Customer Churn Prediction

By Samarjeet Singh Chhabra

Overview

This project focuses on predicting customer churn in a telecommunications company using machine learning techniques, specifically a neural network model. Customer churn, the act of customers switching service providers or canceling their subscriptions, can lead to substantial revenue loss for businesses.

In this project, we've developed a predictive model that identifies customers likely to churn. This enables proactive measures to retain customers, reducing revenue loss. The neural network model used is trained on historical customer data.

Business Context

Customer churn, also known as customer attrition, is a significant concern for businesses in various industries. It refers to customers discontinuing their usage of a company's product or service. In our business, Sunbase, understanding and mitigating customer churn is crucial for long-term sustainability and growth.
The benefits of an effective customer churn prediction model for our business include:

Improved Customer Retention: Identification of at-risk customers enables tailored retention strategies, reducing churn.

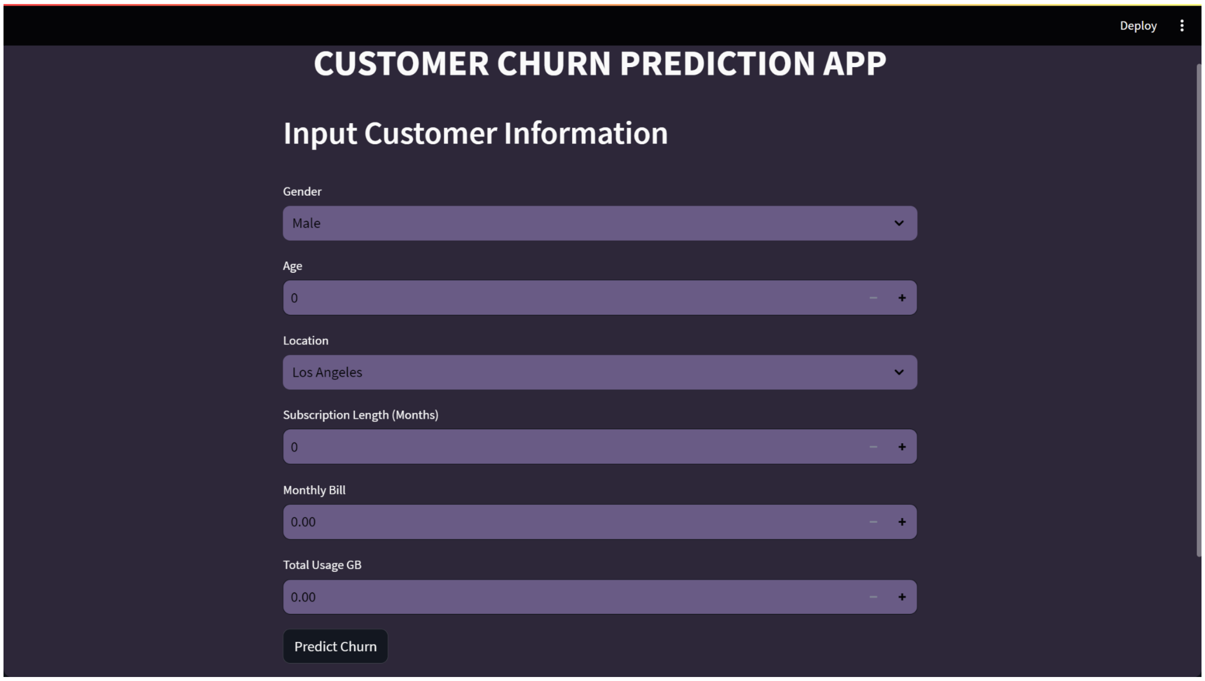
Cost Savings: Lower churn rates result in cost savings related to customer acquisition and onboarding.

Enhanced Customer Satisfaction: Proactive engagement and personalized offers improve the overall customer experience, increasing satisfaction and loyalty.

Revenue Growth: Retaining customers who might otherwise churn contributes to revenue growth and long-term profitability.

Web Application Overview

- Made with Streamlit.
- It is Simple and User friendly.
- Takes appropriate Data type as input and gives Prediction instantly.



CUSTOMER CHURN PREDICTION APP

Input Customer Information

Gender: Male

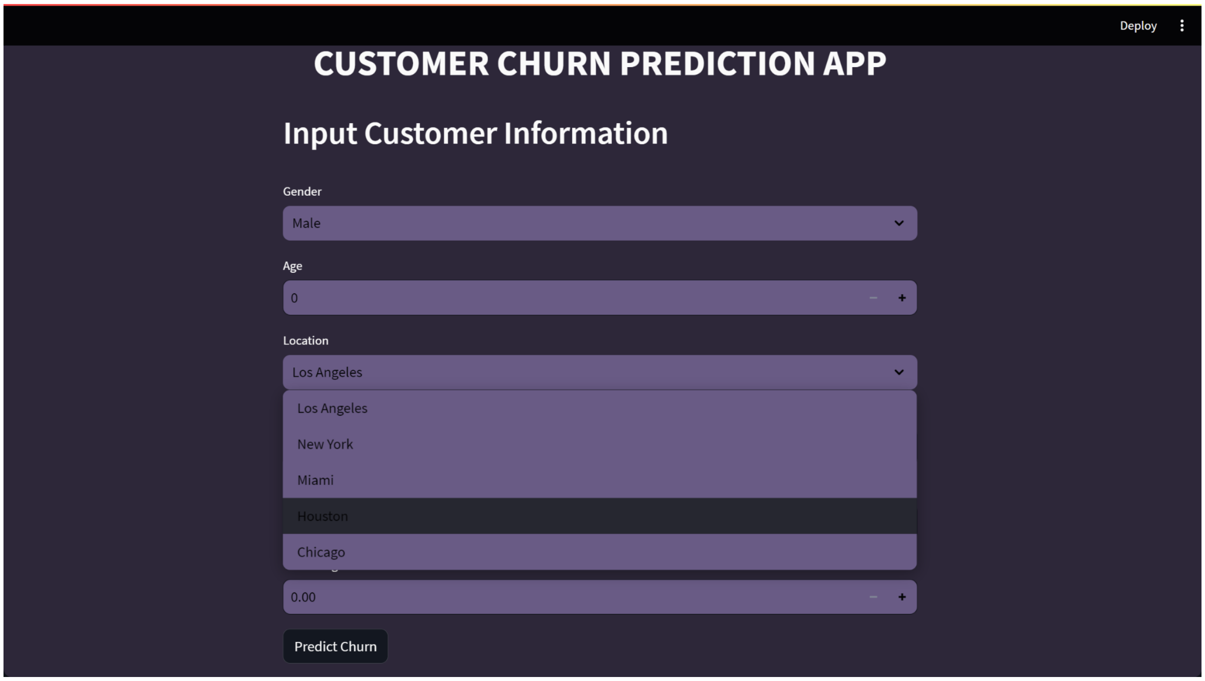
Age: 0

Location: Los Angeles

Subscription Length (Months): 0

Monthly Bill: 0.00

Total Usage GB: 0.00



CUSTOMER CHURN PREDICTION APP

Input Customer Information

Gender: Male

Age: 0

Location: Los Angeles

0.00

Data Understanding

Our dataset comprises 100,000 observations and 9 columns/features. These columns represent essential customer information and their churn status.

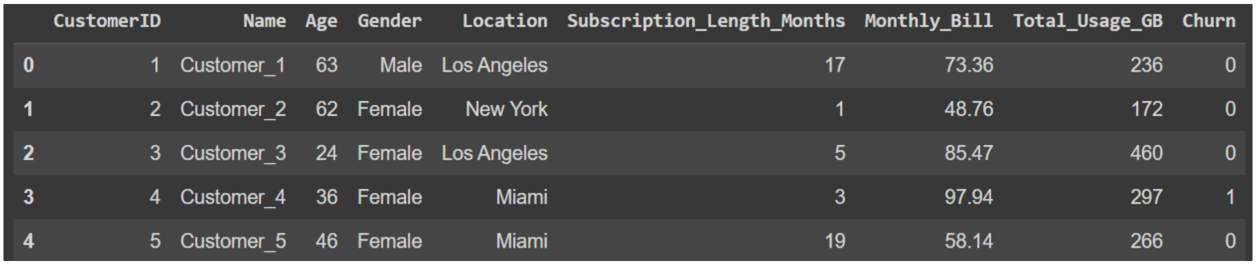
Here's a brief explanation of each column:
CustomerID: A unique identifier for each customer.
Name: The customer's name or identifier.
Age: The age of the customer.
Gender: The customer's gender (Male/Female).
Location: The customer's location (e.g., Los Angeles, New York, Miami).
Subscription_Length_Months: The duration of the customer's subscription in months.
Monthly_Bill: The monthly billing amount for the customer.
Total_Usage_GB: The total data usage in gigabytes (GB) by the customer.
Churn: The target variable, indicating whether the customer has churned (1) or not (0).

- Churn = 1 implies that the customer has discontinued the service.
- Churn = 0 implies that the customer is still using the service.

This dataset serves as the foundation for our customer churn prediction model.

Data Overview

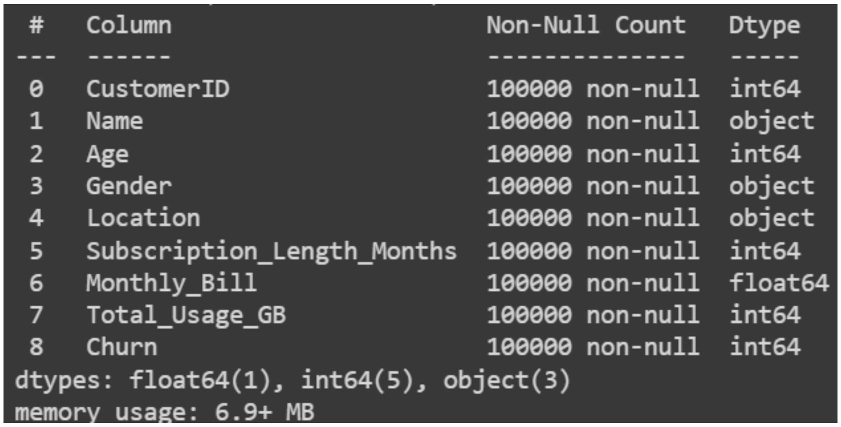
1. Dataset Head



	CustomerID	Name	Age	Gender	Location	Subscription_Length_Months	Monthly_Bill	Total_Usage_GB	Churn
0	1	Customer_1	63	Male	Los Angeles	17	73.36	236	0
1	2	Customer_2	62	Female	New York	1	48.76	172	0
2	3	Customer_3	24	Female	Los Angeles	5	85.47	460	0
3	4	Customer_4	36	Female	Miami	3	97.94	297	1
4	5	Customer_5	46	Female	Miami	19	58.14	266	0

2. Null Values check

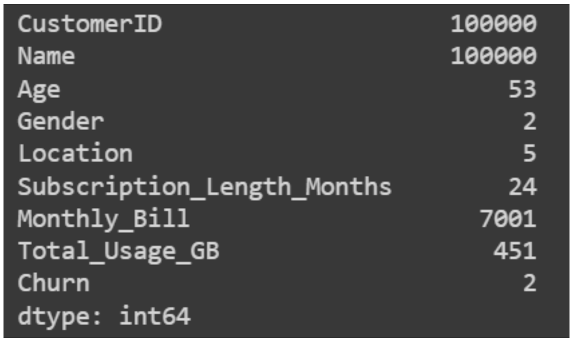
We Have no Null values in our dataset, All columns are non-null.



```
 #   Column                      Non-Null  Count   Dtype
---  ------                      --------------   -----
 0   CustomerID                  100000 non-null  int64
 1   Name                        100000 non-null  object
 2   Age                         100000 non-null  int64
 3   Gender                      100000 non-null  object
 4   Location                    100000 non-null  object
 5   Subscription_Length_Months  100000 non-null  int64
 6   Monthly_Bill                100000 non-null  float64
 7   Total_Usage_GB              100000 non-null  int64
 8   Churn                       100000 non-null  int64
dtypes: float64(1), int64(5), object(3)
memory usage: 6.9+ MB
```

3. Unique values

We have 3 categorical columns and others are numerical.

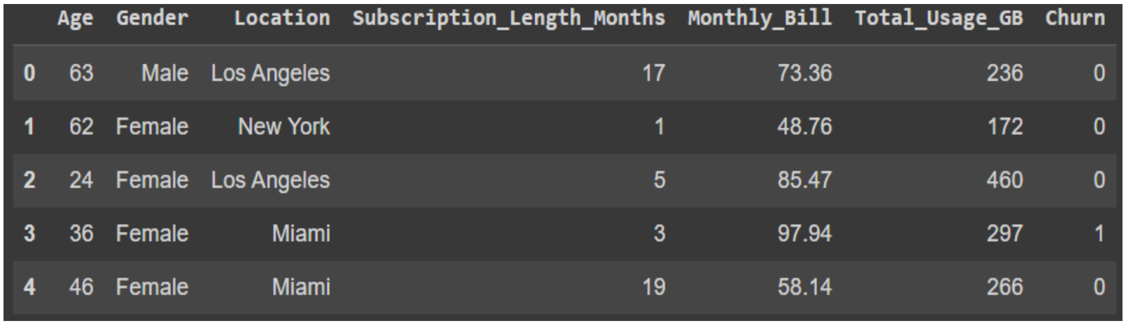


```
CustomerID                    100000
Name                          100000
Age                               53
Gender                             2
Location                           5
Subscription_Length_Months        24
Monthly_Bill                    7001
Total_Usage_GB                   451
Churn                              2
dtype: int64
```

Data Cleanup and Feature engineering

1. Feature Deletion

Since CustomerID and Name have no effect on the target variable so we will drop this column from our dataset.



	Age	Gender	Location	Subscription_Length_Months	Monthly_Bill	Total_Usage_GB	Churn
0	63	Male	Los Angeles	17	73.36	236	0
1	62	Female	New York	1	48.76	172	0
2	24	Female	Los Angeles	5	85.47	460	0
3	36	Female	Miami	3	97.94	297	1
4	46	Female	Miami	19	58.14	266	0

2. Feature Creation

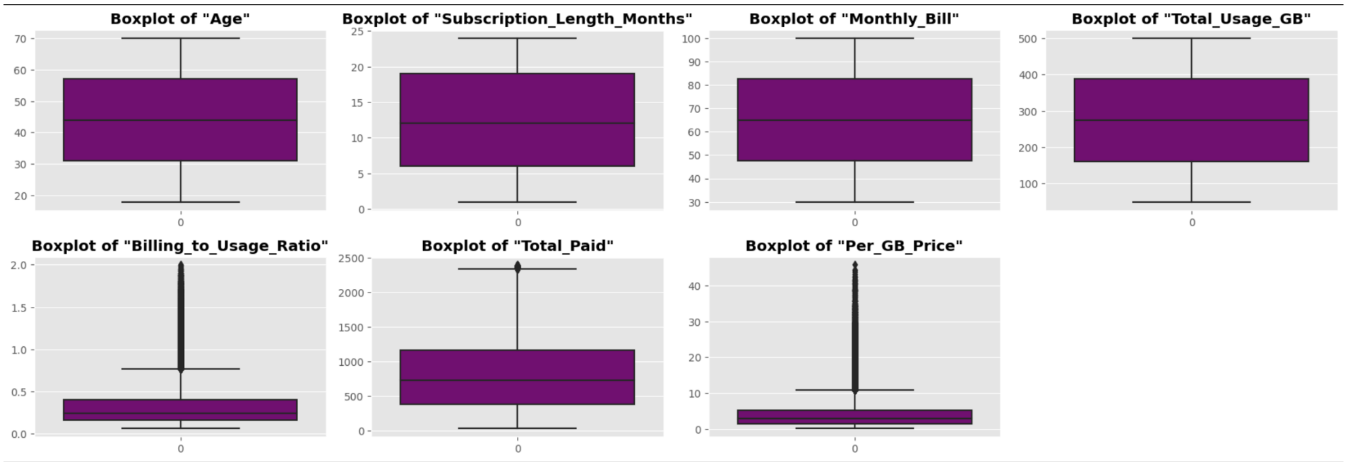
We created Dummies of Gender and Location columns and created new features as Short term and Long term Subscriptions,Age group labels, Billing to Usage Ratio, Total Paid(amount in dollars) and Per_GB_Price. Ended up with 5 rows × 23 columns data frame including target variable Churn.

- **independent_features**= ['Gender_Male', 'Gender_Female', 'Location_Chicago', 'Location_Houston', 'Location_Los Angeles', 'Location_Miami', 'Location_New York', 'Age_0_20', 'Age_21_30', 'Age_31_40', 'Age_41_50', 'Age_51_60', 'Age_61_200', 'Subscription_Category_Short-Term', 'Subscription_Category_Long-Term', 'Age', 'Subscription_Length_Months', 'Monthly_Bill', 'Total_Usage_GB', 'Billing_to_Usage_Ratio', 'Total_Paid', 'Per_GB_Price']

- **dependent_features**=['Churn']

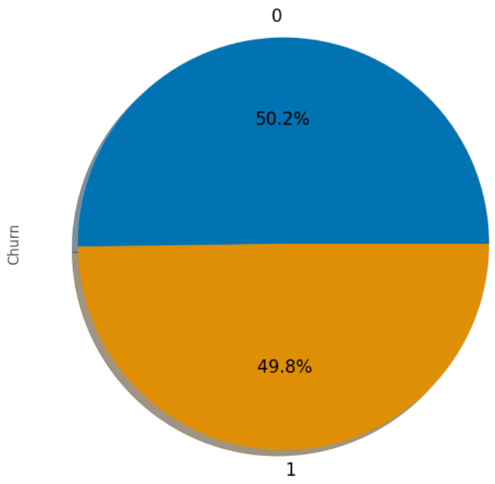
3. Outlier detection

In some cases, outliers can represent meaningful data points and removing them may lead to loss of important information. As we checked and did not find any abnormal range of each column. Also There are no such impossible values in our dataset, ex- 0 Age, 0 Monthly_Bill or values in minus in age. All these situations are errors in data collection.

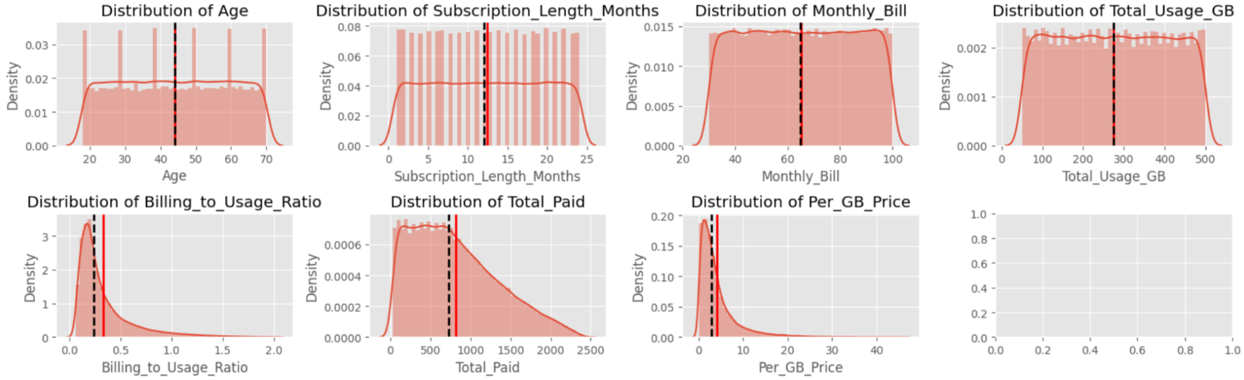


4. Class Imbalance

No class imbalance was seen.



5. Transformation



Applied on features

Billing_to_Usage_Ratio → Log Transformation

Total_Paid → Square Root Transformation

Per_GB_Price → Log Transformation

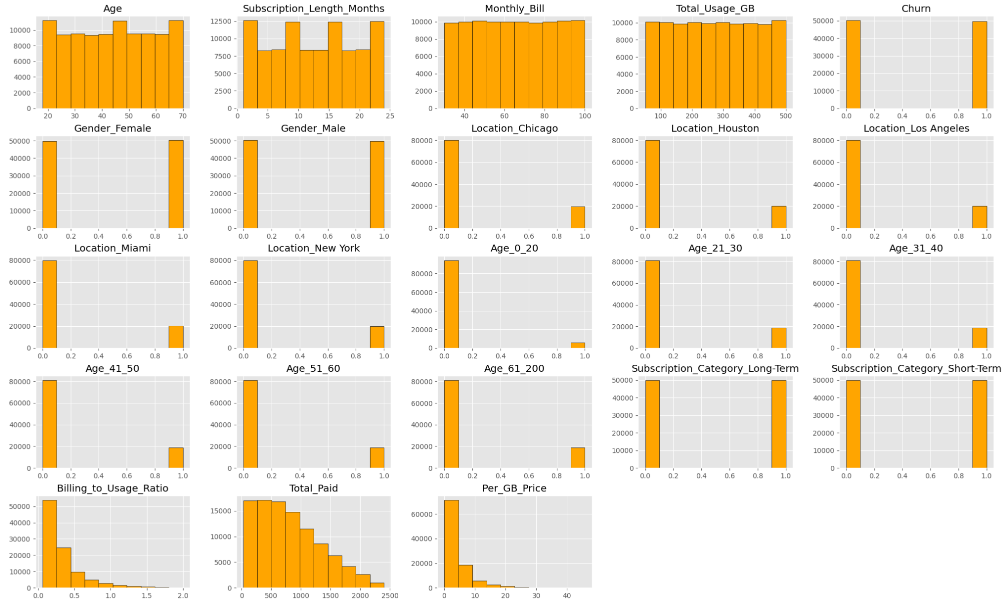
6. Feature Scaling

Applied MinMaxScaler on Train and test data, only on numerical features.

Exploratory Data Analysis

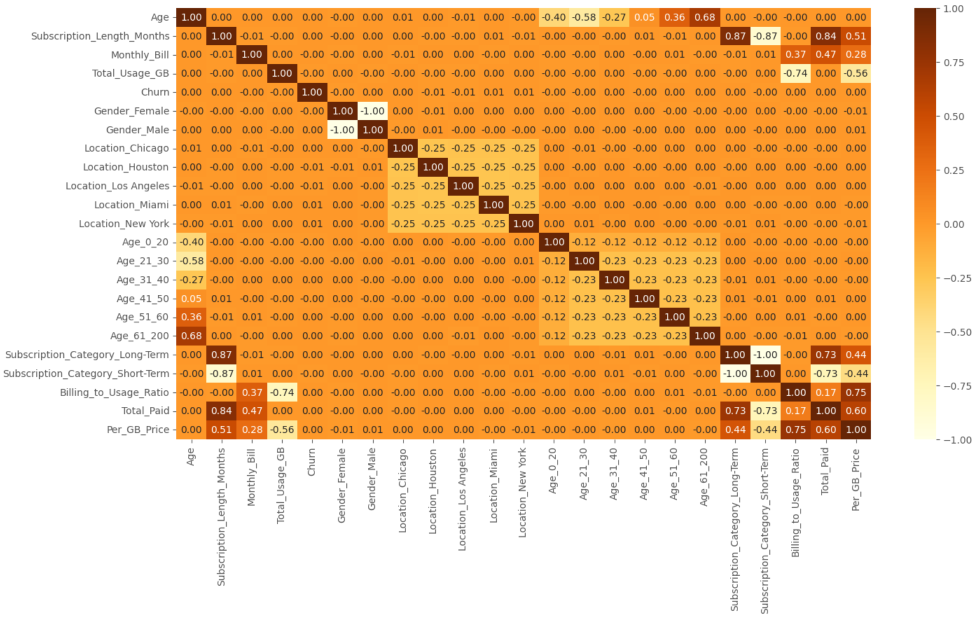
1. Histogram

We can see the distribution of all the features in the dataset.



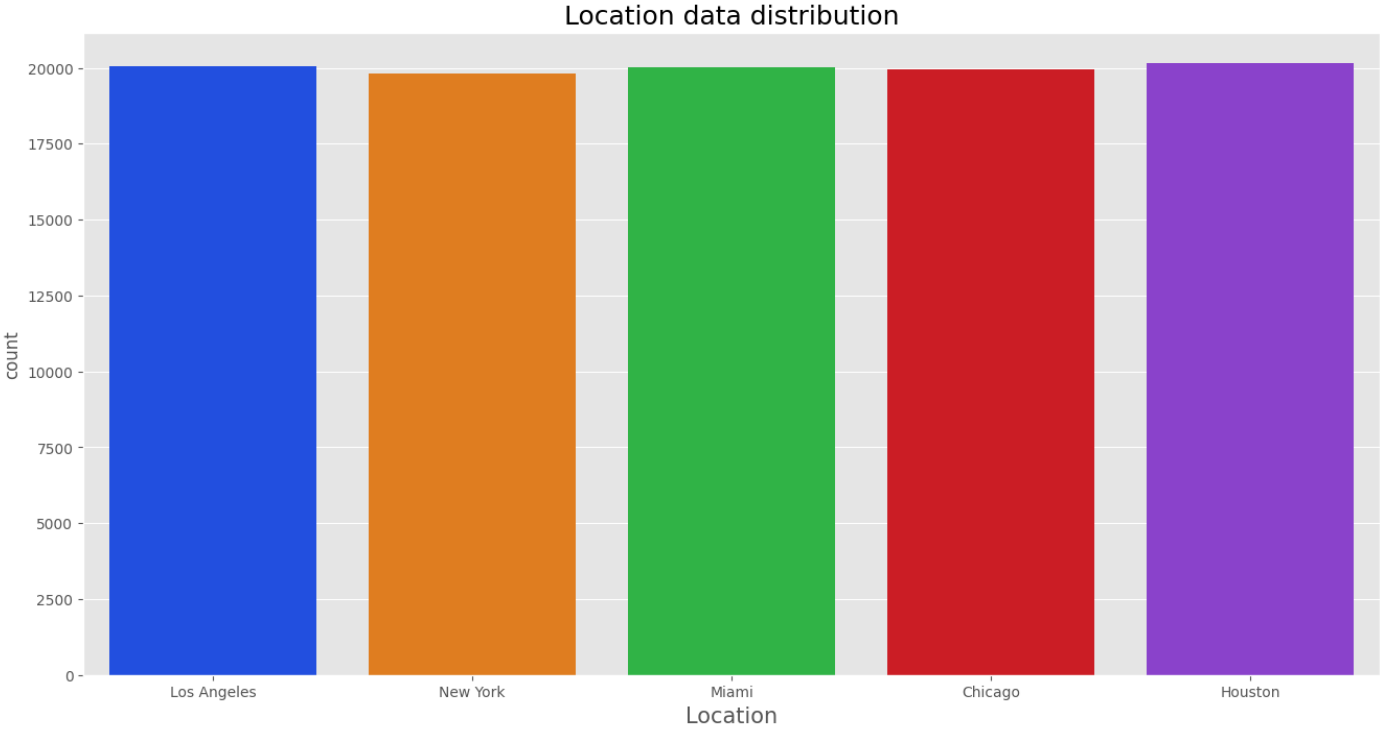
2. Correlation Heatmap

No such correlation found between the actual provided features in the dataset.



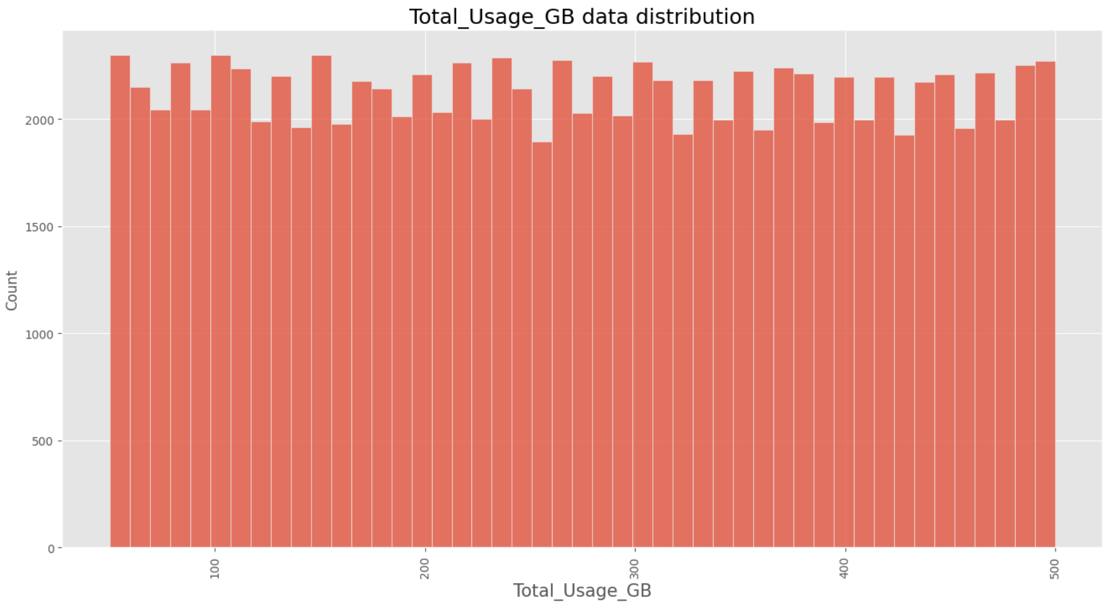
3. Distribution of Location

All locations are almost in equal number of counts in the dataset, but the max available data is of Houston city.



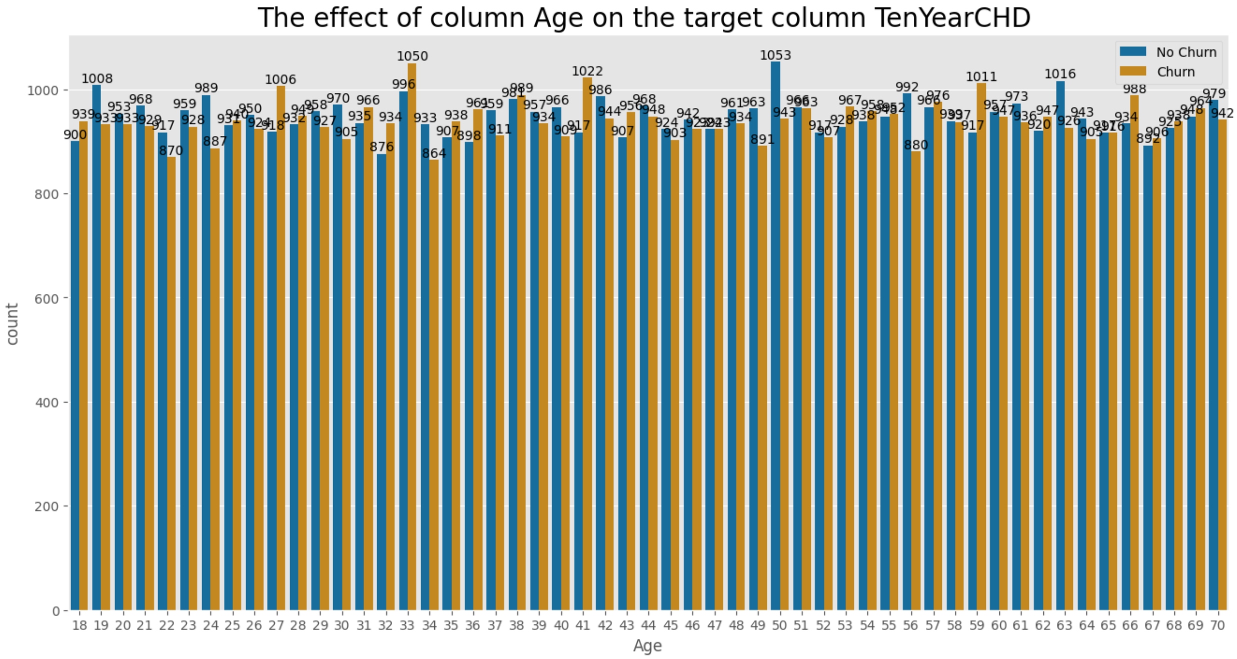
4. Total_Usage_GB

No such noticeable trend is observed.



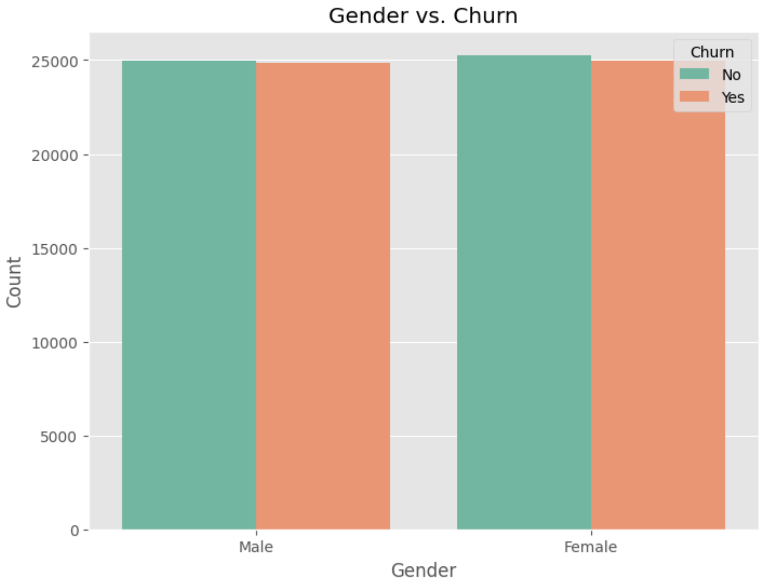
5. Churn vs Age

The age at which maximum people have Churned is 33 and age when least customers churned is 34.



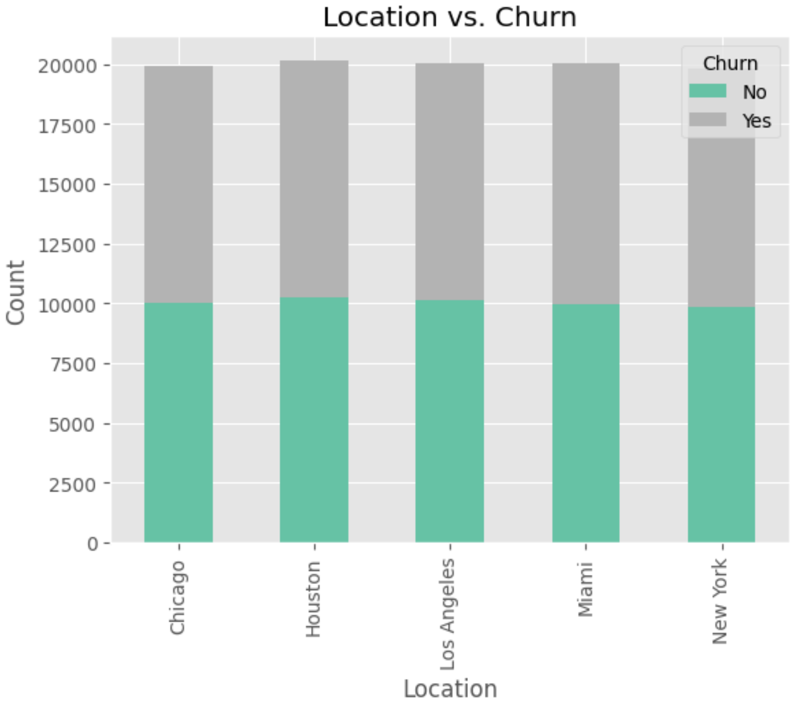
6. Churn vs Gender

We can see in our dataset, Females have churned little more as compared to the total number of male customers who have churned. And in case of Not churned count females are also more.



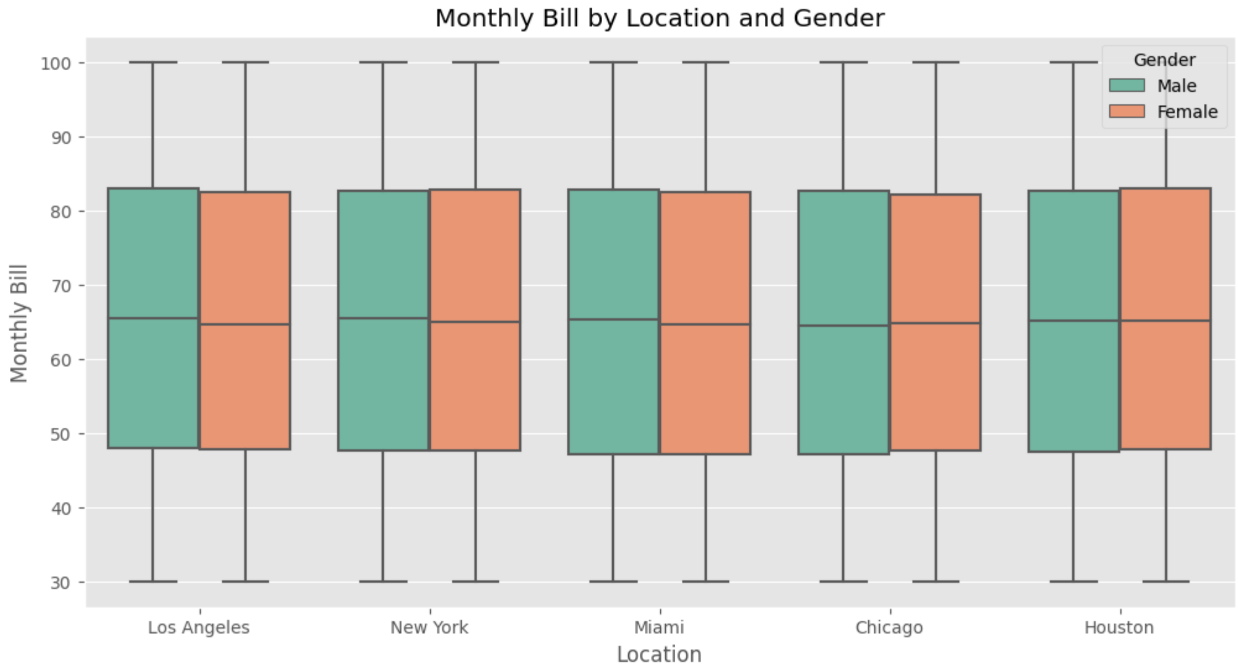
7. Churn vs Location

We can see in our dataset, Maximum customers who churned are from Houston as we have more customer's data from houston, and least count of customers who churned are from New York.



8. Monthly Bill by Location and Gender

Monthly Bill comparison by location and gender



Model Implementation

1. Evaluation Matrix Selection

- Since we are dealing with data related to customer retention and churn problem, we choose precision over recall because False positives in churn prediction can sometimes lead to unnecessary retention efforts, which can be costly. Precision is crucial in situations where the cost of retention efforts is significant and should be spent wisely.
- In other words, it doesn’t matter whether we raise a false alarm but the actual positive cases should not go undetected
- Considering these points in mind, it is decided that I will use ***Precision*** as the model evaluation metric.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

2. Model Selection

- We are working on a binary classification problem.
- Here we can start with a simple model, as a baseline model.
- Try other standard binary classification models and then some Deep learning models. We will also use ensemble models, with hyperparameter tuning to check whether they give better predictions.

  Trained 5 Models which are:-

  1. **Logistic Regression**
  2. **XGboost**
  3. **Random Forest Classifier**
  4. **Support Vector Classifier**
  5. **Neural Network**

3. Train, Test and Validation Data

- **Train** - 90000 rows
- **Test** - 10000 rows
- **Validation** - 20000 rows of Train data

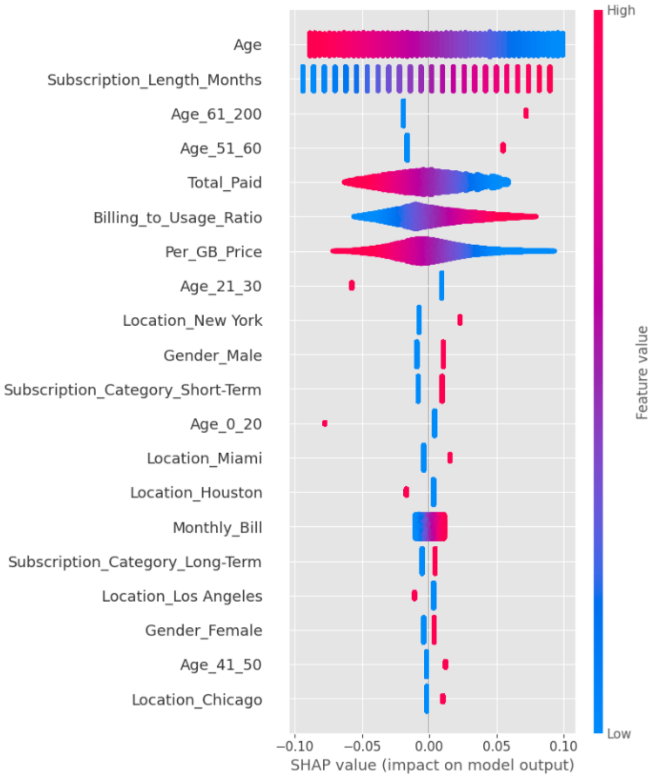
Model Explainability

- Model explainability refers to the concept of being able to understand the machine learning model. For example – If a healthcare model is predicting whether a patient is suffering from a particular disease or not. The medical practitioners need to know what parameters the model is taking into account or if the model contains any bias. So, it is necessary that once the model is deployed in the real world. Then, the model developers can explain the model.

  Popular techniques for model explainability:

  - 1. ELI-5
  - 2. LIME
  - 3. SHAP

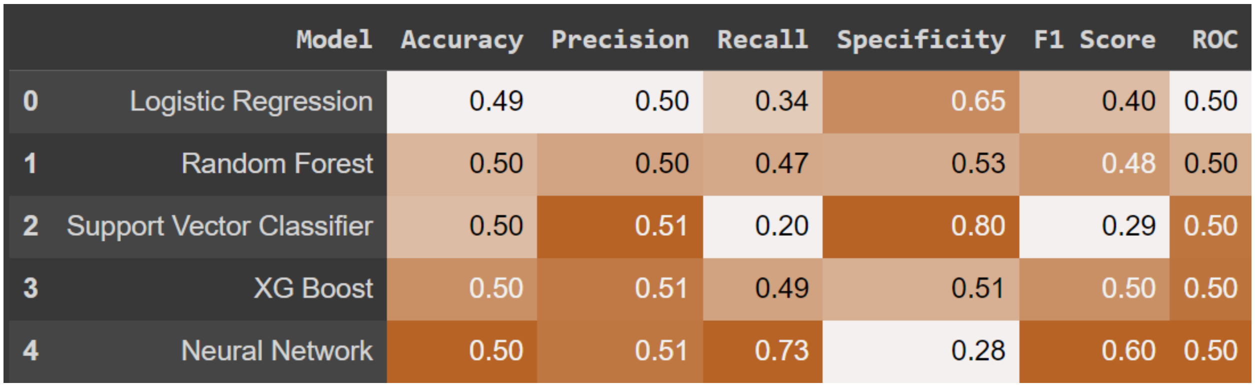
SHAP on Logistic regression model.



Model Evaluation

Report On Test Data

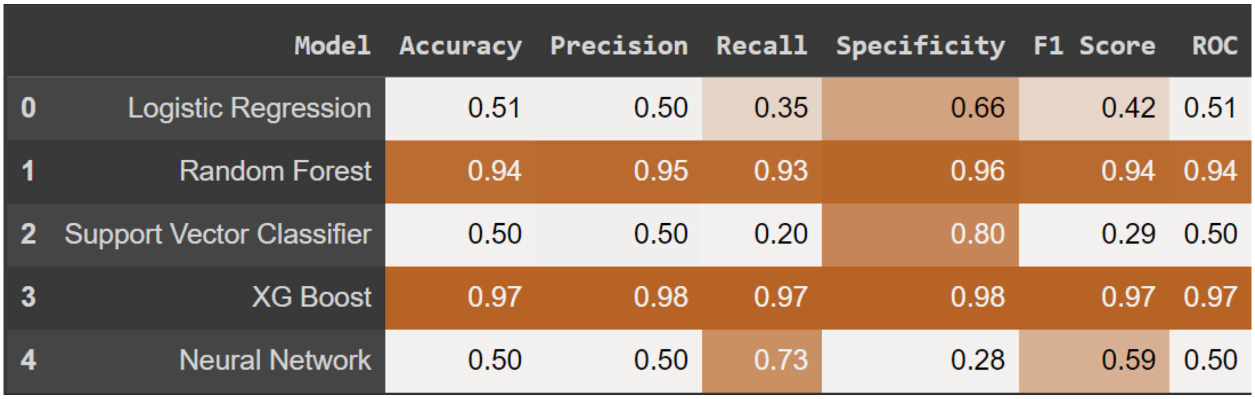
Best **Precision** and **Accuracy** is of ***SVC, XG boost*** and ***Neural Network*** but **F1 Score** is best of **Neural Network** for Test data.
Best model is the **Neural Network** for Test data.



	Model	Accuracy	Precision	Recall	Specificity	F1 Score	ROC
0	Logistic Regression	0.49	0.50	0.34	0.65	0.40	0.50
1	Random Forest	0.50	0.50	0.47	0.53	0.48	0.50
2	Support Vector Classifier	0.50	0.51	0.20	0.80	0.29	0.50
3	XG Boost	0.50	0.51	0.49	0.51	0.50	0.50
4	Neural Network	0.50	0.51	0.73	0.28	0.60	0.50

Report On Train Data

Best **Precision** and **Accuracy** is of ***XG boost*** and **Random Forest** but overall **precision**, **accuracy** and **F1 Score** is best of **Random Forest** for Train data.
Best model is **Random Forest** for Train data.



	Model	Accuracy	Precision	Recall	Specificity	F1 Score	ROC
0	Logistic Regression	0.51	0.50	0.35	0.66	0.42	0.51
1	Random Forest	0.94	0.95	0.93	0.96	0.94	0.94
2	Support Vector Classifier	0.50	0.50	0.20	0.80	0.29	0.50
3	XG Boost	0.97	0.98	0.97	0.98	0.97	0.97
4	Neural Network	0.50	0.50	0.73	0.28	0.59	0.50

Summary and Conclusion

The Best Model for live data is NEURAL NETWORK which shows maximum precision on test data.
Other results and summary are as follows:-

- I trained 5 Models which are Neural Network, Random Forest Classifier, Support Vector Classifier, XGboost and Logistic Regression.
- All performed similarly but the fastest trainable is Neural Network and it takes less time to process output with test data **accuracy** of **0.5006.**
- Neural network gives **Precision** of **0.51** on **class 1** which is ('Churn'='yes') and **f1-score** of **0.50**.
- We recommend the use of **Neural Network** in real world data processing as it gives the best optimal performance in less time.

Challenges faced

- Feature engineering.
- Creating New features
- Fine tuning the models and running Slow XG boost Model.
- Choosing model explainability techniques.