South Asian Churn Data Analysis

BCIS 5110 Programming Languages for Business Analytics

Project Report - 01

Submitted by

Donetra Hemraj Lanjewar	11561188
Garlapati Monish Gandhi	11596204

In partial fulfilment for the award of the degree of

Masters
In
Business Analytics

University of North Texas
G. Brint Ryan College of Business

Abstract

In today’s world, Churn dataset is very useful for the analysis in nearly each domain of customer subscription or customer services sector. It helps the industry to predict the active status of their customers. This report is about customer churn prediction in a mobile operator industry.

The main aim is to get in the dataset to understand the frequency of customer who remain active in their services of a mobile operator and of those who no more turn their status to inactive. The project involves getting in the unique statistics of the different mobile operators with great respect in the two months data. Thus, it acts as a full real time back up of proofs.

The report focuses on the descriptive data in regards to the revenue generation, data of customers, the type of network for each customer, all in respect to the mobile operator. Visualization is done in regards with the parameters that are useful to see the trend in data. The churn status, data and call volume, all kinds of revenue are compared and analyzed to finalize the ideal parameters for a mobile operator to work on so that to keep their customers active.

Finally, prediction is done with 4 types of regression models and a comparison between each is done in regards with the accuracy score given by each of the model. Also, graphs are plotted for each of the prediction model.

Table of Content

List of Tables

List of Figures

Chapter 1

SUMMARY OF ANALYSIS PROJECT - 1

About Churn Analysis

Stakeholders are putting more time and effort into determining the causes within their organizations, how they can precisely predict the kinds of current customers who may stop doing business with them, and what they can do to minimize customer churn as a result of the significant importance of customer churn within a business.

Although the causes of customer churn can vary and are best defined by experts in the field, some common ones include infrequent use of the product, subpar customer service, and cheaper prices elsewhere. Whatever the justification, which might vary depending on the industry, one thing is true across all industries: acquiring new consumers is more expensive than keeping the ones you already have. This directly affects the company's operating expenses and marketing budgets.

In any organization, preventing customer turnover is a key component of customer relationship management (CRM).
In telecom sector, fierce rivalry among telecom carriers makes it imperative for operators to reduce churn if they are to keep their current subscribers.
Churn is the phrase used to characterize customers who switch from one service provider to a rival after breaking their contract. The customers select the provider that advertises new incentives, frequently meeting their wants with less expensive services than their present provider.

Churn prediction has become essential in determining in advance whether a current customer intends to leave the carrier due to the intense competition among telecom companies. The operators can then devise win-back strategies to provide the consumer with some better offers. As the cost of customer acquisition is far higher than the cost of customer retention, churn prevention is a crucial aspect.

The project covers aspects in sequential manner as Data cleaning > Exploratory Data analysis > Diagnostic > Visualisation > Prediction.

1.1 Descriptive Analysis

The descriptive starts with covering the related variation in according to the network age i.e. the duration of customer using or has used the service of a certain operator.
Further the work is continued in accordance to the Churn class.

1.2 Diagnostic Analysis

Next, the project shows statistics related to aggregate values all in respect to the churn status, this includes showing the number of statistical category, mean and median of each. It focuses on understanding the trend of a factor with respect to other factor.

1.3 Visualization

Pictorial representation helps us to understand the information more quickly and in an easier way as compared to the written information which is also time consuming and lengthy. The project covers graphically picturizing the relation for user types, favourite mobile operators, Churn class ,etc.

The project also involves checking for Bi-Variate outliers for major aggregates against Churn class

1.4 Predictive analysis

Predictive analysis using machine learning approaches has been extensively used to accurately classify churners and non-churners. Decision trees are among the most often used methods . Numerous studies have also used Support Vector Machine and Neural Network-based solutions. In recent years, ensemble techniques like voting, stacking, bagging, and boosting have been used more successfully because they take into account employing a group of learners rather than just one, which raises the classification rate.

The project proceeds for the prediction using 4 models namely :

- K-n Folds
- Linear Regression
- Random Forest
- Decision Tree

The prediction is done with respected to the train test model with a 30% split. The project ends with comparing the regression models by plotting a graph and picking up the one with best outputs among all.

CHAPTER 2

PROJECT MOTIVATION / BACKGROUND

Customer churn affects companies in various industries frequently. One needs to make an investment in gaining new customers for the business to expand. Every time a customer goes, a sizable investment is lost. It is necessary to put time and effort into finding replacements. A company can save a ton of money if it can foresee when a client is most likely to depart and give them incentives to stay.

Understanding what keeps consumers engaged is therefore very important information since it may help to create retention strategies and implement operational procedures meant to prevent customers from leaving.

For a great analysis, the dive in motto should be more of interesting. The selected dataset is from everyday happening which makes it even more relatable. Further, the variables about which the dataset revolves is what everyone can relate as they are from basic happenings.

The telecom sector is massive, vibrant, and dynamic, with a massive client base, making customer acquisition and retention critical concerns for its existence and profitability. New entrants emphasis client acquisition, whereas established firms emphasize customer retention.

Customers are encouraged to choose from a wide variety of products and services thanks to globalization, which makes it easier for them to select the best services accessible. Customer satisfaction and customer churn are closely related. Because the cost of attracting new customers is much higher than the cost of retaining the existing customers, operators put an emphasis on various customer-related practices and data to ensure customer retention.

Analysis becomes more interesting when the prediction is real time useful and starts from junk.
Cleaning the data helps to understand it more than normal view of the same.

This project primarily focuses on identifying churn. In order to estimate customer churn, telecom operators frequently need to examine the processes taken in order to identify the most likely reason and justification for why consumers are leaving. Regression models may be able to accomplish this because to their simplicity in interpretation, visualization, and analysis.

Even a 5% increase in consumer retention rates can raise revenue by at least 25%. A company's overall success can be significantly impacted by improved client retention.

If a business has a low customer retention rate, even if it is adding new customers every month, it is still losing out on a ton of additional money.

It costs five times as much to recruit a new client as it does to retain an existing one, even though many organizations emphasize expanding their customer base and devising ways to reduce customer acquisition costs. While attracting new clients should be a top goal, efforts to keep existing clients should also be made.

CHAPTER 3

KEY QUESTIONS

It costs five times as much to recruit a new client as it does to retain an existing one, even though many organizations emphasize expanding their customer base and devising ways to reduce customer acquisition costs. While attracting new clients should be a top goal, efforts to keep existing clients should also be made.

The key findings of the work revolve around Churn Prediction

The project work focuses on finding the statistics, relation, and a more of details from the given aspects of churn.

1. The importance of Normalization in process of data exploration and further analysis.
2. Understanding the relation between the trends of one attribute in respect to other and the extent of affect towards churn.
3. Comparison between different prediction models and find out which one suits best.
4. Use of the results to develop more of focused programs that will reduce the retention.

Chapter 4

DATA SOURCE

The data selection was done from ‘OpenML’

Data source link :
https://www.openml.org/search?type=data&status=active&id=44227&sort=runs

CHAPTER 5

DATA DESCRIPTION

5.1 Columns name and type in Dataset

The selected data is about the mobile operators in South Asia. It includes :

Table 5.1 Data type for each dataset column

Column Name	Datatype for column
Network age	Integer
Aggregate SMS Revenue	Real
Aggregate Total Revenue	Real
Aggregate Data Revenue	Real
Aggregate On-Net Revenue	Integer
Aggregate Off-Net Revenue	Integer
Aggregate Data volume	Real
Aggregate Calls	Integer
Aggregate Complaint Count	Integer
August user type	String
September user type	String
August Favourite operator	String
September Favourite operator	String
Class	String

5.2 About the variables in the Dataset

Table 5.2 Description of columns in dataset

Network age :	The period of time (in days) since the subscriber began utilizing the carrier's services has passed.
Aggregate SMS	The money received by a subscriber from using the SMS service.

Revenue :	
Aggregate Total Revenue :	The carrier's overall monthly income in Rupees for the months of August and September 2015.
Aggregate Data Revenue	The money that the carrier's current subscriber makes through calls and other interactions with clients who are on-network (using the same network as the subscriber).
Aggregate On-Net Revenue :	The money produced by the carrier's current customer when they call or otherwise communicate with customers who are on networks other than their own (off-network).
Aggregate Off-Net Revenue :	The amount of data service that the subscriber uses.
Aggregate Data volume :	Count of calls made by the customer during the course of its service.
Aggregate Calls :	The money the subscriber receives from using the data service.
Aggregate complaint Count :	Count of number of complaints raise by the customer.
August user type :	This information is useful in determining whether the user has a 2G or 3G service subscription for the month of August.
September user type :	This information is useful in determining whether the user has a 2G or 3G service subscription for the month of September.
August Favourite operator :	This information reveals which other network or mobile operator subscribers make the majority of their calls to and may persuade them to switch to that network in order to save money for the month of August.
September Favourite operator :	This information reveals which other network or mobile operator subscribers make the majority of their calls to and may persuade them to switch to that network in order to save money for the month of September.
Class :	Churned status

5.3 Key aspects of dataset

The data is basically taken randomly to form 2000 entries over the month of August and September.

The variables are in respect to the network age with 1000 each for ‘Churned’ and ‘Active’ class. Other than the calculative variables, it covers some major aspects as the type of network user by customer, the favourite mobile operator in a certain month and lastly the churn status.

Types of network used in to the data are :

- 2G
- 3G
- Other

The variable in the name of favourite operator basically directs us to the telecom operator which tends to attract more of the customers in comparison to the other competitors.
The favourite mobile operators in the data are :

- 0
- Mobilink
- Ptcl
- Telenor
- Ufone
- Warid
- Zong

With respect to the above mentioned, '0' here stands for none of the mobile operator being favourite which precisely exists for only the month of august.

At last, the very vital component, about which the data revolves is the 'Class' i.e. the Churn class. It directs us to the group which includes still operational customers and no longer operational group. It includes two categories namely:

- Active
- Churned

Here, the term 'Active' symbolizes customers who are still using the telecom service with a particular distributer and the opposite is termed as 'Churned'.

Chapter 6

DATA TRANSFORMATION / EXPLORATORY

Data Cleaning

The practice of correcting or deleting inaccurate, damaged, improperly formatted, duplicate, or incomplete data from a dataset is known as data cleaning. There are numerous ways for data to be duplicated or incorrectly categorized when merging multiple data sources. Even if results and algorithms appear to be correct, they are unreliable if the data is inaccurate. Because the procedures will differ from dataset to dataset, there is no one definitive way to specify the precise phases in the data cleaning process. But it is essential to create a template for your data cleaning procedure so you can be sure you are carrying it out correctly each time.

6.1 Steps involved in Data cleaning

The initial step in working was data cleaning. The raw file has null values and unwanted symbols.

1. Opening the original file format so that to have an idea of the dataset
2. Opening the same in Excel to get details of data as in whole.
3. Nothing the symbolic terms in any and the occurrence position
4. Worked with the starting empty and unwanted rows.
5. Renamed the column names in the data.
6. Working with negative Network Age values.
7. Defines the symbolic values in the data.
8. Filled the missing values using forward fill.

Followed by doing the above steps, recheck was done to make sure that all the cleaning is done in respect to the unwanted items and Nan values.

6.2 Exploratory Data Analysis

EDA can be done once we know the datatype for each of the value associated or having an idea about. (*Table 5.1)*

6.2.1 Descriptive

The EDA is done with three different classifications as stated below:

1 : Related to age input - To show records and related averages as per user age input.
2 : Related to Churn Class - To show data in relation with Churn Class.

3 : Aggregate Values with churn index - To show data with details of sum, mean and median in respect to churn status.

Each of the above choice is given further options of display out the output

6.2.2 Diagnostic

Diagnostic gives a clear view about the relation between the different variable so that we can get more of helpful insights of the data.

6.3 Visualization

The majority of useful visualization are covered in the project and will be explored as the topic proceeds.

Chapter 7

DATA ANALYSIS

One of the most significant and difficult issues facing organizations worldwide, including credit card companies, cable service providers, SASS, and telecommunication corporations, is customer churn. Client churn indicators can help firms increase customer retention even though they are not the most entertaining to look at.

7.1 Descriptive

The EDA is done with three different classifications as stated below:

1 : Related to age input - To show records and related averages as per user age input.
2 : Related to Churn Class - To show data in relation with Churn Class.
3 : Aggregate Values with churn index - To show data with details of sum, mean and median in respect to churn status.

Each of the above choice is given further options of display out the output

7.1.1 Related to age input

- Display records for particular age and <= age

```
For age = 8
    Network_age  Aggregate_Total_Rev ...  Sep_fav_a   Class
594          8            485.7008  ...      ufone  Churned
```

```
For age <= 8
    Network_age  Aggregate_Total_Rev ...  Sep_fav_a   Class
594          8            485.7008  ...      ufone  Churned
595          5            857.1008  ...    telenor  Churned
596          4           2898.8240  ...      warid  Churned
597          3            307.1204  ...      ufone  Churned
598          2            130.5780  ...      ufone  Churned
599          6            468.4244  ...   mobilink   Active
```

- Display average total revenue generated for particular age <= particular age

```
For age = 8
    Network_age  Aggregate_Total_Rev ...  Sep_fav_a   Class
594          8            485.7008       ...     ufone  Churned
```

```
For age <= 8
    Network_age   Aggregate_Total_Rev  ...  Sep_fav_a    Class
594           8            485.7008          ...      ufone  Churned
595           5            857.1008          ...    telenor  Churned
596           4           2898.8240          ...      warid  Churned
597           3            307.1204          ...      ufone  Churned
598           2            130.5780          ...      ufone  Churned
599           6            468.4244          ...   mobilink   Active
```

- Display average data volume for particular age <= particular age

```
For age = 8
The total average data volume for given age is: 2327900.778
```

```
For age <= 8
The total average data volume till and including given age is: 757917.6526666665
```

- Display average complaint count for particular age <= particular age

```
For age = 8
The total average complaint count for given age is: 1.0
```

```
For age <= 8
The total average complaint count till and including given age is: 1.5
```

- Display average calls for particular age <= particular age

```
For age = 8
The total average calls for given age is: 607.0
```

```
For age <= 8
The total average calls for till and including given age is: 257.0
```

7.1.2 Related to Churn Class

- Mobile operator in each churn class for both month

```
August mobile operator in each churn class
Class            Active  Churned
 Aug_fav_a
0                  43      16
mobilink          139     150
ptcl              179     290
telenor           154     132
ufone             223     223
warid             119      90
zong              143      99
```

```
September mobile operator in each churn class
 Class            Active  Churned
 Sep_fav_a
mobilink           44      94
ptcl              101     322
telenor            25      57
ufone             787     426
warid              24      50
zong               19      51
```

- User type in each churn class for both months

```
August user type in each churn class
 Class         Active  Churned
 Aug_user_type
2G               214     249
3G               595     499
Other            191     252
```

```
September user type in each churn class
 Class         Active  Churned
 Sep_user_type
2G               189     236
3G               628     522
Other            183     242
```

- Median for Network age in each churn class

```
Median for Network age in each Class section:
                 Network_age
 Class
Active           1423.5
Churned           943.0
```

- Average churn rate in each churn class

```
Average churn rate in each Class section:
                  Network_age
 Class
Active            1698.571
Churned           1240.582
```

7.1.3 Aggregate Values with churn index

- Network age in each churn Class

```
                    len         sum         mean       median
         Network_age Network_age Network_age Network_age
 Class
Active        1000     1698571    1698.571      1423.5
Churned       1000     1240582    1240.582       943.0
```

- Types of Revenue generated in each churn class

```
                    len                ...           median
          Aggregate_SMS_Rev  ...  Aggregate_SMS_Rev
 Class                                   ...
Active              1000                 ...              9.60
Churned             1000                 ...             19.71
```

```
                    len               ...           median
        Aggregate_Data_Rev    ...  Aggregate_Data_Rev
 Class                                  ...
Active                 1000             ...             17.50
Churned                1000             ...             11.25
```

```
                    len                  ...           median
        Aggregate_ONNET_REV  ...  Aggregate_ONNET_REV
 Class                                      ...
Active                  1000                ...             2994
Churned                 1000                ...              998
```

```
                      len                  ...            median
        Aggregate_OFFNET_REV  ...  Aggregate_OFFNET_REV
 Class                                       ...
Active                   1000                ...           6162.0
Churned                  1000                ...           4134.5
```

```
                 len               ...          median
        Aggregate_Total_Rev   ...  Aggregate_Total_Rev
 Class                              ...
Active              1000            ...          948.156
Churned             1000            ...          362.394
```

- Data Volume generated in each churn class

```
                 len               ...          median
        Aggregate_Data_Vol    ...  Aggregate_Data_Vol
 Class                             ...
Active              1000           ...     448470.18305
Churned             1000           ...      53479.07615
```

- Complaint count in each churn class

```
                     len                  ...           median
        Aggregate_complaint_count    ...  Aggregate_complaint_count
 Class                                    ...
Active                  1000              ...                1
Churned                 1000              ...                1
```

- Calls in each churn class

```
                 len            sum             mean           median
        Aggregate_Calls  Aggregate_Calls  Aggregate_Calls  Aggregate_Calls
 Class
Active           1000         342944          342.944          181.5
Churned          1000         138877          138.877          48.0
```

7.2 Diagnostic

Diagnostic gives a clear view about the relation between the different variable so that we can get more of helpful insights of the data. Following were calculated to check whether any relation exists among the two:

- Relation between aggregate complaints counts by customer and customer churn

```
Complain count distribution for churned subscriber
  count    1000.00000
mean        1.75800
std         2.46323
min         1.00000
25%         1.00000
50%         1.00000
75%         2.00000
max        49.00000
Name:  Aggregate_complaint_count, dtype: float64
```

```
Complain count distribution for active subscriber
  count    1000.000000
mean        2.091000
std         2.036877
min         1.000000
25%         1.000000
50%         1.000000
75%         2.000000
max        17.000000
Name:  Aggregate_complaint_count, dtype: float64
```

- Complain count distribution for each type of subscriber where complaint count is more than 2

```
Complain count distribition > 2 for churned subscriber
  count    129.000000
mean       5.844961
std        5.210341
min        3.000000
25%        3.000000
50%        4.000000
75%        7.000000
max       49.000000
Name:  Aggregate_complaint_count, dtype: float64
```

```
Complain count distribition > 2 for active subscriber
  count    212.000000
mean       4.962264
std        2.884587
min        3.000000
25%        3.000000
50%        4.000000
75%        6.000000
max       17.000000
Name:  Aggregate_complaint_count, dtype: float64
```

- Relation exists between from how long time customer taking services and customer churn

```
Network age distribution for churned subscriber
  count    1000.000000
mean     1240.582000
std      1140.521108
min        2.000000
25%      240.000000
50%      943.000000
75%      1880.250000
max     5355.000000
Name: Network_age, dtype: float64
```

```
Network age distribition for active subscriber
  count    1000.000000
mean     1698.571000
std      1381.023411
min        6.000000
25%      419.750000
50%      1423.500000
75%      2643.750000
max      5451.000000
Name: Network_age, dtype: float64
```

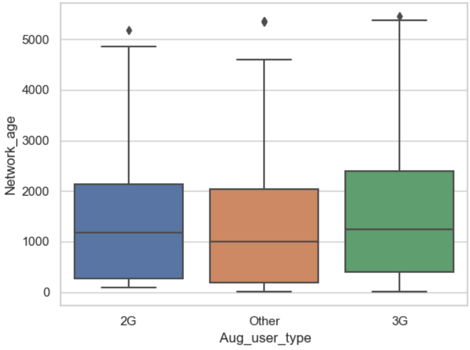
As maximum of the complaint count data lies in as 1 and 2, So one more complaint count distribution was done so that output won't be biased as then the output would be side preferred.

7.3 Visualization

It's a proven thing that the power of visualization is way more in conveying the ideas, so it is used in here.

The pictorial representation is done as following:

- User type for both month against Network age



7. 1 User type for August month vs network Age

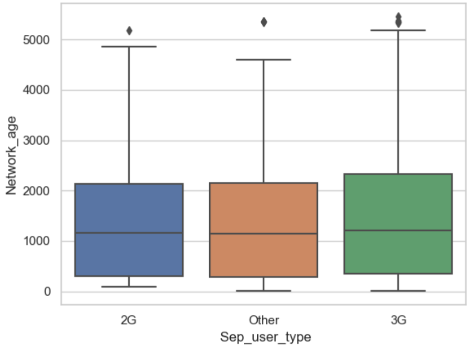


Fig 7. 2 User type for September month vs Network Age

- Favourite mobile operator for both month

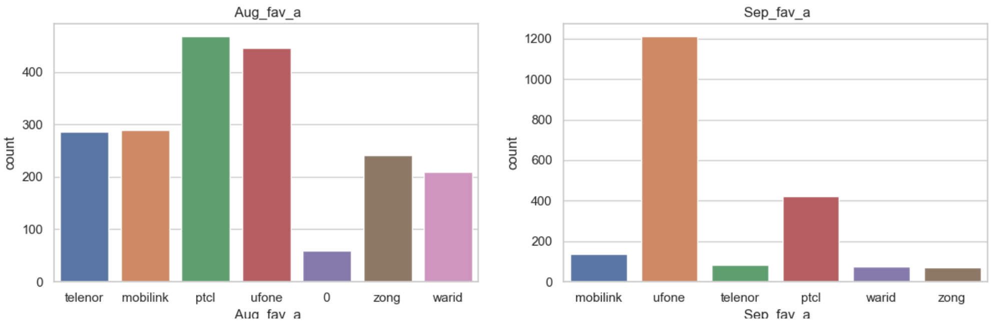


Fig 7. 3 Count of favourite operator for both months

- Box – plot for Aggregate total revenue against the Churn class

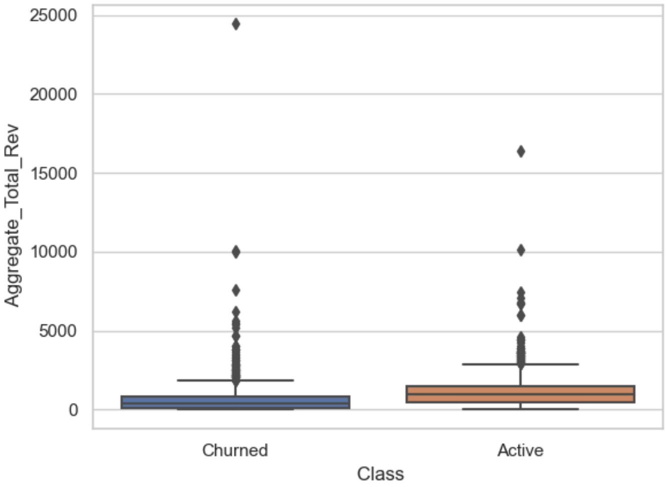


Fig 7. 4 Box plot for Aggregate Total revenue with respect to churn class

The project also involves checking for Bi-Variate outliers for following against Churn class :

- Aggregate calls
- Aggregate On-Net revenue
- Aggregate Off-Net revenue
- Aggregate SMS revenue
- Network age

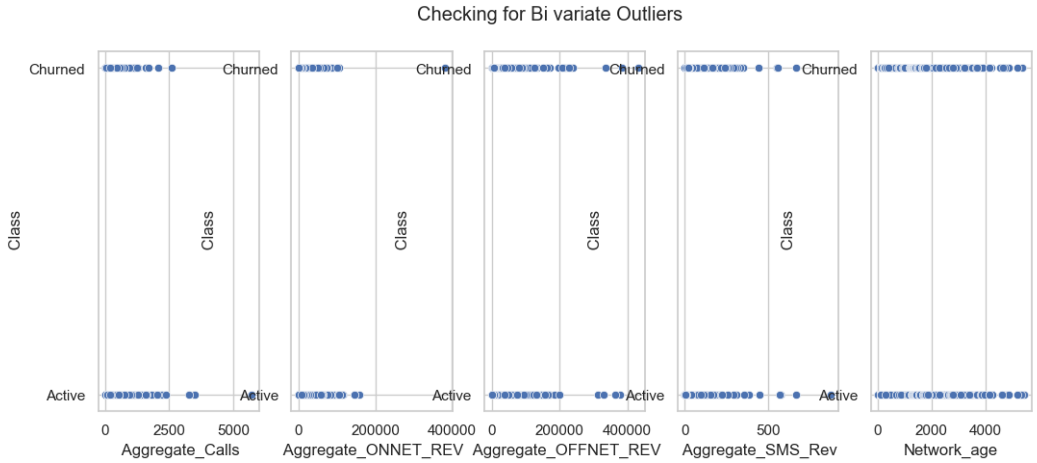


Fig 7. 5 Bi - Variate outliers

Chapter 8

MODELS AND ANALYSIS

A company's profitability will suffer and its ability to grow will be constrained by a high turnover rate. Our ability to estimate customer churn would enable the telco firm to better understand how effectively it is keeping its current customers and what the underlying factors are behind existing customers terminating their contracts (high churn rate).

In order to accurately predict whether or not the customer will churn or not in the data set, the goal of this process is to construct a binary logistic regression machine learning model. This will be followed by an analysis of the impact of major components that have a real impact. The strength of the trained model's generalization (model assessment) on the unknown data set is next tested.

8.1 Classification

Machine learning algorithms under supervision create models that depict data relationships. The goal of classification, a subfield of supervised machine learning, is to determine from a set of features which class or category a given entity falls into.

There are two possible formats for the features or variables:

- Independent variables (inputs or predictors) don't rely on other important traits.
- Dependent variables (outputs or reactions) are determined by the variation in respect to the independent variables. Here, the dependent variable in the churn class

8.2 Regression

Regression and classification issues differ in the type of dependent variables. The outputs of regression issues are typically continuous and unbounded. An illustration would be predicting compensation based on degree and experience. On the other hand, discrete and finite outputs from classification issues are referred to as classes or categories. A classification difficulty can be determining whether or not a worker will be promoted (true or false).

There are primarily two categories of classification issues:

- Binomial or binary classification: there are just two classes available (usually 0 and 1, true and false, or positive and negative)
- Three or more output classes are available in multiclass or multinomial classification.

The project comes under binary classification as the two classes here are ‘Churned’ and ‘Active’

8.3 A Model's Fit (Binary Logistic Regression)

The diabetic data set is split into a train and test split using the train test split function in the ‘sklearn.model’ selection module, and a logistic regression model is then fitted using the ‘statsmodels’ package/library.

8.4 Train and Test Models

Typically, the entire set of data is divided into a train data set and a test data set (general rule of thumb). 70% of the train data are used to train the model, and the remaining 30% are used to assess how well the model generalized to new sets of data.

Further the columns of user type for both the months and mobile operators are converted into categorical values so that the models gets it easy to interpret.

8.5 Model Assessment Using Test Data

The next step after fitting a binary logistic regression model is to test the model with 30% of the test data, which is unseen data.
Predicting the classes in the test data set and creating a confusion matrix are the following steps.

The following were the steps:

- The first step entails importing the ‘NumPy’ library as np and ‘sklearn.metrics' classification report and accuracy score.
- Next, using a model, forecast the churn.
- anticipate() function
- A cut-off value is set (0.5 for binary classification). is treated as a negative condition (-0.5) and a positive condition (+1). (1)
- Create a confusion matrix between the real (neg: 0; pos: 1) and anticipated data using pandas crosstab() (neg:0, pos:1)

8.6 Prediction

The model used for predicting is ‘Random Forest Classifier’

```
Prediction using Random Forest Classifier

0    702
1    702
Name:  Class, dtype: int64
0.74
Accuracy:       74.0%
Recall:         75.53956834532374%
```

8.7 How does the selected model works

Four steps make it work:

- Pick samples at random from the dataset provided.
- Create a decision tree for each sample, then analyze the predictions it produces.
- Cast a vote for each expected outcome.
- As the final prediction, choose the outcome that received the most votes.

8.8 Why Random Forest Classifier is chosen?

A common supervised machine learning approach for Classification and Regression issues is random forest. Distributed Random Forest creates a forest of classification or regression trees instead of just one single tree for classification or regression. Based on a set of traits, each of them generates a type.

8.9 Classification Report

The accuracy of predictions made by a classification algorithm is evaluated using a classification report. how many forecasts come true and how many come truer. The production of the classification report makes use of True Positive, True Negative, False Positive, and False Negative.

- TP / True Positive can be said when a model prediction matches a positive real observation.
- When an actual observation was negative and the model forecast was similarly negative, it is referred to as a TN (True Negative).
- False Positives (FP) occur when a real observation is negative although the model predicts a positive result.
- False Negative, or FN, is when a positive real observation is made despite a negative model forecast.

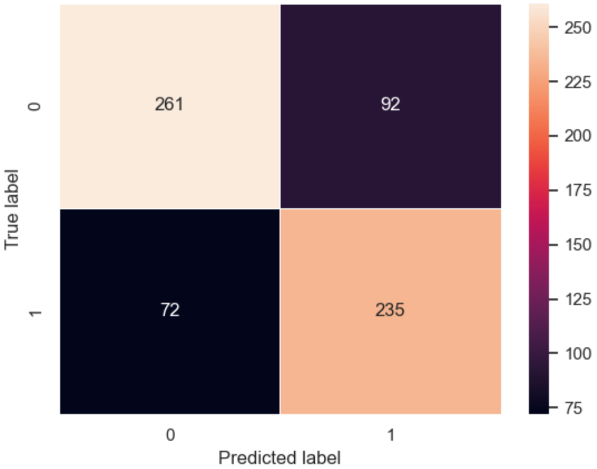


Fig. 8. 1 Heatmap for regression model using Random Forest

8.10 Comparison models

Further the project ends with comparison between 4 regression models namely :

- K -Neighbors Classifier
- Logistic Regression
- Decision tree
- Random Forest Classifier

8.10.1 K – Neighbors Classifier

The K in this classifier's name stands for the k nearest neighbors, where k is an integer value that the user specifies. As a result, this classifier employs learning based on the k nearest neighbors, as the name says. Data determines what value of k should be used.

Accuracy score for K-Neighbors Classifier is 59.5 %

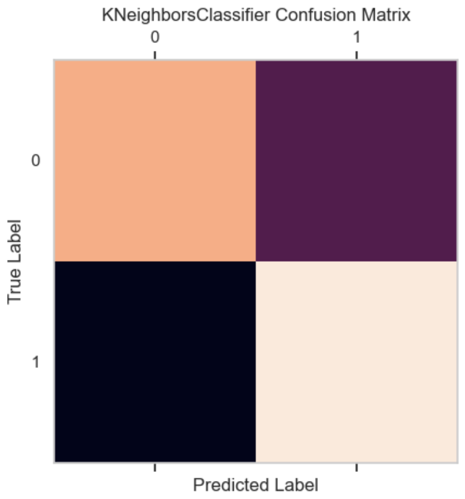


Fig. 8. 2 Confusion Matrix for K - Neighbors Classifier

8.10.2 Logistic Regression

A statistical technique for forecasting binary classes is logistic regression. The result or goal variable has a binary nature. There are just two conceivable classes when something is dichotomous. It can be applied, for instance, to issues with cancer detection. It determines the likelihood that an event will occur.

When the target variable is categorical, linear regression is applied in a specific way. A log of the odds is used as the dependent variable. Using a logit function, logistic regression makes predictions about the likelihood that a binary event will occur.

Accuracy score for Logistic Regression is 63.16 %

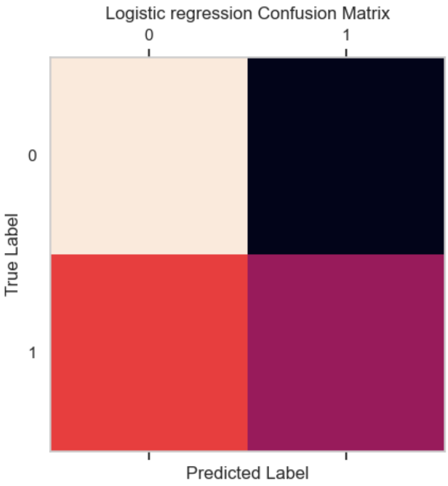


Fig. 8. 3 Confusion Matrix for Logistic Regression

8.10.3 Random Forest Classifier

A supervised learning algorithm is random forests. Both classification and regression can be done with it. Additionally, it is the most user-friendly and adaptable algorithm. There are trees in a forest. A forest is supposed to be stronger the more trees it has. On randomly chosen data samples, random forests generate decision trees, obtain predictions from each tree, and vote for the best option. Additionally, it offers a fairly accurate indicator of the feature's relevance.

Accuracy score for Random Forest Classifier is 75.66 %

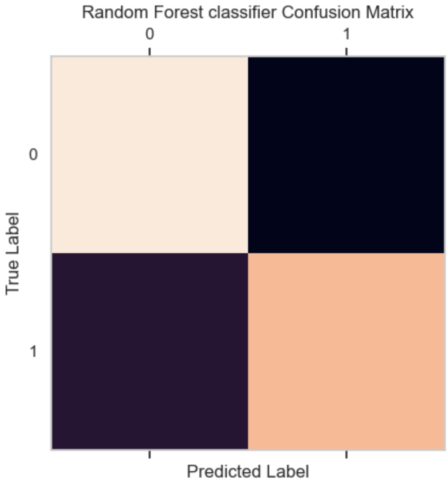


Fig. 8. 4 Confusion Matrix for Random Forest

8.10.4 Decision tree Classifier

A non-parametric supervised learning technique for classification and regression is called a decision tree (DT). The objective is to learn straightforward decision rules derived from the data features in order to build a model that predicts the value of a target variable. A piecewise constant approximation of a tree can be thought of.

Accuracy score for Decision Tree Classifier is 66.83 %

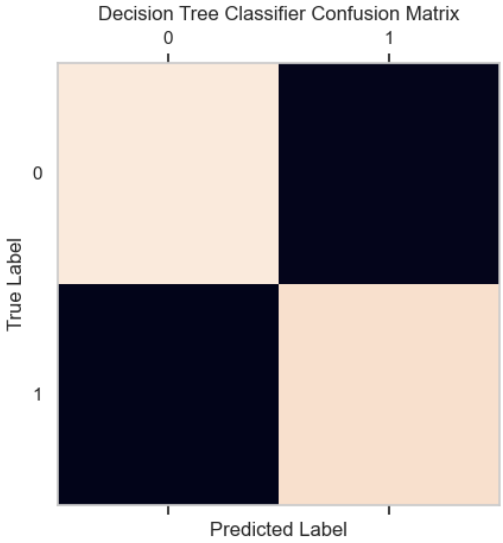


Fig. 8. 5 Confusion Matrix for Decision Tree

8.11 Comparison of regression models

Finally a comparison graph is plotted between the accuracy of reach model as shown below so that the best can be picked up for the further future work.

Based on the comparison of models, the previous decision on using Random Forest Classifier is justified

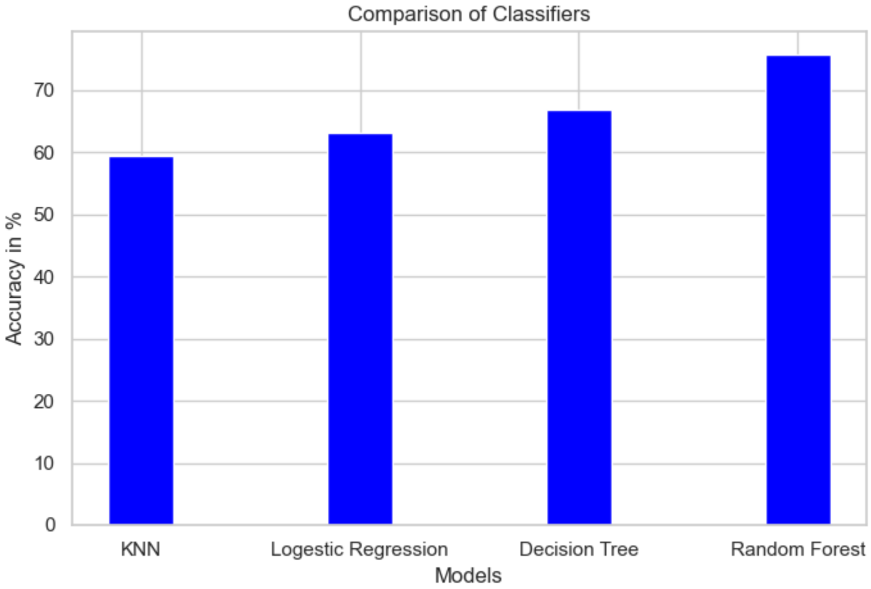


Fig. 8. 6 Comparison of models accuracy

Chapter 9

FINDINGS AND MANAGERIAL IMPLICATIONS

During the course of project, many of the real time factors were clearly depicted. The analysis found that some of the most crucial factors that affect a consumer's decision to churn include – the billings, majorly from the off-net and on-net, secondly, the data and call volume created by the customer, next the type of network a customer is using (2G, 3G, other). The churn consequence influencers model also enumerates the characteristics that lead to general unhappiness and, ultimately, to churn behaviour.

9.1 Answers for key questions

9.1.1. Importance of normalisation of data in the process of data exploration and further analysis.

In order to minimize duplication, prevent problems with data updating, and make queries easier, a data normalization technique divides database management into particular tables and columns.

- When used in the context of machine learning and data science, it transforms database values into a common scale when they are included in numeric columns.
- If you try to conduct any analytics or modeling, the enormous size discrepancies can be a problem.
- By establishing a matching scale across all columns and keeping the distribution constant, this method will take these two columns.
- This technique is better at constructing indexes via smaller, logical tables, which allows for faster searching and sorting, which is one of its key analytical advantages. Additionally, more tables allow for a better use of segments to manage the physical location of the data store.
- After modelling any essential columns, there will be fewer nulls and redundant data, and bias/issues with anomalies are considerably decreased by removing the disparities in scale.

9.1.2. Understanding the relation between trends.

The data is diagnosed for finding out the relation of one factor with respect to other. Also, visualization tools are used to address the relation more easily. The churn trends are seen to be affected severely with respect to the service type used by the customer. In accordance to this, the favourite monthly operator is seen to be changed.

We began with the data to do a number of studies, and from those analyses, we obtained data usage, revenues generated, service type used and fav operator over a certain period. To get a view of affects, regression models were created by statistical and data mining analytics.

9.1.3. Comparison between regression models

The purpose of do the comparison between models is to select one which delivers to best of accuracy level in predicting the churn rate.

The level of accuracy for each model is as stated below :

Table 9.1 Accuracy for each regression model

Model Name	Level of accuracy (in %)
K-Neighbors Classifier	59.5
Logistic Regression	63.16
Random Forest Classifier	75.66
Decision Tree Classifier	66.83

As it can be seen that the Random Forest Classifier provides best of accuracy. Therefore, we can use the respective model to predict the churn rate of customer in a telecom sector with an accuracy of 75.99 %

9.1.4. Use of results

Data normalization, feature selection, and data pre-processing have all been demonstrated to have significant effects. The scores for Average Quality, Churn Risk, and, to some extent, Annoyance may indicate a potential churner. For churn prediction, daily data volumes with the recent history of the customer and essential characteristics like tenure, bill, contract, data usage, kind, etc., are crucial.

9.2 Implications

For an industry to stay up in profits, it needs to do the analysis as following :

9.2.1 Determining the churn period

- In the project, the dataset is for the period of August and September.
- Doing so help to get a detailed view in comparison with unsure period as then the data will be not be properly managed.

9.2.2 Finding out the churn rate

- The study should start with know how many customers one had at the beginning and how many were still there at the end of the same period.
- Clients lost can be found by deducting the final total from the starting one.
- For churn rate, divide the total number of customers one had at the beginning of the time by the number of consumers lost. The turnover rate percentage is then calculated by multiplying the value by 100.

9.2.3 Monitoring the relevant key factors

It's time to keep an eye on the important KPIs (key performance indicators) for that time period. For instance, did the consumer engagement rates increase or decrease? How about product application? Did more support requests come in at that time?

Among the crucial metrics are:

- Activation rate: the proportion of licenses purchased by the business that are actually utilised.
- DAU/MAU rates: the proportion of all users who are active every day (DAU) or every month (MAU) (MAU).
- The Net Promoter Score (NPS), a survey-based measure, shows how customers feel about the product.
- One can go deeper to learn why some of the devoted customers left after obtaining data on the engagement and churn rate KPIs.

9.2.4 Surveys

- A qualitative method of customer churn analysis involves asking each individual customer how they feel about the service and why they are leaving.
- Sending surveys to churned clients and asking them why they cancelled is one approach to do this.
- In order to identify the areas where the industry has weak customer experience, look for common issues in the responses.
- Finding out why these consumers left will not only make it possible to address difficulties for the remaining clients, but it will also make it possible to retarget these clients by addressing the concerns that turned them off.

9.2.5 Addressing the issue

After taking the survey, if results pop out as majority of churned customers faced approximately same issues then it’s a direct implication that many other active clients might be experiencing the same troubles if the problems are widespread. The only factor is that they haven't yet churned, though. This implies that one can identify which accounts are at danger and prepare for possible churn.

Chapter 10

CONCLUSION

Customer churn is a constant problem for the telecom sector since people aren't afraid to switch providers if they can't find what they're searching for. Customer satisfaction and customer churn are closely related. There is no standardized approach that adequately addresses the problems faced by international telecom service providers. Taking all of these factors into account, a study project on predicting customer turnover is based on aggregate values, user type, and favourite operator to estimate the churn rate.

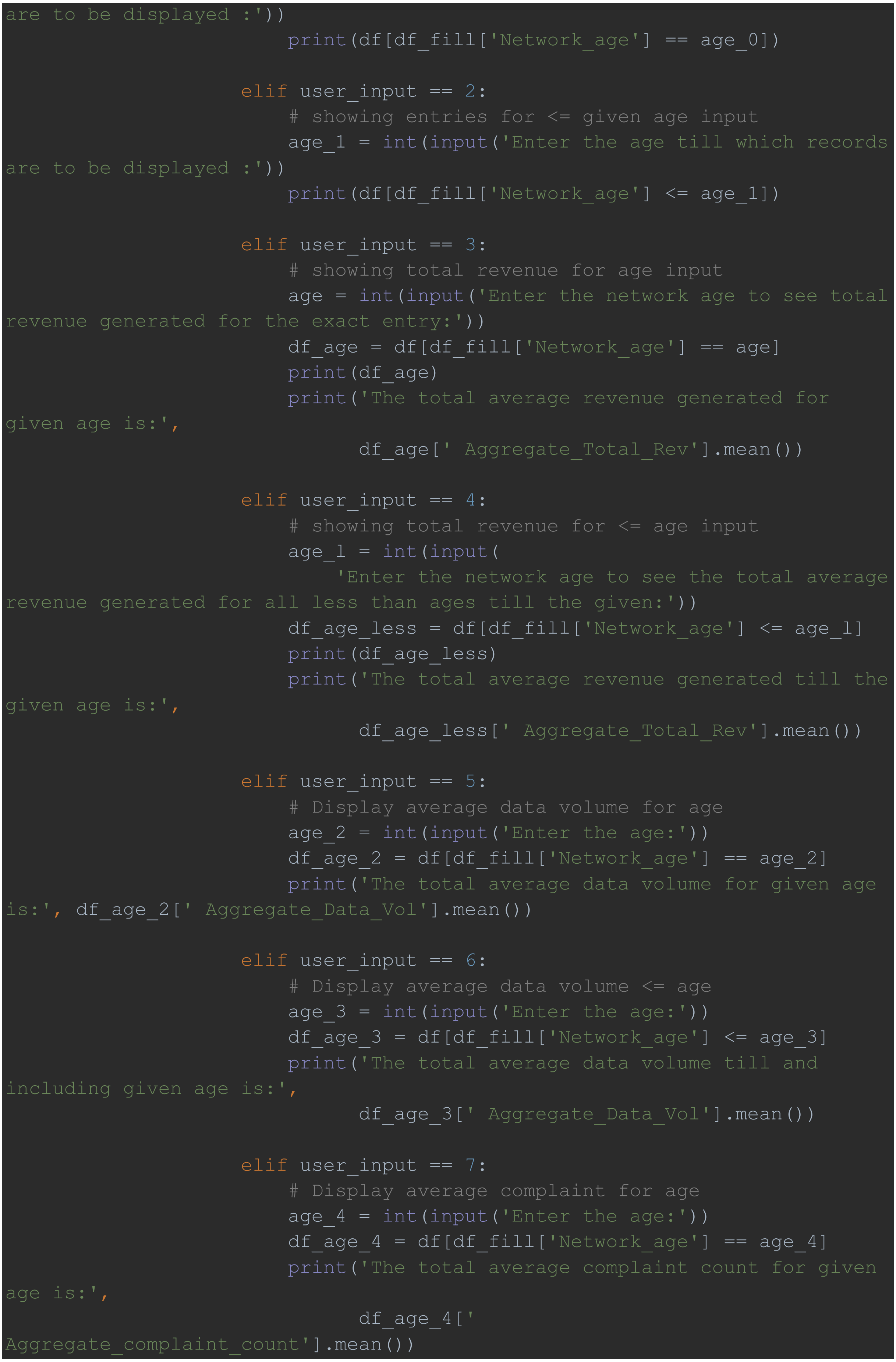
Data as taken from the source with respect to the necessary changes. Factors like mean, count, unique values, frequent value in a column etc are extracted from the dataset. A relational check is done to find the proportionality followed by visual check. At last prediction model is selected from a set of 4. The selection is based on the best accuracy criteria.

The project concludes with predicting the Churn rate of customers for a telecom industry with an accuracy of 75 % which will help predict the potential churners. Based on the prediction, necessary steps can be taken in order to avoid it or lessen the ration at least.

Appendix

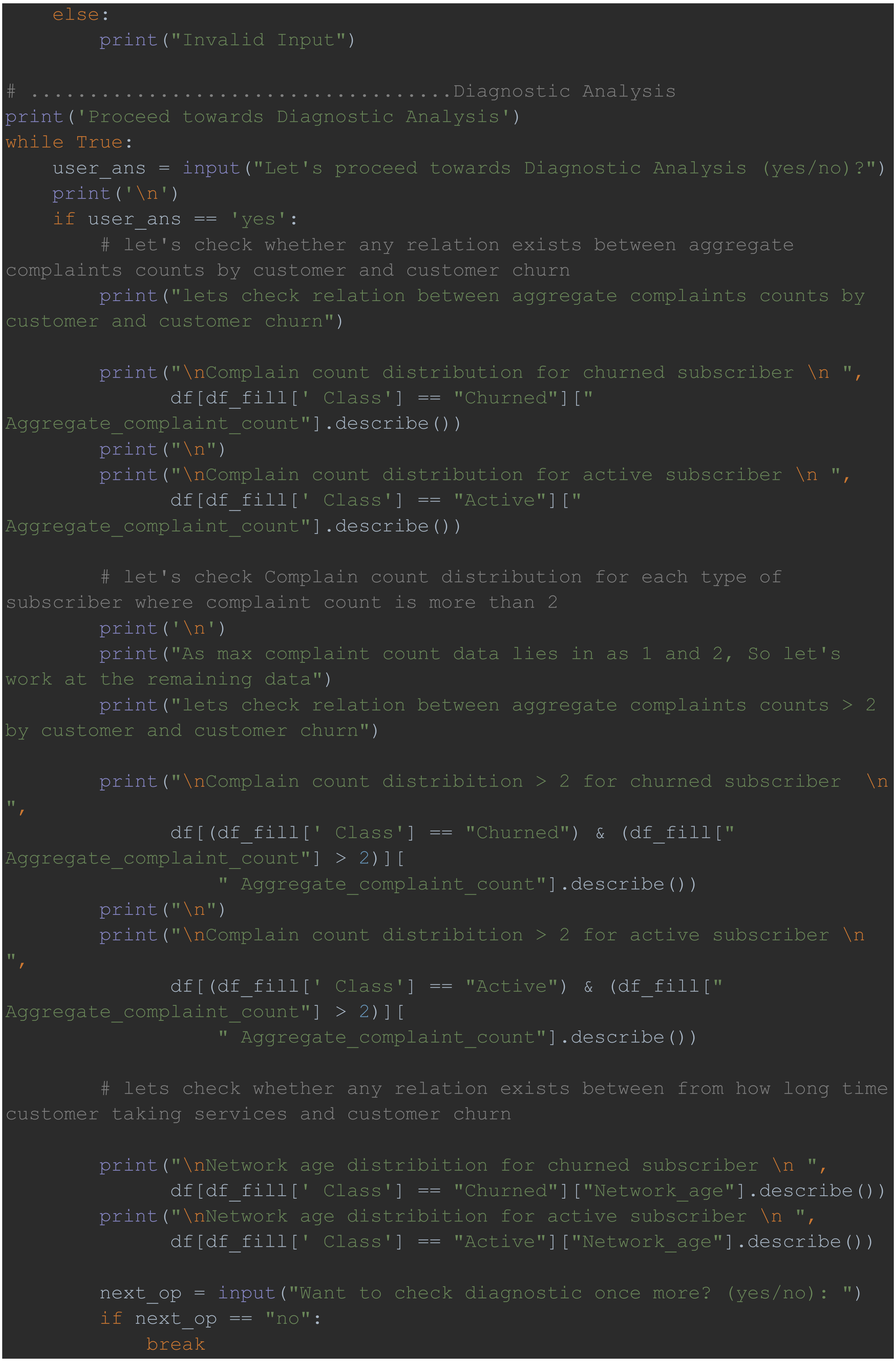
Python Code :

```
import pandas as pd
import numpy as np
import csv
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style="whitegrid")

with open('/Users/donetra/Downloads/project.csv') as file:
    reader = csv.reader(file)
'''for row in reader:
        print(row)'''

df = pd.read_csv('/Users/donetra/Downloads/project.csv')
# print the data frame
# print(df)

# print(type(df))
datatypes = df.dtypes
# print(datatypes)
print(df.columns)
# .......................Working with '?'

missing_values = ["?"]
df = pd.read_csv('/Users/donetra/Downloads/project.csv',
na_values=missing_values)

print(df.describe())

# .......................Converting negative numbers to positive numbers
for age
df['Network_age'] = df['Network_age'].abs()

# .......................Checking null values
print(df.isnull().sum())
df_inter = df.interpolate
df_fill = df.fillna(method='ffill')
# print(df_fill.to_string())

print(type(df_fill))
print('Filling done')

# print(df_fill.columns)
# li_index = list(df_fill.columns)
# print(li_index)
'''Index(['Network_age', ' Aggregate_Total_Rev', ' Aggregate_SMS_Rev',
       ' Aggregate_Data_Rev', ' Aggregate_Data_Vol', ' Aggregate_Calls',
       ' Aggregate_ONNET_REV', ' Aggregate_OFFNET_REV',
       ' Aggregate_complaint_count', ' Aug_user_type', ' Sep_user_type',
       ' Aug_fav_a', ' Sep_fav_a', ' Class'],
      dtype='object')
'''
print('\n')

# ..................Describe Data
print(df_fill.describe())
# ..........................................................Descriptive
statistics

print('Descriptive Analysis')

while True:
    print('The options are as follows :')
    print('1 : Related to age input \nTo show records and related averages
as per user age input')
    print('\n')
    print('2 : Related to Churn Class \nTo show data in relation with Churn
Class (Crossstab)')
    print('\n')
    print(
        '3 : Aggregate Values with churn index \nTo show data with details
of sum, mean and median in respect to churn status (Pivot Table)')
    print('\n')
    print('4 : Exit')
    user_option = int(input('Enter the choice from given options :'))

    if user_option in range(4):
        if user_option == 1:

            while True:
                print('Available options are :\n'
                      '1. Display records for particular age\n'
                      '2. Display records <= particular age\n'
                      '3. Display average total revenue generated for
particular age\n'
                      '4. Display average total revenue generated <=
particular age\n'
                      '5. Display average data volume for particular age\n'
                      '6. Display average data volume <= particular age\n'
                      '7. Display average complaint count for particular
age\n'
                      '8. Display average complaint count<= particular
age\n'
                      '9. Display average calls for particular age\n'
                      '10. Display average calls <= particular age')
                user_input = int(input('Enter the Choice from given sub-
options'))
                if user_input in range(11):
                    if user_input == 1:
                        # showing entries for given age input
                        age_0 = int(input('Enter the age for which records
```



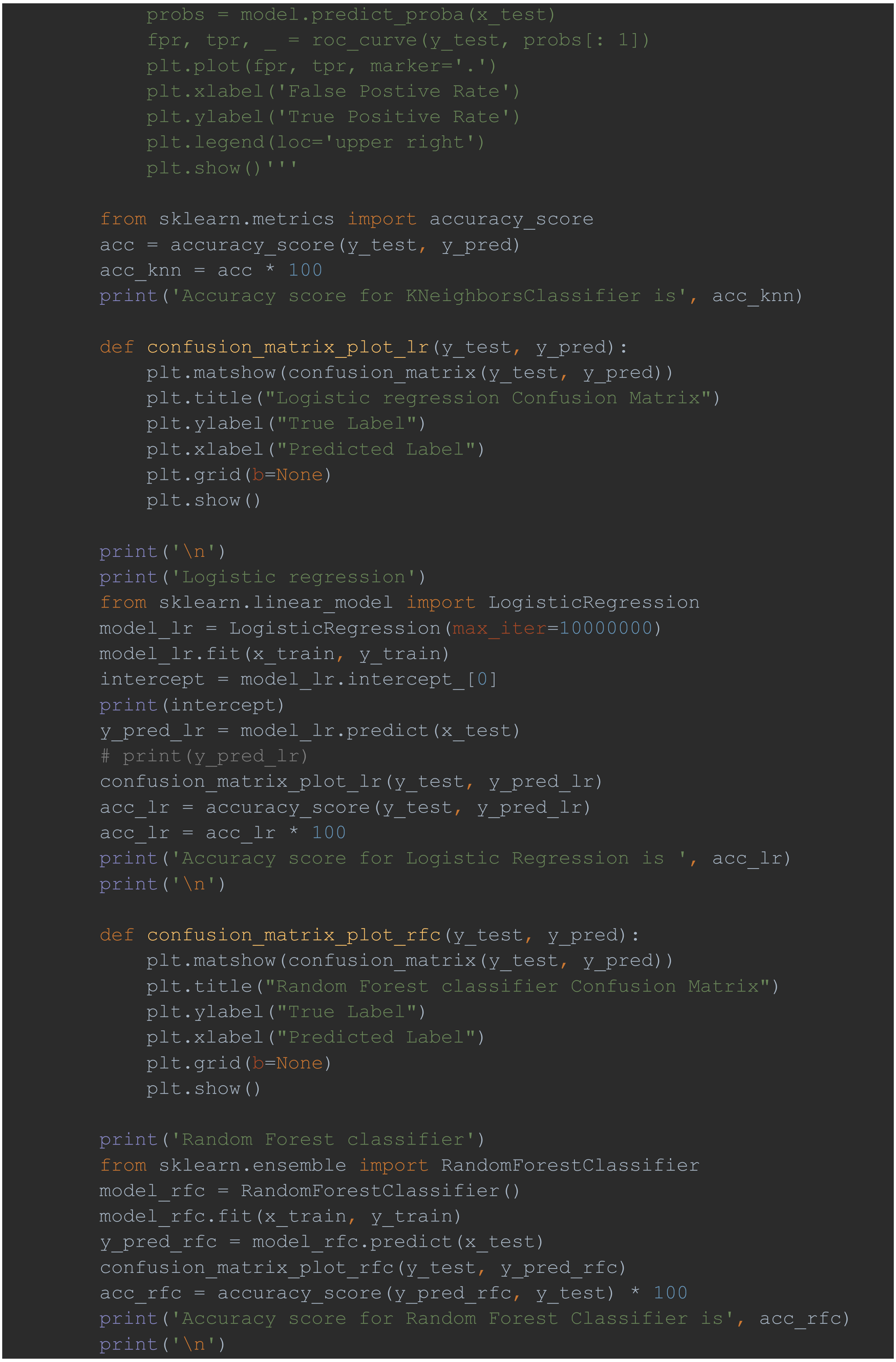
```
are to be displayed :'))
                        print(df[df_fill['Network_age'] == age_0])

                    elif user_input == 2:
                        # showing entries for <= given age input
                        age_1 = int(input('Enter the age till which records
are to be displayed :'))
                        print(df[df_fill['Network_age'] <= age_1])

                    elif user_input == 3:
                        # showing total revenue for age input
                        age = int(input('Enter the network age to see total
revenue generated for the exact entry:'))
                        df_age = df[df_fill['Network_age'] == age]
                        print(df_age)
                        print('The total average revenue generated for
given age is:',
                              df_age[' Aggregate_Total_Rev'].mean())

                    elif user_input == 4:
                        # showing total revenue for <= age input
                        age_l = int(input(
                            'Enter the network age to see the total average
revenue generated for all less than ages till the given:'))
                        df_age_less = df[df_fill['Network_age'] <= age_l]
                        print(df_age_less)
                        print('The total average revenue generated till the
given age is:',
                              df_age_less[' Aggregate_Total_Rev'].mean())

                    elif user_input == 5:
                        # Display average data volume for age
                        age_2 = int(input('Enter the age:'))
                        df_age_2 = df[df_fill['Network_age'] == age_2]
                        print('The total average data volume for given age
is:', df_age_2[' Aggregate_Data_Vol'].mean())

                    elif user_input == 6:
                        # Display average data volume <= age
                        age_3 = int(input('Enter the age:'))
                        df_age_3 = df[df_fill['Network_age'] <= age_3]
                        print('The total average data volume till and
including given age is:',
                              df_age_3[' Aggregate_Data_Vol'].mean())

                    elif user_input == 7:
                        # Display average complaint for age
                        age_4 = int(input('Enter the age:'))
                        df_age_4 = df[df_fill['Network_age'] == age_4]
                        print('The total average complaint count for given
age is:',
                              df_age_4['
Aggregate_complaint_count'].mean())
```

```
                    elif user_input == 8:
                        # Display average complaint <= age
                        age_5 = int(input('Enter the age:'))
                        df_age_5 = df[df_fill['Network_age'] <= age_5]
                        print('The total average complaint count till and
including given age is:',
                              df_age_5['
Aggregate_complaint_count'].mean())

                    elif user_input == 9:
                        # Display average calls for age
                        age_6 = int(input('Enter the age:'))
                        df_age_6 = df[df_fill['Network_age'] == age_6]
                        print('The total average calls for given age is:',
df_age_6[' Aggregate_Calls'].mean())

                    elif user_input == 10:
                        # Display average calls <= age
                        age_7 = int(input('Enter the age:'))
                        df_age_7 = df[df_fill['Network_age'] <= age_7]
                        print('The total average calls for till and
including given age is:',
                              df_age_7[' Aggregate_Calls'].mean())

                    else:
                        print('Invalid input')

                    next_op = input('Want to see more age related data ?
(yes/no): ')
                    if next_op == "no":
                        break

        if user_option == 2:

            while True:
                print('Available options are :\n'
                      '1. August mobile operator in each churn class\n'
                      '2. September mobile operator in each churn class\n'
                      '3. August user type in each churn class\n'
                      '4. September user type in each churn class\n'
                      '5. Median for Network age in each churn class\n'
                      '6. Average churn rate in each churn class')

                user_input_2 = int(input('Enter the Choice from given sub-
options'))
                if user_input_2 in range(7):
                    if user_input_2 == 1:
                        print('August mobile operator in each churn class')
                        print(pd.crosstab(df_fill[' Aug_fav_a'], df_fill['
Class']))

                    elif user_input_2 == 2:
                        print('September mobile operator in each churn
class')
                        print(pd.crosstab(df_fill[' Sep_fav_a'], df_fill['
Class']))

                    elif user_input_2 == 3:
                        print('August user type in each churn class')
                        print(pd.crosstab(df_fill[' Aug_user_type'],
df_fill[' Class']))

                    elif user_input_2 == 4:
                        print('September user type in each churn class')
                        print(pd.crosstab(df_fill[' Sep_user_type'],
df_fill[' Class']))

                    elif user_input_2 == 5:
                        print('Median for Network age in each Class
section: \n{}'.format(
                            df_fill[[' Class', 'Network_age']].groupby('
Class').median()))

                    elif user_input_2 == 6:
                        print('Average churn rate in each Class section:
\n{}'.format(
                            df_fill[[' Class', 'Network_age']].groupby('
Class').mean()))

                    else:
                        print('Invalid input')

                    next_op = input('Want to see more records related to
churn class ? (yes/no): ')
                    if next_op == "no":
                        break

        if user_option == 3:

            while True:
                print('Available options are :\n'
                      '1. Network age in each churn Class\n'
                      '2. SMS revenue generated in each churn class\n'
                      '3. Data revenue generated in each churn class\n'
                      '4. On-net revenue generated in each churn class\n'
                      '5. Off-net revenue generated in each churn class\n'
                      '6. Total revenue generated in each churn class\n'
                      '7. Data Volume generated in each churn class\n'
                      '8. Complaint count in each churn class\n'
                      '9. Calls in each churn class')
                user_input_3 = int(input('Enter the Choice from given sub-
options'))
                if user_input_3 in range(10):
                    if user_input_3 == 1:
                        print(pd.pivot_table(df_fill,
values=['Network_age'], index=[' Class'],
                                             aggfunc=[len, np.sum, np.mean,
np.median]))
                    elif user_input_3 == 2:
                        print(pd.pivot_table(df_fill, values=['
Aggregate_SMS_Rev'], index=[' Class'],
                                             aggfunc=[len, np.sum, np.mean,
np.median]))
                    elif user_input_3 == 3:
                        print(pd.pivot_table(df_fill, values=['
Aggregate_Data_Rev'], index=[' Class'],
                                             aggfunc=[len, np.sum, np.mean,
np.median]))
                    elif user_input_3 == 4:
                        print(pd.pivot_table(df_fill, values=['
Aggregate_ONNET_REV'], index=[' Class'],
                                             aggfunc=[len, np.sum, np.mean,
np.median]))
                    elif user_input_3 == 5:
                        print(pd.pivot_table(df_fill, values=['
Aggregate_OFFNET_REV'], index=[' Class'],
                                             aggfunc=[len, np.sum, np.mean,
np.median]))
                    elif user_input_3 == 6:
                        print(pd.pivot_table(df_fill, values=['
Aggregate_Total_Rev'], index=[' Class'],
                                             aggfunc=[len, np.sum, np.mean,
np.median]))
                    elif user_input_3 == 7:
                        print(pd.pivot_table(df_fill, values=['
Aggregate_Data_Vol'], index=[' Class'],
                                             aggfunc=[len, np.sum, np.mean,
np.median]))
                    elif user_input_3 == 8:
                        print(pd.pivot_table(df_fill, values=['
Aggregate_complaint_count'], index=[' Class'],
                                             aggfunc=[len, np.sum, np.mean,
np.median]))
                    elif user_input_3 == 9:
                        print(pd.pivot_table(df_fill, values=['
Aggregate_Calls'], index=[' Class'],
                                             aggfunc=[len, np.sum, np.mean,
np.median]))
                    else:
                        print('Invalid input')

                    next_op = input("want to show more aggregate data with
churn index? (yes/no): ")
                    if next_op == "no":
                        break
    if user_option == 4:
        break
```

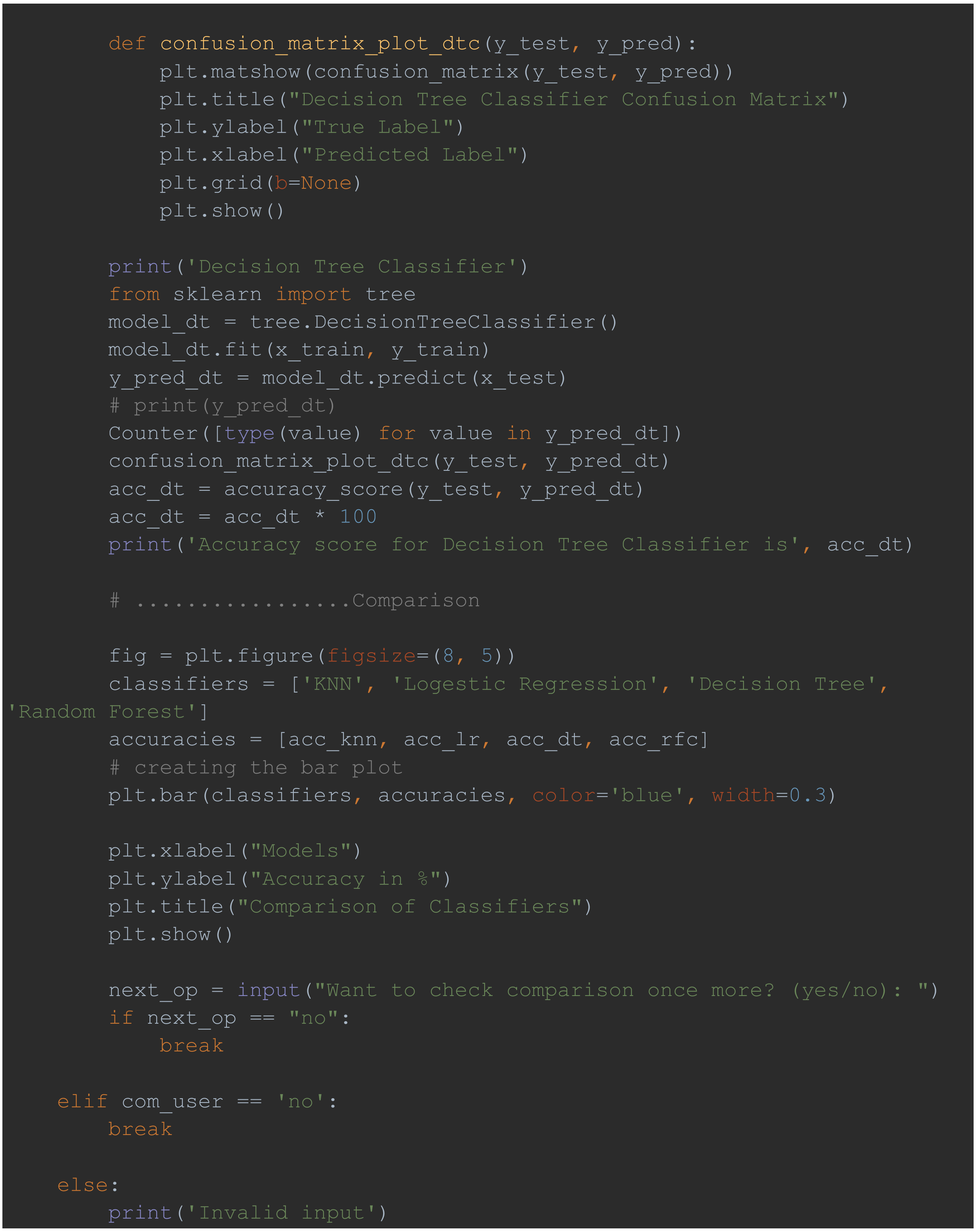


```
    else:
        print("Invalid Input")

# ....................................Diagnostic Analysis
print('Proceed towards Diagnostic Analysis')
while True:
    user_ans = input("Let's proceed towards Diagnostic Analysis (yes/no)?")
    print('\n')
    if user_ans == 'yes':
        # let's check whether any relation exists between aggregate
complaints counts by customer and customer churn
        print("lets check relation between aggregate complaints counts by
customer and customer churn")

        print("\nComplain count distribution for churned subscriber \n ",
              df[df_fill[' Class'] == "Churned"]["
Aggregate_complaint_count"].describe())
        print("\n")
        print("\nComplain count distribution for active subscriber \n ",
              df[df_fill[' Class'] == "Active"]["
Aggregate_complaint_count"].describe())

        # let's check Complain count distribution for each type of
subscriber where complaint count is more than 2
        print('\n')
        print("As max complaint count data lies in as 1 and 2, So let's
work at the remaining data")
        print("lets check relation between aggregate complaints counts > 2
by customer and customer churn")

        print("\nComplain count distribition > 2 for churned subscriber  \n
",
              df[(df_fill[' Class'] == "Churned") & (df_fill["
Aggregate_complaint_count"] > 2)][
                  " Aggregate_complaint_count"].describe())
        print("\n")
        print("\nComplain count distribition > 2 for active subscriber \n
",
              df[(df_fill[' Class'] == "Active") & (df_fill["
Aggregate_complaint_count"] > 2)][
                  " Aggregate_complaint_count"].describe())

        # lets check whether any relation exists between from how long time
customer taking services and customer churn

        print("\nNetwork age distribition for churned subscriber \n ",
              df[df_fill[' Class'] == "Churned"]["Network_age"].describe())
        print("\nNetwork age distribition for active subscriber \n ",
              df[df_fill[' Class'] == "Active"]["Network_age"].describe())

        next_op = input("Want to check diagnostic once more? (yes/no): ")
        if next_op == "no":
            break
    elif user_ans == 'no':
        break

    else:
        print('Invalid inout')

'''Index(['Network_age', ' Aggregate_Total_Rev', ' Aggregate_SMS_Rev',
       ' Aggregate_Data_Rev', ' Aggregate_Data_Vol', ' Aggregate_Calls',
       ' Aggregate_ONNET_REV', ' Aggregate_OFFNET_REV',
       ' Aggregate_complaint_count', ' Aug_user_type', ' Sep_user_type',
       ' Aug_fav_a', ' Sep_fav_a', ' Class'],
      dtype='object')
'''
# .........................................Visualization
while True:
    user_ans_1 = input("Proceed towards Visualization (yes/no)?")
    if user_ans_1 == 'yes':
        cols = [' Aug_fav_a', ' Sep_fav_a']
        numerical = cols

        plt.figure(figsize=(20, 4))

        for i, col in enumerate(numerical):
            ax = plt.subplot(1, len(numerical), i + 1)
            sns.countplot(x=str(col), data=df_fill)
            ax.set_title(f"{col}")
        plt.show()

        sns.boxplot(x=' Class', y=' Aggregate_Total_Rev', data=df_fill)
        plt.show()

        sns.boxplot(x=' Aug_user_type', y='Network_age', data=df_fill)
        plt.show()

        sns.boxplot(x=' Sep_user_type', y='Network_age', data=df_fill)
        plt.show()

        plt.subplot(1, 5, 1)
        sns.scatterplot(y=df_fill[' Class'], x=df_fill[' Aggregate_Calls'])
        plt.subplot(1, 5, 2)
        sns.scatterplot(y=df_fill[' Class'], x=df_fill['
Aggregate_ONNET_REV'])
        plt.subplot(1, 5, 3)
        sns.scatterplot(y=df_fill[' Class'], x=df_fill['
Aggregate_OFFNET_REV'])
        plt.subplot(1, 5, 4)
        sns.scatterplot(y=df_fill[' Class'], x=df_fill['
Aggregate_SMS_Rev'])
        plt.subplot(1, 5, 5)
        sns.scatterplot(y=df_fill[' Class'], x=df_fill['Network_age'])
        plt.suptitle("Checking for Bi variate Outliers")
        plt.show()
        next_op = input("Want to check visualization once more? (yes/no):
")
        if next_op == "no":
            break

    elif user_ans_1 == 'no':
        break

    else:
        print('Invalid inout')

# ..................................Prediction
print('\n')
print('Predictive Analysis')

while True:
    user_prd = input('Proceed towards predictive analysis (yes/no)?')
    if user_prd == 'yes':
        categorical_var = df_fill.drop(['Network_age', '
Aggregate_Total_Rev', ' Aggregate_SMS_Rev',
                                        ' Aggregate_Data_Rev', '
Aggregate_ONNET_REV', ' Aggregate_OFFNET_REV',
                                        ' Aggregate_complaint_count', '
Aggregate_Data_Vol', ' Aggregate_Calls'], axis=1)

        print(categorical_var.head())

        # working with categorical values

        from sklearn import preprocessing

        le = preprocessing.LabelEncoder()
        df_cat = categorical_var.apply(le.fit_transform)
        print(df_cat.head())

        num_features = df[['Network_age', ' Aggregate_Total_Rev', '
Aggregate_SMS_Rev',
                           ' Aggregate_Data_Rev', ' Aggregate_ONNET_REV', '
Aggregate_OFFNET_REV',
                           ' Aggregate_complaint_count', '
Aggregate_Data_Vol', ' Aggregate_Calls']]
        finaldf = pd.merge(num_features, df_cat, left_index=True,
right_index=True)

        # Splitting dataset into train and test models

        from sklearn.model_selection import train_test_split

        finaldf = finaldf.dropna()

        X = finaldf.drop([' Class'], axis=1)
        y = finaldf[' Class']
        X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

        from imblearn.over_sampling import SMOTE

        oversample = SMOTE(k_neighbors=5)
        X_smote, y_smote = oversample.fit_resample(X_train, y_train)
        X_train, y_train = X_smote, y_smote

        # to check equal division of dataset
        print('\n')
        print(y_train.value_counts())

        from sklearn.ensemble import RandomForestClassifier

        rf = RandomForestClassifier(random_state=46)
        rf.fit(X_train, y_train)

        from sklearn.metrics import accuracy_score

        preds = rf.predict(X_test)
        print(accuracy_score(preds, y_test))

        from sklearn.metrics import accuracy_score, recall_score,
confusion_matrix

        print('Accuracy: {}%'.format(100.0 * accuracy_score(preds,
y_test)))
        print('Recall:   {}%'.format(100.0 * recall_score(preds, y_test)))

        sns.heatmap(confusion_matrix(preds, y_test, labels=[0, 1]),
annot=True, fmt="d", linewidths=.5)
        plt.ylabel('True label')
        plt.xlabel('Predicted label')
        plt.show()

        next_op = input("Want to check predictions once more? (yes/no): ")
        if next_op == "no":
            break

    elif user_prd == 'no':
        break

    else:
        print('Invalid input')

# ................................ 4 models comparison

print('Comparing prediction with 4 different Classifier ')
while True:
    com_user = input('Want to proceed towards Campirson in prediction using
4 different classifiers (yes/no) ?')
    if com_user == 'yes':
```

```
X = df_fill.copy()
data = df_fill.copy()
Y = data.pop(' Class')
X.pop(' Class')

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(X, Y,
test_size=0.3, random_state=42)
x_train.shape, x_test.shape, y_train.shape, y_test.shape

x_train.drop(' Aug_user_type', axis=1, inplace=True)
x_train.drop(' Sep_user_type', axis=1, inplace=True)
x_train.drop(" Aug_fav_a", axis=1, inplace=True)
x_train.drop(" Sep_fav_a", axis=1, inplace=True)
x_train

x_test.drop(' Aug_user_type', axis=1, inplace=True)
x_test.drop(' Sep_user_type', axis=1, inplace=True)
x_test.drop(" Aug_fav_a", axis=1, inplace=True)
x_test.drop(" Sep_fav_a", axis=1, inplace=True)
x_test

from collections import Counter
from sklearn.metrics import confusion_matrix
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix


def confusion_matrix_plot_kn(y_test, y_pred):
    plt.matshow(confusion_matrix(y_test, y_pred))
    plt.title("KNeighborsClassifier Confusion Matrix")
    plt.ylabel("True Label")
    plt.xlabel("Predicted Label")
    plt.grid(b=None)
    plt.show()

# Import knearest neighbors Classifier model
# Create KNN Classifier
print('KNeighborsClassifier')
knn = KNeighborsClassifier(n_neighbors=5)
# Train the model using the training sets
knn.fit(x_train, y_train)
# Predict the response for test dataset
y_pred = knn.predict(x_test)
# print(y_pred)
Counter([type(value) for value in y_pred])
confusion_matrix_plot_kn(y_test, y_pred)

from sklearn.metrics._plot.roc_curve import roc_curve

'''def roc(model, x_test, y_test):
```



```
    probs = model.predict_proba(x_test)
    fpr, tpr, _ = roc_curve(y_test, probs[: 1])
    plt.plot(fpr, tpr, marker='.')
    plt.xlabel('False Postive Rate')
    plt.ylabel('True Positive Rate')
    plt.legend(loc='upper right')
    plt.show()'''

from sklearn.metrics import accuracy_score
acc = accuracy_score(y_test, y_pred)
acc_knn = acc * 100
print('Accuracy score for KNeighborsClassifier is', acc_knn)

def confusion_matrix_plot_lr(y_test, y_pred):
    plt.matshow(confusion_matrix(y_test, y_pred))
    plt.title("Logistic regression Confusion Matrix")
    plt.ylabel("True Label")
    plt.xlabel("Predicted Label")
    plt.grid(b=None)
    plt.show()

print('\n')
print('Logistic regression')
from sklearn.linear_model import LogisticRegression
model_lr = LogisticRegression(max_iter=10000000)
model_lr.fit(x_train, y_train)
intercept = model_lr.intercept_[0]
print(intercept)
y_pred_lr = model_lr.predict(x_test)
# print(y_pred_lr)
confusion_matrix_plot_lr(y_test, y_pred_lr)
acc_lr = accuracy_score(y_test, y_pred_lr)
acc_lr = acc_lr * 100
print('Accuracy score for Logistic Regression is ', acc_lr)
print('\n')

def confusion_matrix_plot_rfc(y_test, y_pred):
    plt.matshow(confusion_matrix(y_test, y_pred))
    plt.title("Random Forest classifier Confusion Matrix")
    plt.ylabel("True Label")
    plt.xlabel("Predicted Label")
    plt.grid(b=None)
    plt.show()

print('Random Forest classifier')
from sklearn.ensemble import RandomForestClassifier
model_rfc = RandomForestClassifier()
model_rfc.fit(x_train, y_train)
y_pred_rfc = model_rfc.predict(x_test)
confusion_matrix_plot_rfc(y_test, y_pred_rfc)
acc_rfc = accuracy_score(y_pred_rfc, y_test) * 100
print('Accuracy score for Random Forest Classifier is', acc_rfc)
print('\n')
```



```
        def confusion_matrix_plot_dtc(y_test, y_pred):
            plt.matshow(confusion_matrix(y_test, y_pred))
            plt.title("Decision Tree Classifier Confusion Matrix")
            plt.ylabel("True Label")
            plt.xlabel("Predicted Label")
            plt.grid(b=None)
            plt.show()

        print('Decision Tree Classifier')
        from sklearn import tree
        model_dt = tree.DecisionTreeClassifier()
        model_dt.fit(x_train, y_train)
        y_pred_dt = model_dt.predict(x_test)
        # print(y_pred_dt)
        Counter([type(value) for value in y_pred_dt])
        confusion_matrix_plot_dtc(y_test, y_pred_dt)
        acc_dt = accuracy_score(y_test, y_pred_dt)
        acc_dt = acc_dt * 100
        print('Accuracy score for Decision Tree Classifier is', acc_dt)

        # .................Comparison

        fig = plt.figure(figsize=(8, 5))
        classifiers = ['KNN', 'Logestic Regression', 'Decision Tree',
'Random Forest']
        accuracies = [acc_knn, acc_lr, acc_dt, acc_rfc]
        # creating the bar plot
        plt.bar(classifiers, accuracies, color='blue', width=0.3)

        plt.xlabel("Models")
        plt.ylabel("Accuracy in %")
        plt.title("Comparison of Classifiers")
        plt.show()

        next_op = input("Want to check comparison once more? (yes/no): ")
        if next_op == "no":
            break

    elif com_user == 'no':
        break

    else:
        print('Invalid input')
```

References

1. (OpenML, 2022)
2. (Rockikz, n.d.)
3. (Abdelrahim Kasem Ahmad, 2019)
4. S. A. Qureshi, A. S. Rehman, A. M. Qamar, A. Kamal, and A. Rehman, “Telecommunication subscribers’ churn prediction model using machine learning,” in 2013 Eighth International Conference on Digital Information Management (ICDIM), 2013, pp. 131–136.
5. W. Yu, D. N. Jutla, and S. C. Sivakumar, “A churn-strategy alignment model for managers in mobile telecom,” in Communication Networks and Services Research Conference, 2005. Proceedings of the 3rd Annual, 2005, pp. 48–53.
6. D. Collange, M. Hajji, J. Shaikh, M. Fiedler, and P. Arlos, “User impatience and network performance,” in 2012 8th EURO-NGI Conference on Next Generation Internet (NGI), 2012, pp. 141–148.
7. J. Shaikh, M. Fiedler, and D. Collange, “Quality of Experience from user and network perspectives,” Annals of Telecommunications, 65(1—2):47—57, Jan./Feb. 2010. [Accessed: 12-Feb-2016].
8. M. R. Khan, J. Manoj, A. Singh, and J. Blumenstock, “Behavioral Modeling for Churn Prediction: Early Indicators and Accurate Predictors of Custom Defection and Loyalty,” in 2015 IEEE International Congress on Big Data (BigData Congress), 2015, pp. 677–680.
9. “AUTOCORRELATION” [Online]. Available: http://www.ltrr.arizona.edu/~dmeko/notes_3.pdf. [Accessed: 14-Sep-2016].
10. W. M. C. Bandara, A. S. Perera, and D. Alahakoon, “Churn prediction methodologies in the telecommunications sector: A survey,” in 2013 International Conference on Advances in ICT for Emerging Regions (ICTer), 2013, pp. 172–176.
11. A. Idris and A. Khan, “Ensemble Based Efficient Churn Prediction Model for Telecom,” in 2014 12th International Conference on Frontiers of Information Technology (FIT), 2014, pp. 238–244.
12. “Matthews correlation coefficient,” Wikipedia, the free encyclopedia. 08-Sep-2016.