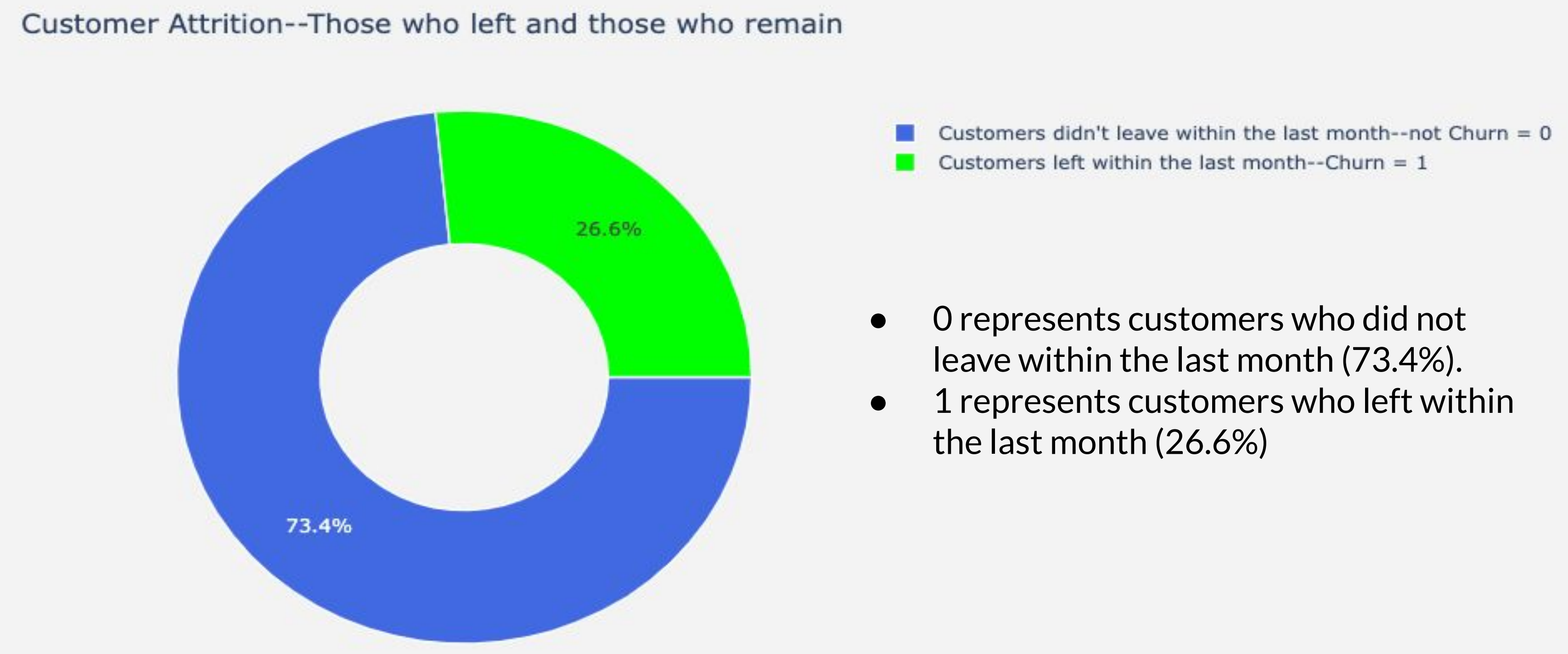
```
#Since it's binary, Use label encoder to assign 0,1 to the binary columns
le = LabelEncoder()
for i in bin_cols :
    telcom[i] = le.fit_transform(telcom[i])

#one-hot encode columns with more than 2 values
telcom = pd.get_dummies(data = telcom, columns = multi_cols )

#standard scale numerical columns
std = StandardScaler()
scaled = std.fit_transform(telcom[num_cols])
scaled = pd.DataFrame(scaled, columns = num_cols)

#dropping original numerical column
df_telcom_og = telcom.copy()
telcom = telcom.drop(columns = num_cols,axis = 1)

#merge dataframe with scaled numerical column
telcom = telcom.merge(scaled, left_index = True, right_index = True, how = "left")
```

Correlation Matrix



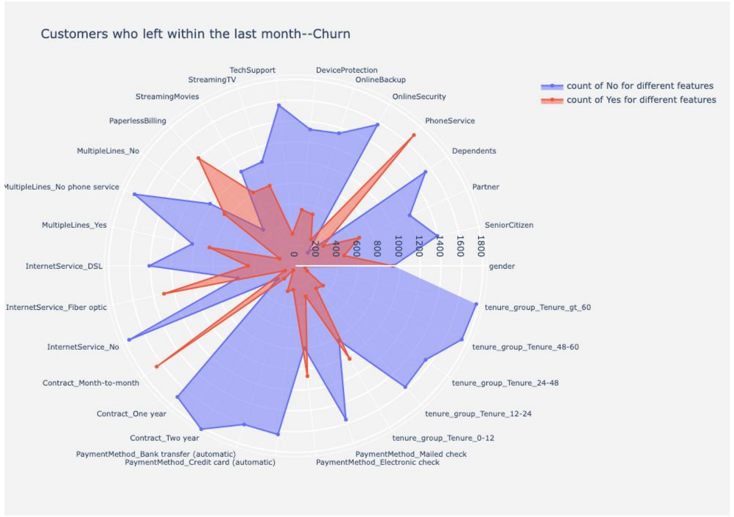
- tenure & totalcharges, monthly charge & internet service fiber optic seem to propose a rather strong linear relationship

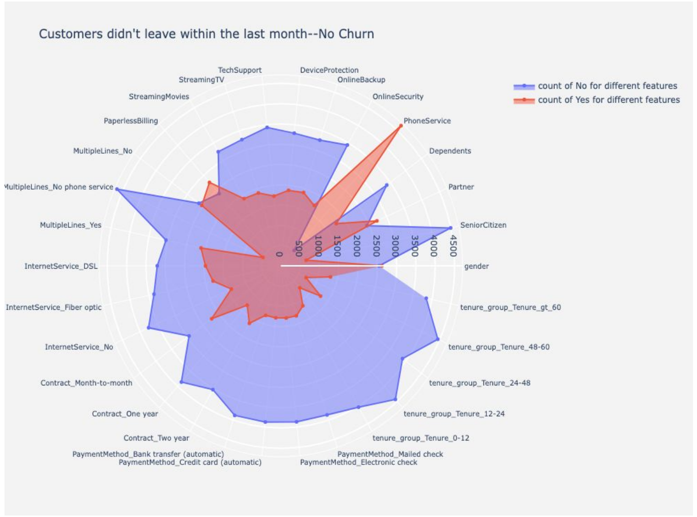
Customer Attrition Plot



- 0 represents customers who did not leave within the last month (73.4%).
- 1 represents customers who left within the last month (26.6%)

Binary Variable Distribution





- The graph shows the comparison of yes and no for customers who left within the last month.

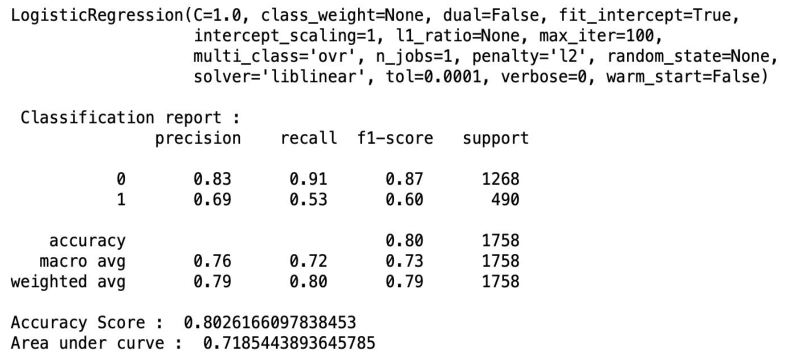
  (To be noticed, 1 and 0 not stands for yes and no for 'gender' here.)

- The graph shows the yes and no count for customers who didn't leave within last month.
- Comparing with the last graph, we can clearly see the difference.
- People who leave generally have month to month contract and use fiber optic internet service.
- They also appear to have tenure fall in the group of 0-12 and tend to use electronic payment methods.

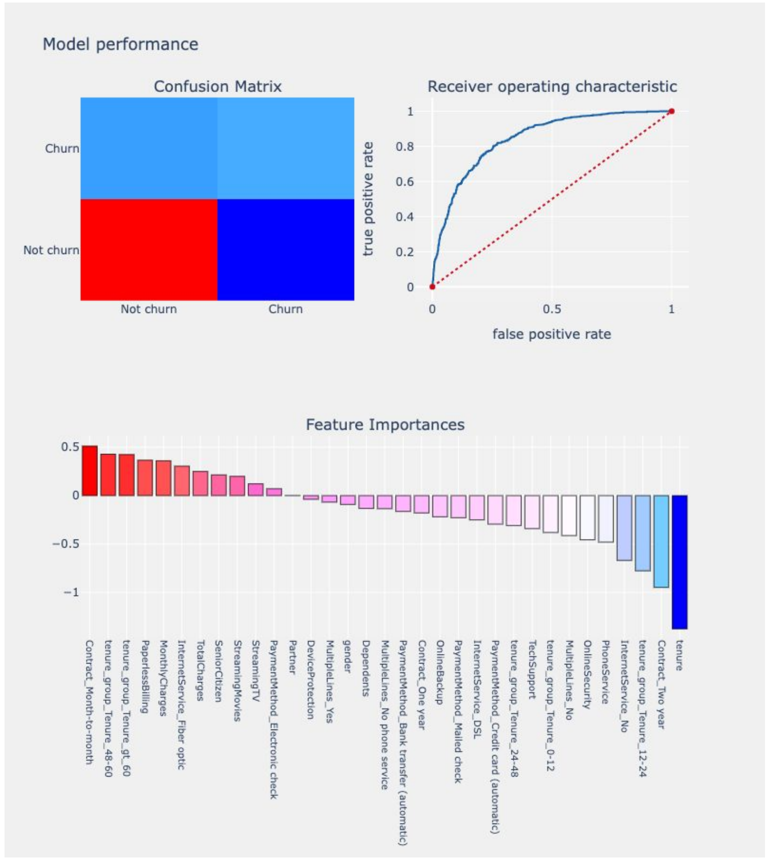
Modeling

- Logistic regression
- KNN Classifier
- Random Forest Classifier

Method 1: Logistic Regression



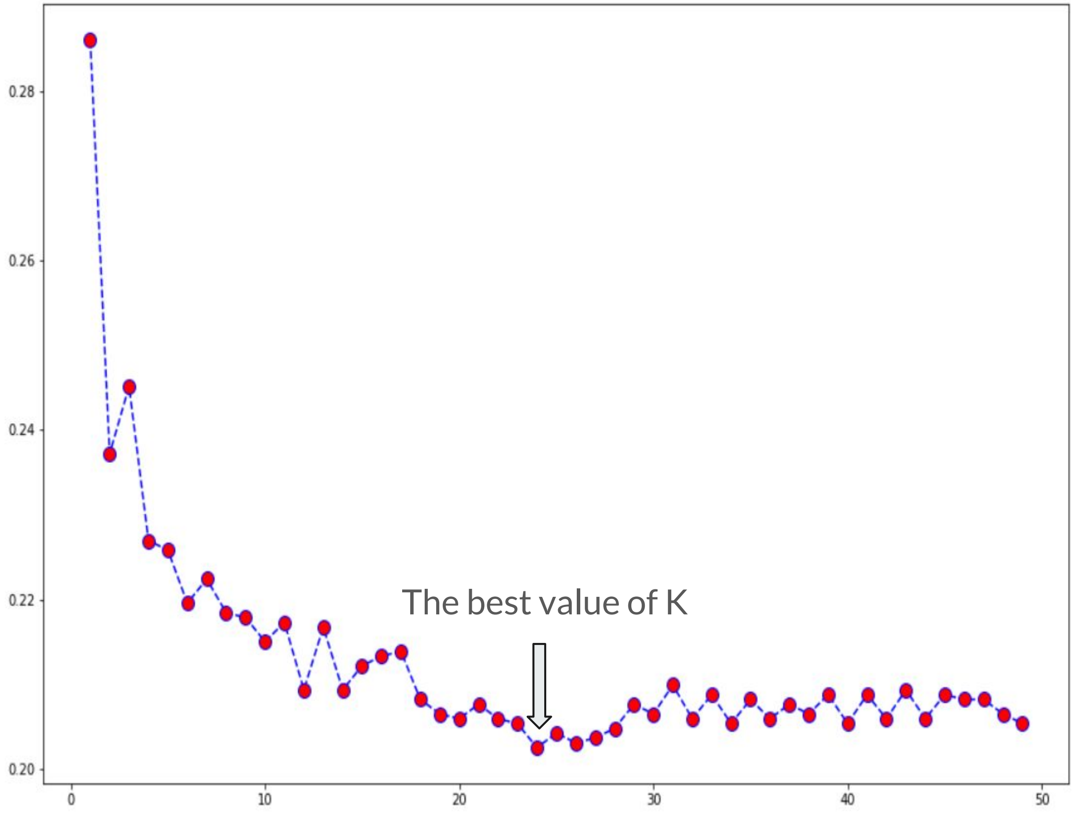
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='ovr', n_jobs=1, penalty='l2', random_state=None,
                   solver='liblinear', tol=0.0001, verbose=0, warm_start=False)

 Classification report :
               precision    recall  f1-score   support

           0       0.83      0.91      0.87      1268
           1       0.69      0.53      0.60       490

    accuracy                           0.80      1758
   macro avg       0.76      0.72      0.73      1758
weighted avg       0.79      0.80      0.79      1758

Accuracy Score :  0.8026166097838453
Area under curve :  0.7185443893645785
```



- Logistic regression models the probabilities for classification problems with two possible outcomes. It's an extension of the linear regression model for classification problems.
- By feature importance, we can see the relative importance of each feature when making the prediction in logistic regression.
- Logistic Regression shows that 'tenure', 'Contract_Two year', and tenure_group_Tenure_12-24', 'InternetService_No', 'PhoneService', and 'OnlineSecurity' are the most important features for predicting the churn decision, which meets our expectations.

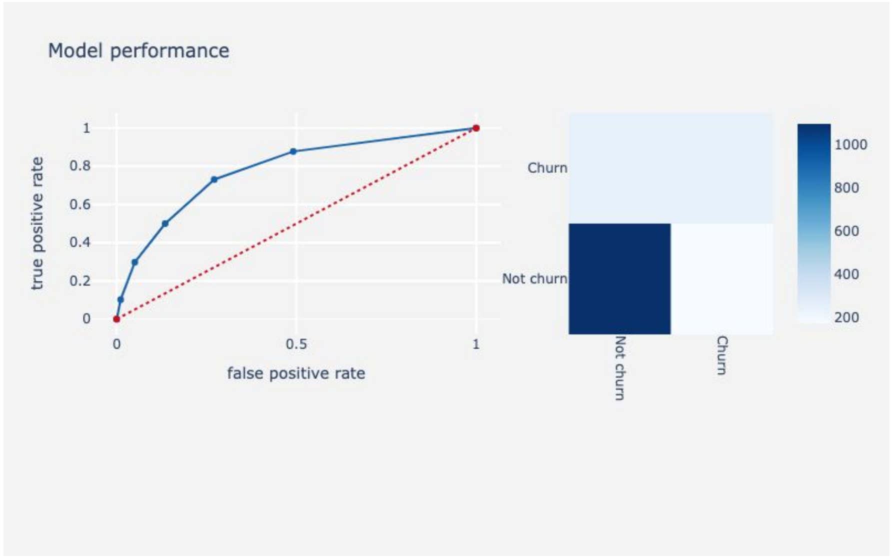
Method 2: KNN Classifier

- The main concept for k-NN depends on calculating the distances between the tested, and the training data samples in order to identify its nearest neighbours. The tested sample is then simply assigned to the class of its nearest neighbour.
- In order to find the best optimal value of K in KNN, we plot a graph to help us identify the best k value to minimize the error rate in range of 0 to 50. Then we find the best value for K is 24.



Method 2: KNN Classifier

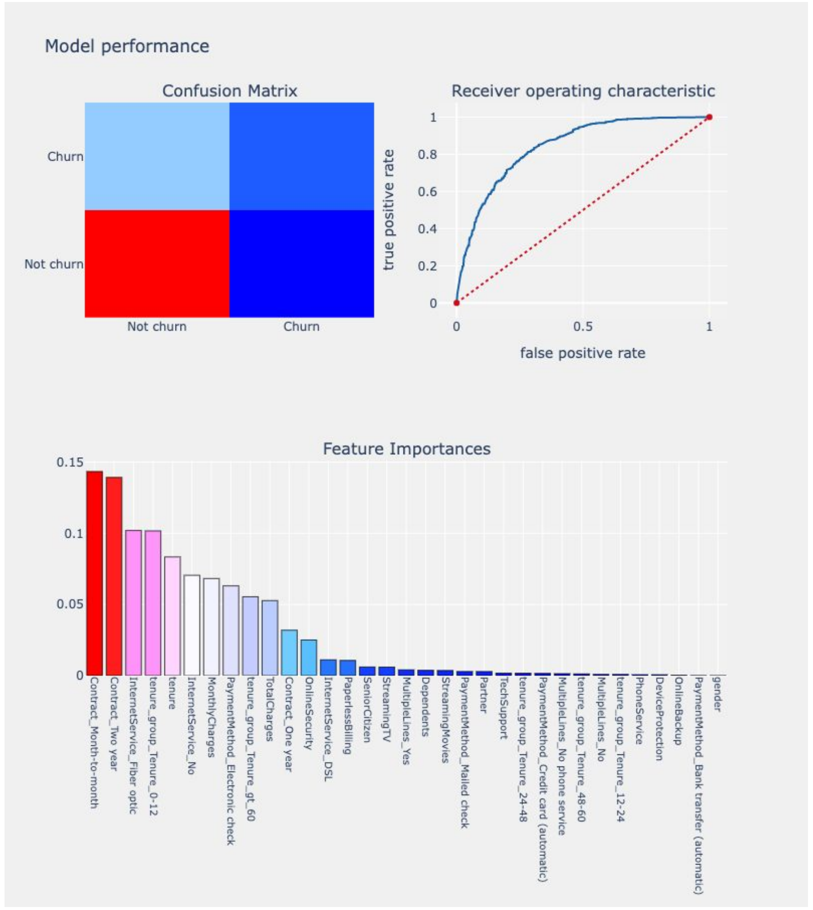
```
K-Nearest Neighbors Classifier

 Classification report :
               precision    recall  f1-score   support

           0       0.82      0.87      0.84      1268
           1       0.59      0.50      0.54       490

    accuracy                           0.76      1758
   macro avg       0.70      0.68      0.69      1758
weighted avg       0.75      0.76      0.76      1758

Accuracy Score :  0.7633674630261661
Area under curve :  0.682570977917981
```



Method 3: Random Forest Classifier

```
RandomForestClassifier(criterion='entropy', max_depth=3)

 Classification report :
               precision    recall  f1-score   support

           0       0.76      0.97      0.86      1268
           1       0.76      0.22      0.34       490

    accuracy                           0.76      1758
   macro avg       0.76      0.59      0.60      1758
weighted avg       0.76      0.76      0.71      1758

Accuracy Score :  0.7622298065984073
Area under curve :  0.5947563252430309
```

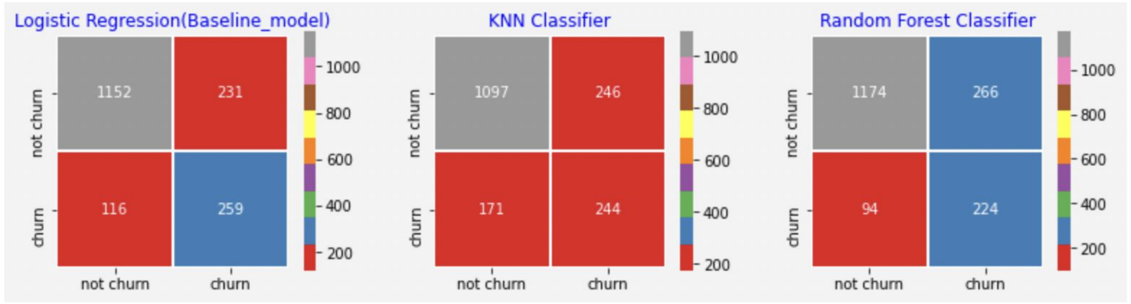
- Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time.
- Random Forest Classifier tells us ‘Contract_Month-to-month’, ‘Contract_Two year’, and ‘tenure’ are the three most important features for predicting the churn decision.



Model Performance Comparison

- Confusion matrix
- ROC curve
- Precision-recall curve
- Comparison of Model Performance output

Confusion Matrix

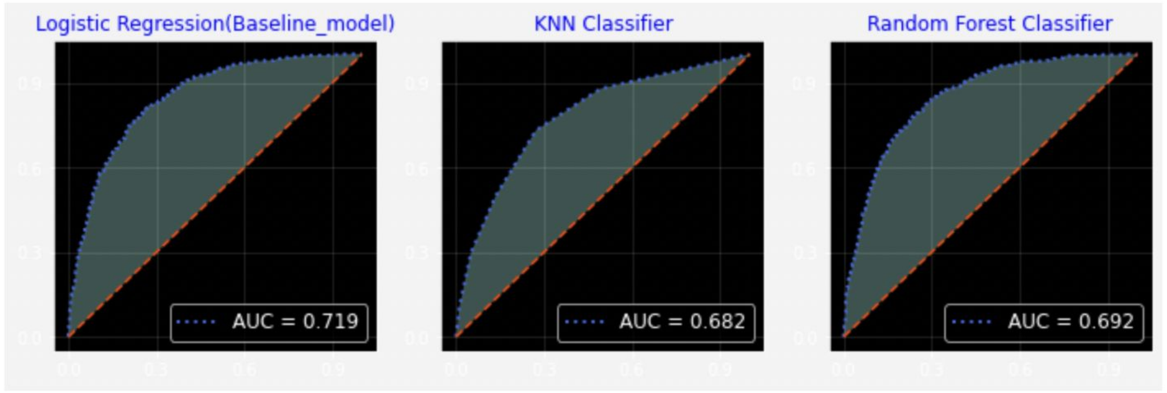


Model	Accuracy_score
Logistic Regression	0.8026
KNN Classifier	0.7634
Random Forest Classifier	0.7952

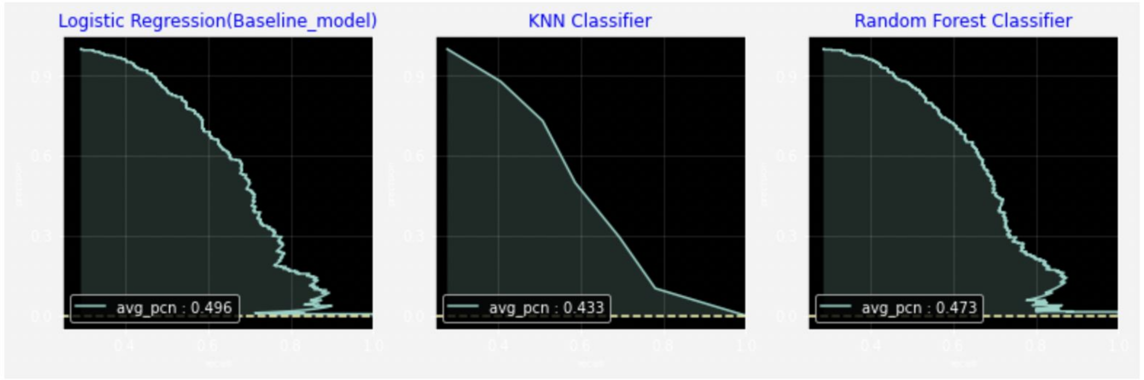
- The Confusion Matrix is the summary of prediction results on a classification problem.
- The total actual 'not churn' in the dataset is the sum of the values on the 'not churn' column.The total actual 'churn' in the dataset is the sum of values in the 'churn' column.
- The correct values are organized in a diagonal line from top left to bottom-right of the matrix.
- Here the logistic regression has the most correct predicting values generally. Its high accuracy score of 0.8026 means there is a high chance for the classifier to be correct.
- Random forest classifier also has a high accuracy score, and its error for predicting 'not churn' is the fewest, while its performance of predicting 'churn' is worse than other classifiers.
- KNN Classifier has relatively poor performance than logistic regression and random forest classifier.
- For each model, more errors were made by predicting 'churn' as 'not churn' than predicting 'not churn' as 'churn'.

ROC Curve

- ROC curve shows the trade-off between TPR and FPR, and the area under the curve (AUC) values for each model.
- The AUC value of the logistic regression is the largest one, whose curve is the closest to the top-left corner, which mean that there is a high chance that the classifier will be able to distinguish the positive class values from the negative class values.
- Therefore, ROC Curve also shows the logistic regression model has a better performance than KNN Classifier and Random Forest Classifier.



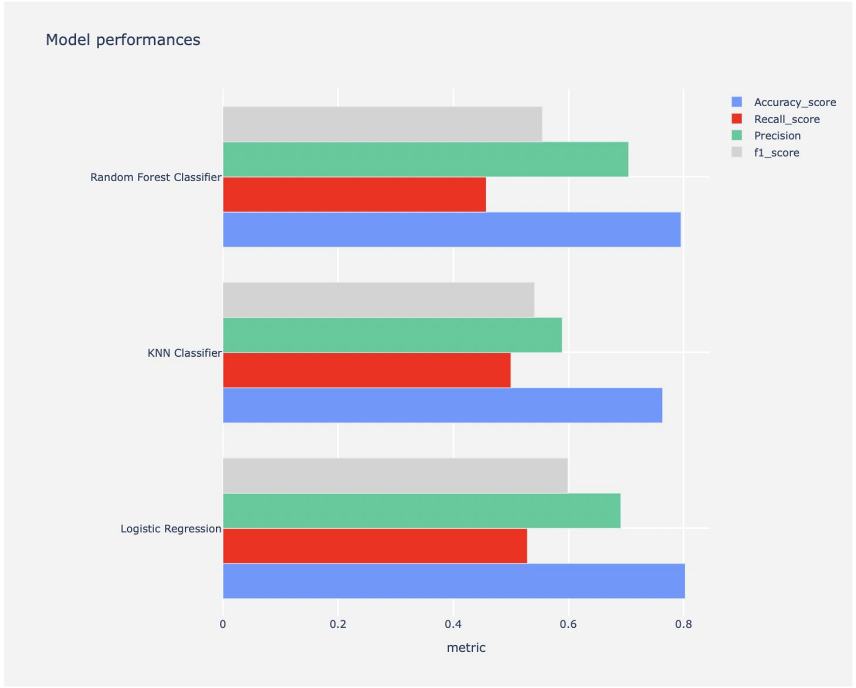
Precision-Recall Curve



- A precision-recall curve shows the relationship between precision (= positive predictive value) and recall (= sensitivity) for every possible cut-off.
- Average PCN of logistic regression is 0.496, which is the largest one, also suggests this model performs better than others.

Model Performance Output for Each Model

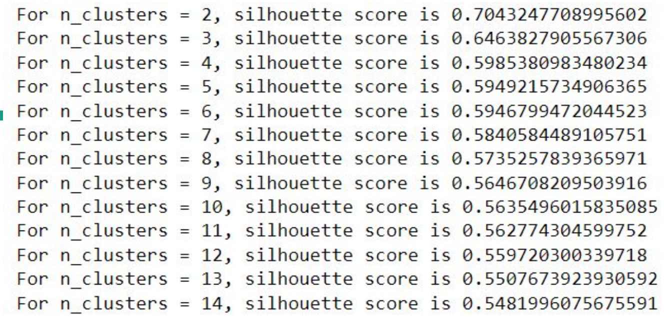
- Logistic regression has the highest accuracy score (80.26%), recall score (52.86%), f1 score (59.88%), and area under curve (71.85%).
- Overall, the performance of the **logistic regression** is better than KNN classifier and random forest classifier. It is the most suitable method of binary classification.



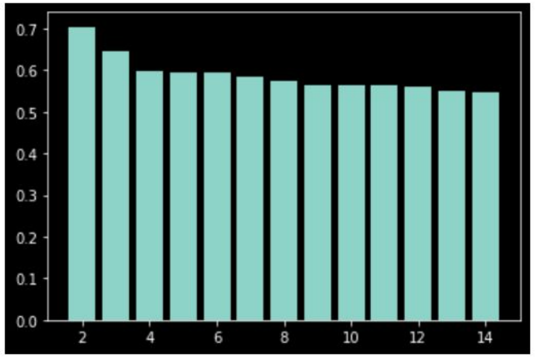
Model	Accuracy_score	Recall_score	Precision	f1_score	Area_under_curve
Logistic Regression	0.8026	0.5286	0.6907	0.5988	0.7185
KNN Classifier	0.7634	0.5	0.5889	0.5408	0.6826
Random Forest Classifier	0.7952	0.4571	0.7044	0.5545	0.6915

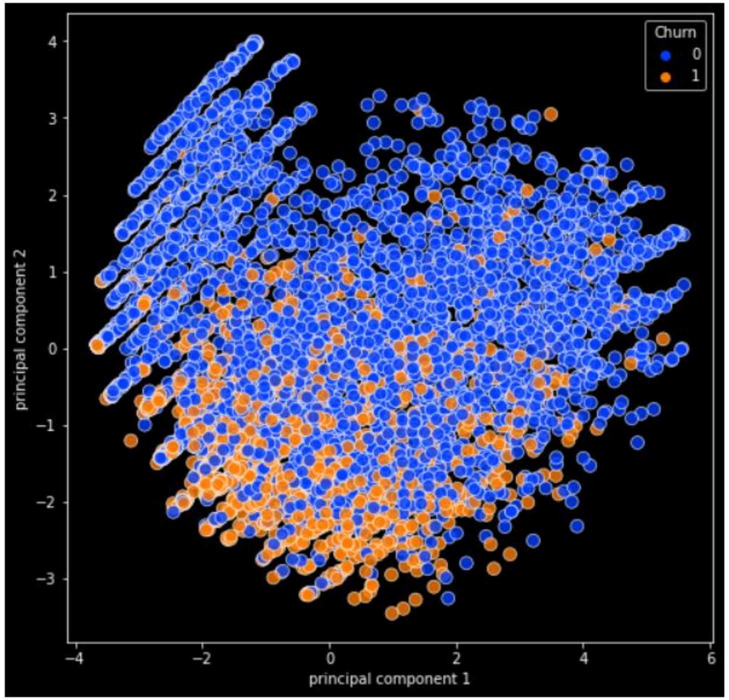
PCA & Clustering

PCA and Clustering



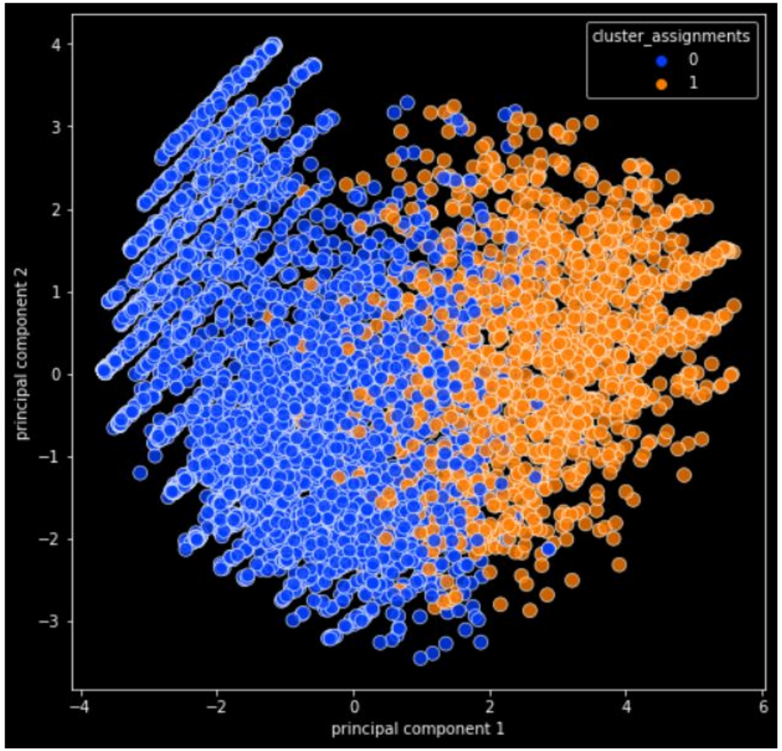
```
For n_clusters = 2, silhouette score is 0.7043247708995602
For n_clusters = 3, silhouette score is 0.6463827905567306
For n_clusters = 4, silhouette score is 0.5985380983480234
For n_clusters = 5, silhouette score is 0.5949215734906365
For n_clusters = 6, silhouette score is 0.5946799472044523
For n_clusters = 7, silhouette score is 0.5840584489105751
For n_clusters = 8, silhouette score is 0.5735257839365971
For n_clusters = 9, silhouette score is 0.5646708209503916
For n_clusters = 10, silhouette score is 0.5635496015835085
For n_clusters = 11, silhouette score is 0.562774304599752
For n_clusters = 12, silhouette score is 0.559720300339718
For n_clusters = 13, silhouette score is 0.5507673923930592
For n_clusters = 14, silhouette score is 0.5481996075675591
```

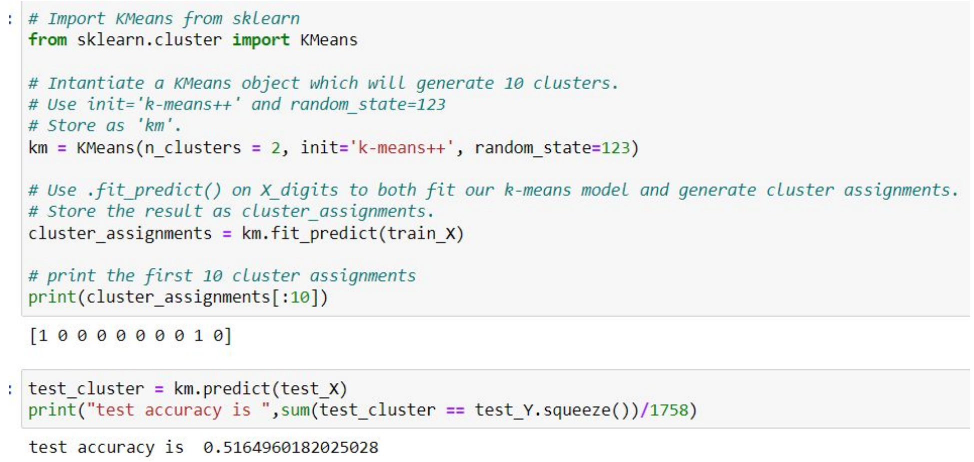




- We want to see if clustering will have better performance than classification.
- By looking at the silhouette score of each number of clusters on unPCAed data, we choose 2 clusters and apply the PCA and Kmeans clustering thereafter

PCA and Clustering





```
# Import KMeans from sklearn
from sklearn.cluster import KMeans

# Intantiate a KMeans object which will generate 10 clusters.
# Use init='k-means++' and random_state=123
# Store as 'km'.
km = KMeans(n_clusters = 2, init='k-means++', random_state=123)

# Use .fit_predict() on X_digits to both fit our k-means model and generate cluster assignments.
# Store the result as cluster_assignments.
cluster_assignments = km.fit_predict(train_X)

# print the first 10 cluster assignments
print(cluster_assignments[:10])
```

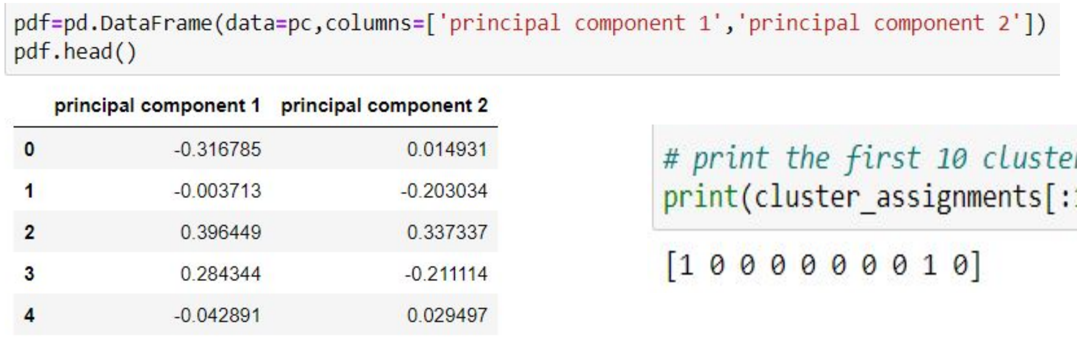
```
[1 0 0 0 0 0 0 0 1 0]
```

```
test_cluster = km.predict(test_X)
print("test accuracy is ",sum(test_cluster == test_Y.squeeze())/1758)
```

```
test accuracy is  0.5164960182025028
```

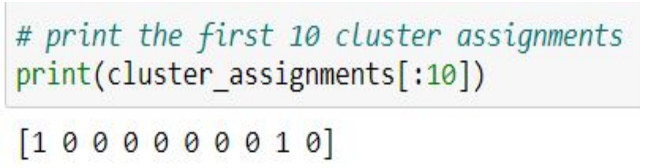
- Fit k-means model on the trained data after PCA.
- The test accuracy is 0.5165, which is relatively low. Thus, **k-means clustering based on PCA performs worse than the classification.**

Kernel PCA and Clustering



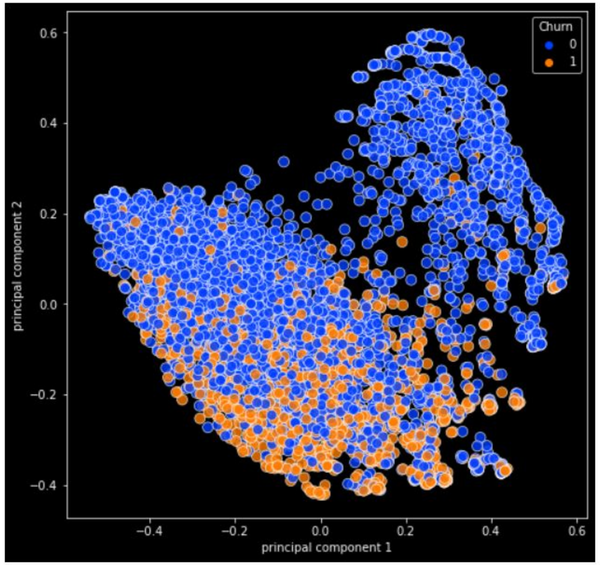
```
pdf=pd.DataFrame(data=pc,columns=['principal component 1','principal component 2'])
pdf.head()
```

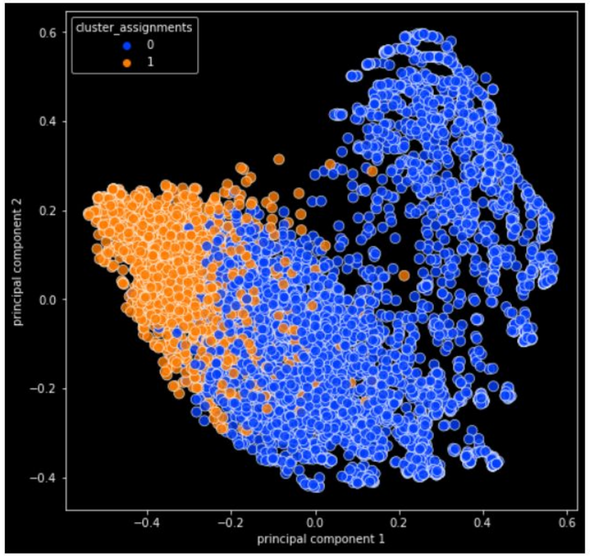
	principal component 1	principal component 2
0	-0.316785	0.014931
1	-0.003713	-0.203034
2	0.396449	0.337337
3	0.284344	-0.211114
4	-0.042891	0.029497



```
# print the first 10 cluster assignments
print(cluster_assignments[:10])
```

```
[1 0 0 0 0 0 0 0 1 0]
```





- We also tried Kernel PCA on data and show the original churn assignment vs cluster assignment image.
- It also shows that **clustering is not effective as the classification.**

Conclusion

- We first cleaned and manipulated our dataset and did data visualization for our dataset.
- Then we built the logistic regression, KNN classifier, and random forest classifier to predict if a customer will leave or remain with existing data.
- Confusion Matrix, ROC, and PRC curves show that the logistic regression has the best performance for our binary classification problem.
- We try to see if clustering will have better performance than classification, but the clustering has a poor performance for predicting our data.
- Hence, we choose logistic regression as our best classifier and get a high accuracy of 0.8026. It shows that 'tenure', 'Contract_Two year', and tenure_group_Tenure_12-24', 'InternetService_No', 'PhoneService', and 'OnlineSecurity' are the most important features for predicting the customer churn decision, which is consistent with our analysis at first.
- Therefore, our **suggestion** for the customer retention program is to **pay attention to the customer's tenure, two-year contract, Internet service number, and security to predict customer churn, and try precision marketing on those target existing customers and retain them. In this way, the cost would be lower than acquiring a new customer.**
- Further steps: we may build more models to compare with, and focus more on analysis of each one.

Reference

https://www.kaggle.com/datasets/blastchar/telco-customer-churn?resource=download

[IBM Sample Data Sets]

Thank you for listening!