ISTANBUL DATA SCIENCE ACADEMY

Customer Churn

Presendeted by: Mert Yigitcan

OUTLINE

ISTANBUL DATA SCIENCE ACADEMY

Abstract

- Why does it matter?
- What are we going to do?

Design

I will build a model to answer questions like "Looking at this customers given data, will he/she churn in X months"

Data

Telco Dataset-Kaggle

21 Initial Features – Mixed

EDA

Visualation of data

Modelling

- What do we want to achieve?
- What is our use case?
- What will be our solution to churn?

ABSTRACT

- Customer churn refers to the process of identifying customer/ clients who will terminate their relations with an organisation.

- The purpose of this project is to build a model to predict if a given customer will churn or not churn using various classification algorithms and techniques

DESIGN

- For the purpose of the business, I ensured that we catch as much churns, so we will make recall and f1 score our priority.

- I built upon that by using various techniques like class imbalance techniques, cross-validation to achive an optimal F1-score which we are about.

ISTANBUL DATA SCIENCE ACADEMY

Data Acquisition & Storage

Data

Telco Dataset-Kaggle

21 Initial Features – Mixed

~7.5K observations

```
In [5]: print(f'num rows: {data.shape[0]} \nnum columns:  {data.shape[1]}')
```

```
num rows: 7043
num columns:  21
```

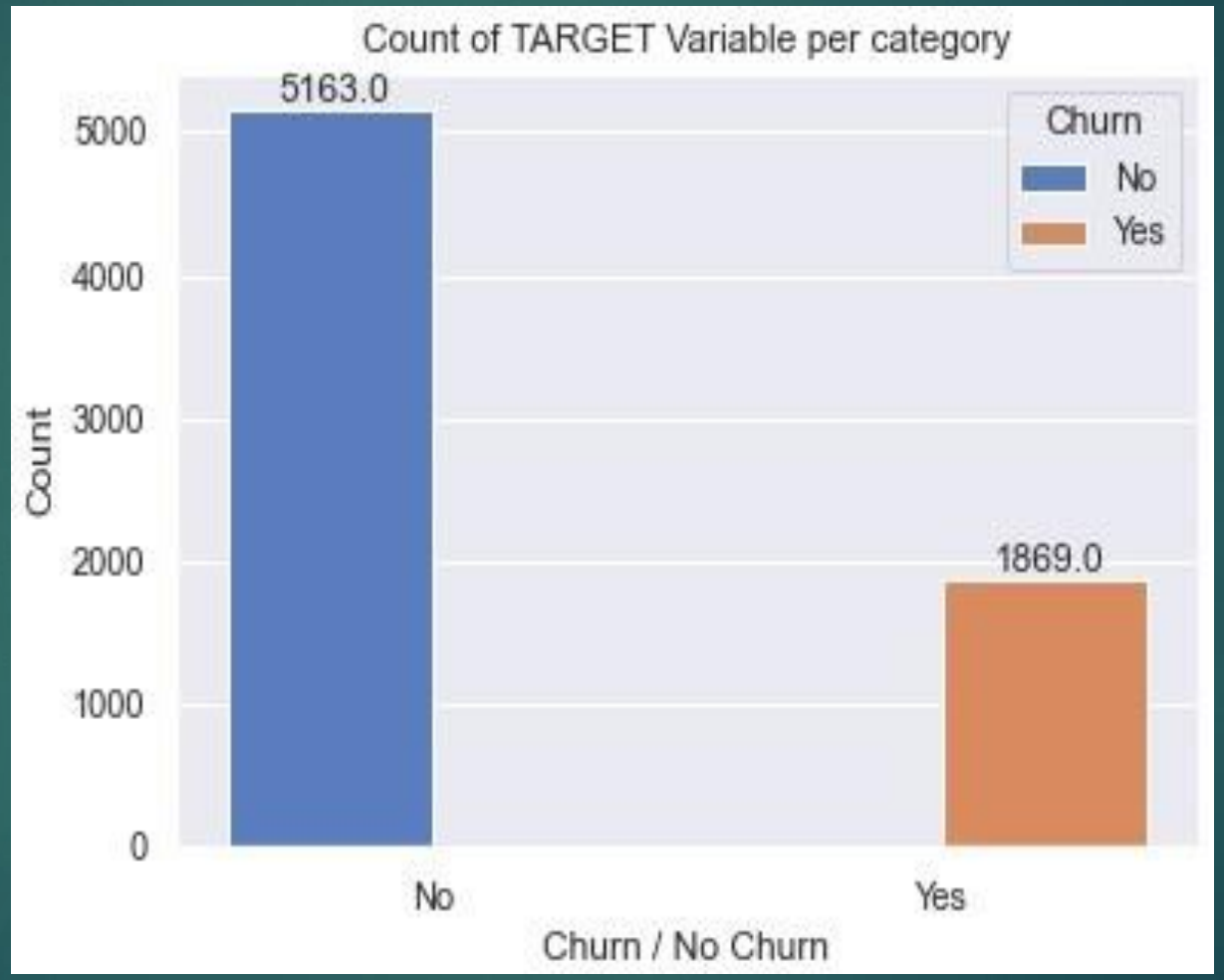
```
data.dropna(how = 'any', inplace = True)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7032 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7032 non-null   object
 1   gender            7032 non-null   object
 2   SeniorCitizen     7032 non-null   int64
 3   Partner           7032 non-null   object
 4   Dependents        7032 non-null   object
 5   tenure            7032 non-null   int64
 6   PhoneService      7032 non-null   object
 7   MultipleLines     7032 non-null   object
 8   InternetService   7032 non-null   object
 9   OnlineSecurity    7032 non-null   object
 10  OnlineBackup      7032 non-null   object
 11  DeviceProtection  7032 non-null   object
 12  TechSupport       7032 non-null   object
 13  StreamingTV       7032 non-null   object
 14  StreamingMovies   7032 non-null   object
 15  Contract          7032 non-null   object
 16  PaperlessBilling  7032 non-null   object
 17  PaymentMethod     7032 non-null   object
 18  MonthlyCharges    7032 non-null   float64
 19  TotalCharges      7032 non-null   float64
 20  Churn             7032 non-null   object
dtypes: float64(2), int64(2), object(17)
```

ISTANBUL DATA SCIENCE ACADEMY

Data Exploration

Imbalance Data

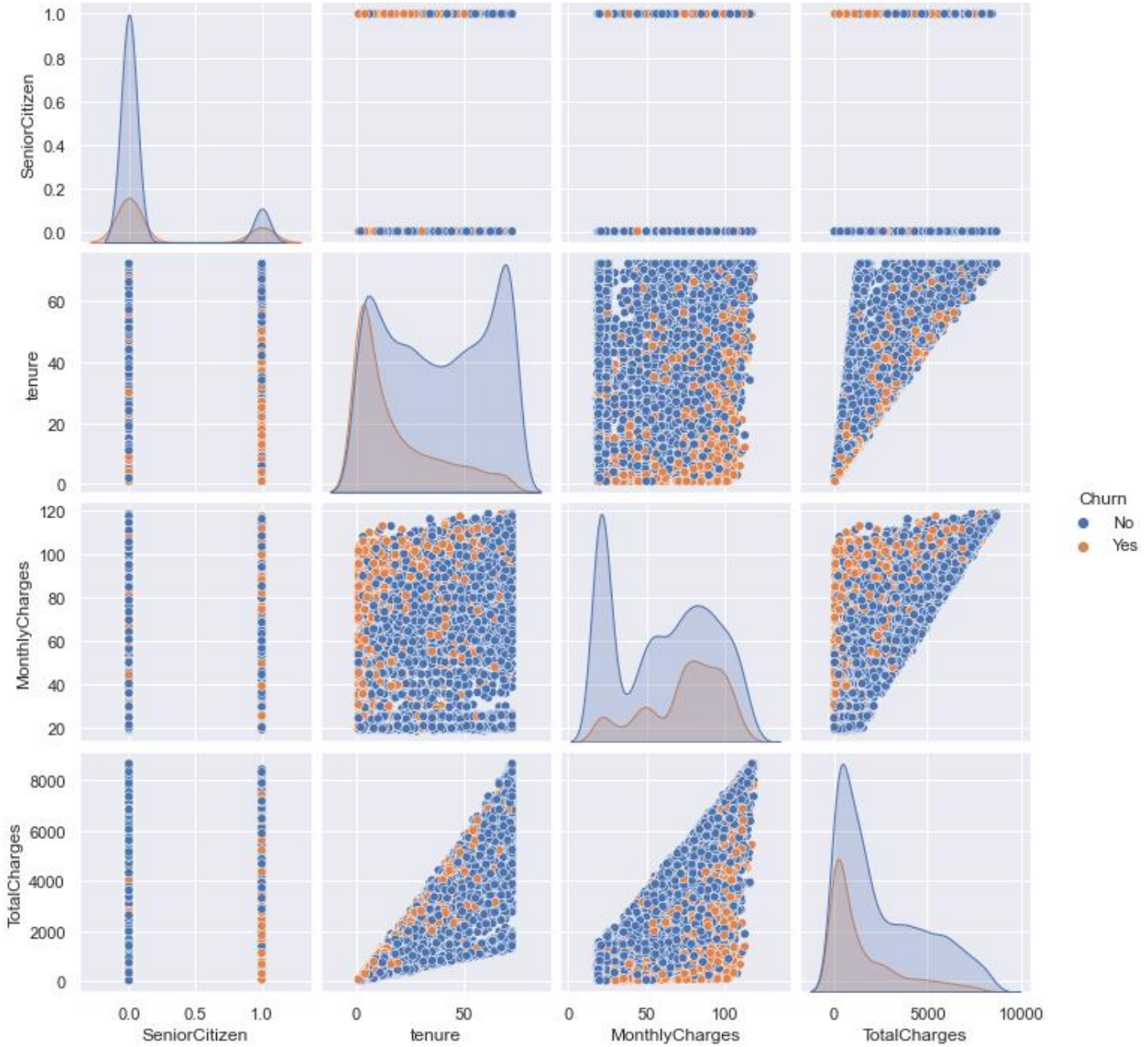



```
No     5163
Yes    1869
Name: Churn, dtype: int64

In precentages
No     73.421502
Yes    26.578498
Name: Churn, dtype: float64
```

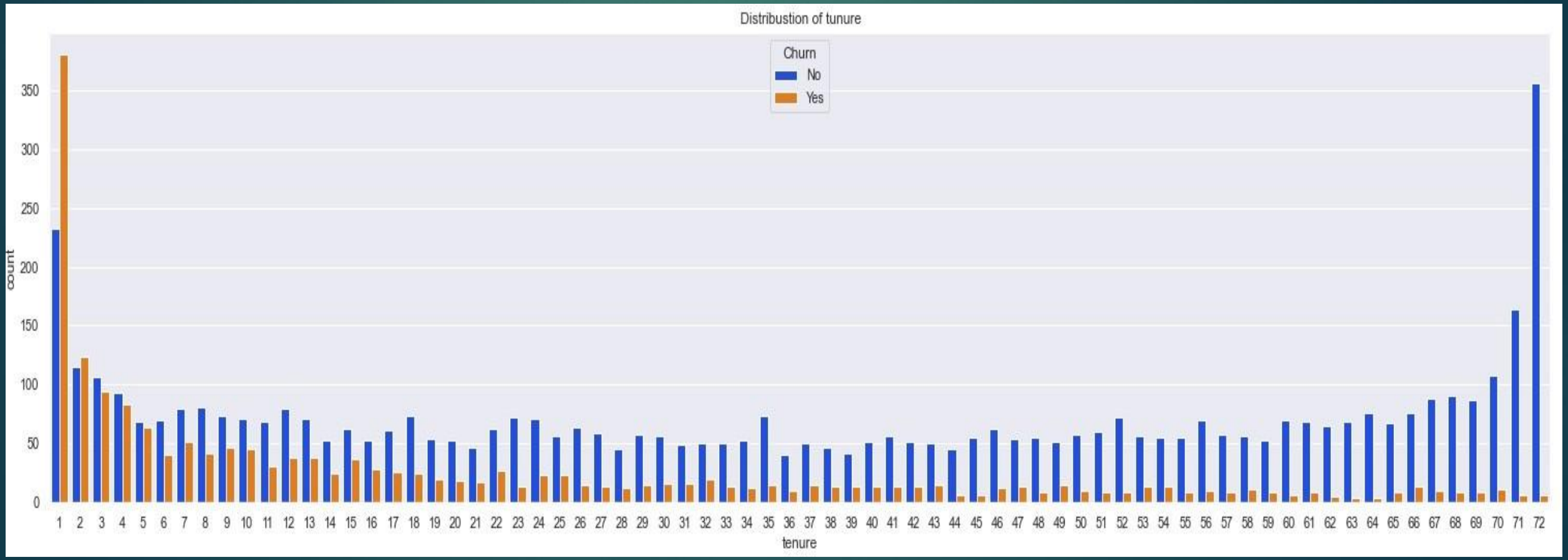
ISTANBUL DATA SCIENCE ACADEMY

Data Exploration

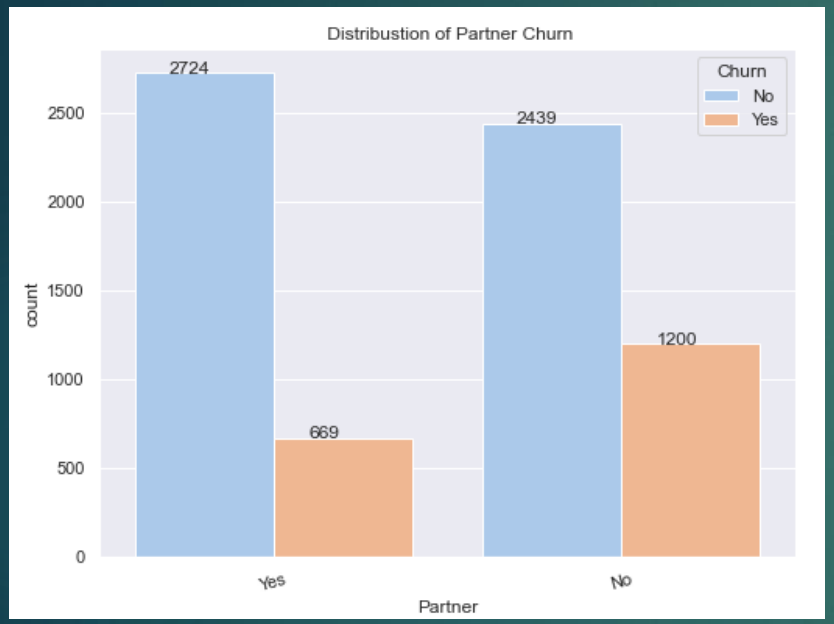
```
In [13]: sns.pairplot(data.drop('customerID', axis=1), hue='Churn')
         plt.show()
```



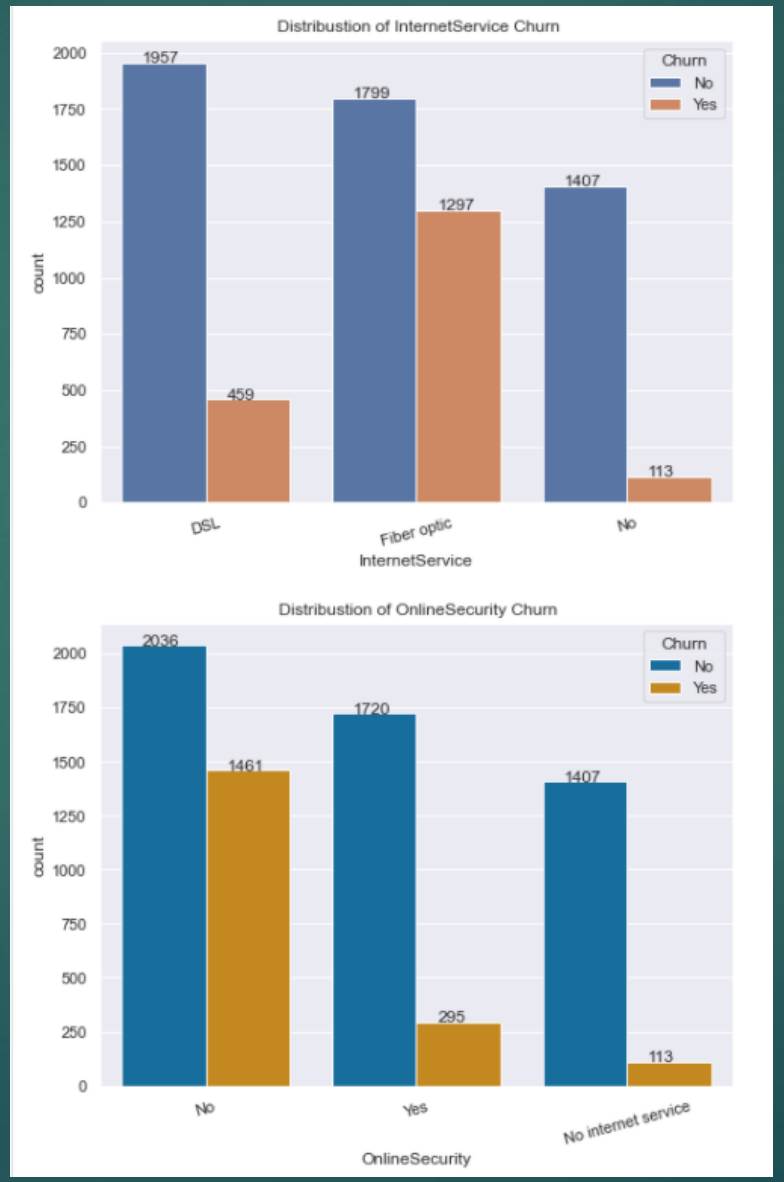
Tenure

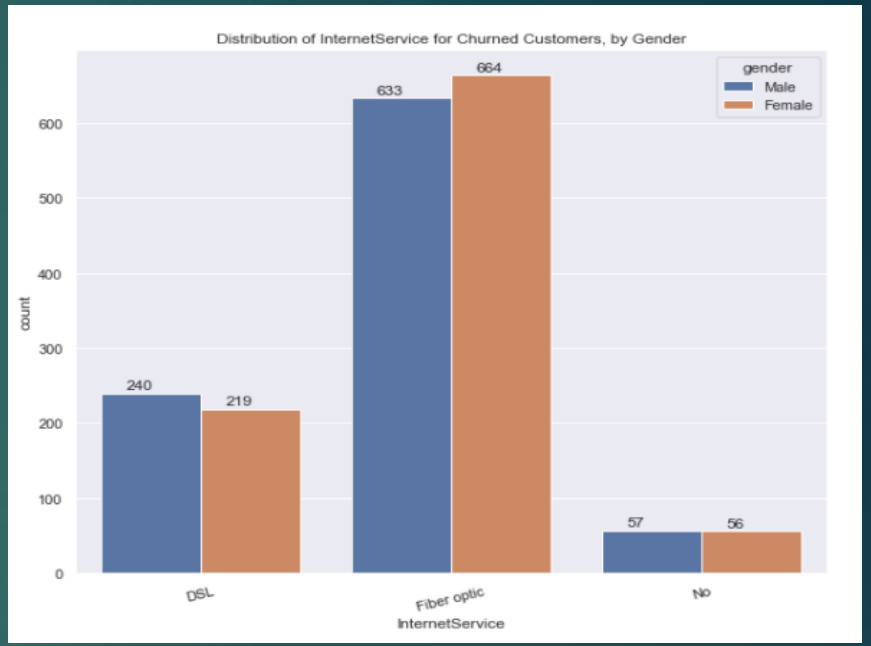

Distribustion of tunure
Churn
No
Yes
count
0
50
100
150
200
250
300
350
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
tenure

Data Exploration



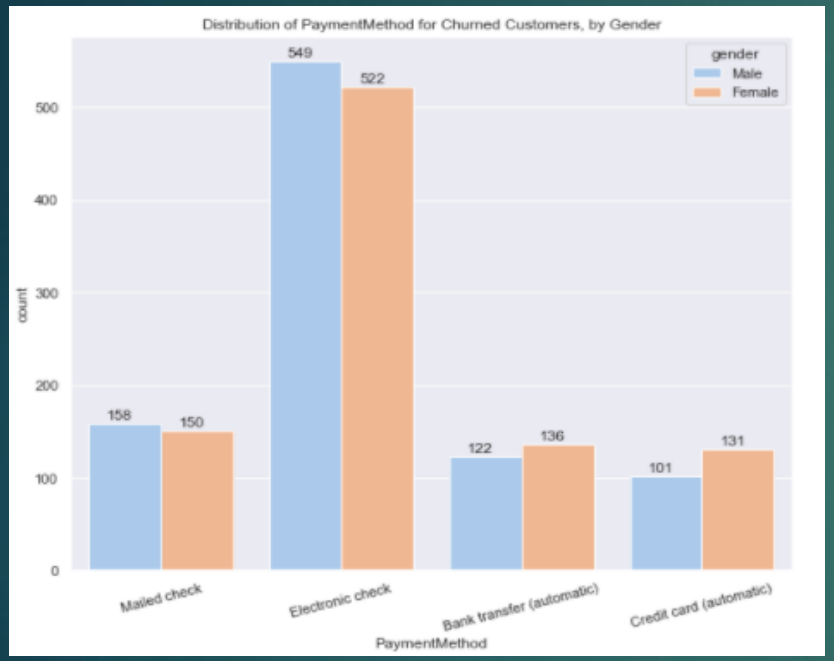
Distrubution of Partner Churn



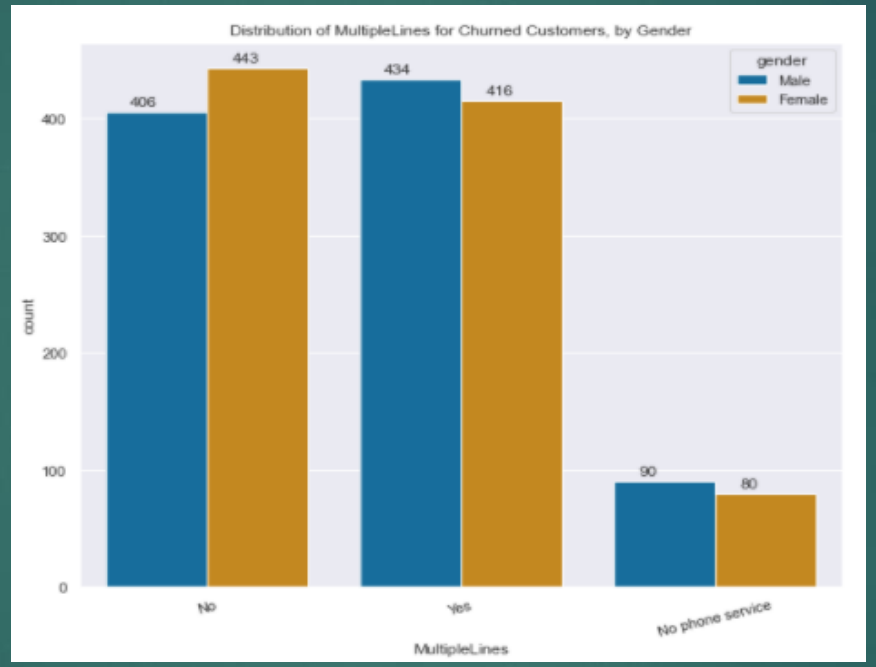


Distrubution of Internet Service for Churned by Gender

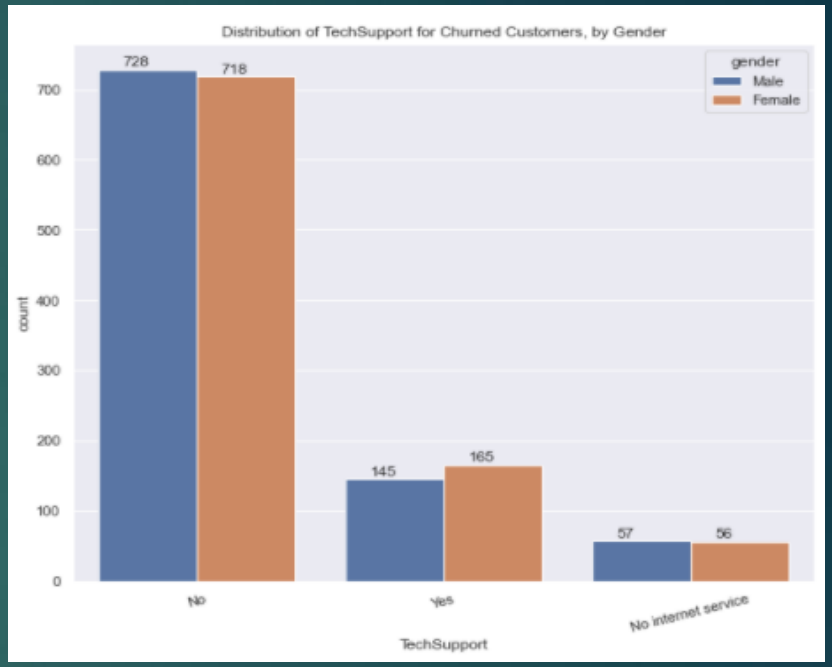
Data Exploration



Distrubution of Payment Method for Churned by Gender



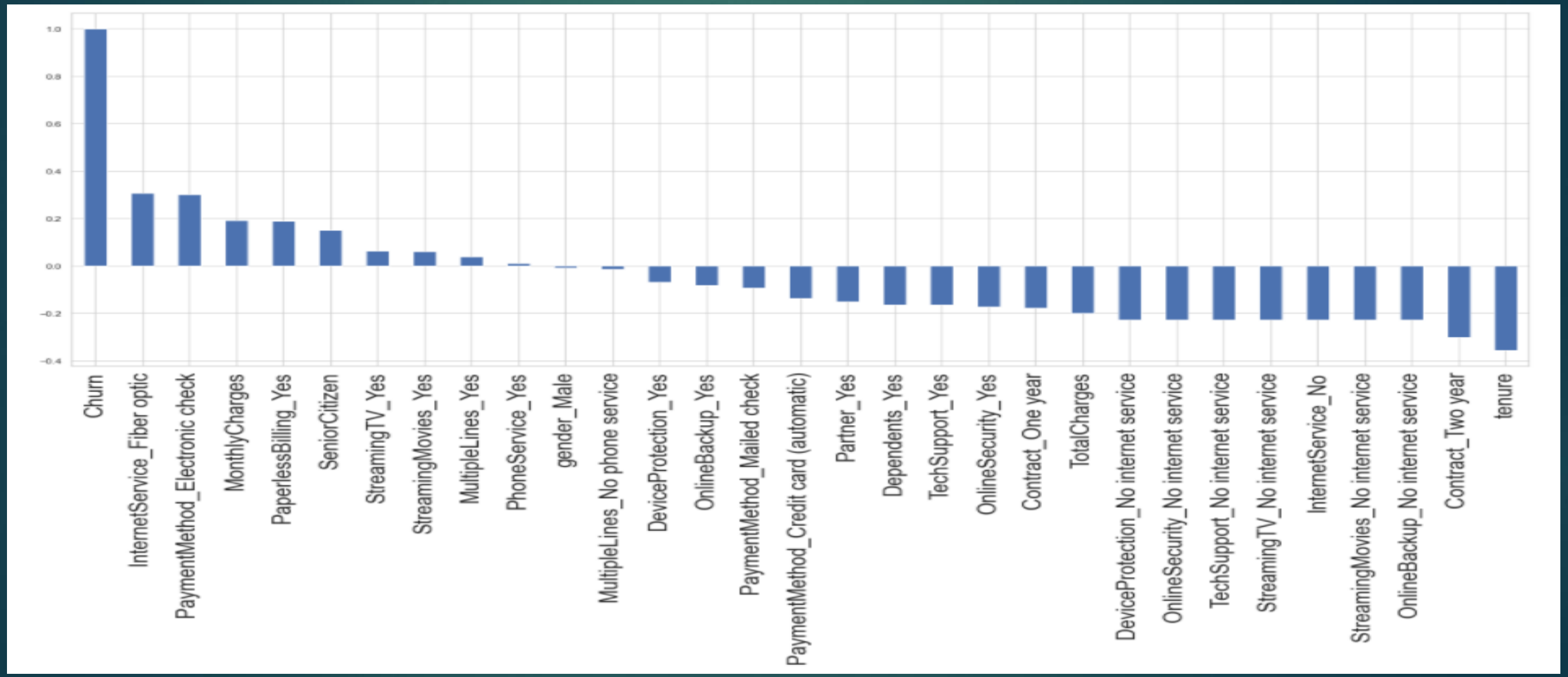
Distrubution of Multiple Lines for Churned Customers by Gender



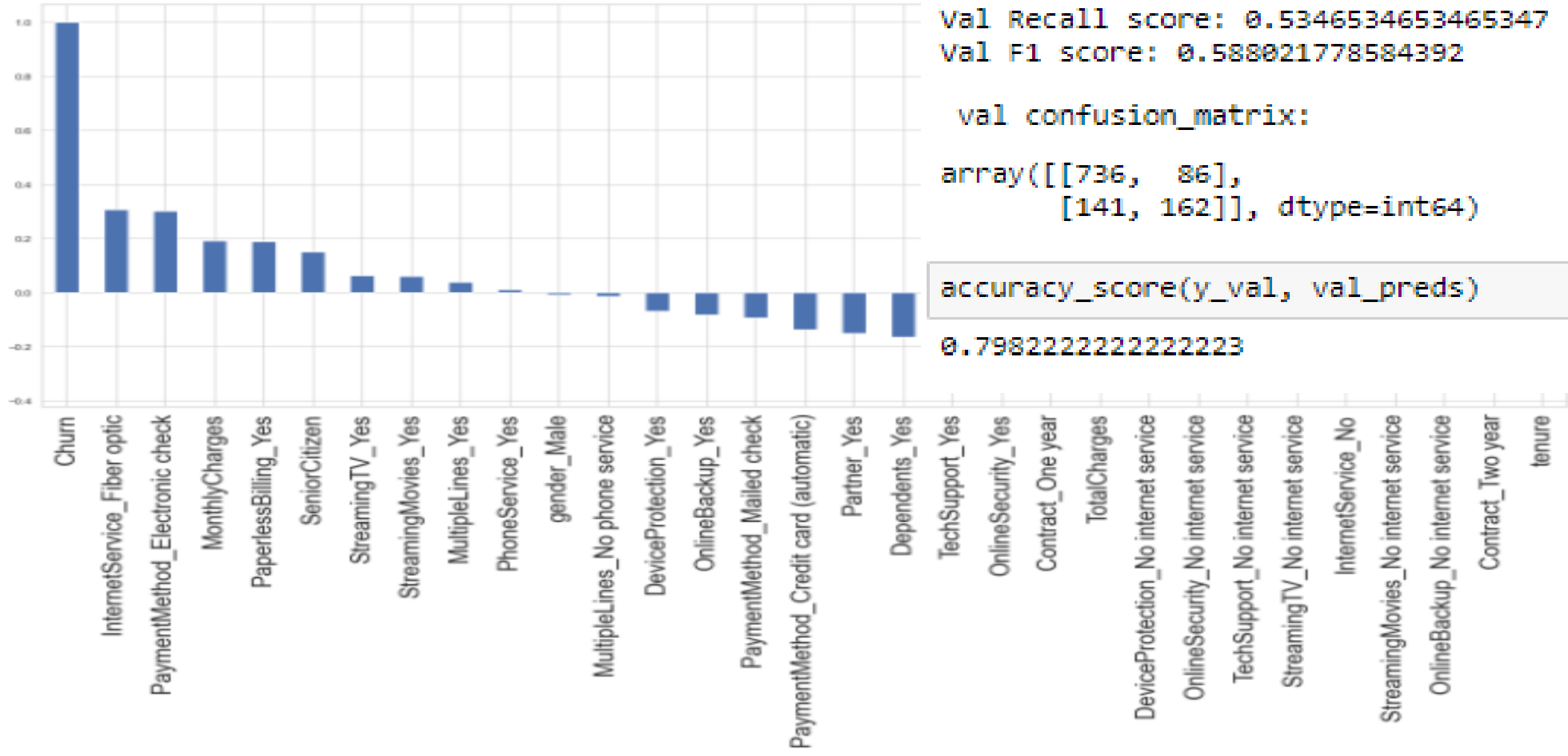
Distrubution of Tech Support fro Churned Customers by Gender

Before Modeling

```
plt.figure(figsize=(25,10))
ab = full_data.corr()['Churn'].sort_values(ascending = False).plot(kind='bar')
plt.xticks(fontsize=25, rotation=90)
```



Modelling




```
Val Recall score: 0.5346534653465347
Val F1 score: 0.5880217785843920

 val confusion_matrix:

array([[736,  86],
       [141, 162]], dtype=int64)
```

```
accuracy_score(y_val, val_preds)
```

```
0.7982222222222223
```

Modelling

ISTANBUL DATA SCIENCE ACADEMY

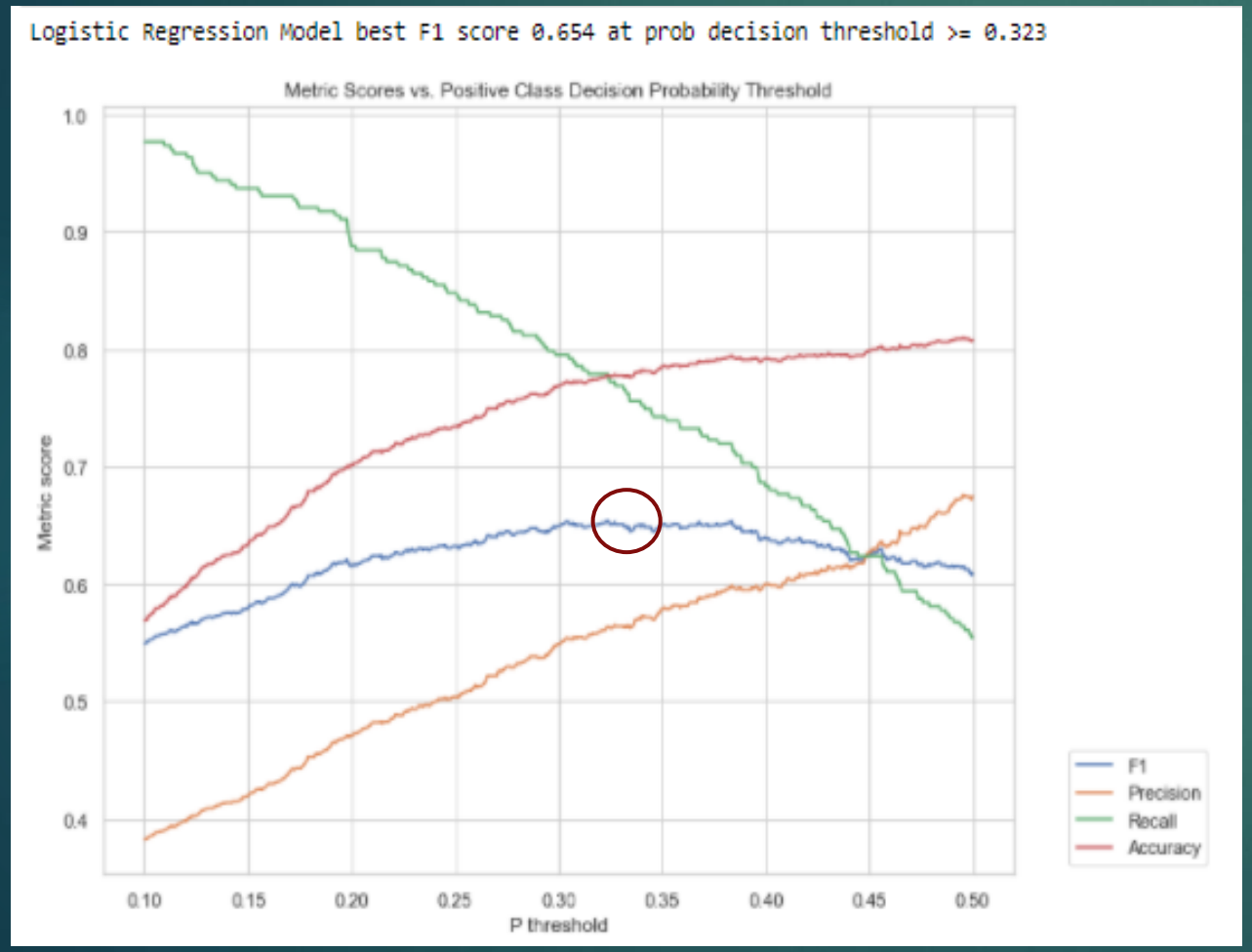
- Logistic Regression as baseline
- Recall: 0.5346
- F1 Score : 0.588
- **…Ooopsss!!**

- Test Different Models
- Every Model:
  - Bring in all our our feature
  - Make them usable by the model (dummy)
  - Cross- Validation
  - Class Imbalance Techniques

• Focus on Imbalance tecniques.

ISTANBUL DATA SCIENCE ACADEMY

```
In [43]: thresh_ps = np.linspace(.10,.50,1000)
         model_val_probs = clf_cv.predict_proba(x_val)[:,1]
```



- F1
- Precession
- Recall
- Accurancy

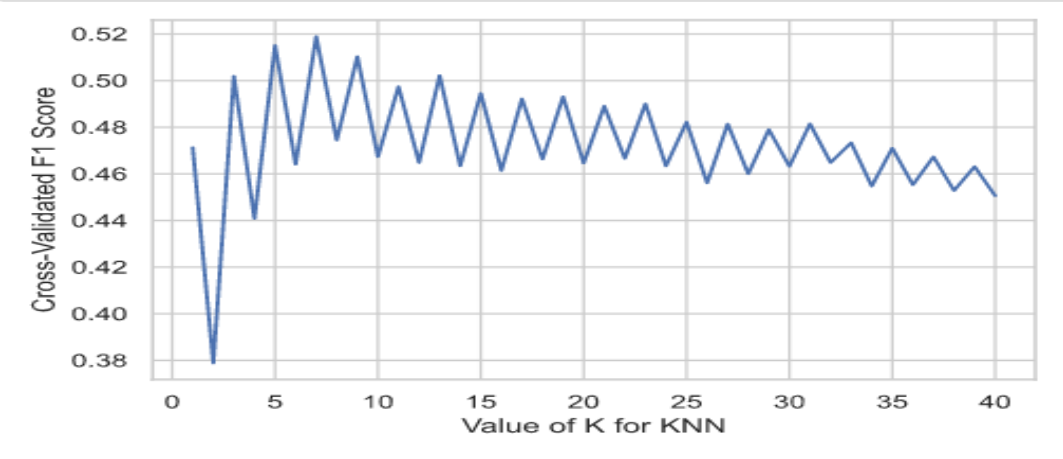
ADASYN

```
Val Recall score: 0.7260726072607261
Val F1 score: 0.6258890469416786

 val confusion_matrix:

array([[642, 180],
       [ 83, 220]], dtype=int64)
```

SMOTE

```
Val Recall score: 0.7161716171617162
Val F1 score: 0.6317321688500728

val confusion matrix:

In [57]: accuracy_score(y_val,smt_preds )
Out[57]: 0.7751111111111111
```

KNeigbhbors



GridSearchCV

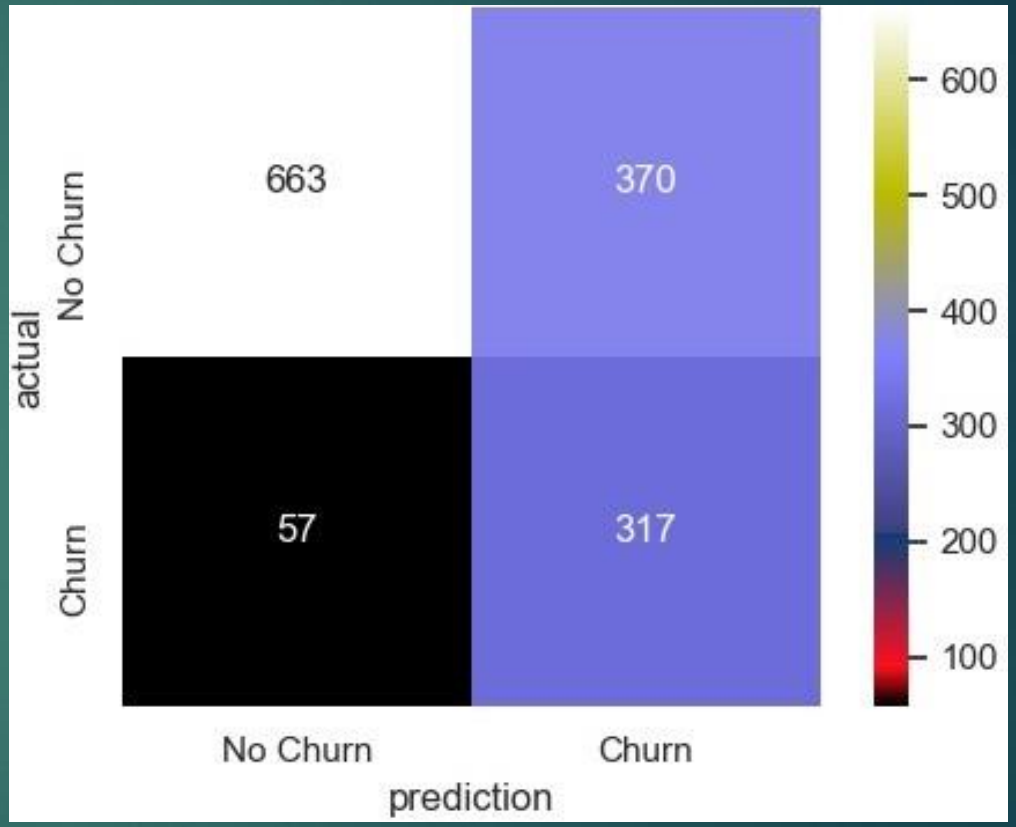
```
In [66]: print("Best params: ", grid.best_params_)
         print("Best estimator: ", grid.best_estimator_)
         print("Best score: ", grid.best_score_)

         Best params:  {'n_neighbors': 7}
         Best estimator:  KNeighborsClassifier(n_neighbors=7)
         Best score:  0.5189494499854447
```

Random Forest

- Combine Val and test
- Fit

- Results:

Threshold @ .21




THANK YOU :)