Report: Customer Churn Prediction

Name: krishnandan sah kanu

Registration No: 12106329

Course: Btech CSE (data science)

School of Computer Science & Engineering

Lovely Professional University, Phagwara, Punjab.

Introduction:

The primary objective of this project is to analyze customer churn data for a fictional telecommunications company and to identify the factors that contribute to customer churn. By understanding these factors, the company can develop strategies to reduce churn, improve customer retention, and enhance overall business performance.

Data Description:

The dataset consists of 7,043 customer records and 21 features.

Data Cleaning & Preprocessing:

For data preprocessing I used python & numpy libraries.

1. **Data information:** Using info(), I am able to identify missing values and data types. The TotalCharges column had an incorrect data type. I converted it from object to float and filled the missing values with the mean.

2. **Missing Values:** Using isnull().sum(), able to identify missing values in every columns. However, there are no any missing values in the datasets.

3. **Duplicated values:** Using duplicated().sum(), able to identify duplicate values in the data. However, there are no duplicate values in the dataset.

4. **Encoding:** From sklearn library Using LabelEncoder() method to convert categorical data into numerical data Because, Machine learning algorithms works only on numerical data.

5. **Data Scaling:** from sklearn using minmaxscaler method to scale our data into small range.

Exploratory Data Analysis (EDA):

For EDA I used matplot & seaborn libraries.

1. **Distribution of Churn:** Using bar chart to represent the distribution of customer who have churned and those who have not. Out of 7048 customers, 1869 have churned while 5174 are still with the company.

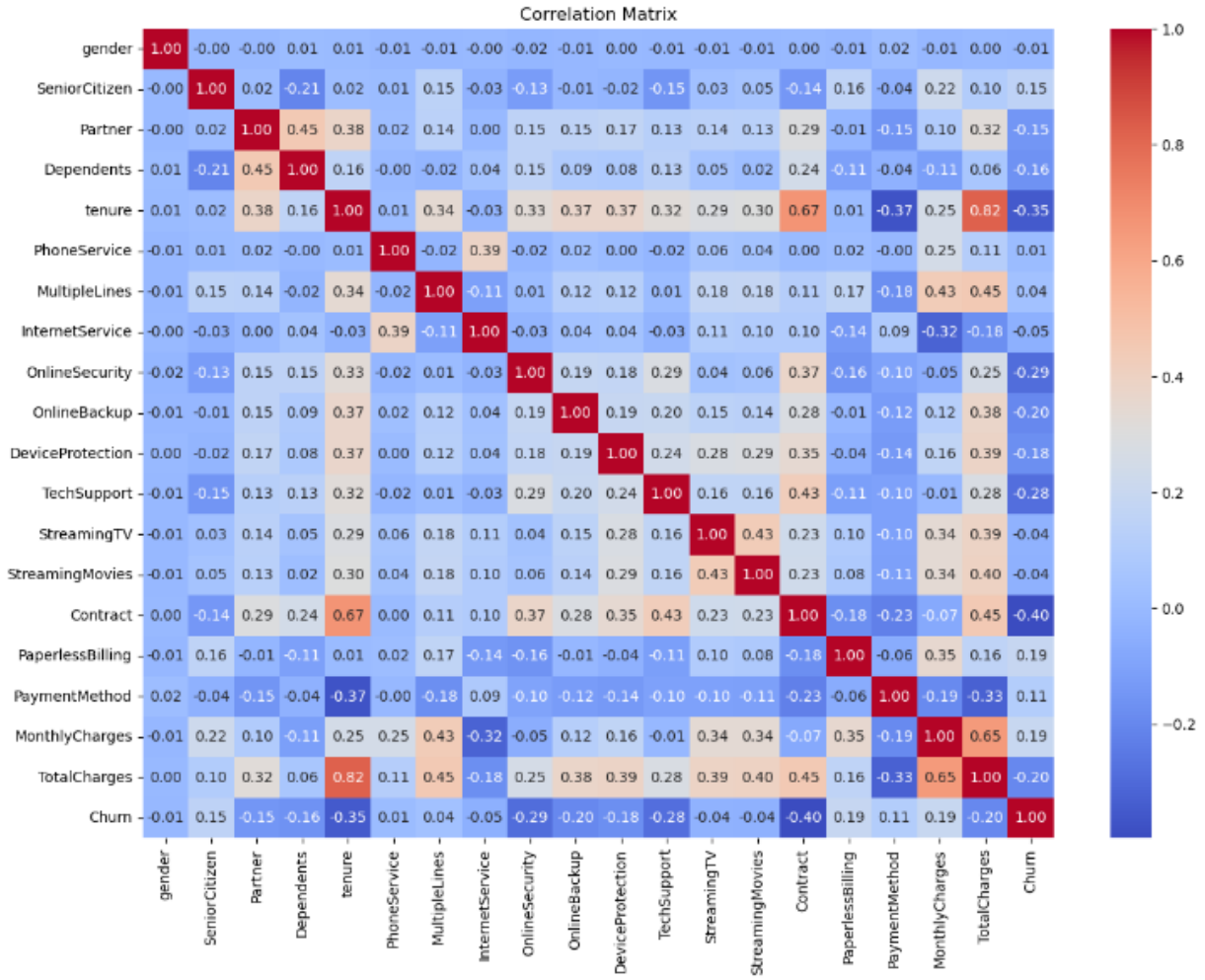
2. **Distribution of Churned Customer by Gender:** Using bar chart to represent the distribution of gender of customer who have churned. Out of 1869 Churned customers, 939 customers are female and 930 are male.

3. **Distribution of Churned Customer by Contract:** Using bar chart to represent the distribution of contract of customer who have churned. Out of 1869 Churned customers, 1655 customers had month-to-month contract, 166 customers had one-year contract & 48 customers had two-year contract.

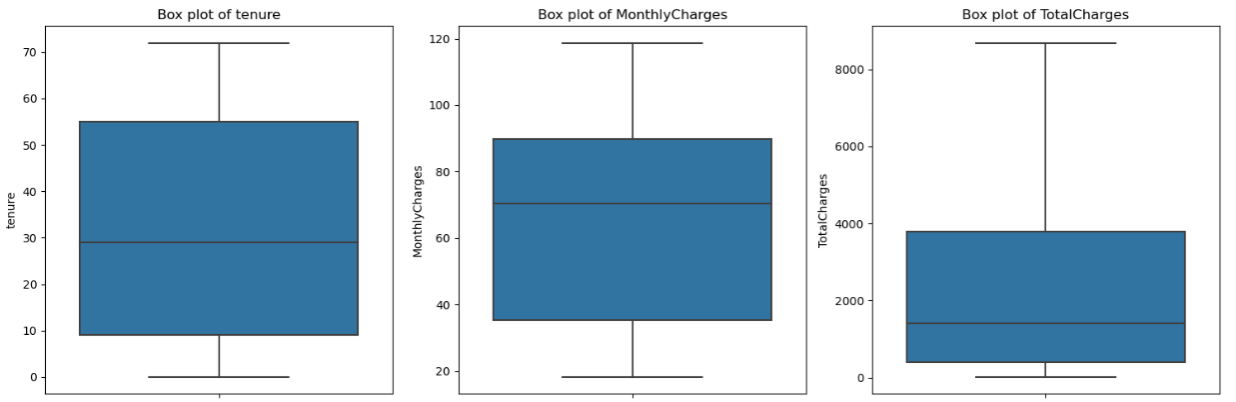
4. **Distribution of Churned Customer by Internet Services:** Using bar chart to represent the distribution of internet service of customer who have churned. Out of 1869 Churned customers, 1297 customers had fiber optic service, 459 customers had Dsl service & 113 customers had no any internet service.

5. **Distribution of Numerical Column:** Using Histogram chart, able to know the distribution of numerical data and skew() function, help to identify the skewness of data. tenure 0.239540
   MonthlyCharges -0.220524
   TotalCharges 0.962394

6. **Correlation:** using heatmap chart, able to identify the correlation between each and every columns.



7. **Outlier Detection:** Using boxplot chart, able to identify the outliers in the dataset. However, there is no outliers in the dataset.



Model Building:

Using sklearn library for model training & testing.

I have run multiple machine learning algorithms like Logistic Regression, SVM, Decision Tree, Random Forest, xg boost, multiple naïve bayes, bagging, adda boost & gradient boost.

Here is the performance of Models:

	Algorithm	Accuracy	Precision
3	LR	0.812633	0.704467
7	GBDT	0.800568	0.697674
2	DT	0.793471	0.681102
5	AdaBoost	0.792761	0.659649
8	xgb	0.793471	0.653333
4	RF	0.782825	0.640000
6	BgC	0.781405	0.637363
1	NB	0.669269	0.436447
0	SVC	0.728176	0.000000

Conclusion:

After applying all algorithms and performing hyperparameter tuning, we found that logistic regression achieved the best performance in terms of accuracy & precision. Therefore, we will deploy our model using this algorithm.

Here is the metrices of Logistic regression Model:

```
Accuracy: 0.8126330731014905
Precision: 0.7058823529411765
Recall: 0.5326370757180157
F1 Score: 0.6071428571428571
Confusion Matrix:
[[941  85]
 [179 204]]
```