Name : Varad Yergole +919960893075

[varadyergole.7@gmail.com](mailto:varadyergole.7@gmail.com)

Customer Churn Prediction Report :

Problem Statement : Develop a machine learning model that predicts customer chunk based on dataset provided.

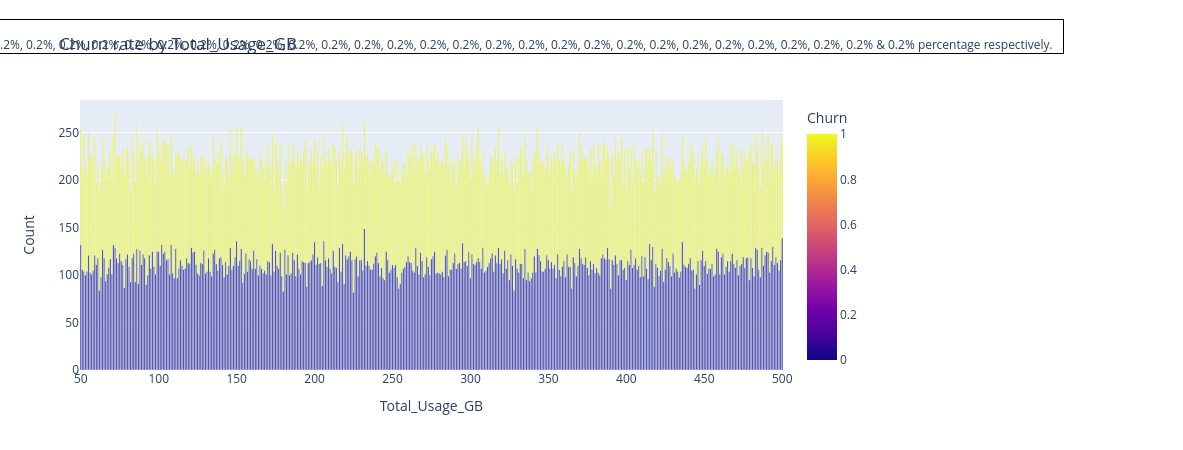
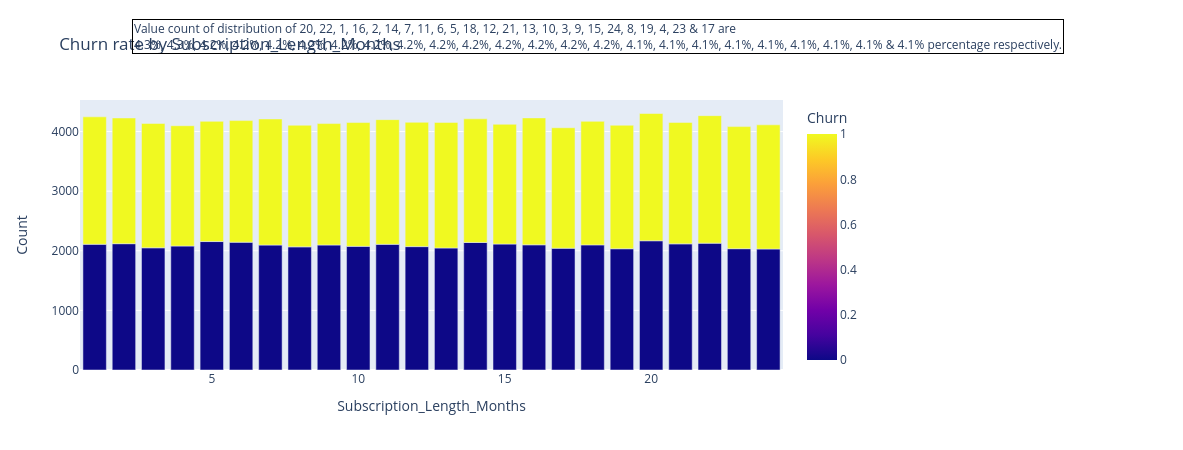
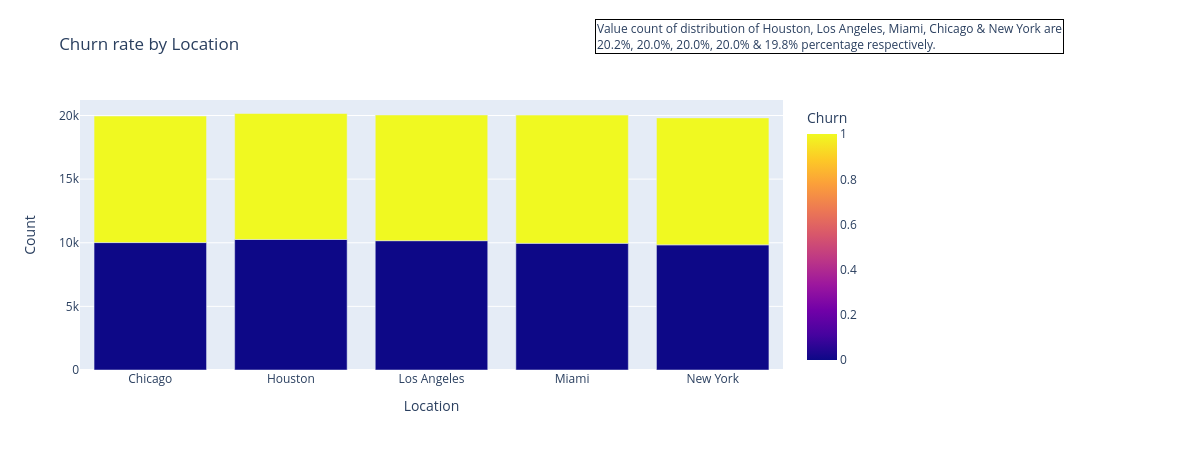
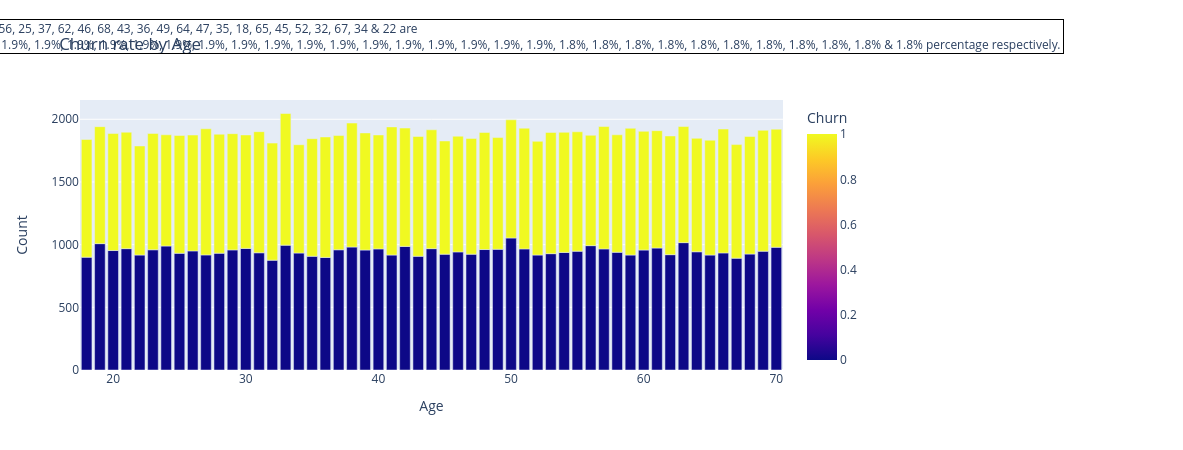
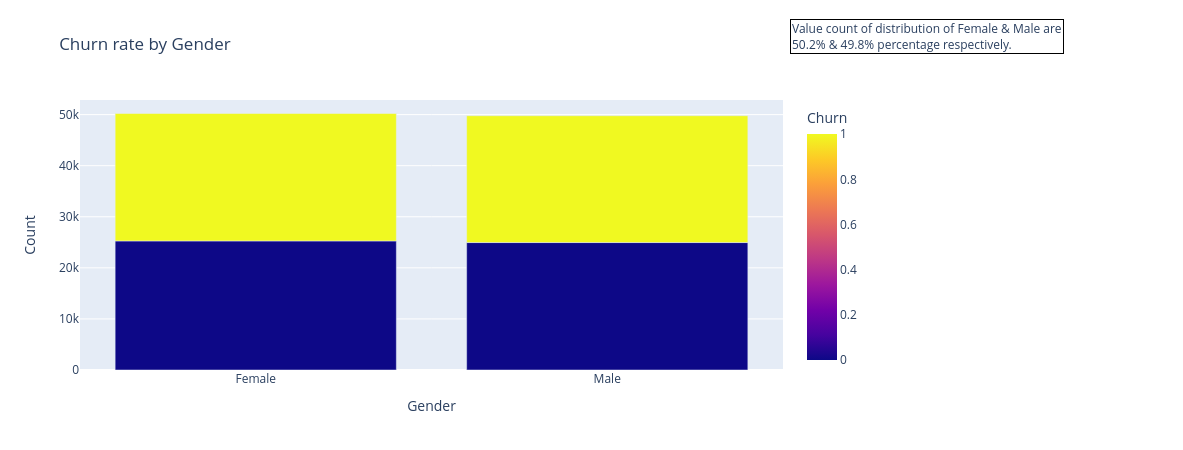
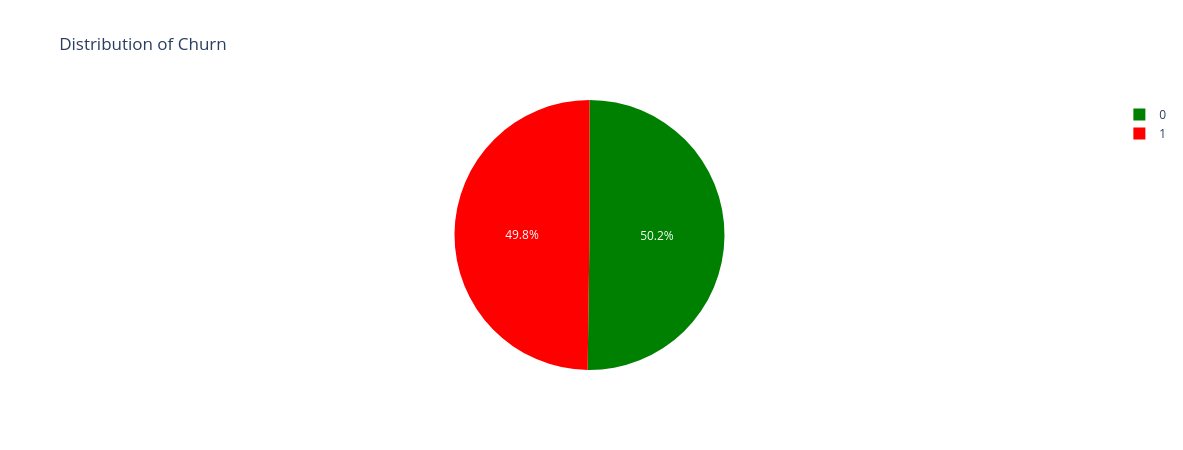
Approach :

1. We first aim to understand the dataset in a better way.
   1. The dataset has the following columns :
      1. CustomerID
      2. Name
      3. Age
      4. Gender
      5. Location
      6. Subscription\_Length\_Months
      7. Monthly\_Bill
      8. Total\_Usage\_GB
      9. Churn

The last (ix. Churn) column suggests us that this is a labelled dataset. So, naturally Supervised learning algorithms should do our job.

* 1. Dataset has 100k entries. We start by cleaning i.e removing those entries that have missing (NaN) values at either of columns. We also drop the duplicate entries.
  2. Next, we aim to encode the categorical labelled columns. We could have had used Binary or One-Hot Encoding for the columns : ‘Name’ , ‘Gender’ , ‘Location’ . But here for ‘Locations’ i preferred simple mapping the locations with numerical index. For eg : while encoding for ‘Gender’ column, i used ‘Male’ : 1 and ‘Female’ : 0 as the numerical representation.
  3. Dropping unnecessary columns : For eg : ‘CustomerID’ column cannot be used for prediction , since the ID will most likely a random/serial generated number from computer.

1. Now lets have a look at the remaining column w.r.t churn :



The most important and common observation among all these plots is that , All features are distributed equally (50% w.r.t churn) among all the customers.So there is no bias w.r.t any particular feature.

We then perform feature scaling , where we normalize each of this feature values (‘Subscription\_Length\_Monthly’ , ‘Monthly\_Bill’ , ‘Total\_Usage\_GB’ . )

\*Due to time constraints , i was not able to dig deeper into feature engineering where we try to find the effect of not one like above but multiple features (a combination of them) that can affect churn.

1. We tested for multiple models (logistic Regression , Random Forest , SVC classification , Decision Tress , Naive-Bayes) . The accuracy in all of this models was about 0.5 , which is more or less expected considering no biassing in the data with respect to the features and churn.

Accuracy, precision, Recall and F1-score are reported in code for each of these algorithms.

Neural Networking model from tensorflow was also tried on google collab, it also gave the same accuracy (0.5).

1. Fine tuning was done for all this above methods. I achieved the best accuracy of 0.51 from tuning the hyper-parameters. Cross validation and Hyper-parameter tuning were also explored. Both almost produce same result.

1. The model is deployed with help of streamlit . Simulation of the same in local environment is given below.

