Telco Customer Churn Analysis Report

Berru Lafcı | 1901042681

.

I. Introduction

In the competitive landscape of telecommunications, understanding and predicting customer churn is crucial for retaining a stable customer base and ensuring business sustainability. This code aims to conduct a comprehensive analysis of customer churn for a telecommunications company and build predictive models to identify potential churners. The dataset utilized in this analysis contains a rich set of information about customers, covering attributes, subscribed services, account details, and demographic information.

II. Dataset Overview

The dataset includes 7,043 rows representing individual customers and 21 columns encompassing various features. These features include customer attributes, details about subscribed services (phone, internet, security, streaming, etc.), account information (tenure, contract, billing method, charges), and demographic details (gender, age range, dependents, etc.). The primary target variable is "Churn," indicating whether a customer has left within the last month.

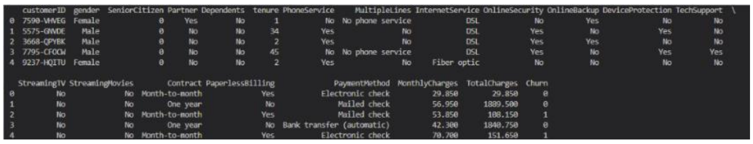
III. Libraries and Data Import

To facilitate the analysis and machine learning tasks, essential libraries such as NumPy, Pandas, Matplotlib, Seaborn, scikit-learn, and others are imported. The dataset is loaded from a CSV file named "Telco-Customer-Churn.csv."

IV. Data Analysis

A. General Information about Dataset

Using df.dtypes, it was observed that the TotalCharges feature has object type. That's why the to_numeric function was implemented. Additionally, binary encoding was applied to Churn, which is the target value. Using the df.isnull().sum() function, it was analyzed how many missing values were in each feature.



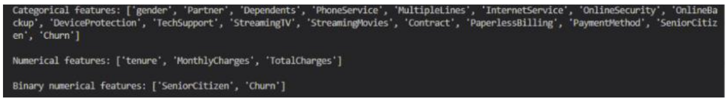
```
  customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService     MultipleLines InternetService OnlineSecurity OnlineBackup DeviceProtection TechSupport  \
0  7590-VHVEG  Female              0     Yes         No       1           No  No phone service             DSL             No          Yes               No          No
1  5575-GNVDE    Male              0      No         No      34          Yes                No             DSL            Yes           No              Yes          No
2  3668-QPYBK    Male              0      No         No       2          Yes                No             DSL            Yes          Yes               No          No
3  7795-CFOCW    Male              0      No         No      45           No  No phone service             DSL            Yes           No              Yes         Yes
4  9237-HQITU  Female              0      No         No       2          Yes                No     Fiber optic             No           No               No          No

  StreamingTV StreamingMovies        Contract PaperlessBilling             PaymentMethod  MonthlyCharges  TotalCharges  Churn
0          No              No  Month-to-month              Yes          Electronic check          29.850        29.850      0
1          No              No        One year               No              Mailed check          56.950      1889.500      0
2          No              No  Month-to-month              Yes              Mailed check          53.850       108.150      1
3          No              No        One year               No Bank transfer (automatic)          42.300      1840.750      0
4          No              No  Month-to-month              Yes          Electronic check          70.700       151.650      1
```

customerID	0
gender	0
SeniorCitizen	0
Partner	0
Dependents	0
tenure	0
PhoneService	0
MultipleLines	0
InternetService	0
OnlineSecurity	0
OnlineBackup	0
DeviceProtection	0
TechSupport	0
StreamingTV	0
StreamingMovies	0
Contract	0
PaperlessBilling	0
PaymentMethod	0
MonthlyCharges	0
TotalCharges	11
Churn	0

There are 11 missing values in TotalCharges feature.

B. Handling categorical and numerical variables

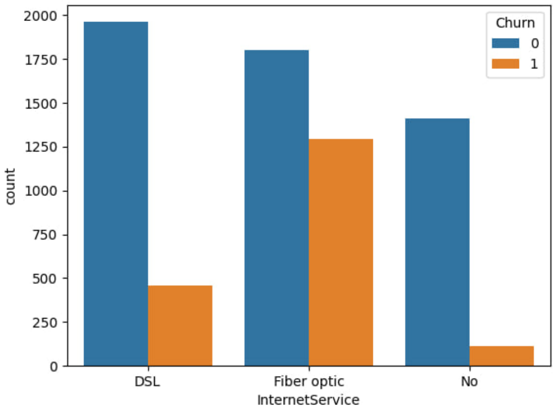
Categorical features are identified by examining columns with object data types. Binary numerical features, characterized by having less than 10 unique values and not being of object data type, are also extracted. Similarly, cardinal features, defined by having more than 20 unique values and being of object data type, are identified. The final set of categorical features is formed by combining traditional categorical features, binary numerical features, and excluding cardinal features. Numeric features and numeric features excluding binary ones are also identified separately. The outcome is presented through printed statements, showcasing the categorized features for further analysis.



```
Categorical features: ['gender', 'Partner', 'Dependents', 'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBa
ckup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'SeniorCitiz
en', 'Churn']

Numerical features: ['tenure', 'MonthlyCharges', 'TotalCharges']

Binary numerical features: ['SeniorCitizen', 'Churn']
```

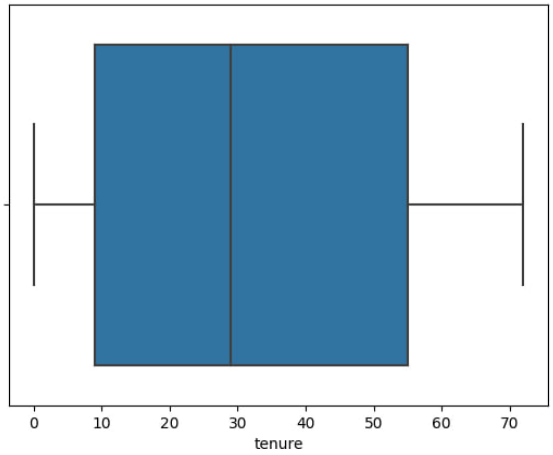
C. *Analyzing Categorical Features*

Analyzing by churn with using countplot. For example,

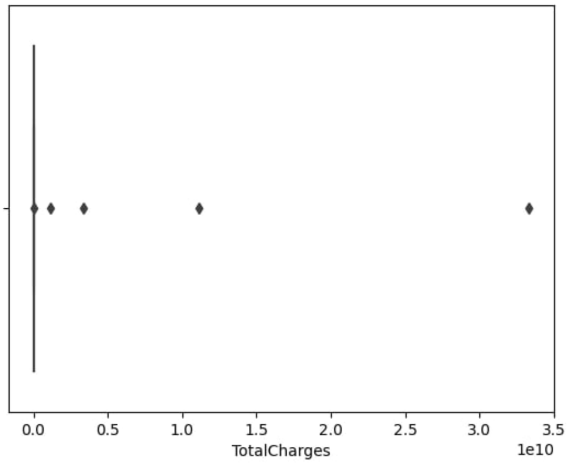


D. *Analyzing Numerical Features*

Analyzing with using boxplot. Also observing outliers. For example,



There are exactly 5 outliers in TotalCharges as it can be seen with the dots on the plot.

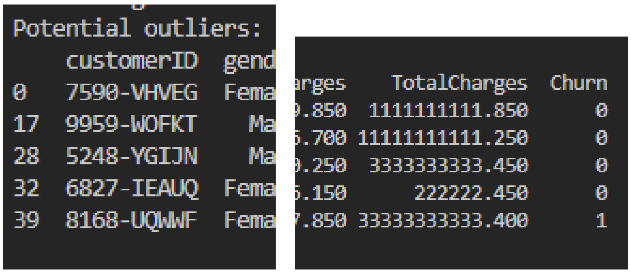


V. Preprocessing

A. *Detecting and handling potential outliers in numerical columns*

The interquartile range (IQR) method is employed to identify potential outliers. First and third quartiles (Q1 and Q3) are calculated, and the IQR is obtained by subtracting Q1 from Q3. The lower and upper bounds for potential outliers are then defined as values falling below Q1 - 1.5 times IQR and above Q3 + 1.5 times IQR.

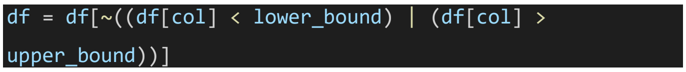
Subsequently, potential outliers are identified by comparing each value in the column against the defined bounds. The identified outliers are printed for examination:



```
Potential outliers:
    customerID  gend
0   7590-VHVEG  Fema
17  9959-WOFKT    Ma
28  5248-YGIJN    Ma
32  6827-IEAUQ  Fema
39  8168-UQWWF  Fema
```

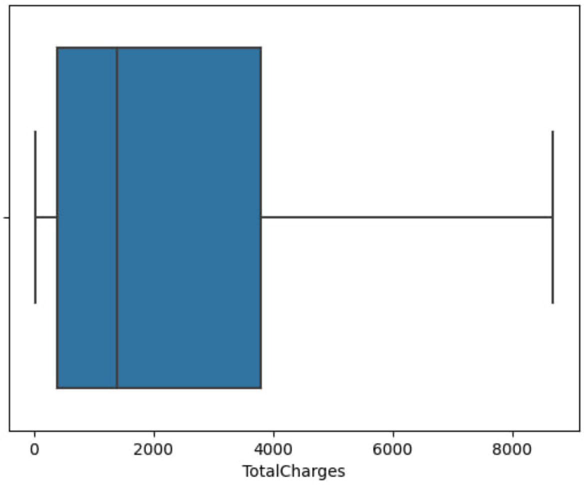
```
arges     TotalCharges  Churn
9.850  1111111111.850      0
5.700 11111111111.250      0
9.250  3333333333.450      0
5.150       222222.450      0
7.850 33333333333.400      1
```

To address the outliers, a filtered version of the dataset is created by excluding rows where the column values fall outside the determined bounds:



```
df = df[~((df[col] < lower_bound) | (df[col] >
upper_bound))]
```

This process is repeated for each numerical column. Finally, a boxplot is generated for each column after outlier removal to visually inspect the impact on the distribution.



B. *Analyzing and Handling Missing Values*

The code calculates the number of missing values (n_miss) and the missing ratio (ratio) for each column in the na_columns list.

	n_miss	ratio
TotalCharges	11	0.160

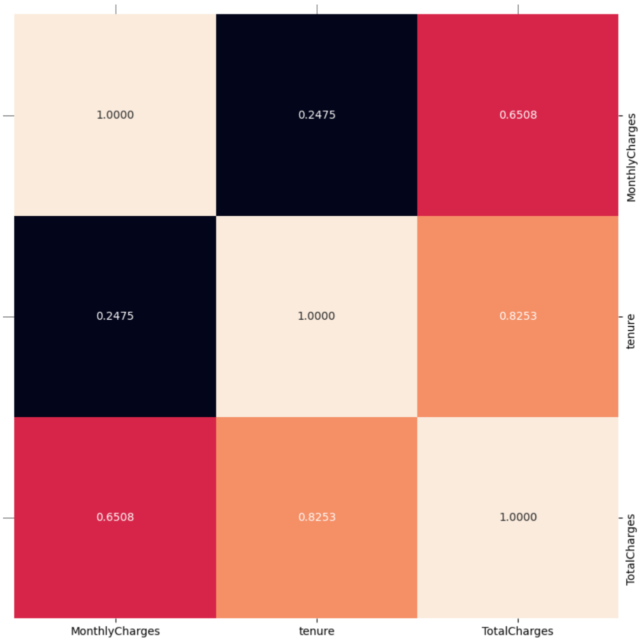
Then, the code captures the columns with missing values in the "TotalCharges" column because we found the missing values only in this feature. Subsequently, it fills the missing values in the "TotalCharges" column with the median value of the available non-missing entries. This approach is a common technique for handling missing data, providing a summary of missing value statistics.

customerID	0
gender	0
SeniorCitizen	0
Partner	0
Dependents	0
tenure	0
PhoneService	0
MultipleLines	0
InternetService	0
OnlineSecurity	0
OnlineBackup	0
DeviceProtection	0
TechSupport	0
StreamingTV	0
StreamingMovies	0
Contract	0
PaperlessBilling	0
PaymentMethod	0
MonthlyCharges	0
TotalCharges	0
Churn	0

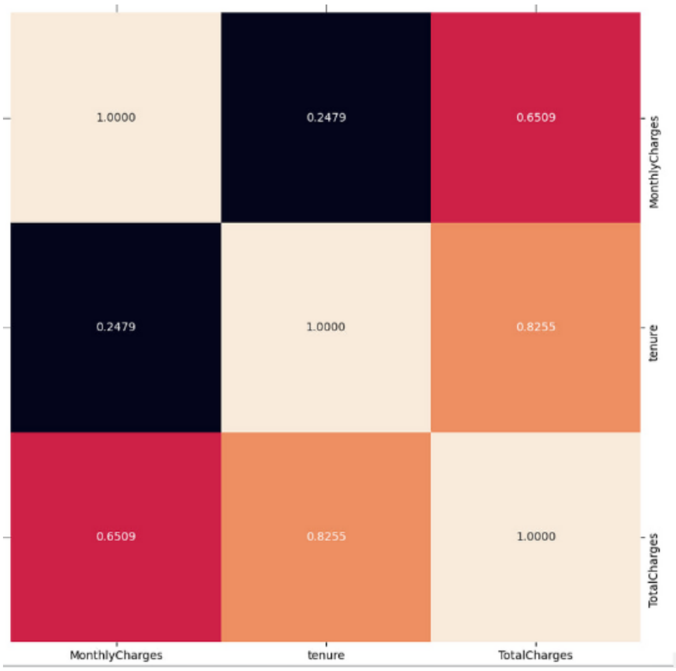
VI. Correlation

This part is focused on visualizing the correlation between numerical features in the dataset. The correlation matrix provides information about the linear relationship between pairs of numerical features.

As we see from matrix, all numerical features have positive correlation with each other which means they affect each other in the same way. Here this is correlation matrix with outliers:



And here is correlation matrix without outliers:



Outliers can inflate the covariance between variables, leading to higher correlation coefficients. By eliminating outliers, the covariance and correlation values may decrease, reflecting a more accurate representation of the relationships between variables in the absence of extreme data points.

The correlation matrix is calculated using the .corr() method in Pandas, which computes the Pearson correlation coefficients between pairs of numerical columns in the dataframe. The

Pearson correlation coefficient (r) is a measure of linear correlation between two variables and ranges from -1 to 1.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

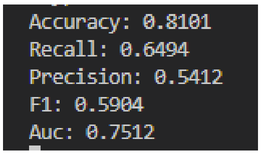
VII. Implementing First Model

I chose to use the GradientBoosting algorithm for the first modeling. Gradient Boosting is an ensemble learning method that builds a sequence of weak learners (typically decision trees) to create a strong learner. It combines the predictions of individual models to improve overall performance. Since the target variable is "Churn" which is a binary classification task, Gradient Boosting is a suitable algorithm for such predictive modeling tasks.

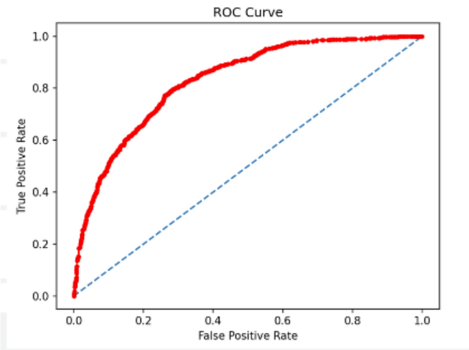
Gradient Boosting and many other machine learning algorithms require numerical input, and they may struggle with categorical variables. One-Hot Encoding is employed to convert categorical variables into a format that is compatible with these algorithms. The "one_hot_encoder" function returns the dataframe with the original categorical columns replaced by their one-hot encoded counterparts.

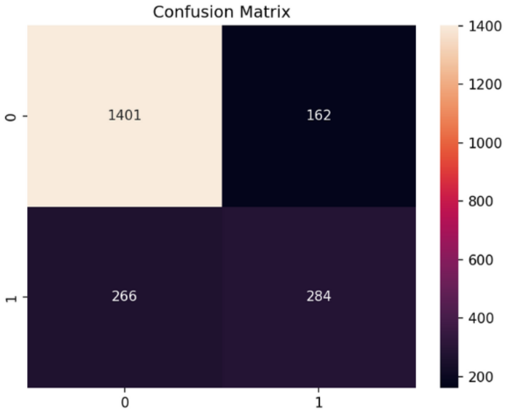
The encoded "dff" is then used to define the target variable (y) as the "Churn" column, and the feature matrix (X) by dropping the "Churn" column and the "customerID" column. The dataset is split into training and testing sets using the "train_test_split" function.

Here is the first model evaluation values:



```
Accuracy: 0.8101
Recall: 0.6494
Precision: 0.5412
F1: 0.5904
Auc: 0.7512
```

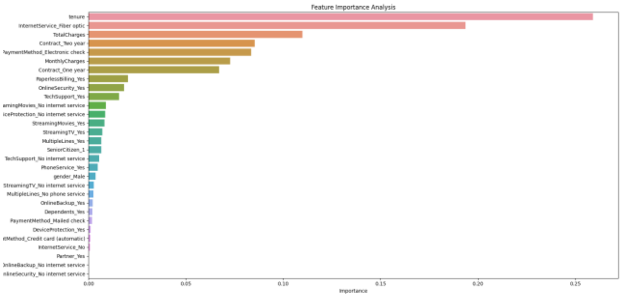




VIII. Feature Importance

The model assigns importance scores to each feature, indicating their contribution to the model's predictive performance. The feature names are obtained from the columns of the training set (X_train).

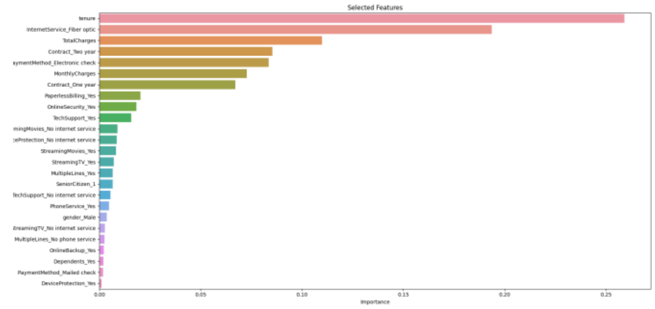
In my analysis, I saw that tenure, InternetService_FıberOptic and TotalCharges features are the most important ones that effects the model's performance.



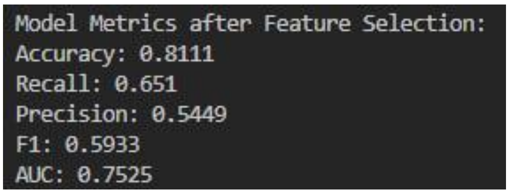
IX. Feature Selection

The goal of feature selection is to identify and retain the most relevant features while discarding less influential ones, which can enhance model interpretability, reduce overfitting, and potentially improve model generalization to new data.

I fistly tried 0.1 as threshold but, it eliminate so much feature. Then I used 0.001 as threshold to determine the importance level. Features with importance scores equal or greater than this threshold are then selected and stored in the selected_features list. Subsequently, the training and testing datasets, denoted as X_train and X_test, are updated to exclusively include the chosen features. Following this, the model employs these chosen features to make predictions on the test set (X_test_selected).



Even though we lost a couple of data, the performance increased a little bit:



```
Model Metrics after Feature Selection:
Accuracy: 0.8111
Recall: 0.651
Precision: 0.5449
F1: 0.5933
AUC: 0.7525
```

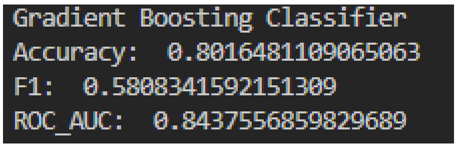
X. Making New Features

I thought of adding new features as the next step to improve accuracy. Also, these features aim to provide additional insights into customer behavior and characteristics. For instance, the "NEW_Age_Status" feature categorizes customers as "Senior" or "Young" based on the "SeniorCitizen" attribute. Similarly, "NEW_Engaged" identifies customers with one or two-year contracts.

customerID	object
gender	object
SeniorCitizen	int64
Partner	object
Dependents	object
tenure	int64
PhoneService	object
MultipleLines	object
InternetService	object
OnlineSecurity	object
OnlineBackup	object
DeviceProtection	object
TechSupport	object
StreamingTV	object
StreamingMovies	object
Contract	object
PaperlessBilling	object
PaymentMethod	object
MonthlyCharges	float64
TotalCharges	float64
Churn	int64
NEW_Age_Status	object
NEW_TotalServices	int64
NEW_Engaged	int64
NEW_Payment_Status	object
NEW_Tenure_Status	object
NEW_MonthlyCharges_Status	object
NEW_TotalCharges_Status	object
NEW_OnlineSecurity_OnlineBackup	object
NEW_StreamingTV_StreamingMovies	object
NEW_AVG_Service_Fee	float64

XI. Implementing Other Models

With an enriched dataset post-feature engineering, a secondary Gradient Boosting Classifier is implemented. Model performance metrics are once again showcased. In this part, I additionally used cross validation to increase accuracy:

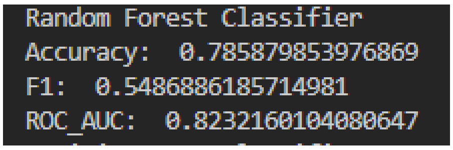


```
Gradient Boosting Classifier
Accuracy:  0.8016481109065063
F1:  0.5808341592151309
ROC_AUC:  0.8437556859829689
```

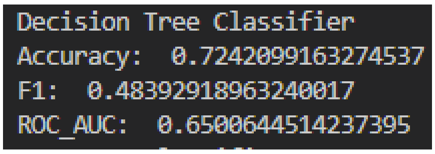
The reason of decreasing can be:

- If the new features added do not provide meaningful information or are irrelevant to the target variable (Churn in this case), they may introduce noise and confusion, leading to a decrease in model performance.
- They can potentially lead to overfitting, where the model becomes too specific to the training data and performs poorly on new, unseen data.

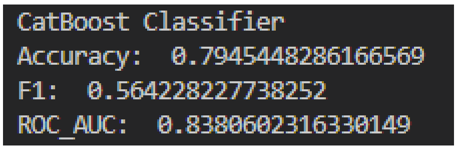
Then I predicted with other classifier methods to compare performances:



```
Random Forest Classifier
Accuracy:  0.785879853976869
F1:  0.5486886185714981
ROC_AUC:  0.8232160104080647
```

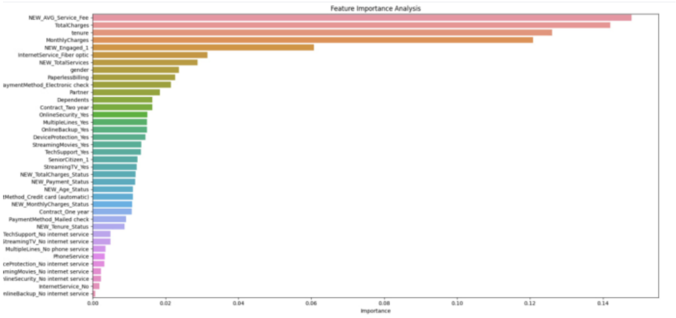


```
Decision Tree Classifier
Accuracy:  0.7242099163274537
F1:  0.48392918963240017
ROC_AUC:  0.6500644514237395
```



```
CatBoost Classifier
Accuracy:  0.7945448286166569
F1:  0.564228227738252
ROC_AUC:  0.8380602316330149
```

New feature importance plot:



XII. Evaluation

In conclusion, this code offers a deep dive into the realm of customer churn analysis for a telecommunications company. It makes data cleaning, exploratory analysis, modeling, and feature engineering to construct a comprehensive understanding of customer behavior. The resultant models not only predict churn but also provide interpretability through feature importance analysis. The code serves as a robust framework for ongoing analysis and adaptation to evolving business requirements.

XIII. Youtube Link

https://youtu.be/E_E7dXGsXWk?si=Es4MkUl1rswce2WO

XIV. Reference

- https://www.kaggle.com/datasets/blastchar/telco-customer-churn/data