

CUSTOMER CHURN PREDICTION

OBJECTIVE: Develop a machine learning model to predict customer churn based on historical customer data.

WORKING ON RAW DATA:

1. Firstly, Checking the missing values and treating/filling it by using median or imputation methods.
2. In addition to that, converting the categorical data to numeric value by using the advanced target encoding technique i.e **K- Fold Target Encoding**.
3. Furthermore, After doing the pre-required steps, building the predictive model without doing any preprocessing things (**Transformations, Normalization/scaling**) to know the nature of the data.
4. Finally, Predicting Whether the customer is going to renew the subscription or the customer is going to deactivate the subscription plan.

After building the model, The accuracy of each model is approx **50%**. Performance of the model is pathetic. Need to analyze the data and get into the insights. know the cause of getting 50% accuracy and rectify the errors(using Hyperparameter tuning) to improve the model performance.

EXPLORATORY DATA ANALYSIS:

CHECKING OUTLIERS:

OUTLIERS: Outliers are the data points that are significantly different from the rest of the dataset. They are often abnormal observations that skew the data distribution, and arise due to inconsistent data entry, or erroneous observations

Checking the outliers by using the IQR method and visualizing the outliers by using the boxplot.

If we detect outliers in the data. We can treat/can replace the value by using median.

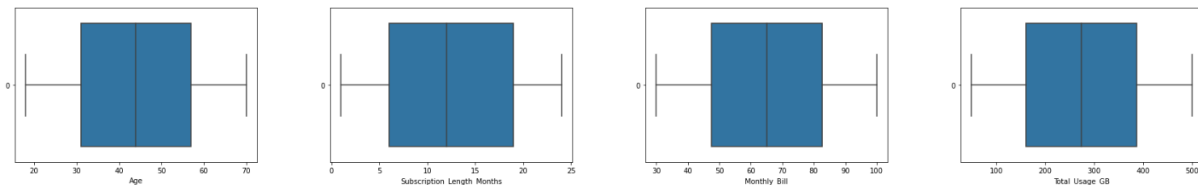
IQR METHOD:

```
1 columns = churn_outlier_detection_data.columns
2
3 for j in columns:
4     outliers_j = []
5     def outliers_iqr(sample):
6         data = sorted(sample)
7         q1 = np.percentile(sample,25)
8         q3 = np.percentile(sample,75)
9         IQR = q3 - q1
10        lwr_bound = q1 - (1.5*IQR)
11        upr_bound = q3 + (1.5*IQR)
12        for i in sample:
13            if (i<lwr_bound or i>upr_bound):
14                outliers_j.append(i)
15        return outliers_j
16    sample_outliers = outliers_iqr(x[j])
17    print("Outliers using IQR Method in {} : {}".format(j),sample_outliers)
```

```
Outliers using IQR Method in Age : []
Outliers using IQR Method in Subscription_Length_Months : []
Outliers using IQR Method in Monthly_Bill : []
Outliers using IQR Method in Total_Usage_GB : []
```

BOXPLOT METHOD:

```
1 # Initialize figure with 4 subplots in a row
2 fig, ax = plt.subplots(1, 4, figsize=(30,4))
3
4 # add padding between the subplots
5 plt.subplots_adjust(wspace=0.3)
6
7 for index,column in enumerate(churn_outlier_detection_data.columns):
8     sns.boxplot(data=churn_outlier_detection_data[column], ax=ax[index],orient='h')
9     ax[index].set_xlabel(column)
10
11 plt.show()
```



DISTRIBUTION OF THE DATA:

When analyzing data, it's important to understand the distribution of the data. The distribution refers to how the data is spread out or clustered around certain values or ranges. By examining the distribution, we can gain insights into the characteristics and patterns of the data, which can be useful in making informed decisions and predictions.

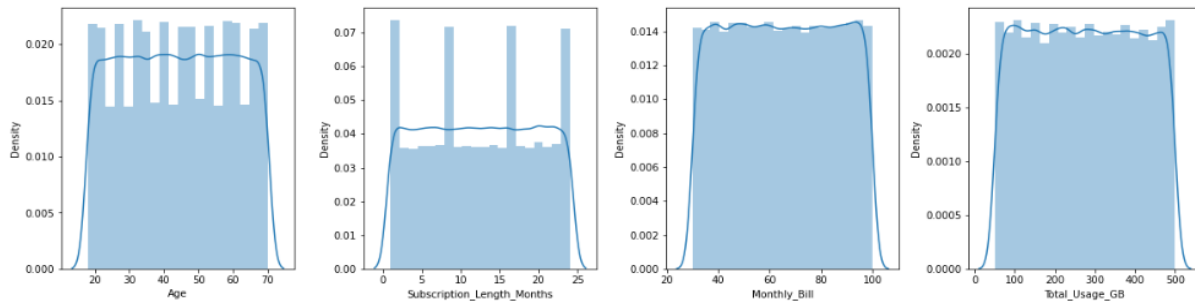
There are various types of data distributions, each with its own unique properties and implications. Understanding these distributions is a fundamental aspect of data analysis and can help us make more accurate and meaningful interpretations of the data.

SKEWNESS: Skewness is a measure of the asymmetry of a Distribution.

KURTOSIS: Kurtosis is a measure of the tailedness of a distribution.

PLOTS :

```
1 #Plotting the distribution of the data
2
3 churn_data_for_distribution = churn_data.drop(['CustomerID','Name','Location','Churn','Gender'],axis=1)
4
5 fig,ax = plt.subplots(1,4,figsize=(20,5))
6
7 plt.subplots_adjust(wspace=0.3)
8
9 for index,column in enumerate(churn_data_for_distribution.columns):
10     sns.distplot(churn_data_for_distribution[column], ax=ax[index],bins=20)
11     ax[index].set_xlabel(column)
12
13 plt.show()
```



SKEWNESS :

```
1 churn_data_for_distribution = churn_data.drop(['CustomerID','Name','Location','Churn','Gender'],axis=1)
2
3 def Skewness(data):
4     for column in data.columns:
5         skewness = data[column].skew()
6
7         if (skewness > -0.5 and skewness < 0.5):
8             print(f'Distribution of the {column} data is symmetrical and value is {skewness}')
9         elif ((skewness > -1) and (skewness < -0.5)) or (skewness < -1):
10             print(f'Distribution of the {column} data is negative skewed and value is {skewness}')
11         else:
12             print(f'Distribution of the {column} data is positive skewed and value is {skewness}')
13
14 Skewness(churn_data_for_distribution)
```

Distribution of the Age data is symmetrical and value is -0.002688580535339188
Distribution of the Subscription_Length_Months data is symmetrical and value is -0.0016554824859520909
Distribution of the Monthly_Bill data is symmetrical and value is -0.00032573400100265417
Distribution of the Total_Usage_GB data is symmetrical and value is 0.007113380196458567

KURTOSIS :

```
1 churn_data_for_distribution = churn_data.drop(['CustomerID','Name','Location','Churn','Gender'],axis=1)
2
3 def Kurtosis(data):
4     for column in data.columns:
5         kurtosis = data[column].kurtosis()
6
7         if (kurtosis > 3):
8             print(f'{column} is having a very long tail and thick tails and value is {kurtosis} i.e Leptokurtic')
9         elif (kurtosis < 3):
10             print(f'{column} is having a thin tail and stretched around the center and value is {kurtosis} i.e Platykurtic')
11         elif (kurtosis == 3):
12             print(f'{column} is having moderate in breadth, and curves are a medium peaked heights and value is {kurtosis} i.e Mesokurtic')
13
14 Kurtosis(churn_data_for_distribution)
```

Age is having a thin tail and stretched around the center and value is -1.1981801069446432 i.e Platykurtic
Subscription_Length_Months is having a thin tail and stretched around the center and value is -1.2056102064968002 i.e Platykurtic
Monthly_Bill is having a thin tail and stretched around the center and value is -1.2045683662909459 i.e Platykurtic
Total_Usage_GB is having a thin tail and stretched around the center and value is -1.201277818271164 i.e Platykurtic

PLOTTING THE GRAPH TO CHECK THE MIXED DATA:

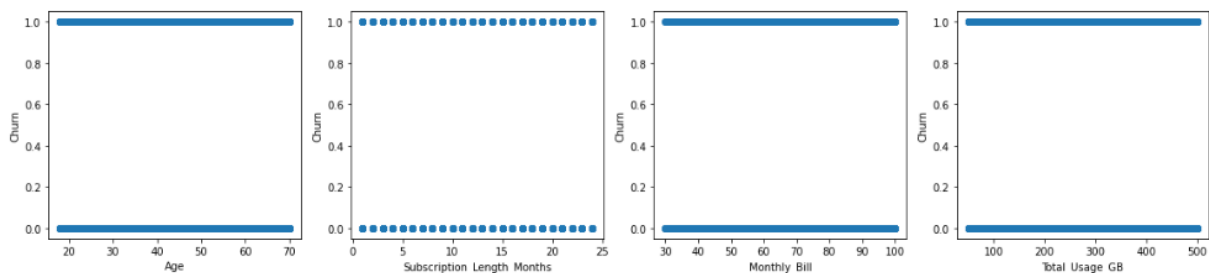
Below plot is to analyze the data whether data points are mixed or separated.

```
1 # plot for differentiating the target variable with independent variable to know the
2 # importance and predictive power of the variable
3
4 fig, axes = plt.subplots(2, 3, figsize=(12, 5))
5 not_churn = x[churn_data.Churn==0]
6 churn = x[churn_data.Churn==1]
7 ax = axes.ravel()
8
9
10 # Used the independent variables length i.e x
11 for i in range(len(x.columns)):
12     bins=40
13     ax[i].hist(churn.iloc[:,i], color='r', bins=bins, alpha=.5)
14     ax[i].hist(not_churn.iloc[:,i], color='b', bins=bins, alpha=.3)
15     ax[i].set_title(x.columns[i])
16 plt.tight_layout()
17 plt.show()
```



PLOTTING THE GRAPH WITH RESPECT TO TARGET VARIABLES:

```
1 #Scatter plot
2 Target_var = churn_data['Churn']
3
4 fig = plt.figure(figsize=(20,4))
5
6 for i,column in enumerate(churn_data_for_distribution.columns):
7     plt.subplot(1,4,i+1)
8     plt.scatter(churn_data_for_distribution[column],Target_var)
9     plt.xlabel(column)
10    plt.ylabel(Target_var.name)
11
```



1. Each variable has mixed data. It doesn't differentiate whether it is a churn or not.

2. Each variable can't differentiate the target variable. By seeing the above visualization we cannot expect good results in the outcomes.

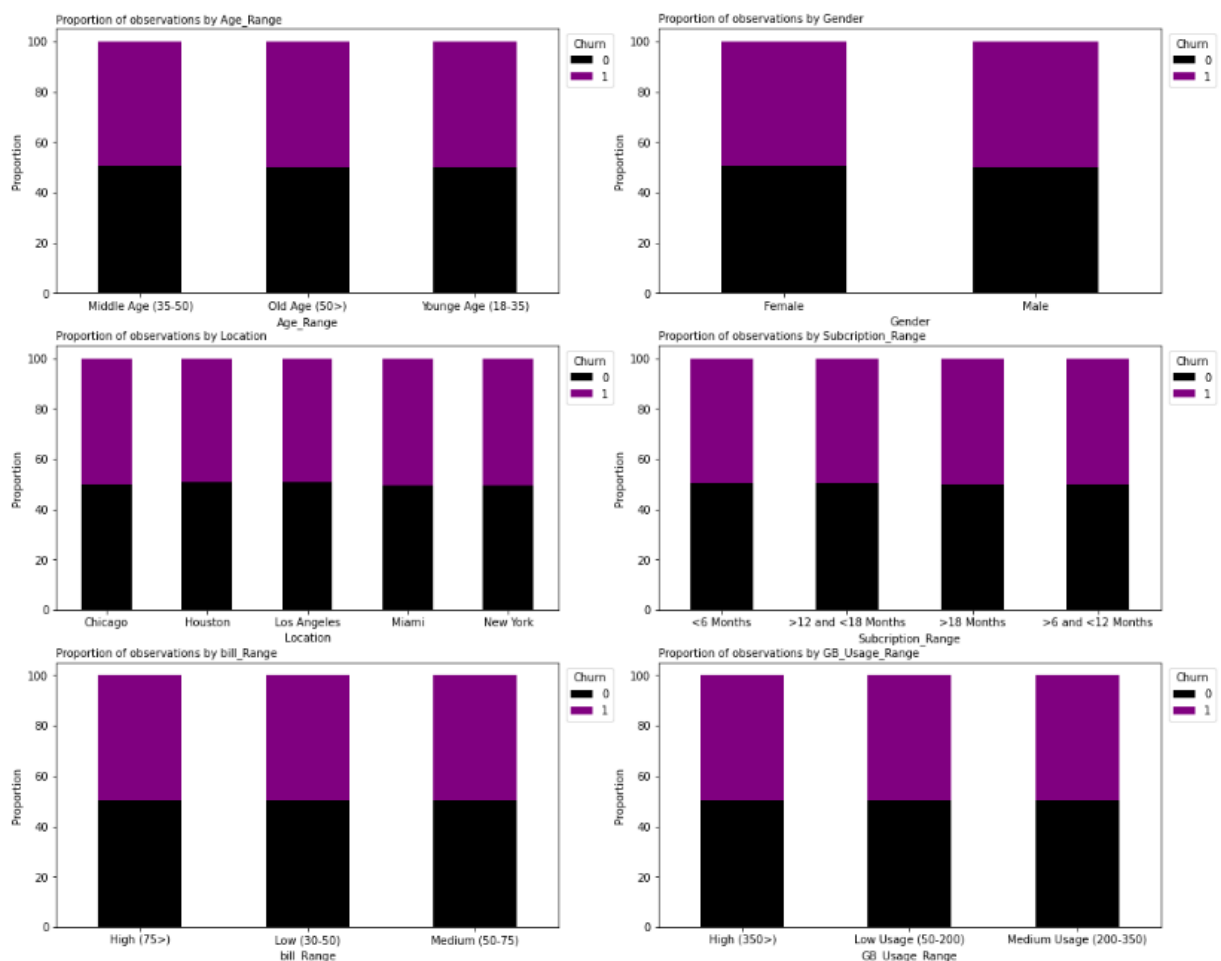
NOTE: Each variable and record has almost the same values for both labels i.e 0 and 1. It doesn't distinguish whether it is a 0 or 1 because it has mixed data for both labels.

PLOTTING THE GRAPH TO KNOW THE PREDICTIVE POWER OF EACH VARIABLES:

As shown below, each bar is a category of the independent variable, and it is subdivided to show the proportion of each response class

We can infer the following conclusions by analyzing the attributes:

We do not expect Age, Gender, Location, Subscription Range, Bill Range and Gb usage to have significant predictive power. A similar percentage of churn is shown for different levels.



MODEL BUILDING:

DATA: Balanced data, Follows almost normal distribution. Doesn't have outliers and missing values

NATURE OF DATA: Each variable and each record has almost the same values for both labels i.e 0 and 1. So, It's very hard to distinguish whether it is a 0 or 1. So we cannot expect good results in the outcomes.

MODEL: So, Tried with diff models and got approx 50% accuracy for each model. Tried the hyperparameter tuning by using the GridSearch but still the model is giving 50% accuracy.

The reason to select **DecisionTreeClassifier** is "Every model trained with train data and predicted with train data and noted the accuracy of each model. Only DecisionTreeClassifier reached the approx 98% when we trained and predicted with train data. So, Chooosed the final model **as DecisionTreeClassifier**.

CICD PIPELINE:

Loading the data directly from github. Created own modules and packages(Separate pipelines for both train data and for validation data) for own functionality i.e For Splitting the data, For converting the categorical to numerical values and For normalization.

Automated the code to save the test data which is coming from the user interface web app and saving the test data into the same file in **github** for future purpose i.e Saving the test data to train the model continuously on test data as well.

WEB PAGE:

Created a front-end web page by using HTML,CSS to get the data from the user. Apply data validation and pattern check on the text boxes to get the right data to the model.

WEB APP LINK: <https://tetflix.onrender.com>

TECHNOLOGIES USED:

Programming Languages:	Python Front-End : HTML, CSS.
Libraries:	Numpy, Pandas, Seaborn, Sklearn, Keras
Algorithms:	Machine Learning Algorithms
Repo:	Github
Web Framework:	Flask
Cloud Deployment:	RENDER.COM (Cloud Application Hosting)

FRONT END WEB PAGE:

TETFLIX

ENJOY THE EXPERIENCE OF REAL WORLD

ENTER CUSTOMER ID (ONLY NUMBERS)

ENTER CUSTOMER NAME

ENTER AGE

ENTER SUBSCRIPTION PLAN






ENTER MONTHLY BILL

ENTER TOTAL USAGE GB

SELECT GENDER

SELECT LOCATION

GET THE DETAILS



TETFLIX: Tetflix offers a superior internet experience to explore your digital life. Tetflix fiber is the technology of the future. It offers the ultimate broadband experience to surf, stream, game and work.