**Course Title:**

**Applications of Artificial Intelligence**

**Assignment No. 2**

**Course Number**

**EAI6010**

**Term and Year: Winter B:**

**Start and End Dates:**

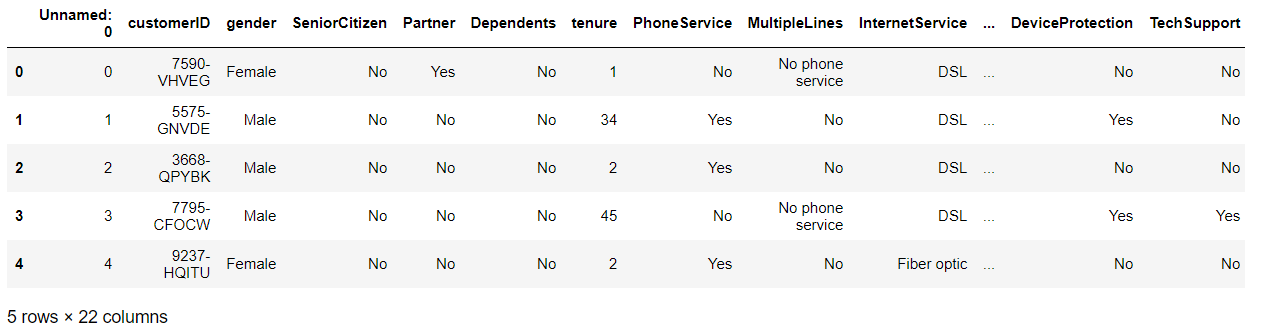
**March 1 - April 10**

****

**Name: Nikhil Sanjay Thorat**

**Steps involved in building the Customer Churn Prediction Model:**

1. **IMPORTING THE DATASET AND NECESSARY LIBRARIES:**

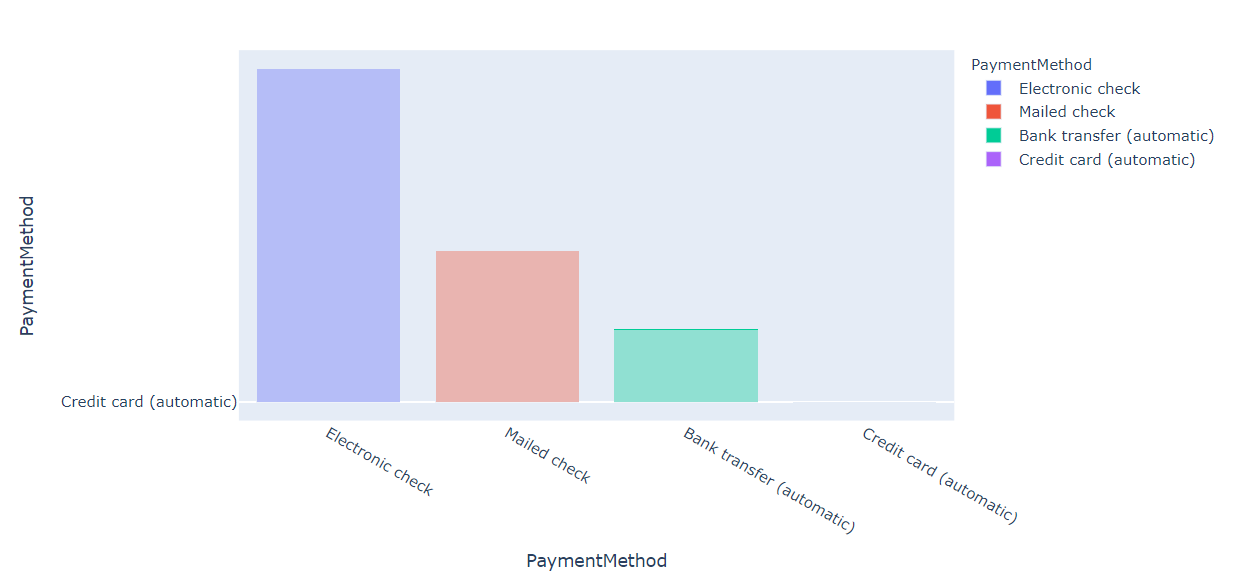


1. **DATA CLEANING:**

* **REMOVED THE NULL VALUES OF THE DATASET**
* **FIXED THE DUPLICATE VALUES**
* **SEPERATED CATEGORICAL AND CONTINUOUS VALUES**
* **FIXED THE DATA TYPES OF THE VARIABLES IN THE DATASET**
* **THERE WERE NO OUTLIERS PRESENT IN THE DATASET. GENERALLY IF OUTLIERS ARE PRESENT WE CAN REPLACE IT WITH CAPPING METHODS**

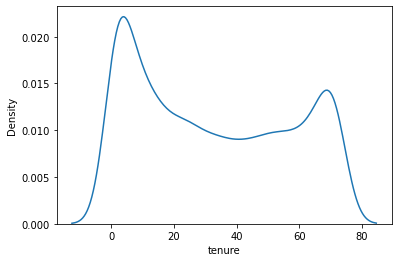
1. **PERFORMING EDA ON THE DATASET:**

**Count plot for payment methods.**



Count plot of various payment methods in the dataset. We can see that most of the customers prefer using electronic checks for payments.

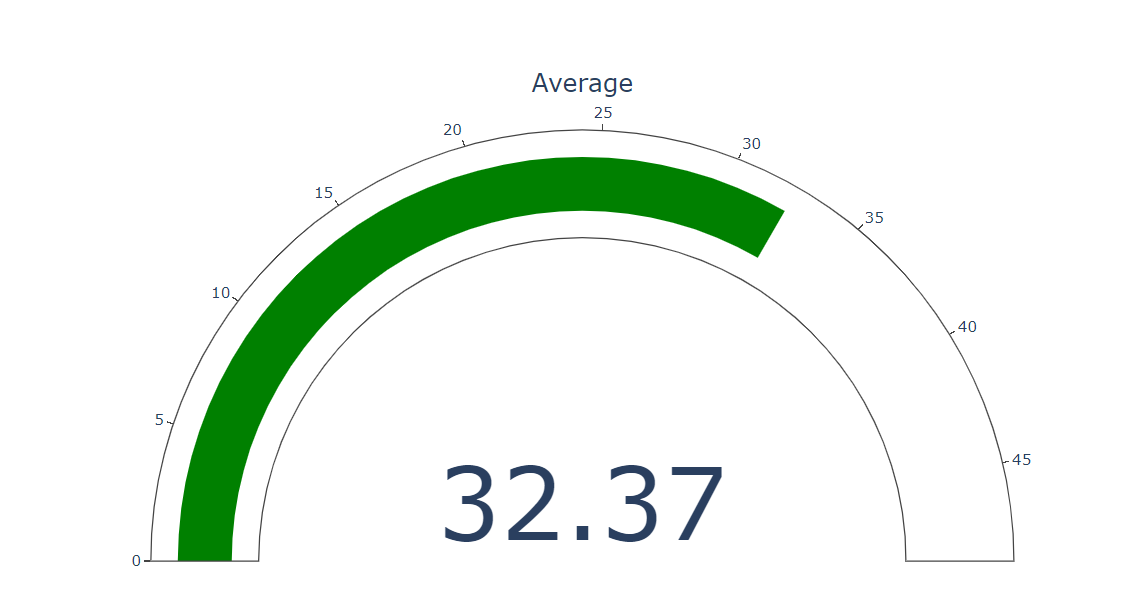
**Density plot for tenure.**

****

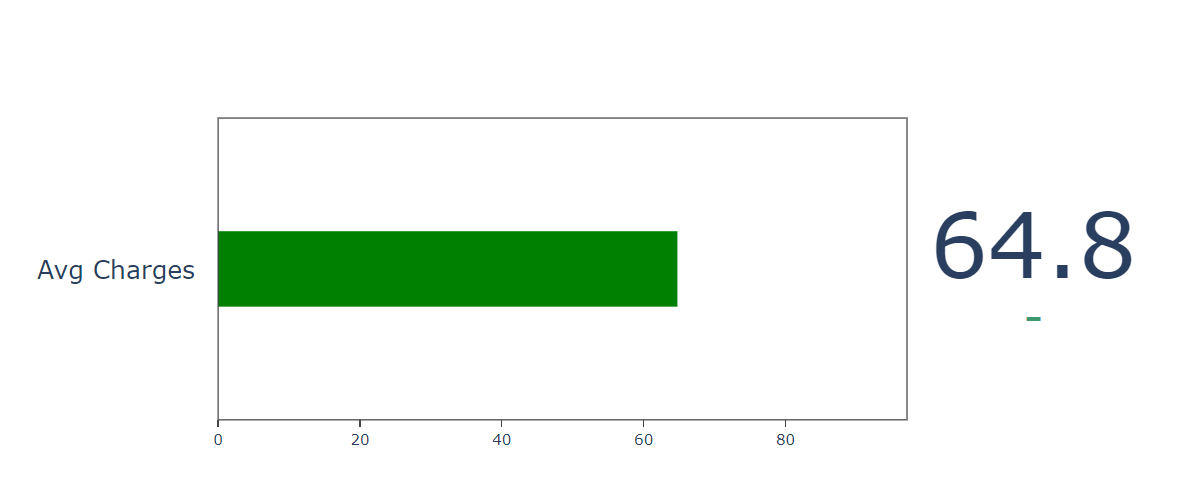
From the above density plot we can see that the tenure mostly ranges from 10 to 70 values.

**Important KPI’s for dashboarding:**

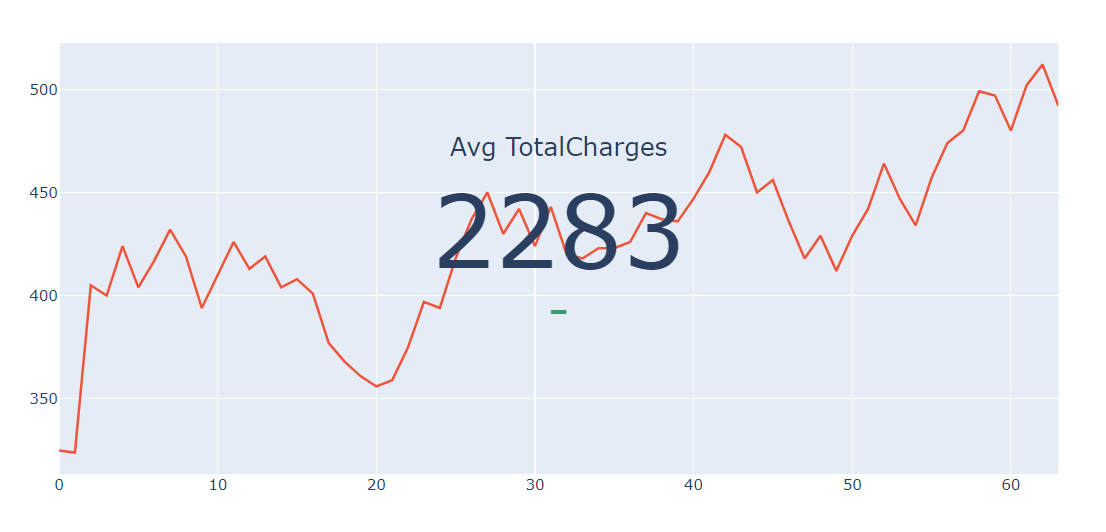
**Average Tenure:**



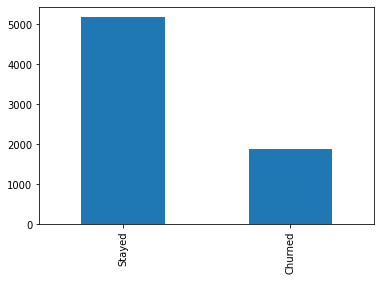
**Average Monthly charges**



**Average Total Charges**

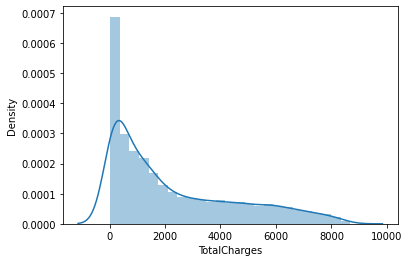


**Count Plot for target variable**

****

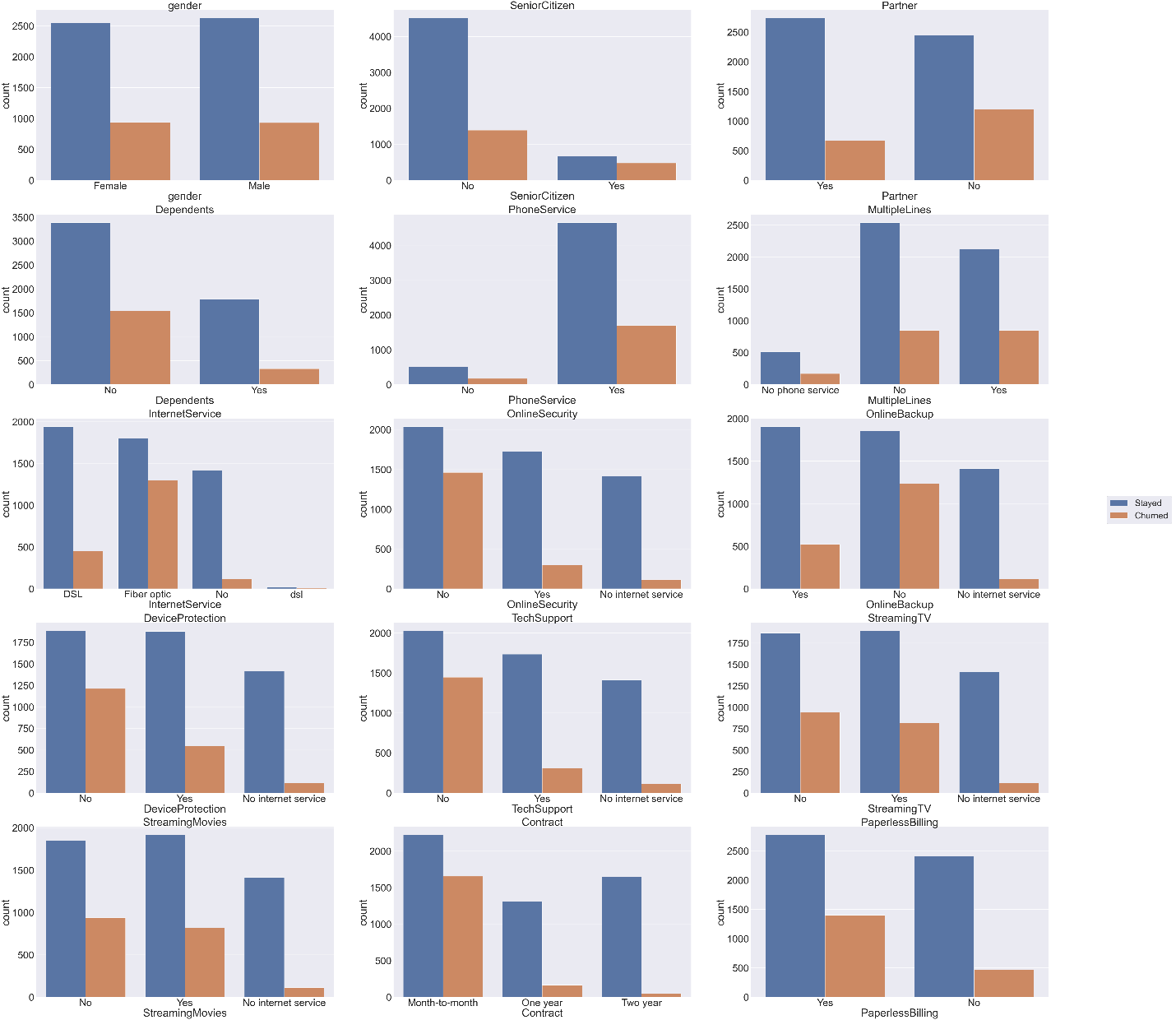
The number of people who stayed are more as compared to number of people who churned.

**Density Plot for Total Charges**

****

The density of most of the total charges range from 0 to 4500.

**Count plot for all the features with respect to churn**

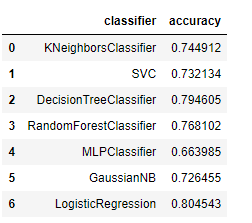
****

The above mentioned diagram shows the count of various features with respect to the target variables.

1. **SPLITTING THE DATASET INTO TRAIN TEST SPLIT USING SKLEARN PACKAGES**

The data was split into 80% training and 20% of testing dataset and a various models were fitted on the same dataset.

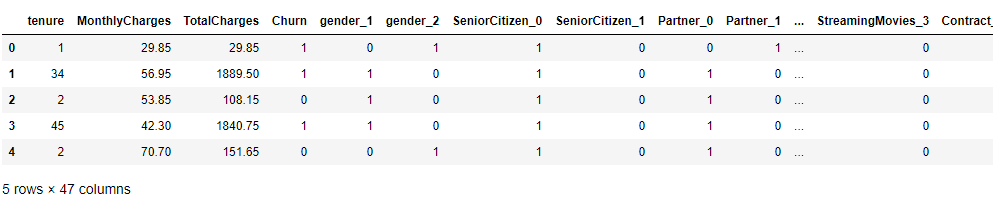
1. **APPLIED VARIOUS MACHINE LEARNING MODELS WITHOUT SCALING THE FEATURE VARIBALES**



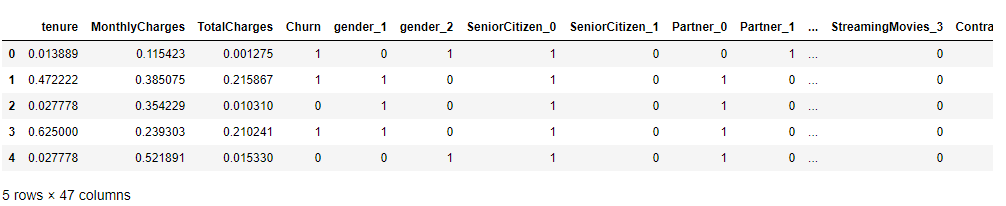
The highest accuracy obtained was for Logistic regression which was 80%.

1. **PERFORMED FEATURE ENGINEERING TO INCREASE THE ACCURACY OF THE MODEL**

Using get\_dummies function dummy variables were created for the dataset for all the categorical columns and that data was later passed into the model.

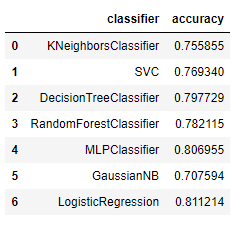


1. **PERFORMED MIN MAX SCALING ON THE DATASET**



Using min max scaling we can normalize the features variables of the dataset. In this dataset we had three continuous variables which are tenure, monthly charges and total charges which were normalized.

1. **APPLYING MACHINE LEARNING ALGORITHMS AFTER SCALING THE DATASET**



The results obtained were more accurate as compared to the results which were obtained without scaling the variables.

1. **USING HYPERPARAMETERS TO TUNE THE MODELS TO OBTAIN MORE ACCURATE RESULTS**

**Hyperparameter Tuning and grid-search**

Almost all algorithms have hyperparameters that can be tuned to fine-tune their performance, reduce over-fitting and better capture the patterns in the dataset. Having a good understanding and intuition of how algorithms work is essential to fully utilize hyperparameter tuning for the purposes of improving model performance and testing different modeling strategies.

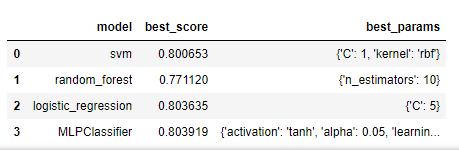
Hyperparameters are significant because they control the conduct of the training algorithm and have a significant impact on the performance model is being prepared. Productively search the space of conceivable hyperparameters. Simple to deal with an enormous arrangement of analyses for hyperparameter tuning.

Grid-search is a hyperparameter tuning algorithm that sequentially goes through every possible combination of hyperparameter combination it is fed in space. For example, for hyperparameters parameter 1 and parameter 2 - it would mean testing out all possible combinations of their values:

Grid-search can be done using the GridSearchCV() function - it takes in as arguments:

* The model being used.
* The possible parameters to test - inputted as a dictionary.
* cv: The number of cross-validation folds.
* verbose: More detailed output if 2

**RESULTS:**



The results obtained after tuning the model were not having a significant raise in the accuracy and almost all the models showed an accuracy of around 80%.

1. **APPLYING ENSEMBLE LEARNING TO THE DATASET**

Ensemble modelling is an interaction where various different base models are utilized to foresee a result. The reason for utilizing ensemble models is to lessen the speculation error of the prediction. The base models are different and free, the prediction error diminishes when the ensemble approach is utilized.

The results obtained for ensemble models were:

* Decision Tree Classifier : 80%
* Adaboost Classifier: 80%
* Logistic Regression Classifier: 80%

1. **CONCLUSION AND RESULTS:**

After trying various methods on the dataset to increase accuracy the maximum accuracy obtained was around 81% which was using Logistic regression. Whereas other models also gave a similar accuracy around 80%. Therefore we can conclude that we can use above mentioned models for classification of the customer chur prediction.

1. **TABLEAU DASHBOARD RESULTS:**

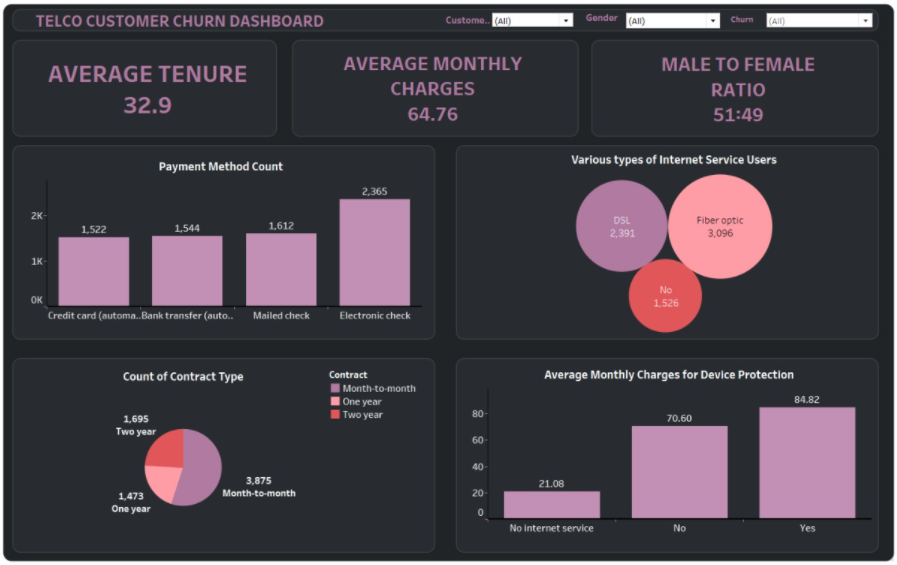


TABLEAU LINK: [Telco Customer Churn Dashboard - Nikhil Sanjay Thorat | Tableau Public](https://public.tableau.com/profile/nikhil.sanjay.thorat#!/vizhome/TelcoCustomerChurnDashboard_16155910468380/Dashboard1)

1. **WEBAPP FOR USING THE MODEL:**

The model was linked to a front end page which was designed using HTML and the webapp was deployed using Heroku so that the end user can test the model by giving various feature inputs. As the current model has a lot of variables which cannot be inserted in a webapp, therefore I performed feature selection on the dataset and took a few top variables and designed a webapp which predicted customer churn.

**Webapp Link:** [Home Page (churn-customer.herokuapp.com)](https://churn-customer.herokuapp.com/)

