

Predictive Modelling

Final Project: Telco Customer Churn

Presented By **Alamelu Ramanathan**

Problem Statement

Telecommunication companies face significant financial losses when customers discontinue their service (churn). It is crucial for telecom companies to predict customer churn early to implement strategies for customer retention and improve business performance.

Key Objectives

- **Predict Customer Churn:** Develop a machine learning model to predict which customers are likely to churn based on historical data.
- **Understand Key Drivers:** Identify the key factors (features) that are most indicative of churn (e.g., subscribed services, payment history, contract type).
- **Actionable Insights:** Provide recommendations for strategies that the company could implement to retain customers likely to churn.

Exploratory Data Analysis

Analyse the information in the Telco dataset , find the missing / null values if any and find ways to handle them. Dataset has 21 columns that could be grouped under the following categories.

- Customer demographics,
- Subscribed Services and
- Tenure And Payment information

TotalCharges has blank information for fewer records which on further analysis indicate that those rows belong to new customers which could be safely removed from the training data as it won't affect the model.

```
#2. Exploratory data analysis

#understand the dataset
print(ds.info())
print('\n Data dimension', ds.shape)

#check for the missing information
print('\n', ds.isna().sum())

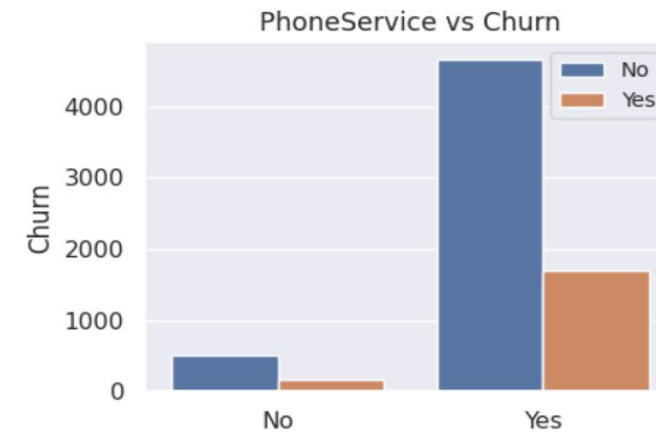
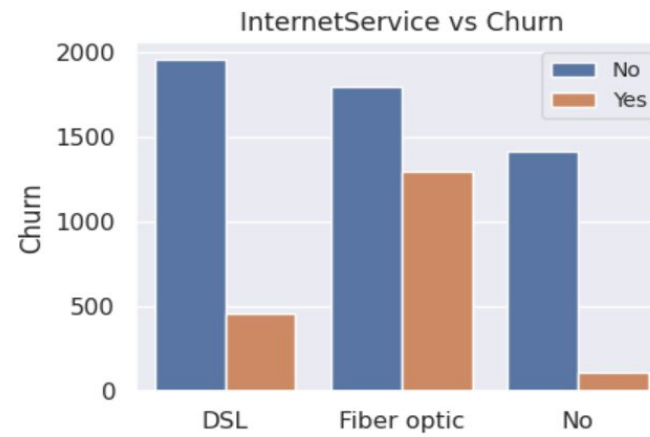
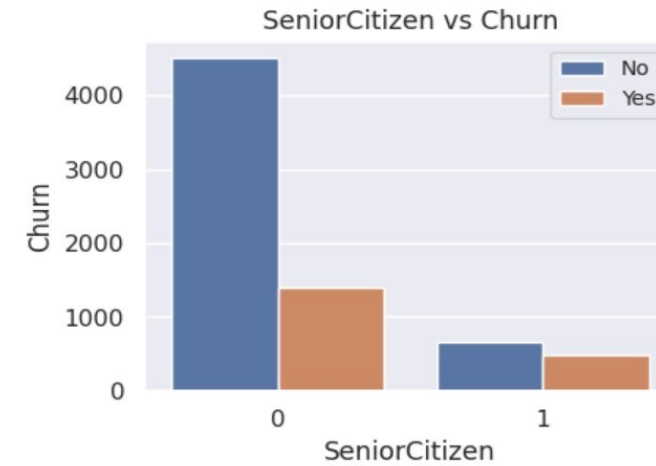
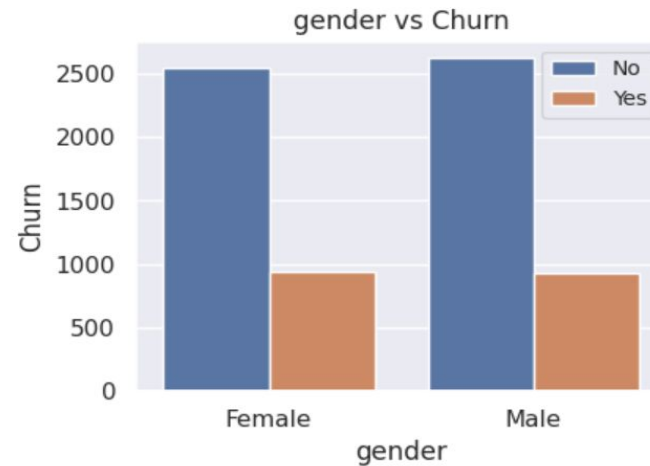
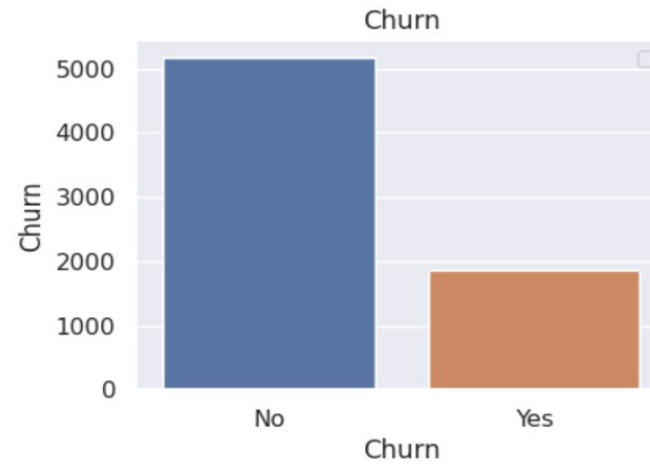
#check availability of data in total charges
column
missingTotalCharge =
ds.loc[ds['TotalCharges']== " "]
print(missingTotalCharge.shape)
print(missingTotalCharge[["tenure", "Contract",
"TotalCharges"]])

#Analyse the value in the target column "Churn"
ds['Churn'].value_counts().to_frame().T
```

Visual Insights

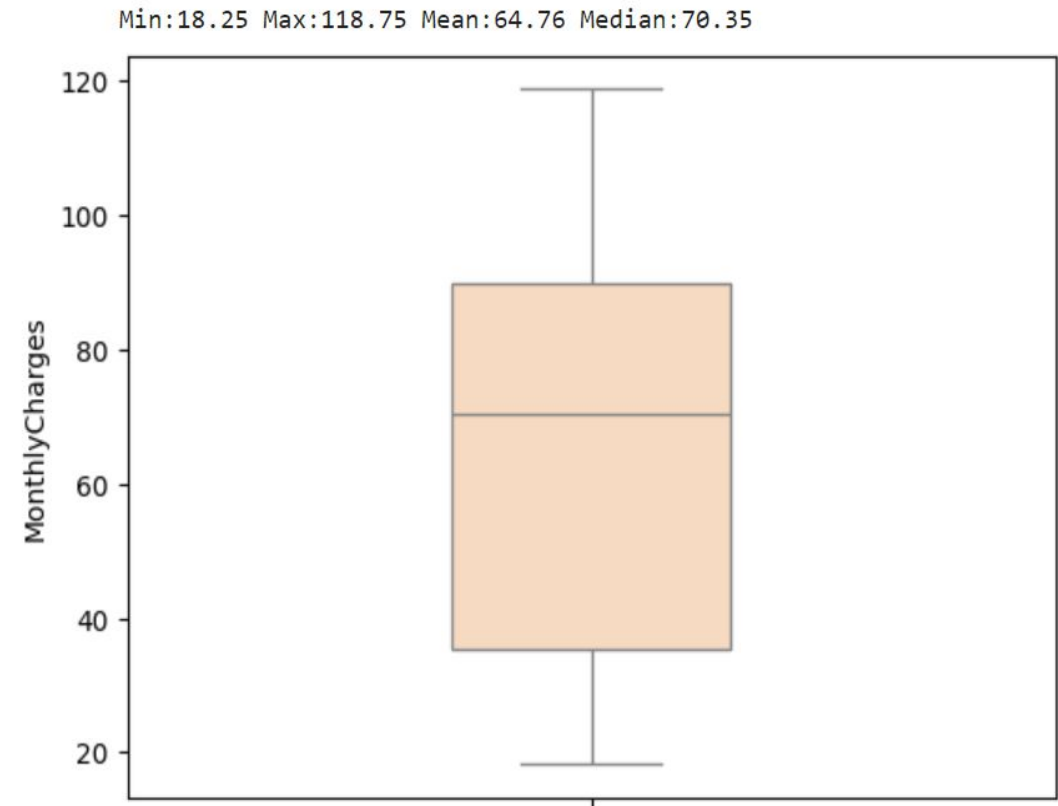
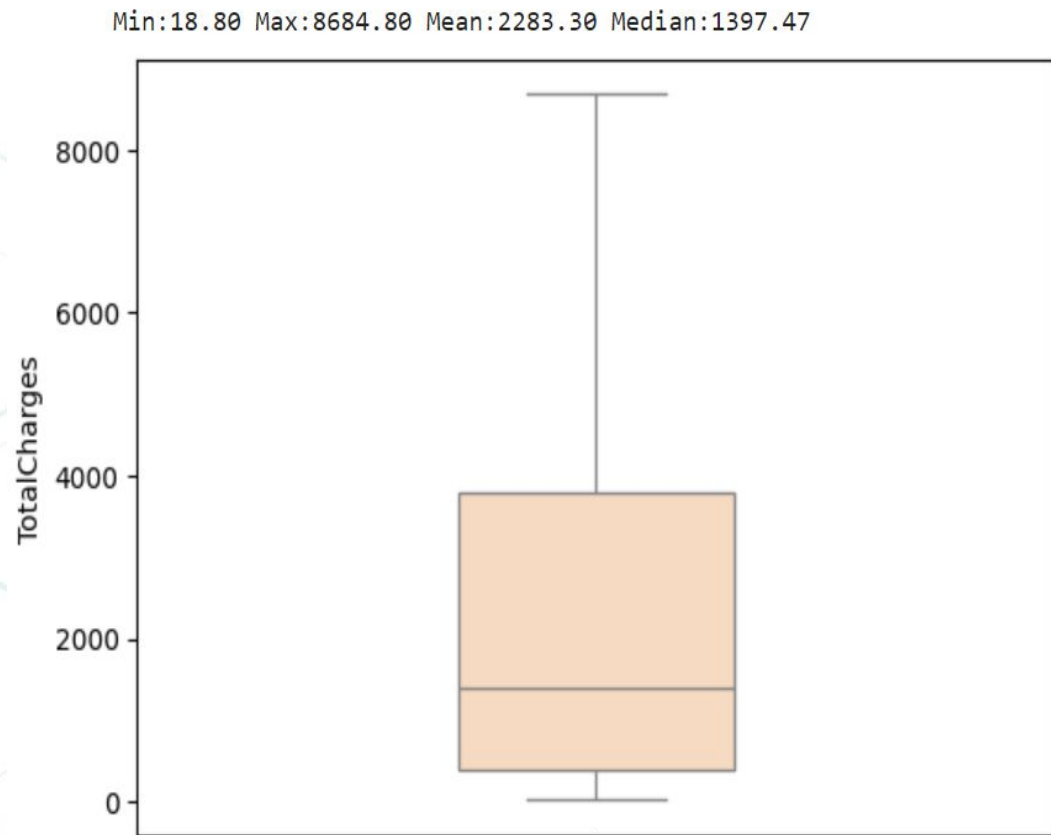
Analysing the features against the target feature 'Churn' gives insights like,

- Long term contract has fewer Churns compared to Month to month.
- InternetService option plays a vital role in the churn rate.
- Gender doesn't have much impact on the target.



Visual Insights on ordinal features

Box plot for the ordinal features to identify the outliers if any and no significant outliers found for total charges and monthly charges.

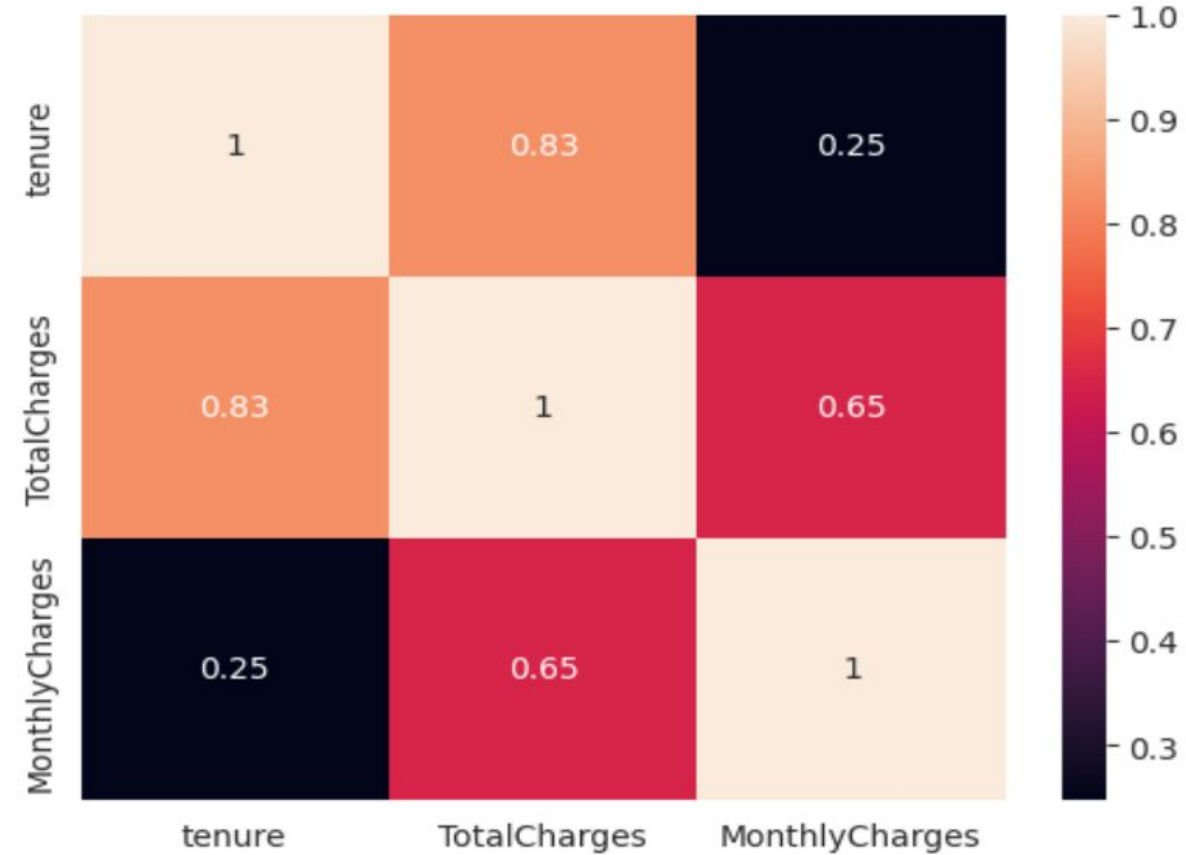


Visual Insights on ordinal features

Heatmap to find the correlation between ordinal features

- Monthly Charges,
- Total Charges and
- tenure.

Tenure and total charges seem to have a significant correlation.



Data Preprocessing

```
#This function prepares the data ready for supervised machine learning algorithms
def wrangle(ds):
    ds = ds[ds['TotalCharges']!=" "]
    #Convert the total charges column to numeric
    ds['TotalCharges'] = pd.to_numeric(ds['TotalCharges'])
    print(ds.dtypes["TotalCharges"])
    #delete the column customer ID
    ds.drop('customerID', axis =1, inplace = True)
    #combine the categorical value " No service " to "No"
    ds.loc[ds['MultipleLines']=="No phone service", 'MultipleLines'] = 'No'
    ds.loc[ds['OnlineSecurity']=="No internet service", 'OnlineSecurity'] = 'No'
    ds.loc[ds['OnlineBackup']=="No internet service", 'OnlineBackup'] = 'No'
    ds.loc[ds['DeviceProtection']=="No internet service", 'DeviceProtection'] = 'No'
    ds.loc[ds['TechSupport']=="No internet service", 'TechSupport'] = 'No'
    ds.loc[ds['StreamingTV']=="No internet service", 'StreamingTV'] = 'No'
    ds.loc[ds['StreamingMovies']=="No internet service", 'StreamingMovies'] = 'No'

    #Encode the categorical data
    label_encoder = preprocessing.LabelEncoder()
    ds['Churn']= label_encoder.fit_transform(ds['Churn'])
```


Model Selection

Classifier algorithms are well suited for predicting the categorical variable 'Churn'.

Following Machine learning algorithms for classification problems are evaluated,

- Logistic Regression,
- DecisionTree Classifier and
- Random forest Regression and
- Cross Validation to evaluate the models
- GridSearchCV for hyper parameter tuning.

Training and Test data, 2 ways

1. Used Train test split from sklearn library to prepare training data and test data

```
#4. Split the data to train and test
```

```
y= tuned_ds['Churn']
```

```
X= tuned_ds.drop(columns='Churn', axis =1)
```

```
X_train,X_test,Y_train, Y_test = train_test_split(X, y, test_size = 0.3,stratify=y, random_state= 42)  
print(X_train.shape,X_test.shape, Y_train.shape, Y_test.shape)
```

2. Cross validation technique to train the models with K Folds(5 Folds)

```
#6 cross validation with the same data with 5 Folds and models
```

```
for model in models:
```

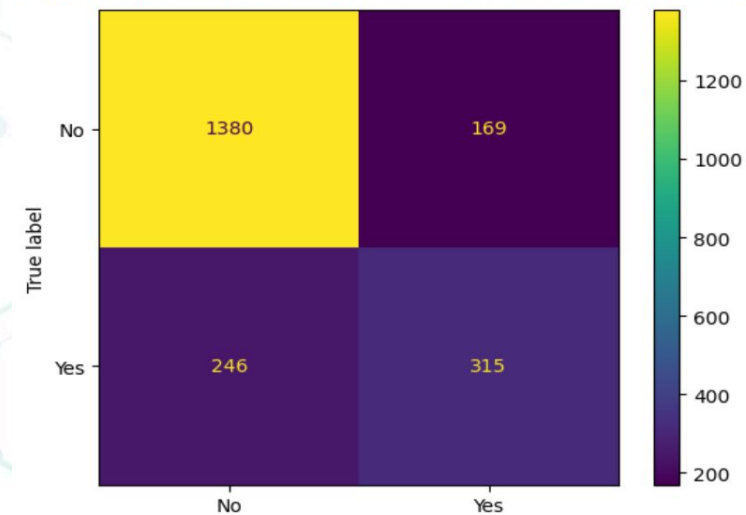
```
    cv_score = cross_val_score(model, X, y, cv= 5)
```

```
    print(cv_score)
```

Training and testing the models

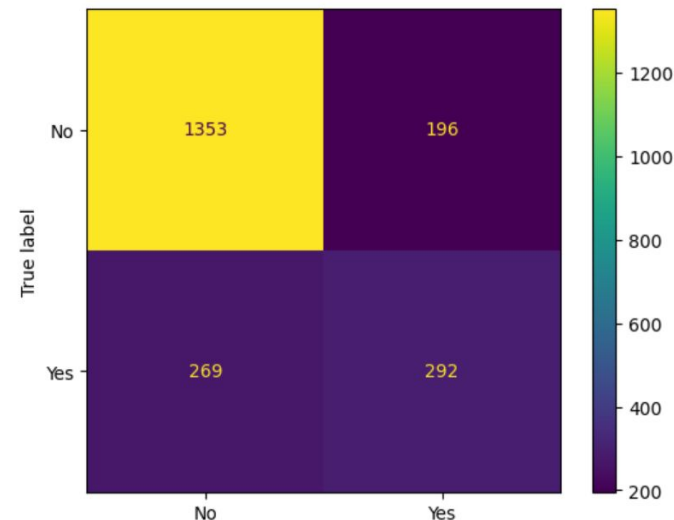
```
models=[LogisticRegression(solver='liblinear'), DecisionTreeClassifier(max_depth=maxDepth), RandomForestClassifier()]
plt.figure(figsize=(5,5), dpi=100)
for model in models :
    model.fit(X_train, Y_train)
    Y_pred= model.predict(X_test)
    Y_pred_proba=model.predict_proba(X_test)
```

LogisticRegression(solver='liblinear') with the accuracy of 80.33 %



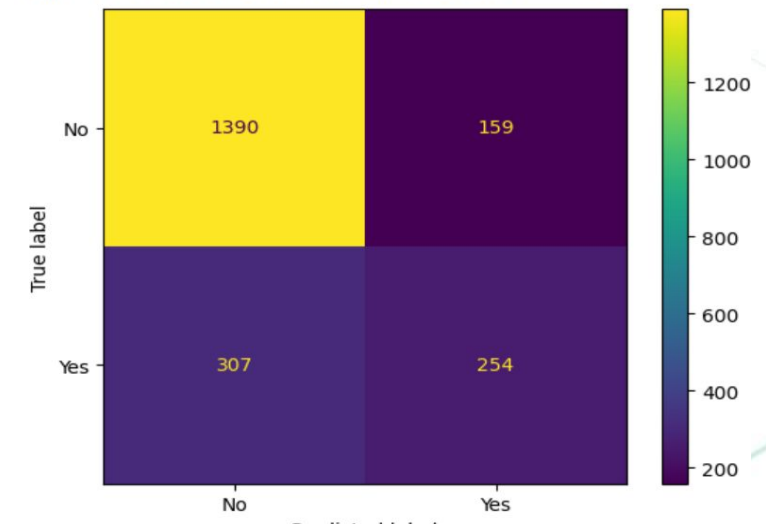
	precision	recall	f1-score	support
0	0.85	0.89	0.87	1549
1	0.65	0.56	0.60	561
accuracy			0.80	2110
macro avg	0.75	0.73	0.74	2110
weighted avg	0.80	0.80	0.80	2110

DecisionTreeClassifier(max_depth=7) with the accuracy of 77.96 %



	precision	recall	f1-score	support
0	0.83	0.87	0.85	1549
1	0.60	0.52	0.56	561
accuracy			0.78	2110
macro avg	0.72	0.70	0.71	2110
weighted avg	0.77	0.78	0.77	2110

RandomForestClassifier() with the accuracy of 77.91 %



	precision	recall	f1-score	support
0	0.82	0.90	0.86	1549
1	0.62	0.45	0.52	561
accuracy			0.78	2110
macro avg	0.72	0.68	0.69	2110
weighted avg	0.76	0.78	0.77	2110

Cross Validation with 5 Fold

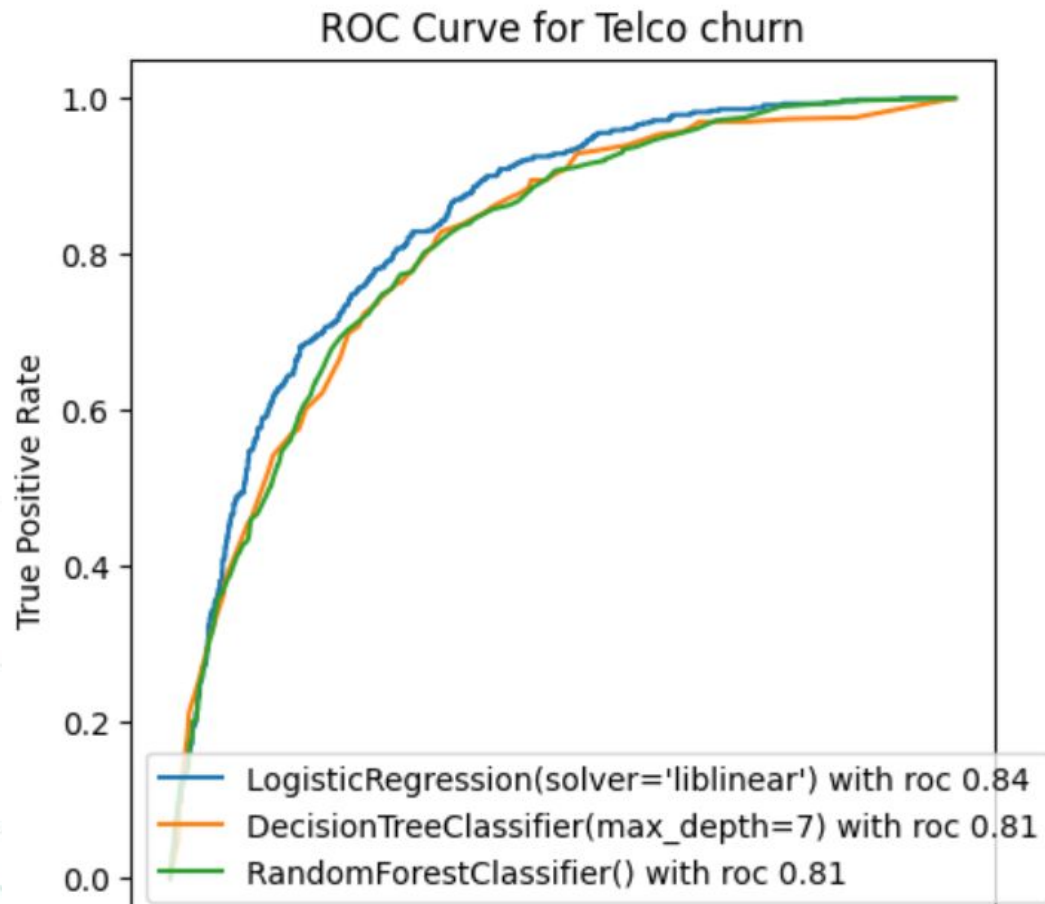
```
#6 cross validation with the same data with 5 Folds and models
for model in models:
    cv_score = cross_val_score(model, X, y, cv= 5)
    print(cv_score)
    accuracy = (sum(cv_score)/len(cv_score))*100
    print("The accuracy score of the model ", model, ' = ', round(accuracy,2))
```

```
⇒ [0.80597015 0.81307747 0.79587482 0.80654339 0.80369844]
The accuracy score of the model LogisticRegression(solver='liblinear') = 80.5
[0.78891258 0.79530917 0.77524893 0.78733997 0.7972973 ]
The accuracy score of the model DecisionTreeClassifier(max_depth=7) = 78.88
[0.7960199 0.78535892 0.76031294 0.79018492 0.79089616]
The accuracy score of the model RandomForestClassifier() = 78.46
```


Model Selection

Metrics from the logistic regression 'Accuracy', 'Recall' and 'AUC' is slightly higher than the other models.

Logistic Regression is best suited machine learning algorithm for this problem



Hyperparameter Tuning

Parameters like solver and the max_iter are tuned using GridSearchCV to get the highest accuracy possible.

Hyperparameter tuning

```
#Implement GridSearchCV
param = {
    'solver': ['lbfgs', 'newton-cg', 'liblinear', 'sag', 'saga'],
    'max_iter' : [100, 1000, 2000]}

model= LogisticRegression()
classifier = GridSearchCV(model, param, cv=5)
classifier.fit(X_train, Y_train)
print(classifier.best_params_)
print(f'{classifier.estimator} with {classifier.best_params_} gives the best accuracy : {classifier.best_score_*100:.2f}%')
```

```
{'max_iter': 100, 'solver': 'liblinear'}
LogisticRegression() with {'max_iter': 100, 'solver': 'liblinear'} gives the best accuracy : 80.03%
```

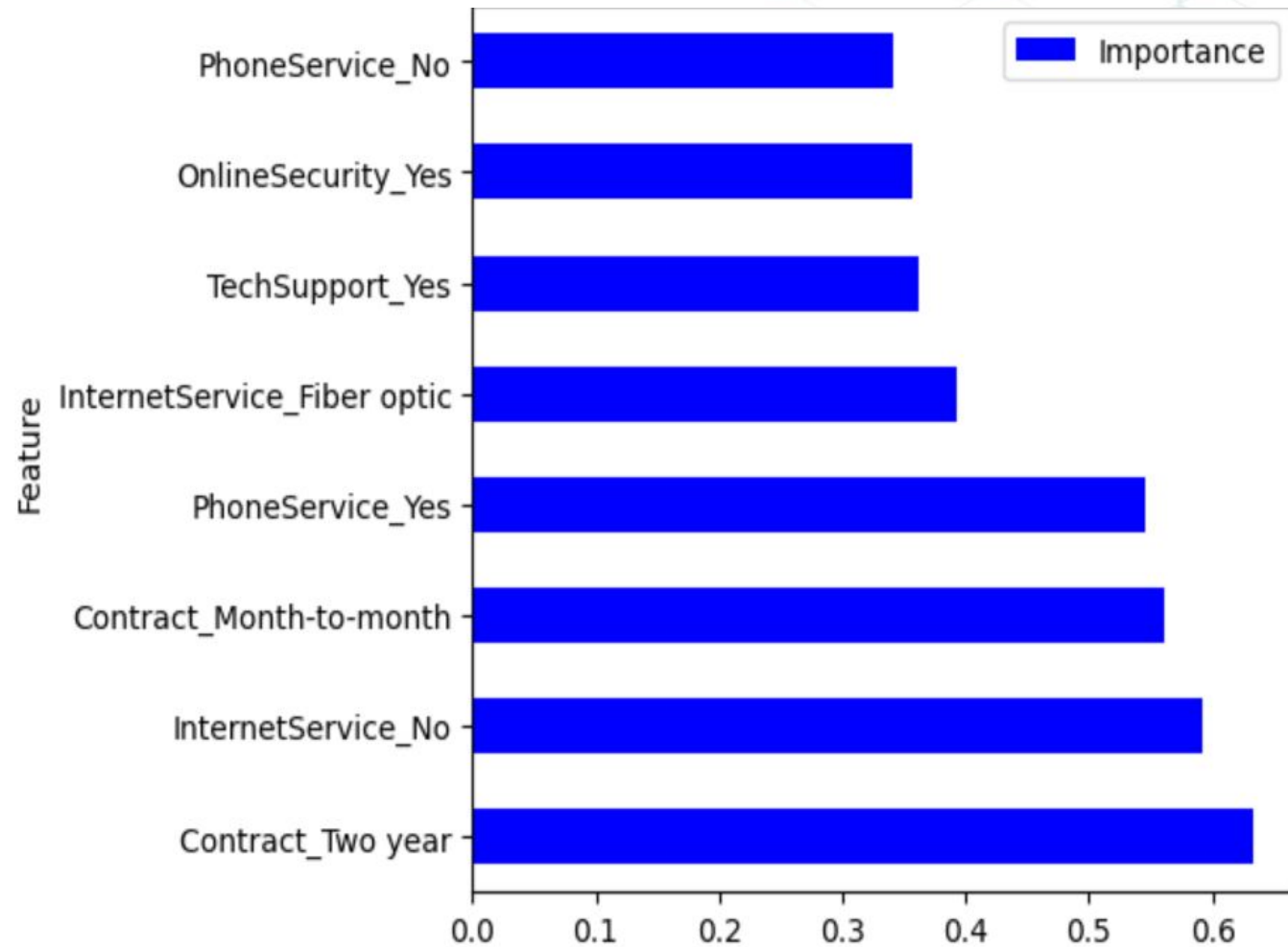

Feature Importance

Feature Importance

```
5] #plot feature importance from the selected model
model = LogisticRegression(solver='liblinear', max_iter=100)
model.fit(X_train, Y_train)

coefficients = model.coef_[0]

feature_importance = pd.DataFrame({'Feature': X.columns, 'Importance': np.abs(coefficients)})
feature_importance = feature_importance.sort_values('Importance', ascending=False)
feature_importance=feature_importance.head(8)
feature_importance.plot(x='Feature', y='Importance', kind='barh', figsize=(5, 5), color='blue')
```



Conclusion

- Machine learning algorithm, Logistic regression is well suited for this problem among the other classifier algorithms with 80% accuracy.
- Contract Type and Services play a significant role in churning of the customers.
- Enrolling the customers on the verge of Churning , the other long lasting customers and the new customers to the following is suggested for business improvement.
 - Retention reward campaign,
 - Loyalty reward program and
 - Onboarding discounts.

Alamelu Ramanathan
alamurm@gmail.com

Thank You!

Jayanth Rasamsetti

340S, Lemon Avenue, Walnut, California, USA 91789
4th Floor, T-Hub, IIIT-Hyderabad, Gachibowli, India



Microsoft
for Startups

jay@pixeltests.com

