Northeastern University

# Customer Retention Analysis

## Project Report

INFO 7390: Advances in Data Sciences & Architecture

April 28th, 2018

**Team 4**

| | |
|---|---|
| Eklavya Saxena | 001850025 |
| Ankur Jain | 001206900 |
| Amandeep Singh | 001271649 |

**Supervisor**

Prof. Sri Krishnamurthy

# Table of Contents

# Abstract

Retaining customers is one of the most critical challenges in the service industry. Using customer information, this study investigates determinants of customer churn in any service provider firm. Results indicate that service quality-related factors influence customer churn; however, customers participating in various contract programs are also more likely to churn, which raises questions about program effectiveness. Furthermore, heavy users also tend to churn. In order to analyze partial and total defection, this study defines changes in a customer's status from active use (using the service on a regular basis) to non-use (deciding not to use it temporarily without having churned yet) or suspended (being suspended by the service provider) as partial defection and from active use to churn as total defection. Thus, mediating effects of a customer's partial defection on the relationship between the churn determinants and total defection are analyzed and their implications are discussed. Results indicate that some churn determinants influence customer churn, either directly or indirectly through a customer's status change, or both; therefore, a customer's status change explains the relationship between churn determinants and the probability.

# 1.Introduction

The subject of customer retention, loyalty, and churn is receiving attention in many industries. This is important in the customer lifetime value context. A company will have a sense of how much is really being lost because of the customer churn and the scale of the efforts that would be appropriate for retention campaign. The mass marketing approach cannot succeed in the diversity of consumer business today. Customer value analysis along with customer churn predictions will help marketing programs target more specific groups of customers.

This project will present a customer churn analysis in Telecommunication sector. The goal of this project is twofold. First the churning customers are analyzed in context of services. The second objective is a forecast of churning customers based on various predictive models using supervised approach.

# 2. Need for Customer churn prediction

Our data in this project was of company operating in a Telecommunication sector. In this sector a company must operate on a long-term customer strategy, young customers are recognized as being unprofitable in the early stage in lifecycle but will become profitable later on. So as the customer relationships last, maybe decades, the company must address the value of a potential loss of a customer. The customer lifetime value analysis will help to face this challenge.

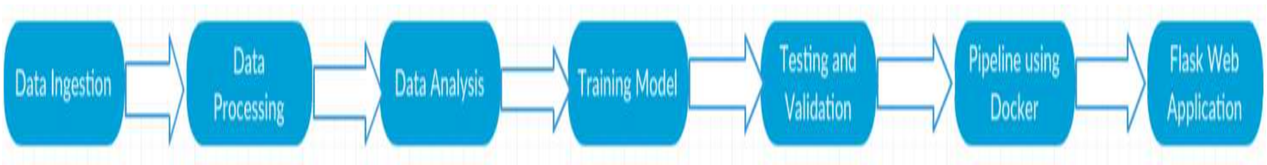
# 3. Data Source

In this study a customer data from a Telecommunication is used and analyzed.

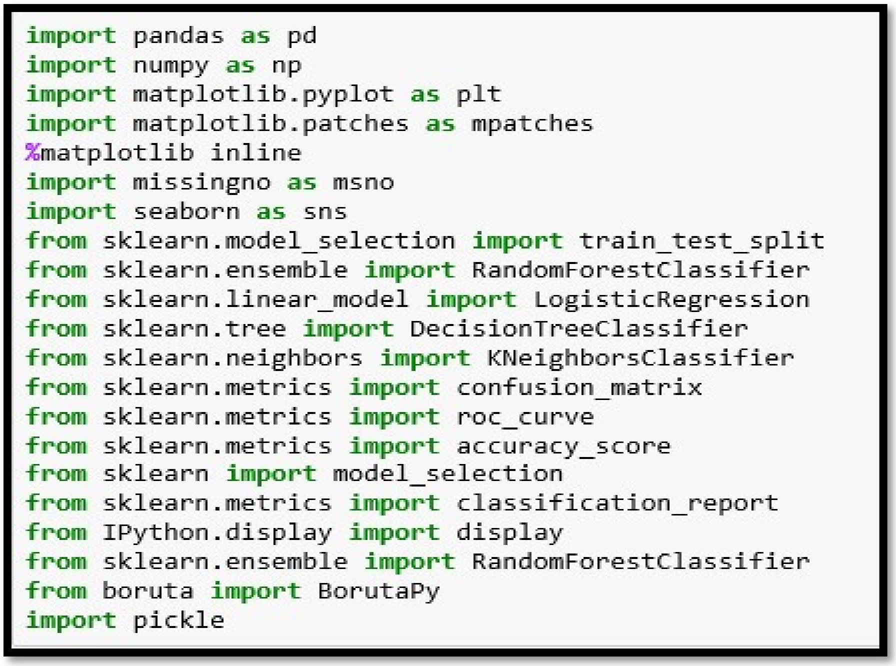
The dataset consists of 7043 records/rows and 21 fields/columns

- Customers who left within the last month – the column is called Churn
- Target Variable – Churn: Binary Classification {Yes or No}
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies
- Customer account information – how long they’ve been a customer, contract, payment method, paperless billing, monthly charges, and total charges
- Demographic info about customers – gender, age range, and if they have partners and dependents

# 4. Approach



## Important Libraries

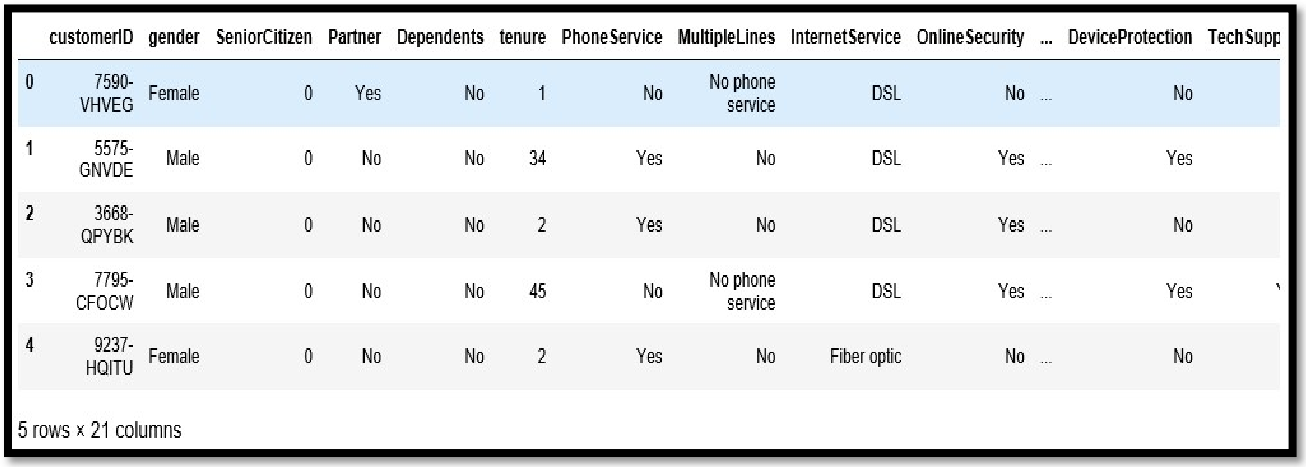


```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
%matplotlib inline
import missingno as msno
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_curve
from sklearn.metrics import accuracy_score
from sklearn import model_selection
from sklearn.metrics import classification_report
from IPython.display import display
from sklearn.ensemble import RandomForestClassifier
from boruta import BorutaPy
import pickle
```

Above Libraries are used for analysis and modelling.

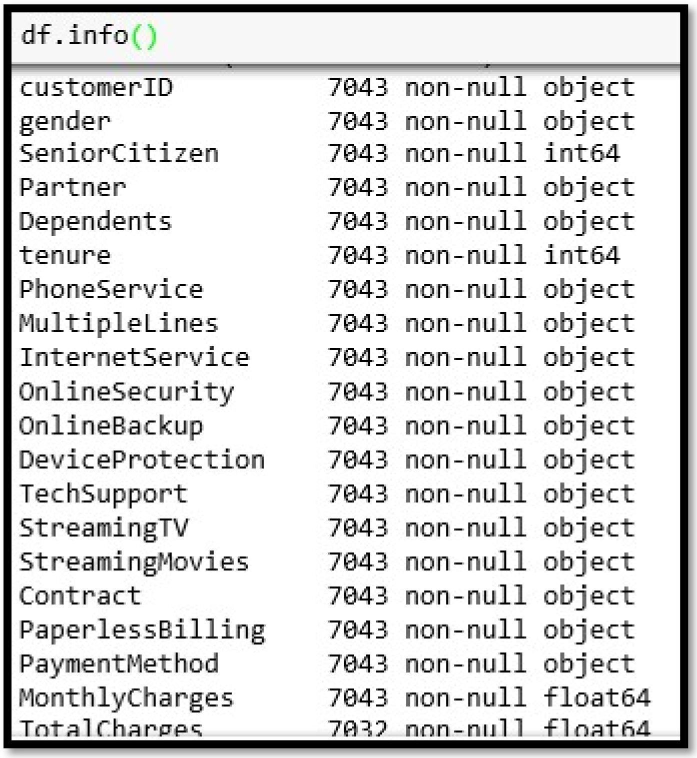
The input dataset has 21 fields, and it has 11 missing values.

The below picture describes the structure of dataset



| | customerID | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLines | InternetService | OnlineSecurity | ... | DeviceProtection | TechSupp |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 7590-VHVEG | Female | 0 | Yes | No | 1 | No | No phone service | DSL | No | ... | No | |
| 1 | 5575-GNVDE | Male | 0 | No | No | 34 | Yes | No | DSL | Yes | ... | Yes | |
| 2 | 3668-QPYBK | Male | 0 | No | No | 2 | Yes | No | DSL | Yes | ... | No | |
| 3 | 7795-CFOCW | Male | 0 | No | No | 45 | No | No phone service | DSL | Yes | ... | Yes | |
| 4 | 9237-HQITU | Female | 0 | No | No | 2 | Yes | No | Fiber optic | No | ... | No | |

5 rows × 21 columns



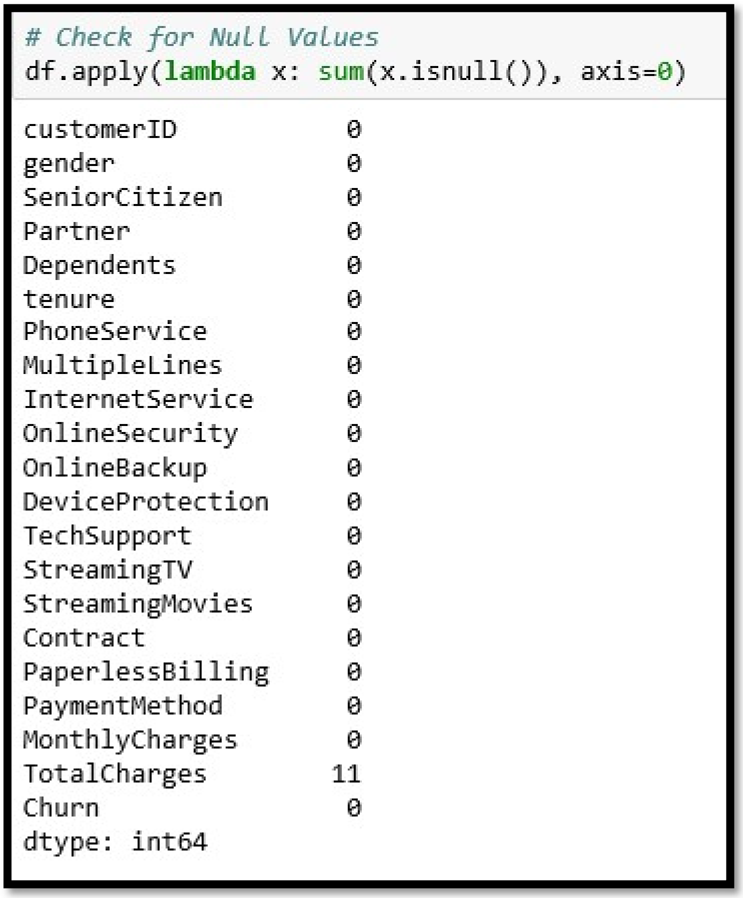
```
df.info()
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
tenure              7043 non-null int64
PhoneService        7043 non-null object
MultipleLines       7043 non-null object
InternetService     7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null object
StreamingMovies     7043 non-null object
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
TotalCharges        7032 non-null float64
```

# 5.Data Wrangling

Data wrangling, is the process of transforming and mapping data from one "raw" data form into another format with the intent of making it more appropriate.

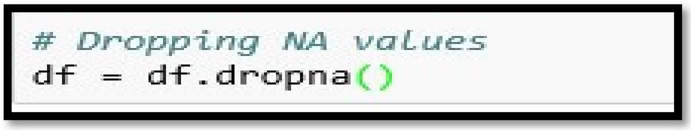
## 5.1 Handling Missing Data

This data consists of 11 missing values in TotalCharges column, so to handle this it has been replaced by NaN values and later these NaN values are deleted from the data to get clean data.



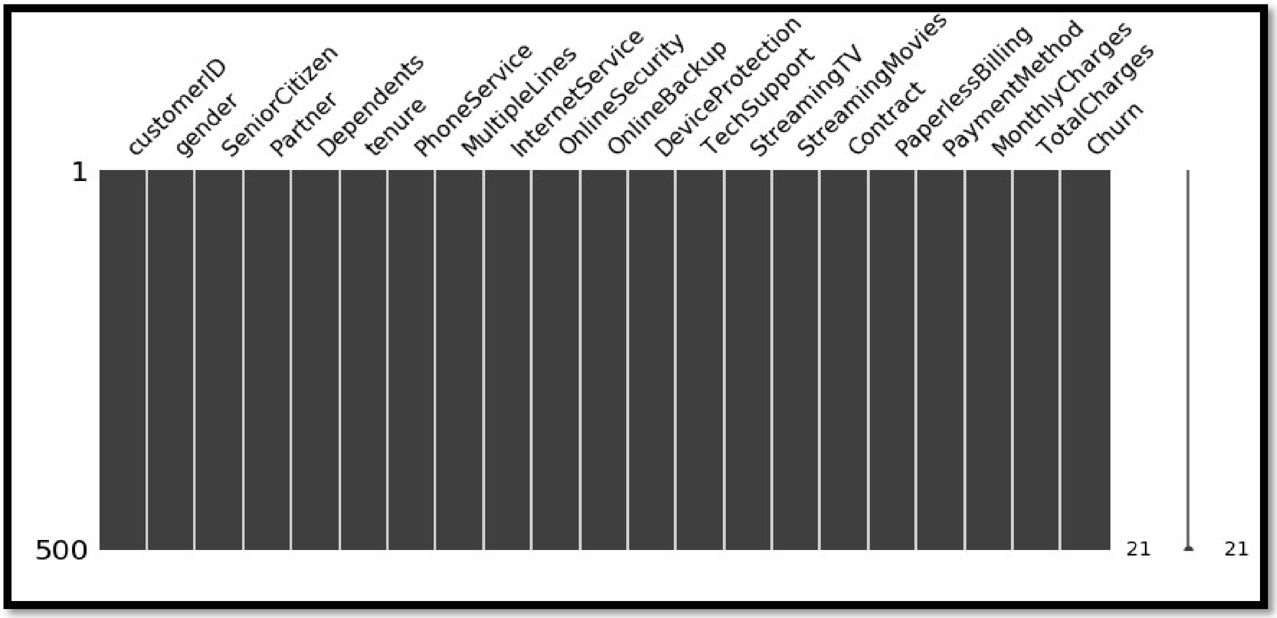
```
# Check for Null Values
df.apply(lambda x: sum(x.isnull()), axis=0)
```

```
customerID           0
gender               0
SeniorCitizen        0
Partner              0
Dependents           0
tenure               0
PhoneService         0
MultipleLines        0
InternetService      0
OnlineSecurity       0
OnlineBackup         0
DeviceProtection     0
TechSupport          0
StreamingTV          0
StreamingMovies      0
Contract             0
PaperlessBilling     0
PaymentMethod        0
MonthlyCharges       0
TotalCharges        11
Churn                0
dtype: int64
```



```
# Dropping NA values
df = df.dropna()
```

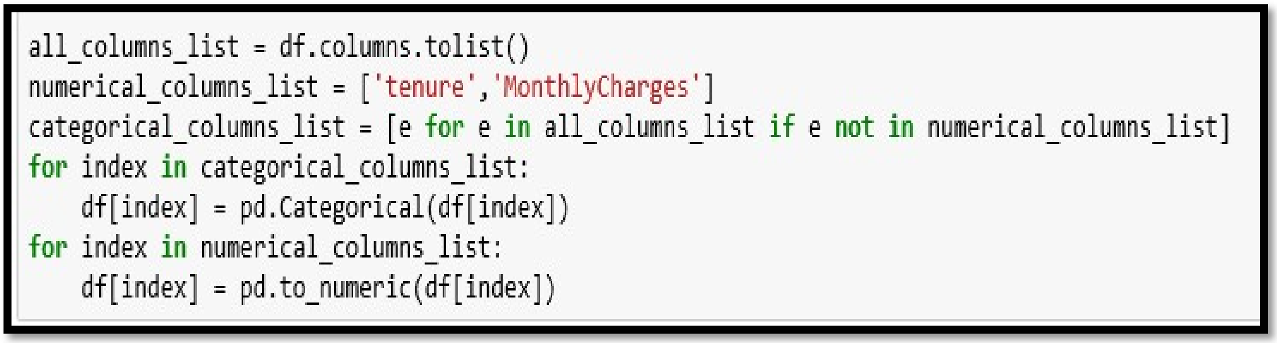
## Visualizing Missingness of Data



From the above plot it is clear that there is no missing data in the dataset,

## 5.2 Data Manipulation

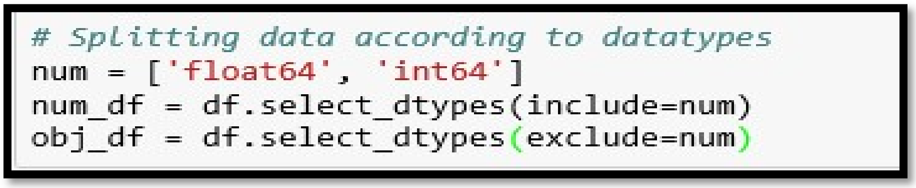
In this dataset there are 18 variables which are categorical, so converting these variables into categorical datatype for further analysis



```
all_columns_list = df.columns.tolist()
numerical_columns_list = ['tenure','MonthlyCharges']
categorical_columns_list = [e for e in all_columns_list if e not in numerical_columns_list]
for index in categorical_columns_list:
    df[index] = pd.Categorical(df[index])
for index in numerical_columns_list:
    df[index] = pd.to_numeric(df[index])
```

Since this dataset consists of both numeric and categorical variables, so splitting of these variables into groups of their specific datatype will facilitate the analysis

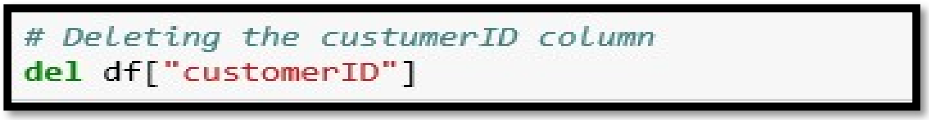
Splitting the Numeric and Object variables



```
# Splitting data according to datatypes
num = ['float64', 'int64']
num_df = df.select_dtypes(include=num)
obj_df = df.select_dtypes(exclude=num)
```

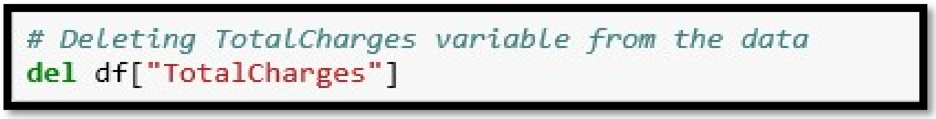
## 5.3 Variable Reduction

Started with 21 variables in this dataset, it was observed the CustomerId is unique for every customer record and cannot influence churn prediction, which results in removal of this variable from the dataset



```
# Deleting the custumerID column
del df["customerID"]
```

After exploring data, it was observed that TotalCharges are derived from MonthlyCharges and approximately equal to product of MonthlyCharges and tenure. So deleting this variable from dataset



```
# Deleting TotalCharges variable from the data
del df["TotalCharges"]
```

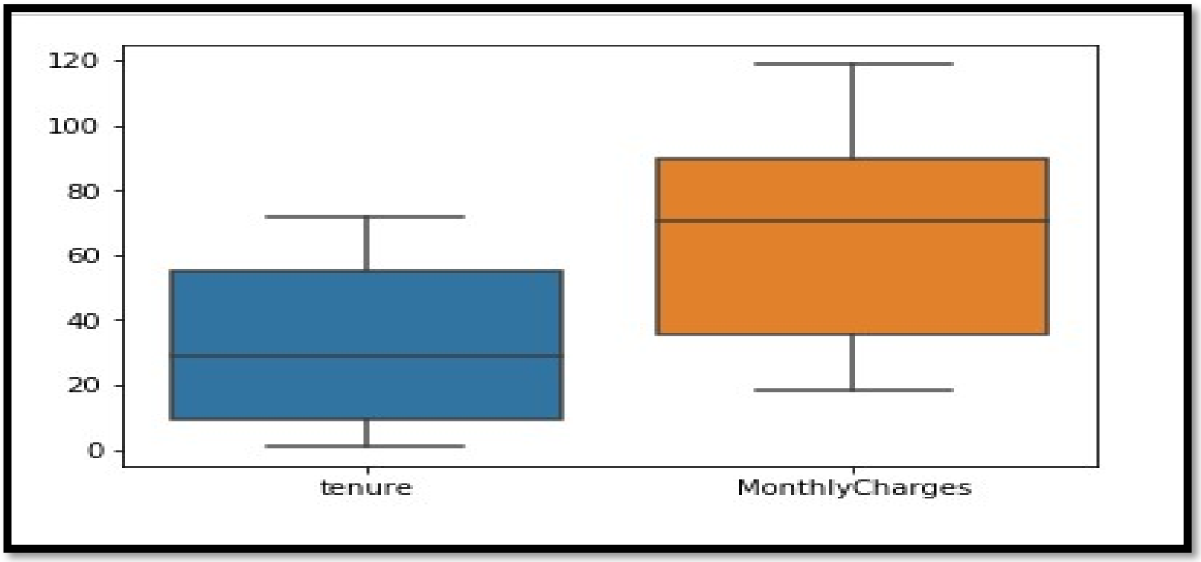
# 6. Exploratory Data Analysis

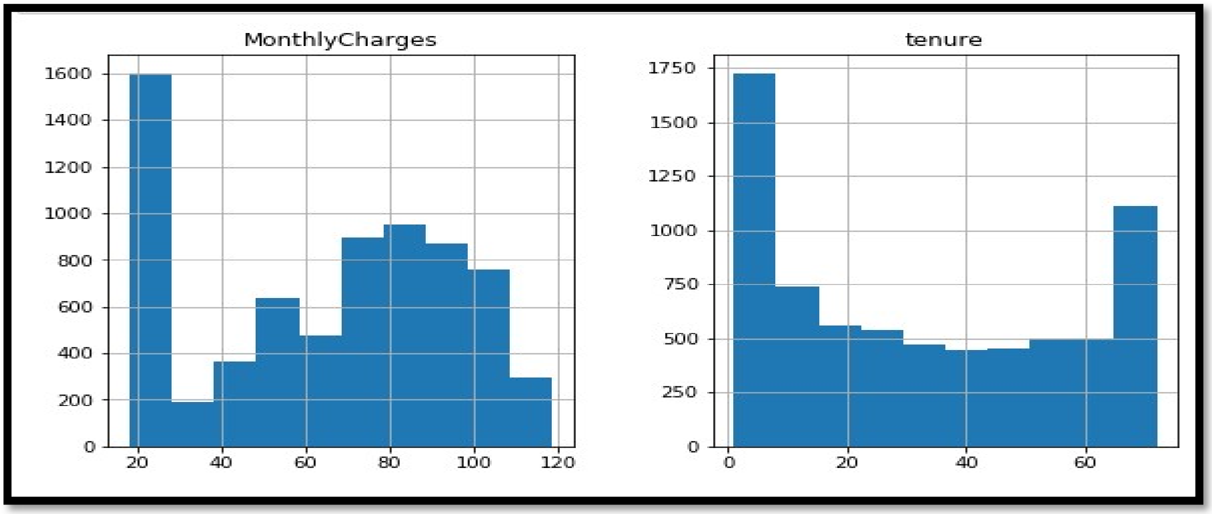
Exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

## 6.1 Univariate Analysis

A univariate frequency analysis was used to pinpoint value distributions, missing values and outliers. For Univariate analysis following plots are used

1.Boxplots
2.Histograms

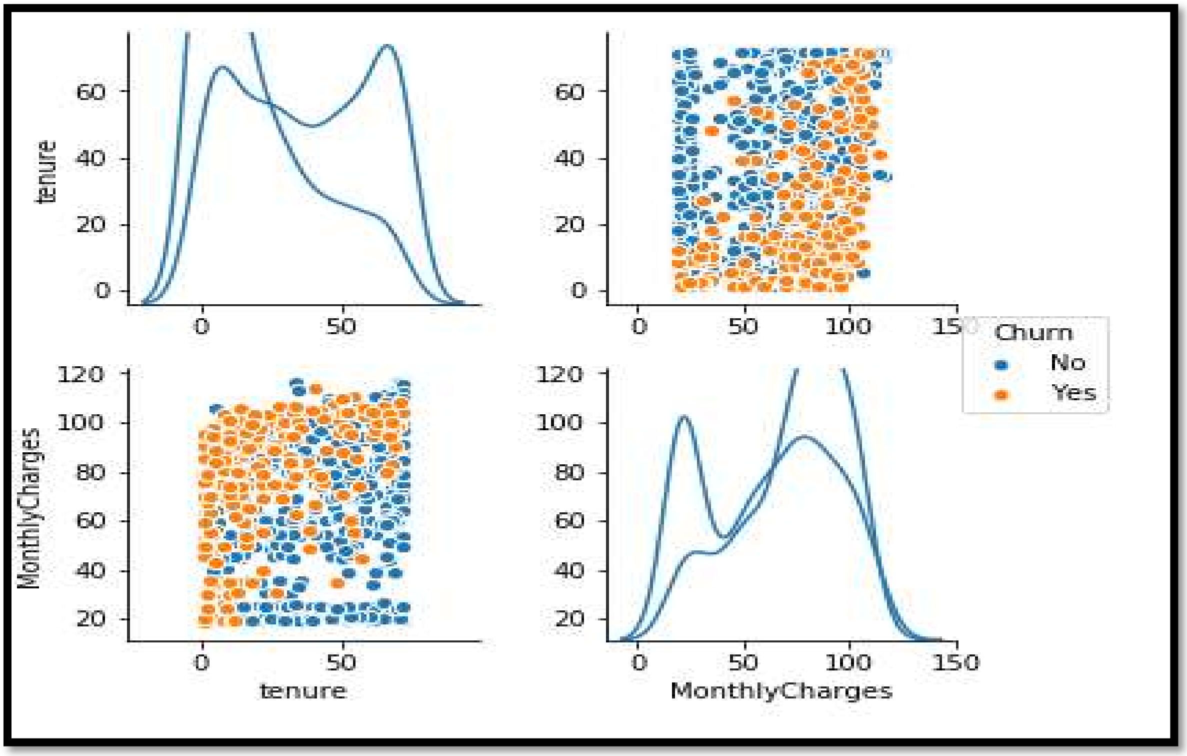




The above plots show that

- Numeric variables are not normally distributed.

Further, checking how they are related to the target variable



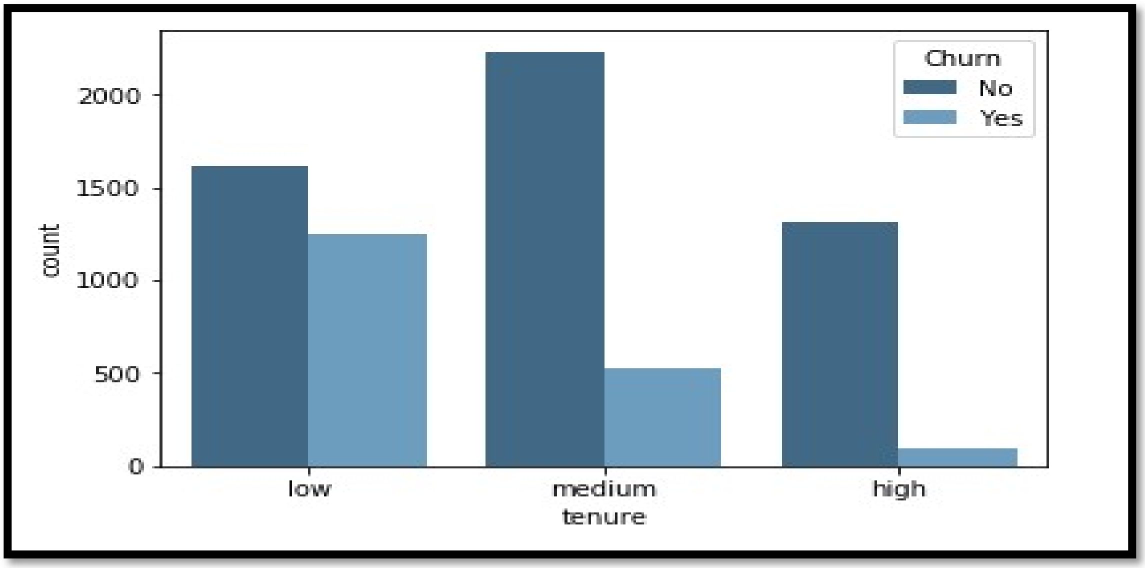
- It is clear that the numeric variables of dataset including MonthlyCharges and tenure are evenly distributed in the context of Churn and are good predictors of Churn variable.

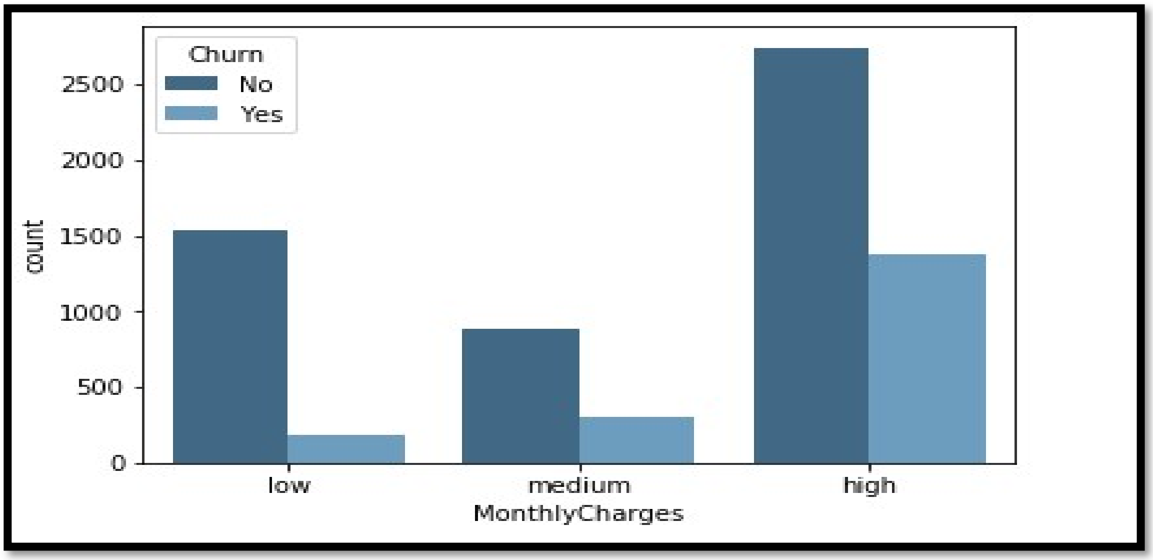
## 6.2 Bivariate Analysis

Bivariate Distribution helps to identify how each numeric variable is distributed with respect to target variable 'Churn'

The numerical variables are divided in 3 bins trying to separate them in a way to distinguish the low churn rate areas from the high churn rate areas, justified by count plots showing how distribution is different in each bin.

- Countplots are used for Bivariate analysis of variables





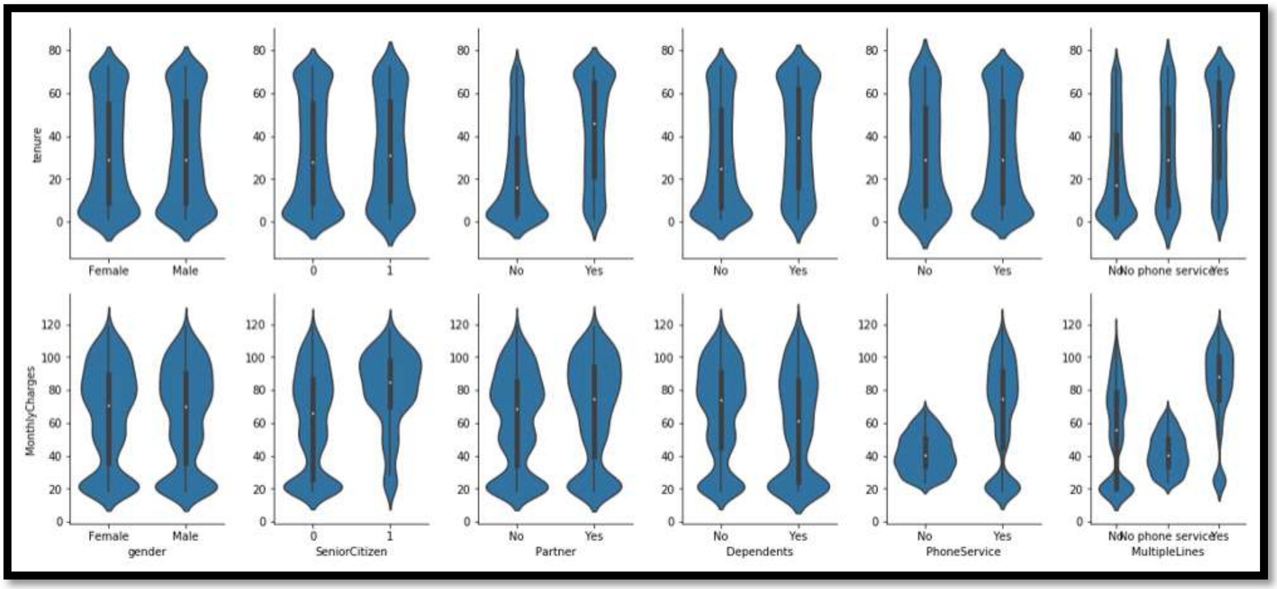
From the above following can be deducted

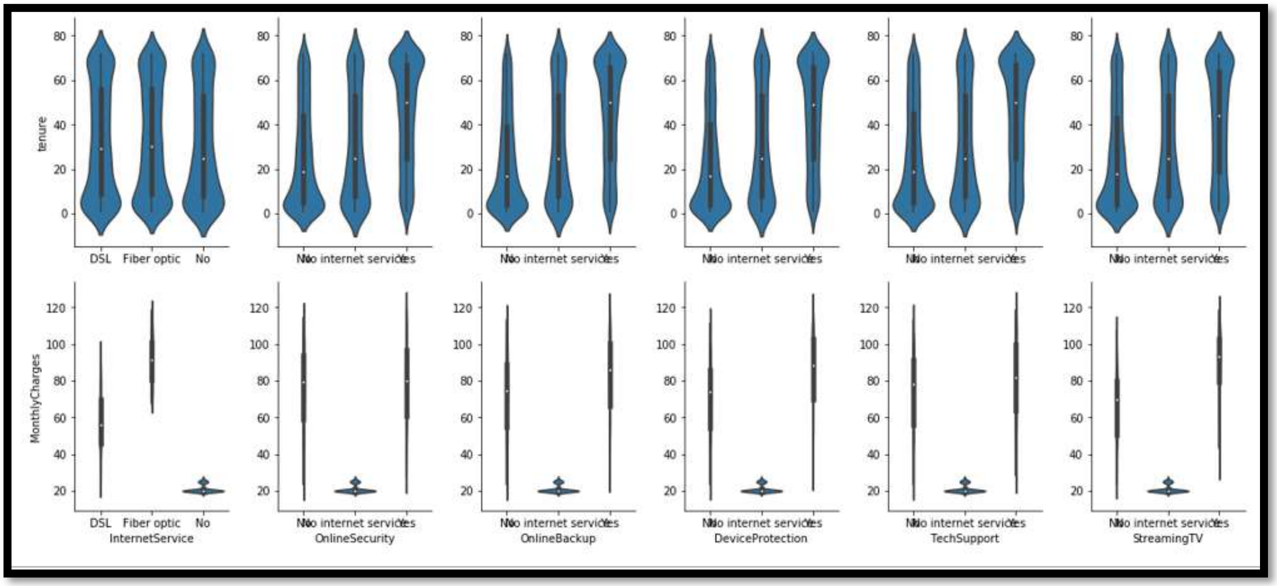
- Less than a year tenure category has highest churn
- Customers paying high Monthly charges tends to churn more
- 2 to 5 years tenure category results in highest revenue loss
- There is not much variation in medium bin which means customer remain loyal as they cross a year

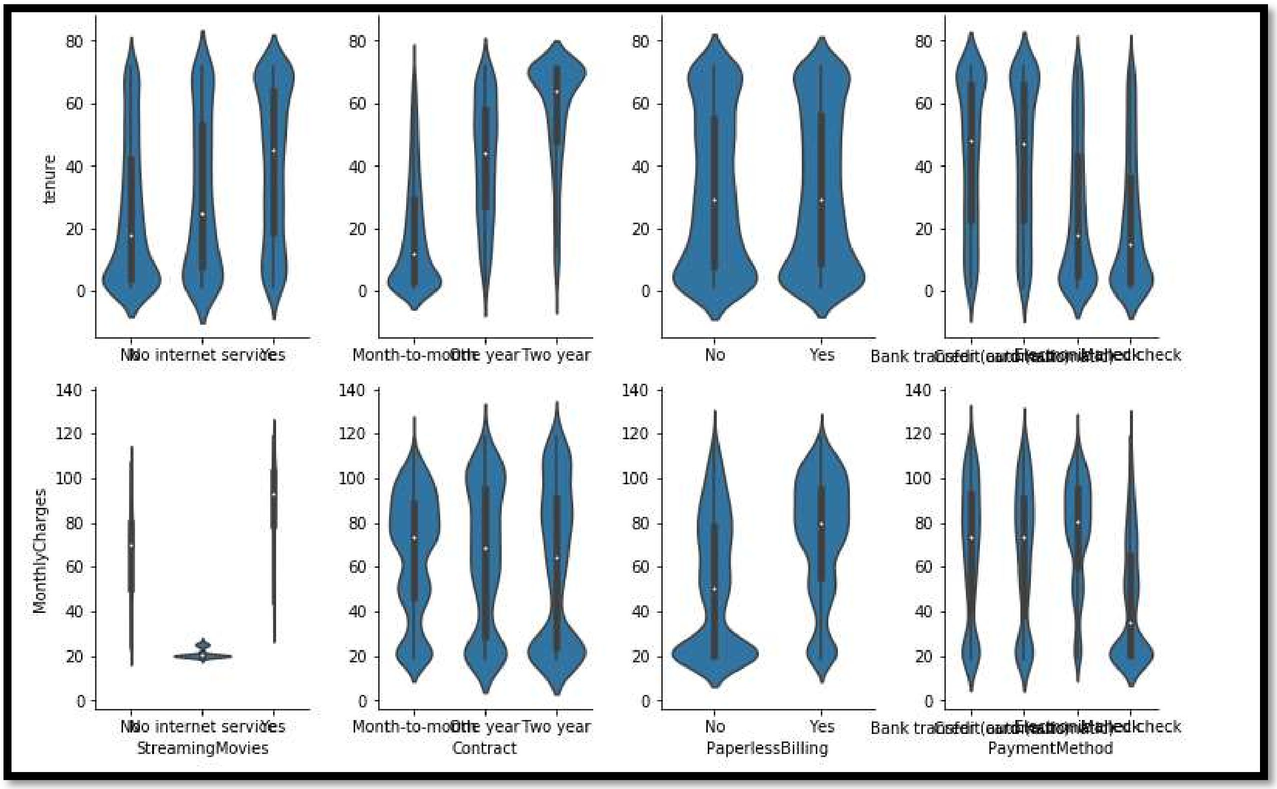
## 6.3 Numerical vs Categorical Variable Analysis

This part of analysis helps to explore the dependency of variables on each other.

Since there are too many categorical variables, splitting them into 3 different sets and then plotting each set with the numerical variables







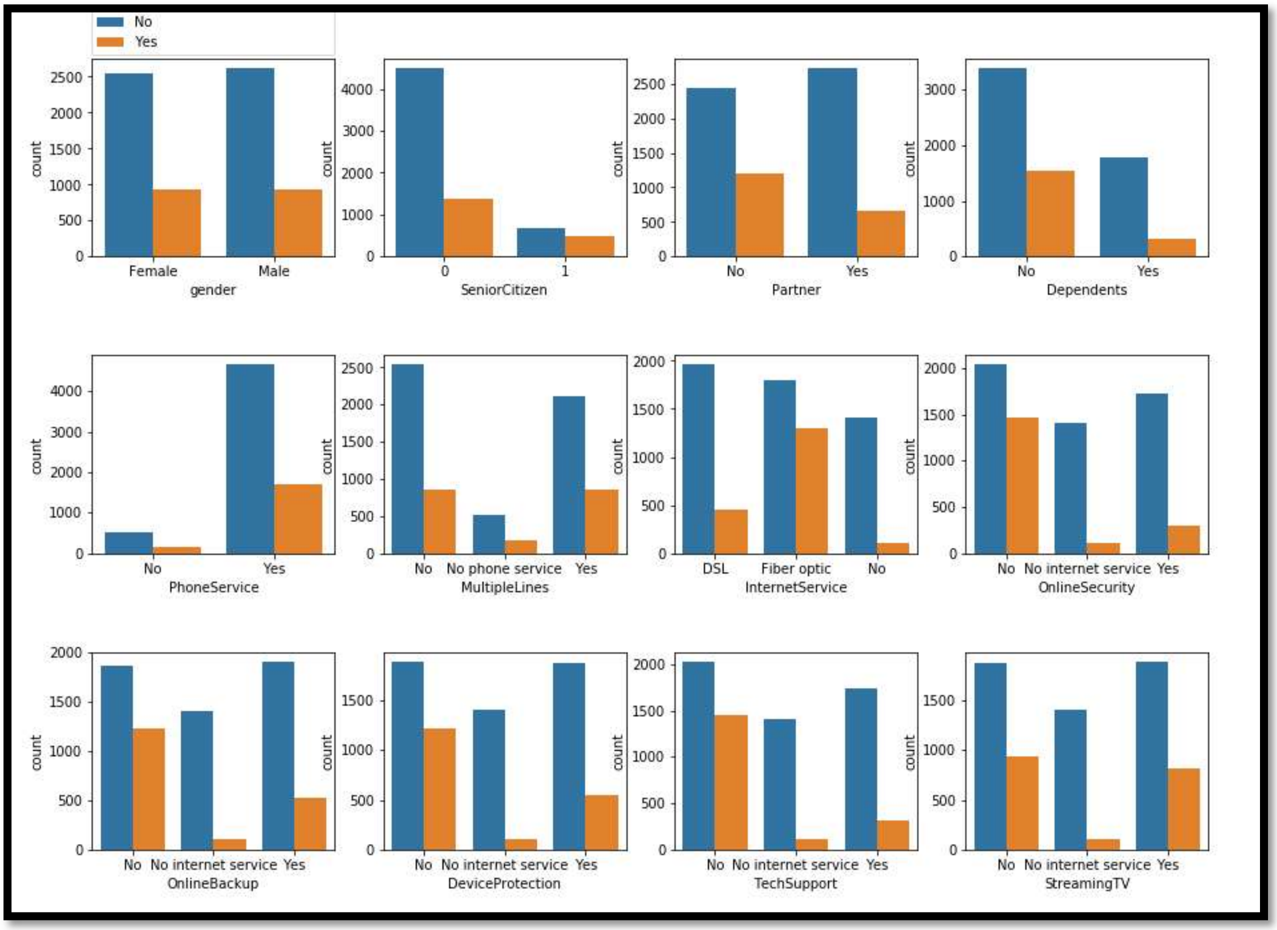
Above plots are helpful to deduce following results:

- Customers with Eletronic or Mailed check payment method have a lower tenure
- Variable like gender does not influence tenure and monthly payment
- The importance of Fibre optic service on monthly revenue is very clear

The above section of analysis of numeric and categorical variables are very important to design the campaigns for the firm to focus on the specific areas which needs to be enhanced in terms of service quality or affordable price to reduce the customer's churn rate.

## 6.4 Categorical Variable Analysis

Main target for analysis of categorical values is to focus on the variables that delivers best results in context of churn of clients



The above plots give a better picture of variables that are important for Churn prediction

- Month-to-month contract is a strong indicator if the client might leave
- Electronic check payment method also provides clear view to the client stability
- On the other hand, Senior citizen is a good predictor but only represents a small amount of clients

# 7. Feature Engineering

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. Feature engineering is fundamental to the application of machine learning.

Feature engineering is an informal topic, but it is considered essential in applied machine learning.

In the dataset:

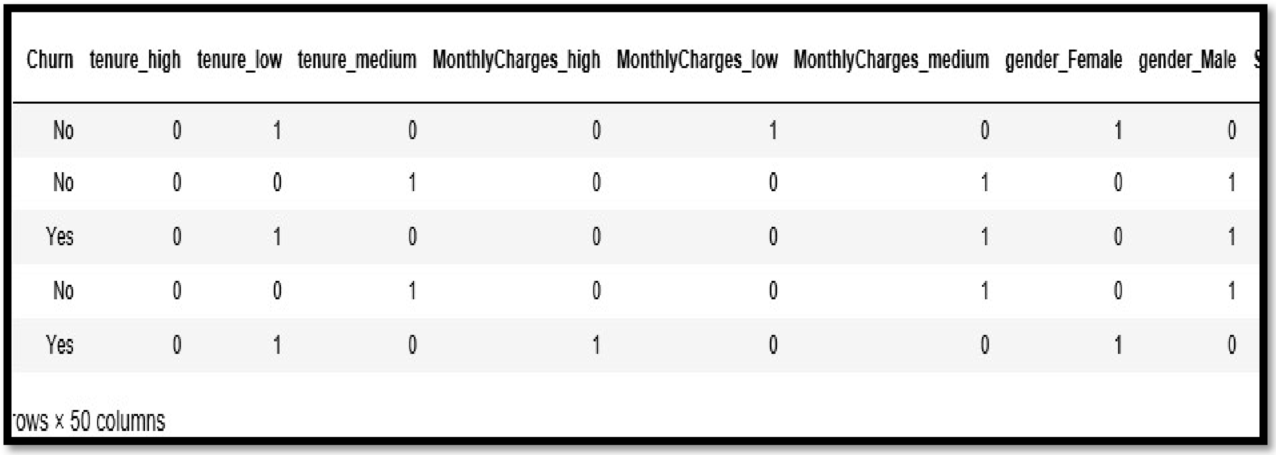
- There are columns with a scope of transformation, like SeniorCitizen
- There is no null data in the dataset as it has been removed initially
- Variable like CustomerId is redundant as it does not help in prediction of the churn of the clients, already dropped
- As per Numerical variable analysis TotalCharges and Monthly Charges are correlated and MonthlyCharges is approximately equal to product of MonthlyCharges and tenure which makes MonthlyCharges redundant, already dropped

## 7.1 Dummy Variables

A dummy variable (also known as an indicator variable, design variable, Boolean indicator, binary variable, or qualitative variable) is one that takes the value 0 or 1 to indicate the absence or presence of some categorical effect that may be expected to shift the outcome. Dummy variables are used as devices to sort data into mutually exclusive categories. A dummy variable can thus be thought of as a truth value represented as a numerical value 0 or 1

In this project we have created dummy variables for all variables except target variable Churn

Below show the dummy variables of the dataset



| Churn | tenure_high | tenure_low | tenure_medium | MonthlyCharges_high | MonthlyCharges_low | MonthlyCharges_medium | gender_Female | gender_Male |
|---|---|---|---|---|---|---|---|---|
| No | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| No | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| Yes | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| No | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| Yes | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |

ows × 50 columns

- Dummy variables are used as devices to sort data into mutually exclusive categories

- Converting the categorial variables into dummy variables (extensive categories) indicate the occurrence of major prediction values

# 8. Feature Selection

In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction.

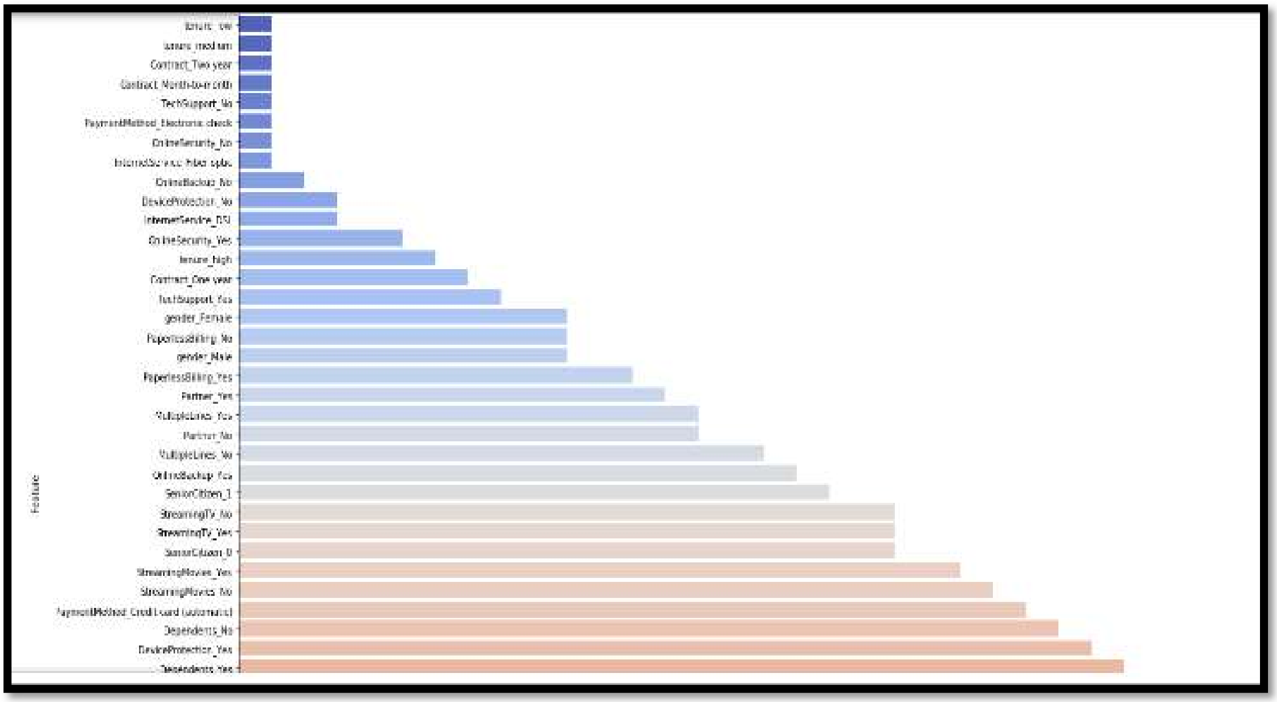
The central premise when using a feature selection technique is that the data contains many features that are either redundant or irrelevant and can thus be removed without incurring much loss of information.

We have implemented Boruta Algorithm in this project for feature selection.

## 8.1 Boruta Algorithm

The Boruta algorithm is a wrapper built around the random forest classification algorithm. It tries to capture all the important, interesting features you might have in your dataset with respect to an outcome variable.

After implementing Boruta library, the below graph shows the rank of the predictors



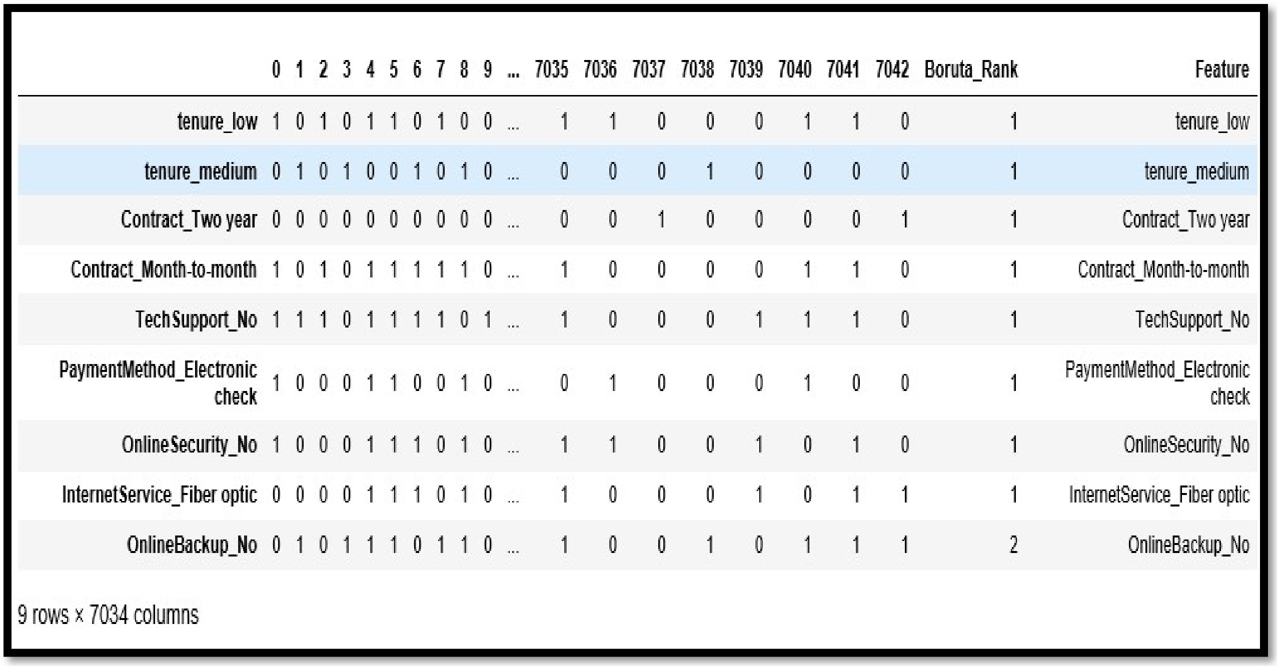
Feature selection technique using Boruta are used for four reasons:

- Simplification of models to make them easier to interpret by researchers/users
- Shorter training times
- Enhanced generalization by reducing Overfitting

## 8.2 Defining Hyperparameters

From above feature selection method, it is easy to define some hyperparameters which are selected on the basis of their rank with respect to other variables

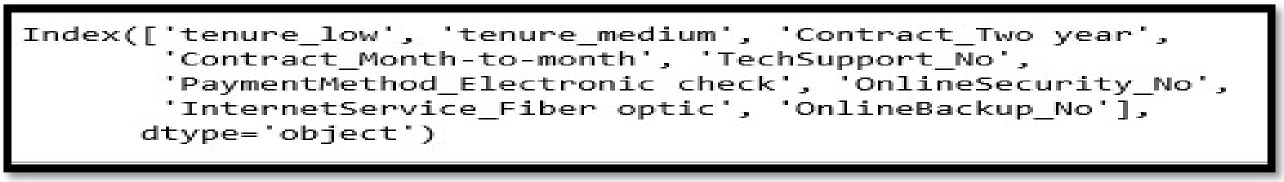
To select enough and concise number of parameters, variables of **Rank 1 & 2** will suffice the need of effective input for prediction algorithms.



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 7035 | 7036 | 7037 | 7038 | 7039 | 7040 | 7041 | 7042 | Boruta_Rank | Feature |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| tenure_low | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | ... | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | tenure_low |
| tenure_medium | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | tenure_medium |
| Contract_Two year | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | Contract_Two year |
| Contract_Month-to-month | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | ... | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | Contract_Month-to-month |
| TechSupport_No | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | ... | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | TechSupport_No |
| PaymentMethod_Electronic check | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | ... | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | PaymentMethod_Electronic check |
| OnlineSecurity_No | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | ... | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | OnlineSecurity_No |
| InternetService_Fiber optic | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | ... | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | InternetService_Fiber optic |
| OnlineBackup_No | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | ... | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 2 | OnlineBackup_No |

9 rows × 7034 columns

The above selected features will be used to train the different models using supervised approach for predicting Clients Churn. Below is the image of selected features



```
Index(['tenure_low', 'tenure_medium', 'Contract_Two year',
       'Contract_Month-to-month', 'TechSupport_No',
       'PaymentMethod_Electronic check', 'OnlineSecurity_No',
       'InternetService_Fiber optic', 'OnlineBackup_No'],
      dtype='object')
```

# 9. Prediction Algorithms

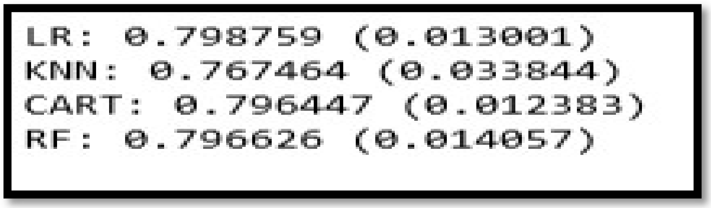
**Split Training and Testing Dataset**

For this problem, we split whole data into 80% training and 20% testing

To predict the customer Churn, we have used 4 following algorithms

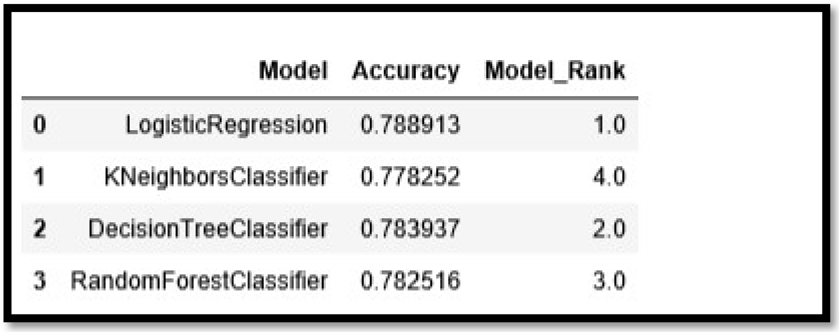
- Logistic Regression
- KNeighborsClassifier
- DecisionTreeClassifier
- RandomForestClassifier

By training above models for training data, we get the below metrics



```
LR: 0.798759 (0.013001)
KNN: 0.767464 (0.033844)
CART: 0.796447 (0.012383)
RF: 0.796626 (0.014057)
```

Now implementing above models on test data for predictions, we get the following metrics by rank, where Ranks are given to the models as per their accuracy score



| | Model | Accuracy | Model_Rank |
|---|---|---|---|
| 0 | LogisticRegression | 0.788913 | 1.0 |
| 1 | KNeighborsClassifier | 0.778252 | 4.0 |
| 2 | DecisionTreeClassifier | 0.783937 | 2.0 |
| 3 | RandomForestClassifier | 0.782516 | 3.0 |

From the above it can be concluded that the best model for this problem is **Logistic Regression** with better overall accuracy among all models.

# 10. Pipeline

The entire framework from converting raw data to data usable by Machine Learning algorithm, training an algorithm, and finally using the output of this algorithm to perform actions in the real-world is called **pipeline**.

For this project, Luigi pipeline is used which is a Python-based framework for expressing data pipelines.

Luigi pipeline is used for

- Establishing consistency in data
- Developing a common ETL (Extract, Transform and Load) process

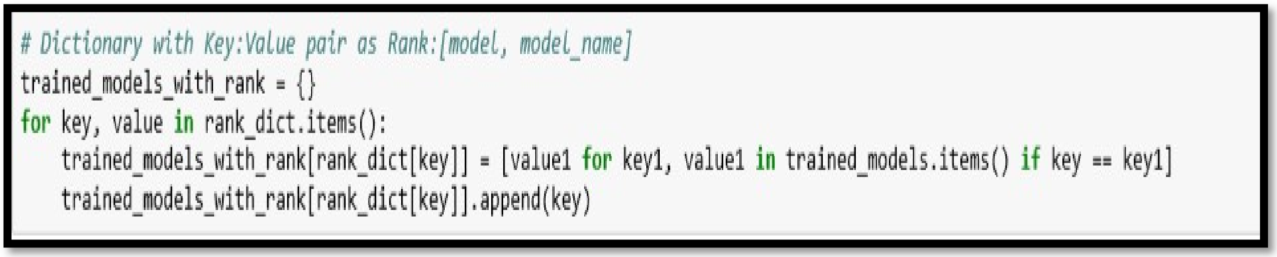
# 10. Deployment

After you have a set of models that perform well, you can operationalize them for other applications to consume. Depending on the business requirements, predictions are made either in real time basis. Deployment is a crucial step which enables the model to be easily consumed from various applications.

## 10.1 Serialization

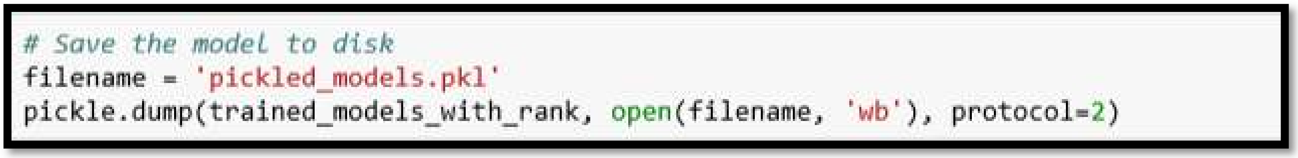
In Data Science, in the context of data storage, **Serialization** is the process of translating data structures or object state into a format that can be stored (for example, in a file or memory buffer) or transmitted (For example, across a network) and reconstructed later (possibly in a different computer environment).

**Pickle** is the standard way of serializing objects in Python. We can use the pickle operation to serialize our machine learning algorithms and save the serialized format to a file. Later we can load this file to deserialize our model and use it to make new predictions.



```
# Dictionary with Key:Value pair as Rank:[model, model_name]
trained_models_with_rank = {}
for key, value in rank_dict.items():
    trained_models_with_rank[rank_dict[key]] = [value1 for key1, value1 in trained_models.items() if key == key1]
    trained_models_with_rank[rank_dict[key]].append(key)
```

- The above will store the Models rank as keys and Model_var as values



```
# Save the model to disk
filename = 'pickled_models.pkl'
pickle.dump(trained_models_with_rank, open(filename, 'wb'), protocol=2)
```

- Above will save the pickle format in a file.

## 10.2 Amazon S3

Amazon S3 is a web service that provides storage through web services.

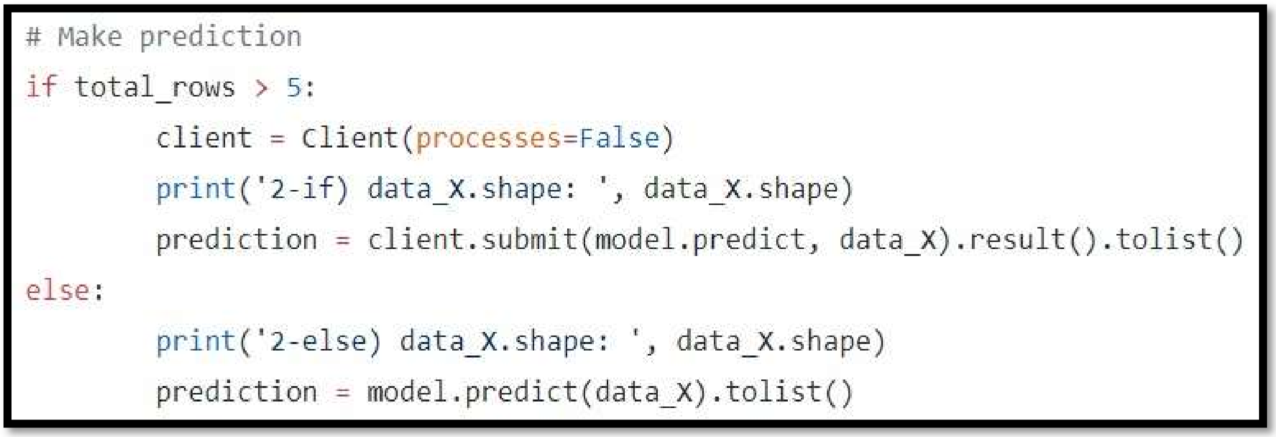
In this project, Once, the pickle script is written, pickle file is uploaded on Amazon S3.

## 10.3 Dask

Parallel computing or Parallel Processing is a type of computing architecture in which several processors execute or process an application or computation simultaneously. Parallel computing helps in performing large computations by dividing the workload between more than one processor, all of which work through the computation at the same time.

In this project to scale up the computation, Dask Frame work is used. Dask scales up from a single node to thousand-node clusters.

The below picture shows the implementation of Dask



```
# Make prediction
if total_rows > 5:
        client = Client(processes=False)
        print('2-if) data_X.shape: ', data_X.shape)
        prediction = client.submit(model.predict, data_X).result().tolist()
else:
        print('2-else) data_X.shape: ', data_X.shape)
        prediction = model.predict(data_X).tolist()
```

## 10.4 Docker

It is a computer program that performs operating-system-level virtualization. Docker image automates the task and makes it OS independent. The below snippet is of the docker file. The link to the docker image created can be accessed via:

https://hub.docker.com/r/eklavyasaxena/final_model_development/

## 10.5 Git

Implementation and documentation of the project can be accessed via:

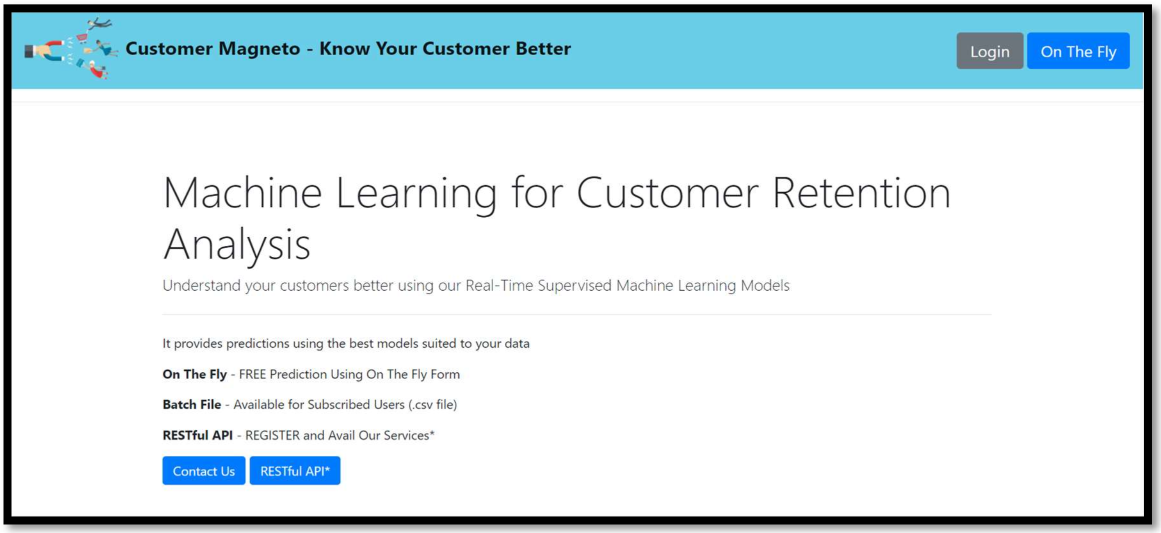
https://github.com/eklavyasaxena/FinalProject_ADS_Team4.git

# 11. Web Application (http://159.89.40.49/)

In computing, a web application is a client-server program which the client (including the user interface and client-side logic) runs in a web browser.

We have used Flask platform to create a web application for user-server interface.

Flask is a micro web framework written in python and based on the Werkzeug toolkit and Jinja2 template engine. Flask is called a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.
Below image shows the working of our application



Customer Magneto - Know Your Customer Better

Machine Learning for Customer Retention Analysis

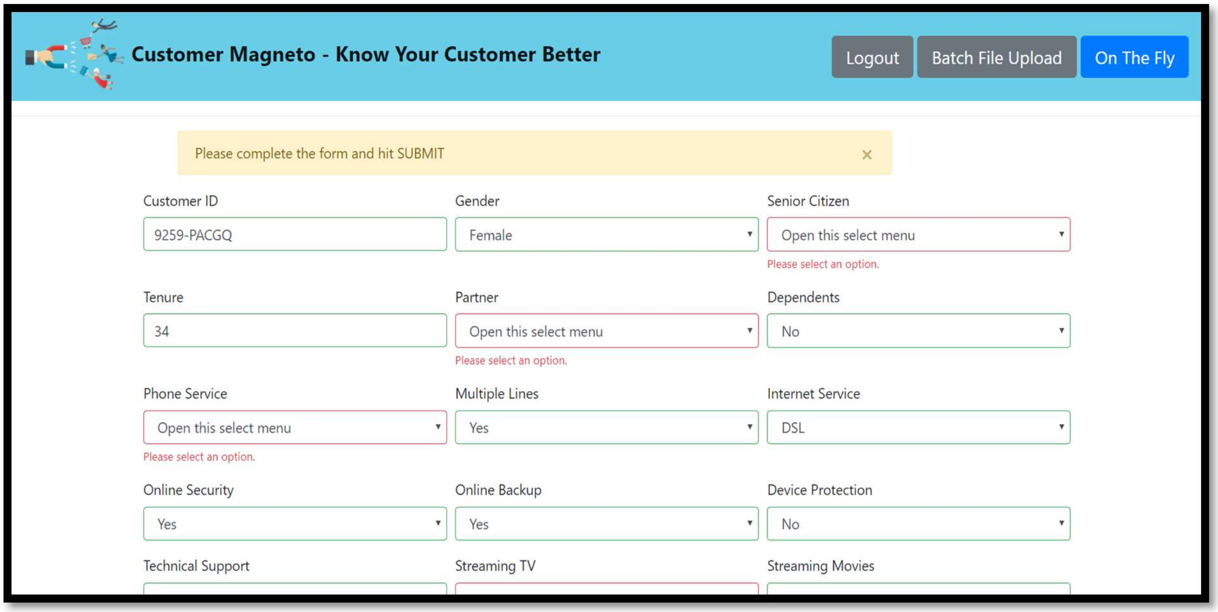
Understand your customers better using our Real-Time Supervised Machine Learning Models

It provides predictions using the best models suited to your data

**On The Fly** - FREE Prediction Using On The Fly Form

**Batch File** - Available for Subscribed Users (.csv file)

**RESTful API** - REGISTER and Avail Our Services*



Please complete the form and hit SUBMIT ×

Customer ID: 9259-PACGQ

Gender: Female

Senior Citizen: Open this select menu
Please select an option.

Tenure: 34

Partner: Open this select menu
Please select an option.

Dependents: No

Phone Service: Open this select menu
Please select an option.

Multiple Lines: Yes

Internet Service: DSL

Online Security: Yes

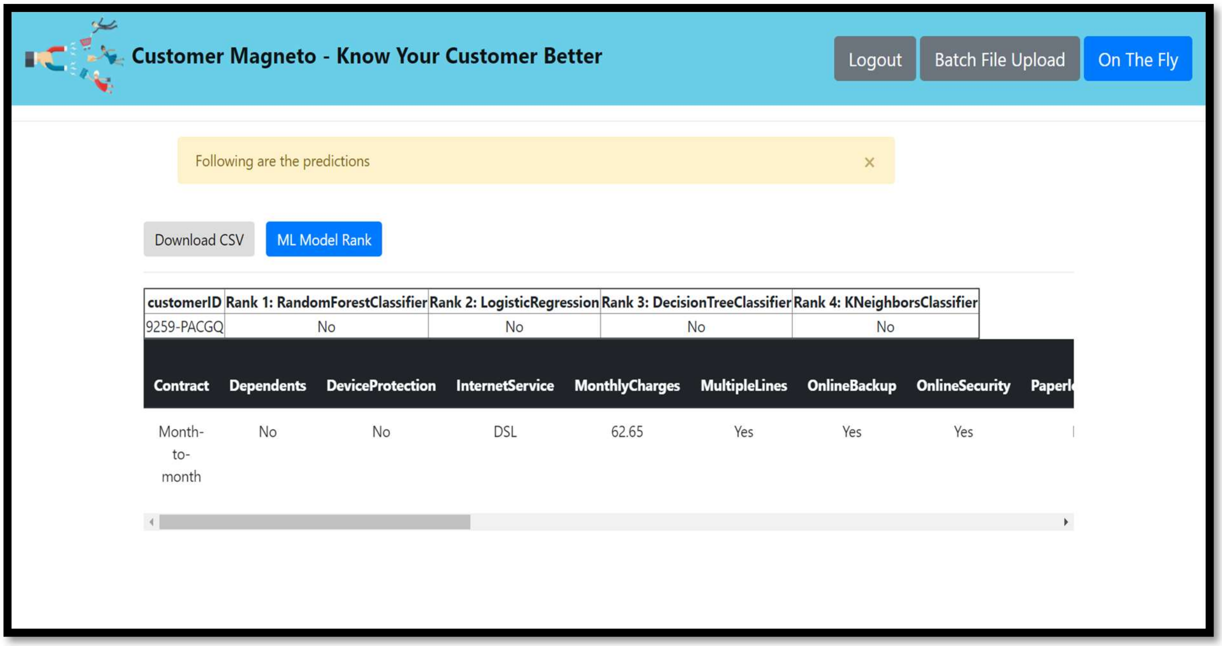
Online Backup: Yes

Device Protection: No

Technical Support

Streaming TV

Streaming Movies



Following are the predictions ×

| customerID | Rank 1: RandomForestClassifier | Rank 2: LogisticRegression | Rank 3: DecisionTreeClassifier | Rank 4: KNeighborsClassifier |
|---|---|---|---|---|
| 9259-PACGQ | No | No | No | No |

| Contract | Dependents | DeviceProtection | InternetService | MonthlyCharges | MultipleLines | OnlineBackup | OnlineSecurity | Paperl |
|---|---|---|---|---|---|---|---|---|
| Month-to-month | No | No | DSL | 62.65 | Yes | Yes | Yes | |

This Application mainly aims for

- Business aiming to prevent customer churn
- Evaluating revenue loss due to churn
- Services in a business that requires quality improvement
- Services that are most influential to customer churn in context of cost
- Business solutions to minimize customer churn in terms of longer contracts and quality service

# 12. Conclusion

In this project a customer churn analysis was presented in a Telecommunication sector. The analysis focused on churn prediction based on various predictive models using supervised approach. The different models predicted the actual churners relatively well. Models did work almost as the random probabilities. There is no difference between the models input which indicates nature of the churning customer profile. Accuracy measure formulate one standard model that could be used as the predictive model in the future. The findings of this study indicate that, in case of logistic regression model, the user should update the model to be able to produce predictions with high accuracy.

All these results together give us an important tool when the company wants to decide which clients to focus their resources to try to avoid them leaving so you can leverage your resources in the best way possible.

## Solutions Proposed

- By using best model, prioritize the concerns of churning customers first.
- Leverage the time to improve quality of services, of the high cost services like Fibre optic.
- Collect customer feedback and act it on immediately to prevent customer churn.

## Future Work

- Customer profile should be included in the data for a company's perspective whether the churning customer are worth retaining or not.
- A probabilistic model can be introduced to list the clients with highest chance of leaving.

# References

https://www.ibm.com/

https://towardsdatascience.com/predict-customer-churn-with-r-9e62357d47b4

http://www.vtt.fi/inf/julkaisut/muut/2006/customer_churn_case_study.pdf

https://github.com/Atreya22/luigi_rosmann_sales

https://pythontips.com/2013/08/02/what-is-pickle-in-python/