PIXELTESTS

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.

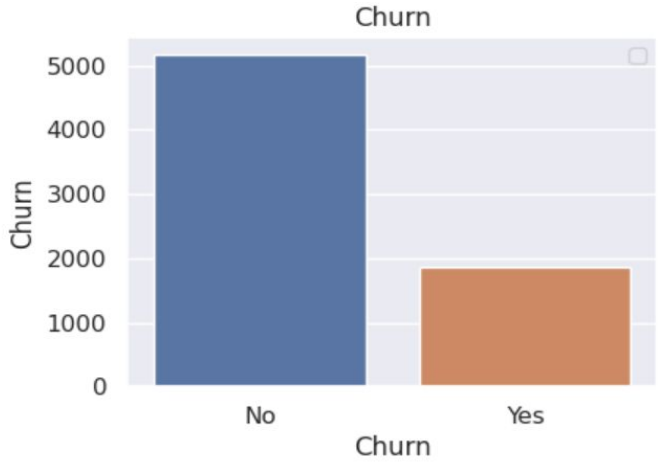
```
#2. Exploratory data analysis

#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
```

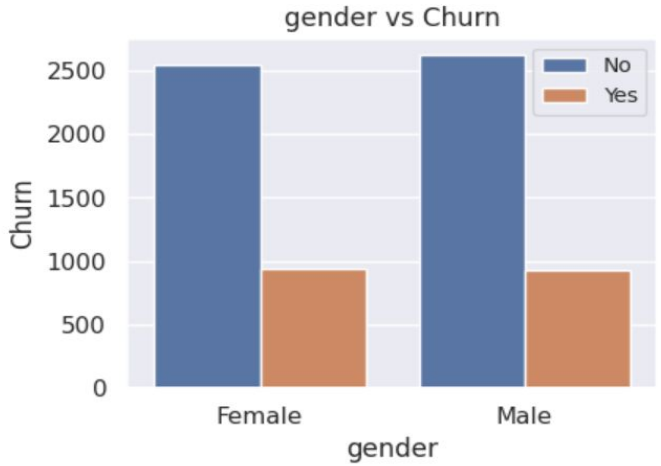
PIXELTESTS

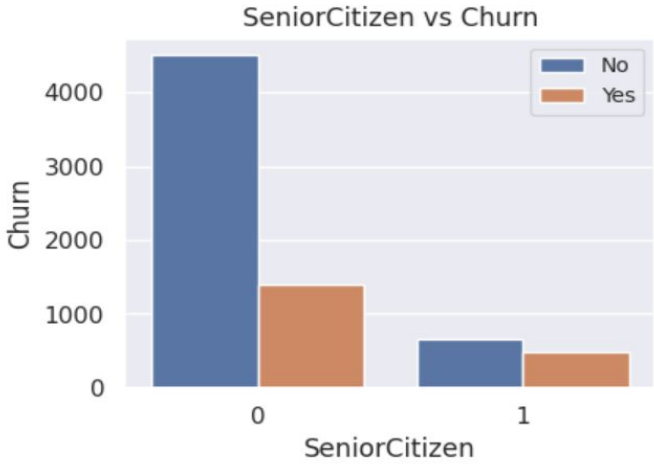
Visual Insights

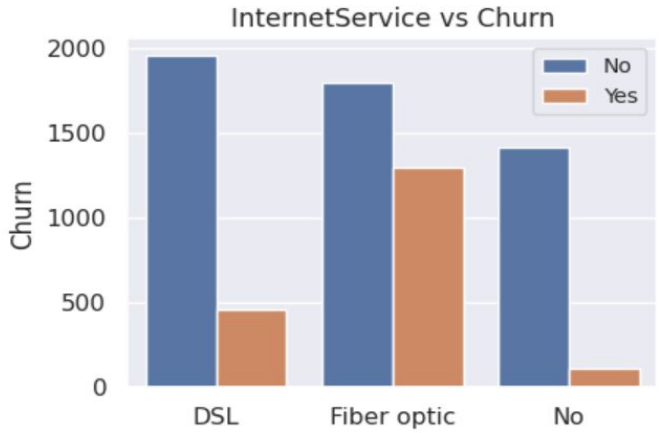
Analysing the features against the target feature ‘Churn’ gives insights like,

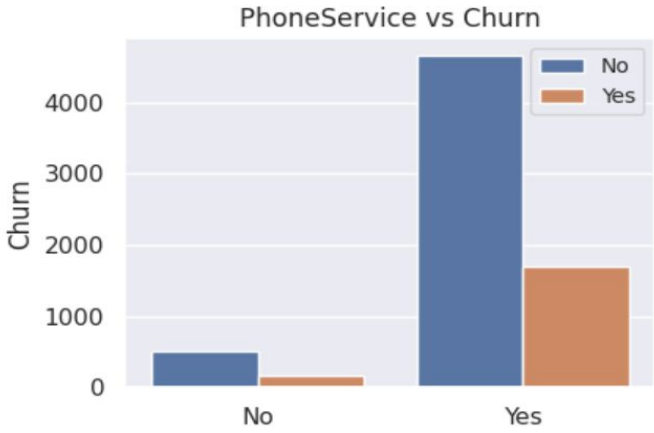
- Long term contract has fewer Churns compared to Month to month.
- InternetService option plays a vital role in the churn rate.
- Gender doesn't have much impact on the target.







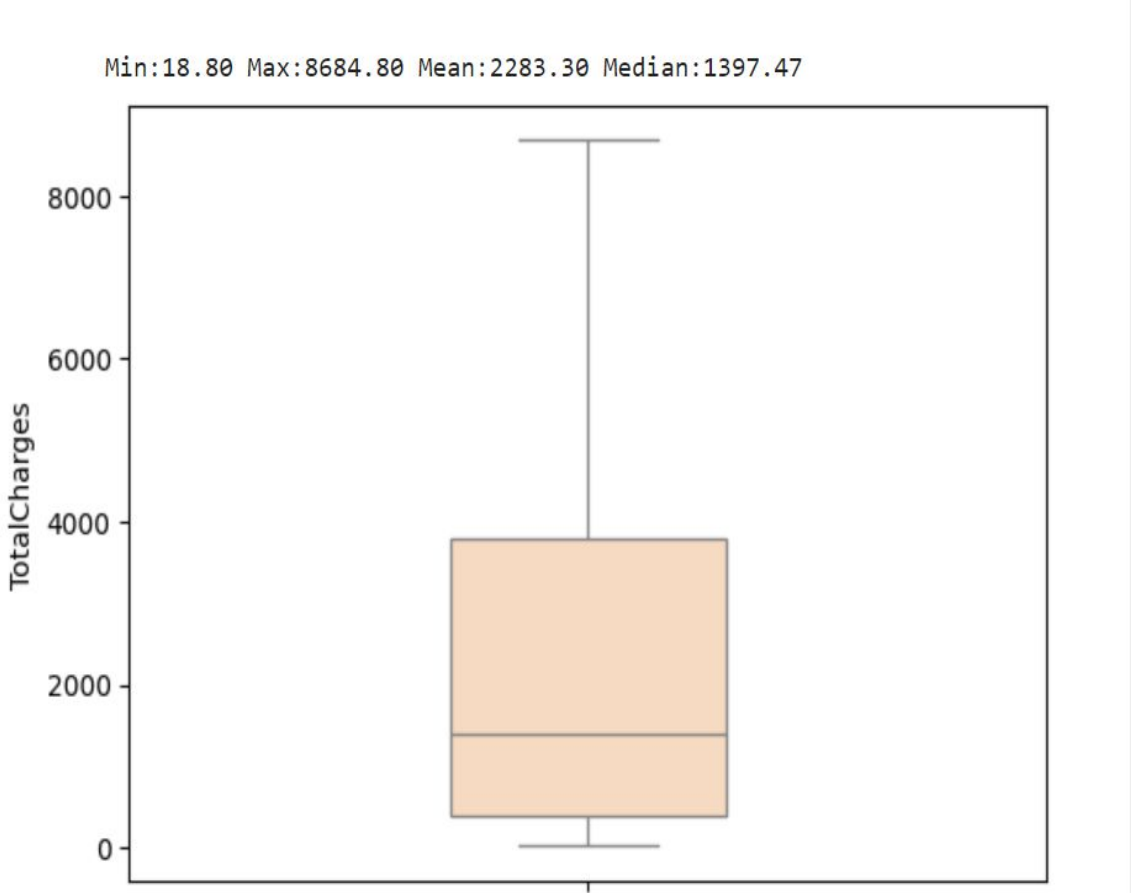


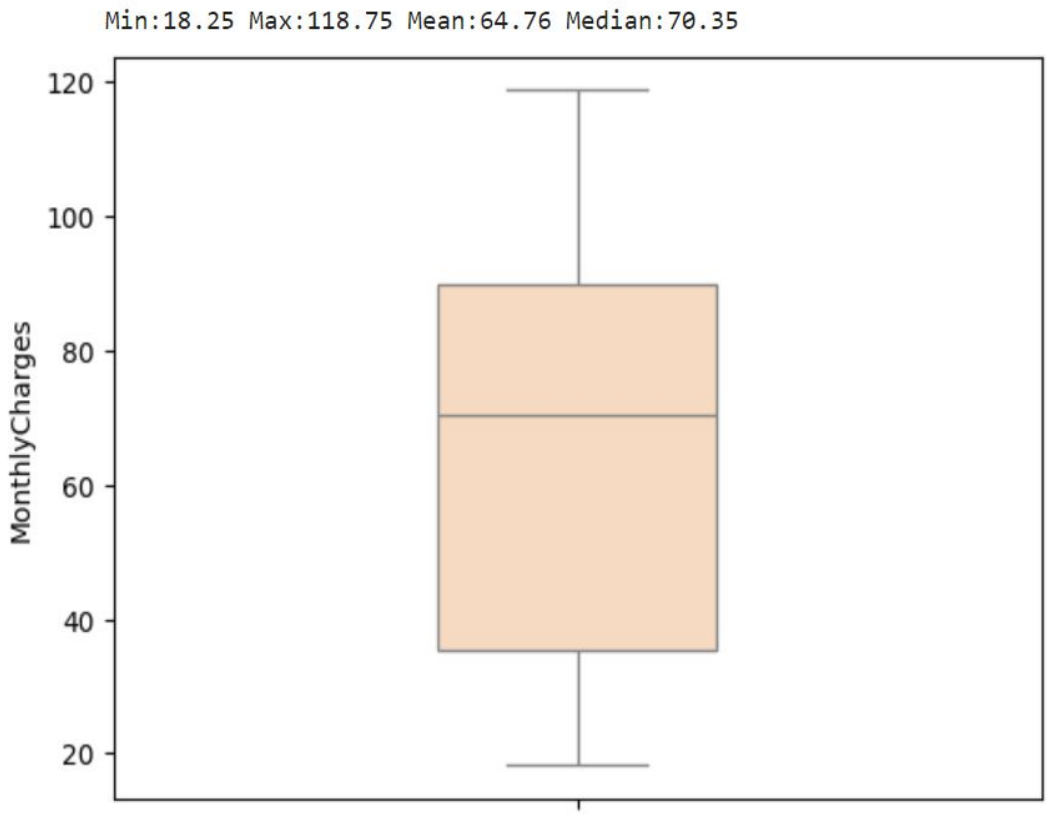


PIXELTESTS

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.



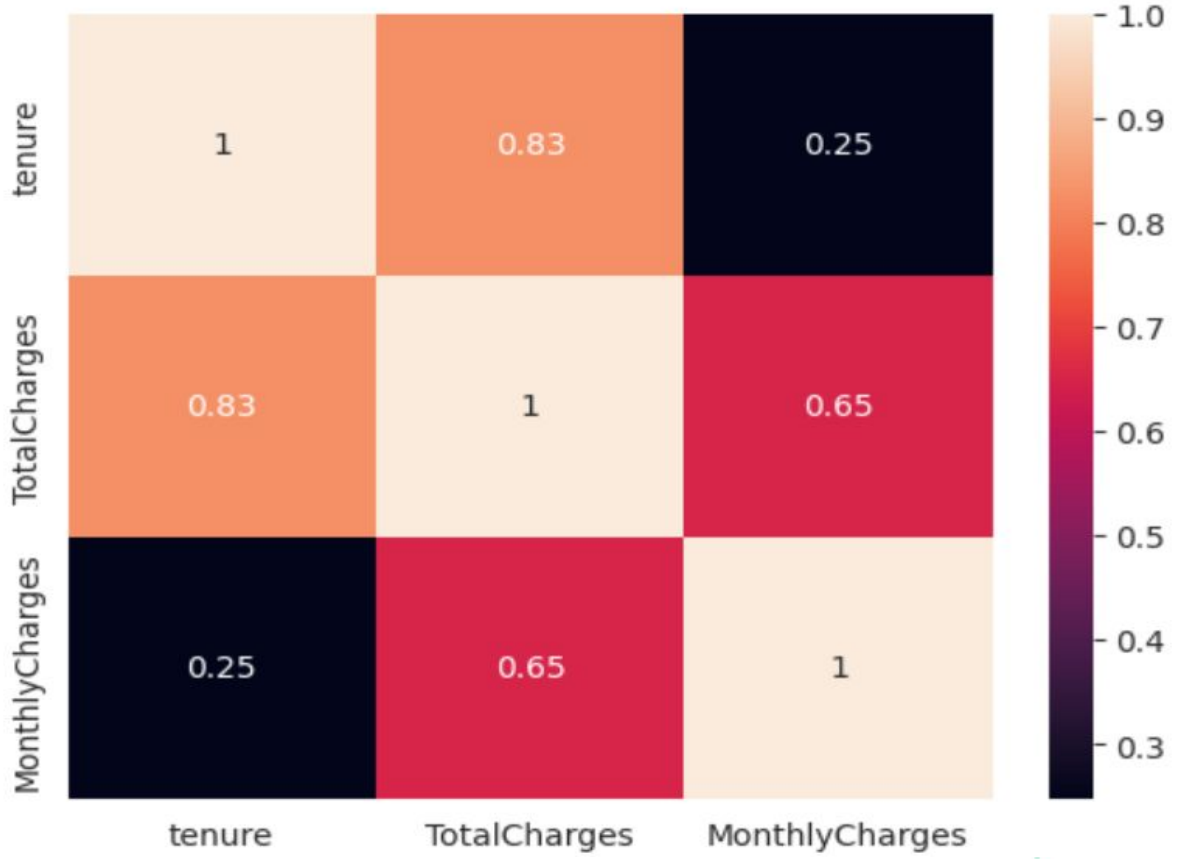


Visual Insights on ordinal features

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,

- ➢ Removes the unimportant columns
- ➢ Handles the missing values
- ➢ Encodes the categorical data

Makes the data suitable for the supervised machine learning algorithm.

```
[6]  #This function prepares the Telco data for ML algorithms
     from sklearn.preprocessing import LabelEncoder
     def wrangle(ds):
       ds=ds.loc[ds['TotalCharges']!= " "]
       #Convert the numerical columns
       ds['TotalCharges']= pd.to_numeric(ds['TotalCharges'])
       print(type(NUM_FEATURES))
       NUM_FEATURES.append('TotalCharges')
       #Customer ID could be dropper
       ds.drop('customerID', axis=1, inplace= True)

       #Encode the target Vriables
       enc=LabelEncoder()
       ds[TARGET]=enc.fit_transform(ds[TARGET])
       #Encode the other caegorical columns
       ds=pd.get_dummies(ds, dtype=int)

       return ds
     encoded_ds = wrangle (ds)
```

Model Selection

Classifier algorithms are well suited for predicting the categotical 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

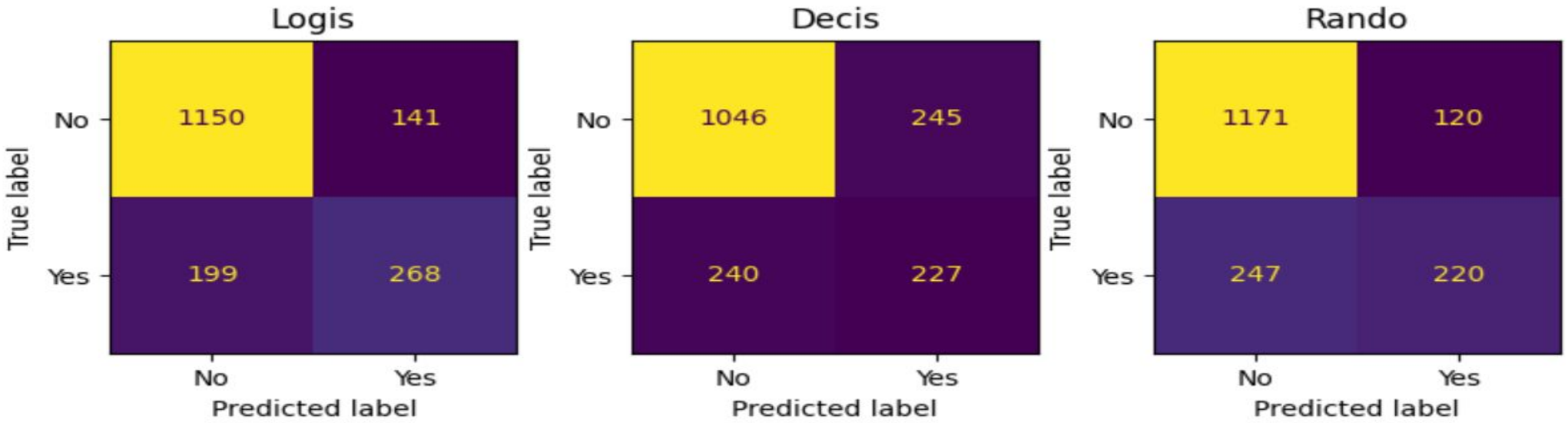
```
#4. Split the data to train and test
y= tuned_ds['Churn']
X= tuned_ds.drop(columns='Churn', axis =1)

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)
```



Baseline accuracy : 73.42

	Accuracy	precision	recall	f1-score
LogisticRegression	80.66	65.53	57.39	61.19
DecisionTree	72.41	48.09	48.61	48.35
RandomForest	79.12	64.71	47.11	54.52

Cross Validation with 5 Fold

```python
#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))
```

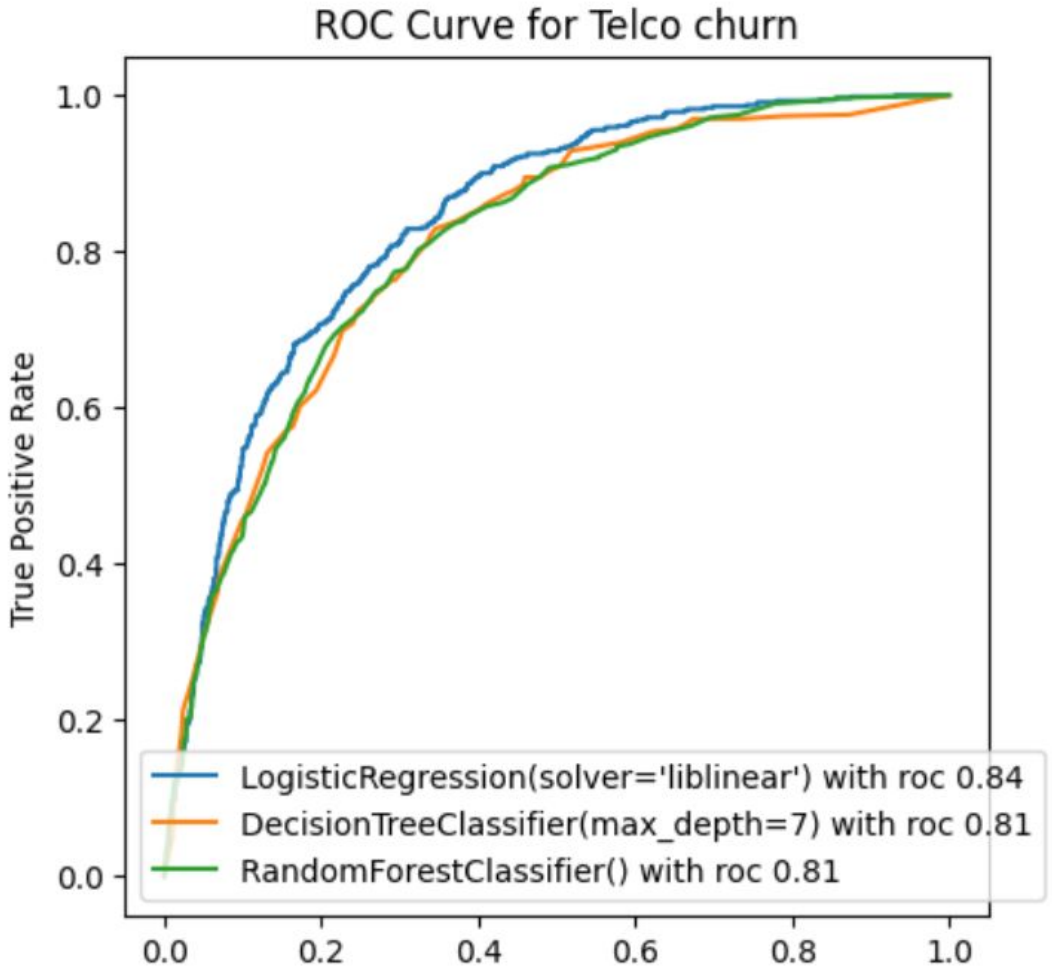
```
[0.80597015 0.81307747 0.79587482 0.80654339 0.80369844]
The accuracy score of the model  LogisticRegression(solver='liblinear')  =  80.5
[0.78891258 0.79530917 0.77524893 0.78733997 0.7972973 ]
The accuracy score of the model  DecisionTreeClassifier(max_depth=7)  =  78.88
[0.7960199  0.78535892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

PIXELTESTS

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

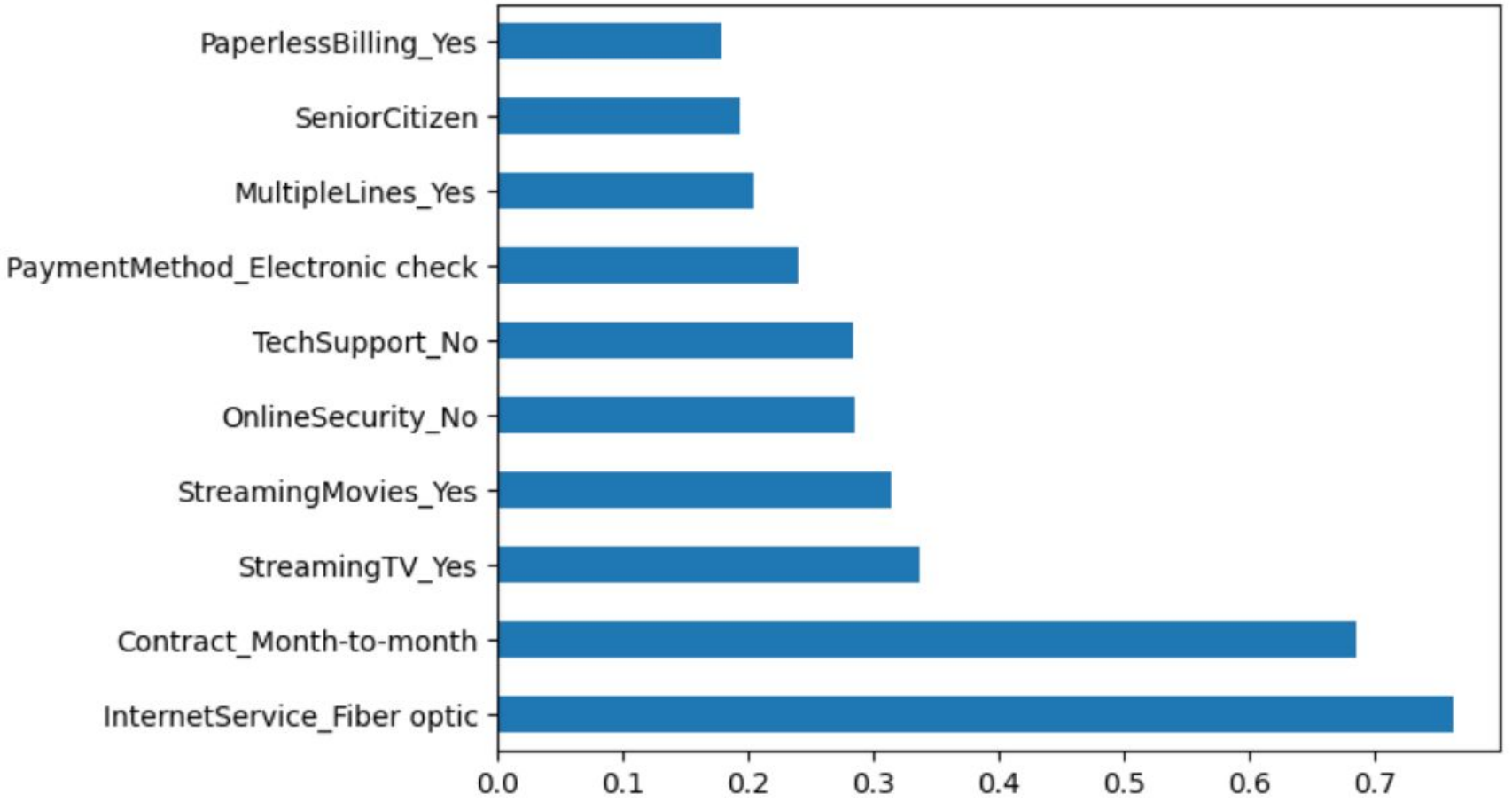
Logistic regression parameters like solver and the max_iter are tuned using GridSearchCV to get the highest accuracy possible.

```
#Implement GridSearchCV
from sklearn.model_selection import 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_res, Y_train_res)
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': 'newton-cg'}
LogisticRegression() with {'max_iter': 100, 'solver': 'newton-cg'} gives the best accuracy : 81.30%
```

PIXELTESTS

Feature Importance



Conclusion

- Machine learning algorithm, Logistic regression is well suited for this problem among the other classifier algorithms with 81% 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

NVIDIA
INCEPTION PROGRAM