Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Informatics

Diploma Thesis

Analysis telecom data using Business Intelligence

VIJAYA KRISHNA YADLAPALLI

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

FacultyofEconomicsandManagement

DIPLOMATHESISASSIGNMENT

Vijaya Krishna Yadlapalli

Informatic

Thesis

AnalysistelecomdatausingBusinessIntelligence

Objectives of thesis

Customer attrition, also known as customer churn, customer turnover, or customer defection, is the loss of clients or customers. The objective of this thesis is to analyse the telecom data and identify churn customers, that is, customers most likely to cancel subscription to a fictitious telecom company.

The partial objectives of the thesis are to analyse churn Customers based on

- gender distribution,
- age distribution,
- ratio of churn and non-churn.

Methodology

The dataset for this thesis was obtained from the public repository. The dataset relates features of account and usage for churn and non-churn clients. In the context of this thesis, this is a problem of supervised classification will be used for the development of predictive models and evaluation of accuracy and performance. It seeks to find the most appropriate model for the business. The students going to analysis the problem using BI tool: Python and Power-BI and will visualize the data using dashboards, charts and bar graphs.

The proposed extent of the thesis

60 – 80 pages

Keywords

business intelligence, customer analysis, data analysis, reporting, python

Recommended information sources

ECKERSON, Wayne W. Performance dashboards: measuring, monitoring, and managing your business. John Wiley & Sons, 2010.

CHAUDHURI, Surajit; DAYAL, Umeshwar; NARASAYYA, Vivek. An overview of business intelligence technology. Communications of the ACM, 2011, 54.8: 88-98.

KIMBALL, Ralph; ROSS, Margy. The data warehouse toolkit: the complete guide to dimensional modeling. John Wiley & Sons, 2011.

TYRYCHTR, J. – VASILENKO, A. Business Intelligence in Agribusiness - Fundamental Concepts and Research. Brno: KONVOJ, spol. s r. o. , 2015, 100s. ISBN 978-80-7302-170-2.

Expected date of thesis defence

The Diploma Thesis Supervisor

doc.Ing. Jan Tyrychtr, Ph.D.

Supervising

department

Department of Information Engineering

Electronic approval: 23. 11. 2021   Electronic approval: 25. 11. 2021

Ing. Mar n Pelikán, Ph.D.   **Ing. Mar n Pelikán, Ph.D.**

Head of department   Dean

Prague on 01. 04. 2022

Declaration

I confirm that the work on my diploma thesis is called "Analysis telecom data using Business Intelligence" and I utilized the resources mentioned at end of the thesis. As the writer of the diploma thesis, I announce that the thesis doesn't break the copyrights of any individual.

In Prague on 1.04.2022   _____

Acknowledgment

 I would like to thank my supervisor doc. Ing. Jan Tyrychtr, Ph.D. for his recommendations and help during my work on this thesis in the course of tough times.

Analysis Telecom data using Business Intelligence

Abstract

Telecom companies have a competitive market as gaining new customers is very difficult compared to retaining an existing customers. As retaining advertisements can be used to prevent them, but we need some efficient solution to stop the churn. That's why we need churn perdition and with this prediction, we can get an idea what new changes they need to do.

In my thesis, I focused on IBM the telecom dataset. The data was containing the numerical and categorical features. The dataset has one column which is showing the customer is churning or not. The dataset is containing customer basic information, account information and services they are using.

To analyse the data, I used python libraries: pandas, Numpy, and matplotlib. I applied data pre-processing, transformation, and cleaning on data. After analysing the data, I made hypotheses that tenure, monthlycharges, totalcharges and contracts are the main reason for churn.

Then, I applied the machine learning algorithms like SVM, Random Forest, Logistic regression, and KNN and analysed the important features based on the model. Random Forest has supported my hypotheses, but SVM and Logistic Regression model don't supported my hypotheses.

Keywords: business intelligence, customer analysis, data analysis, reporting, python

Abstraktní

Telekomunikační společnosti mají konkurenční trh, protože získat nové zákazníky je velmi obtížné ve srovnání s udržením stávajících zákazníků. Protože zadržování reklam lze použít k tomu, abychom jim zabránili, ale potřebujeme nějaké účinné řešení, abychom zastavili tok. To je důvod, proč potřebujeme churn perdition a díky této předpovědi můžeme získat představu, jaké nové změny potřebují udělat.

Ve své diplomové práci jsem se zaměřil na telekomunikační datovou sadu IBM. Data obsahovala číselné a kategoriální znaky. Datová sada má jeden sloupec, který ukazuje, že zákazník míchá nebo ne. Dataset obsahuje základní informace o zákaznících, informace o účtu a služby, které využívá.

K analýze dat jsem použil knihovny python: pandy, Numpy a matplotlib. já aplikované předzpracování dat, transformace a čištění dat. Po analýze dat jsem vytvořil hypotézy, že hlavním důvodem odchodu jsou držba, měsíční poplatky, celkové poplatky a smlouvy.

Poté jsem aplikoval algoritmy strojového učení, jako je SVM, Random Forest, Logistická regrese a KNN, a analyzoval důležité funkce založené na modelu. Random Forest podporuje mé hypotézy, ale SVM a model logistické regrese mé hypotézy nepodporují.

Klíčová slova: business intelligence, zákaznická analýza, analýza dat, reporting, python

Abbreviation:

OLAP: Online analytical processing

OLTP: online transactional processing

BI: Business Intelligence

ETL: Extract, Transform, Load

LR: Logistic Regression

KNN: K Nearest Neighbours

ANN: Artificial Neural Network

SVM: Support Vector Machine

TP: True Positive

TN: True Negative

FP: False Positive

FN: False Negative

Contents

List of Figures:

List of Images:

List of Tables:

1. Introduction

Customers are always important for every organization to grow better in terms of revenue and size how big it is, especially how the company is adding new clients and losing them. Mainly, customer churn plays a crucial role as we need to analyse why customers are opting out as it is very easy to convince existing customers than new customers to join especially in the telecom sector where it is very hard to convince new customers due to competition available in the market and their services or offerings. Telecoms companies are customer-centric as they always tend to make changes to attract new clients and to expand their business innovations, but it shouldn't impact staying customers. The technology that needs to be moving forward as it always requires advancements to solve new kinds of problems those are facing by organizations. As the telecom sector is thriving due to the latest technologies such as 5G and upcoming 6G, every company is eager to try these technologies and, in some countries, the 5G rollout started much earlier. So, due to these technological advancements the sector always looks at the customer share like how many customers are adding in day by day or monthly basis just to check if they are losing them. It always depends on customer's tastes in each organization and companies need to analyse the feedback and data they are receiving from customers to improve services that suit user tastes in terms of pricing, services, and improvements based on feedback it all depends on using business intelligence (BI) tools it is easier than ever to perform analysis on customers earlier data. Data analysis can help to distinguish the customers in order change to offerings based on their expectations with the help of power tools of BI using their organizations can make decisions to improve services that suit customer interests. Moreover, we can forecast with help of past data, how business is going to be in the future.

1.1. Motivation and Goals

Telcom organizations need to draw in new clients additionally to avoid a contract termination or extension to extend their revenues and business. viewing to churn, multiple factors are responsible for those customers terminating their contracts, for example, a decent price offer, a more interesting & useful package, a poor/costly service experience are often reasons for churn.

To predict customer churn, churn analysis is extremely useful and also provides the

responsible factors. With the churn matric, we will get the kids of the customer is using services and the way many has cancelled the services if the company has 10 million customers and an 800k contract is expiring by the last of the month, then the monthly churn rate is 8%.

Telecom companies apply machine learning models to predict churn on individual customer bases and take countermeasures like discounts, special offers, or other satisfaction to retain their customers. A customer churn analysis could be a specific classification problem within the field of supervised learning.

Many requirements are there in a churn prediction. Below are some requirements.

[Balle et al., 2013]

- Performance – the performance of the model with different data. This is required for the business to take the correct steps at the correct time.

- Flexibility – the model needs to be flexible and needs to predict with a good forecast % with different data.

- Scalability - the model needs to deal very perfectly with a high amount of data.

- Targeting: the model can predict concrete data which is responsible for churn.

- Precision and Recall – the model needs a high level of recall means as much as possible to define churn and a medium-high level of precision mean fewer wrong prediction.

1.2. Dissertation Structure

Apart from this chapter, this thesis contains more chapters.

In chapter 2, a review of the existing literature on the topic is conducted and some related work is presented. Some methods are explained, and Churn type and about BI and ETL process.

Chapter 3 mentioned tools and libraries used to analyse churn prediction. I explained what the use of the tool is and where I used it in this project.

Chapter 4 aims to describe the source and columns of information about the data. The data contained information about demographic info, account info, activate different services, and left customer or not information.

Chapter 5 aims to describe all the analyses performed on data. Like: pre-processing, cleaning the data, analysis of the data, transformation steps performed on the data, and lastly predicting the churn & important features based on different models and matrices. I checked the model performance and accuracy.

This chapter is containing the below pipelines:

Figure 1 Process flow of the project

Problem Description

Reading the data

Analysis the data

Train-Test split

Model Evolution

Model Selection, Prediction, Assessment

If the model is not good, then Train-test it again
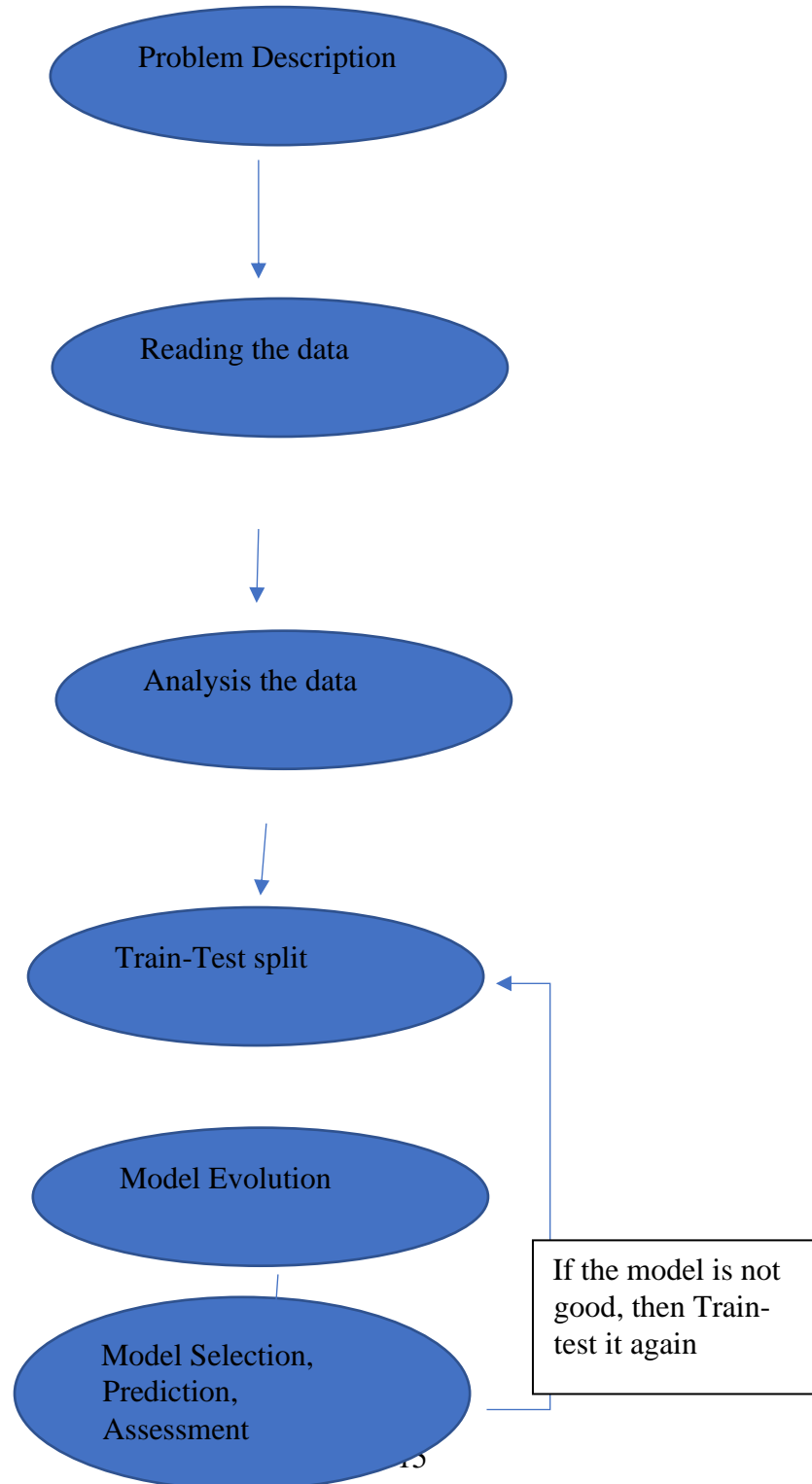
2. Literature Review

In this chapter, a review of the bibliographic content found is conducted. the most focus is that the approaches and techniques applied to the churn prediction model in telecom businesses.

2.1.   Churn Customer

When a user or customer who was using services/subscriptions, stopped business and relationship with the company, is known as a Churn Customer.

Types of Customer Churn [Retana et al., 2016, Olaleke et al., 2014]

Contractual Churn: When a customer terminated services in between a running contract is known as a Contractual Churn. like: Cable TV etc.

Voluntary Churn: When a user willingly cancels a service like a Mobile connection.

Non-Contractual Churn: When a customer isn't under a contract for a service and decides to cancel the service, for instance, Consumer Loyalty in retail stores.

Involuntary Churn: When a churn occurs with no request from the customer, for example, MasterCard expiration.
Reasons for Voluntary Churn
Lack of usage
Poor service
better price

2.2. Data analysis

Data is raw, unorganized which might be anything because it requires further processing so as make it the information it should have meant. The analysis will work like processing to alter into meaningful insights as the organization is required to create business decisions supporting the issues that have been addressed or to introduce new change organization-wide. thanks to this thesis can automate this process using analysis tools, s, especially in the telecom sector where we'd like to cope with many data and even data, is purchased from third-party companies to use for analytics. Using data analytics organizations can predict the problems and bottlenecks supported by past or current data because it enables them to avoid them as early as possible.

2.3. Business intelligence

Business intelligence has been used first time in 1865 by author Richard Millar Devens. In the 1990s it was the popular solution, and many people were using who was an expert in BI. But now BI tools it became available for non-experts to work with the data. [https://www.ibm.com/topics/business-intelligence (accessed 30 March 2022).]

BI especially is being used for forecasting past data to see the trends as BI is playing an important role in the analysis of unstructured and structured data from multiple resources and can be combined to make analysis. that can be done using history and current data to make critical decisions, and insights to improve the organization how it operates in the future moreover, same data can be shared across platforms. The data starts from a data warehouse where all data will be stored from multiple sources to process it using ETL. ETL tools can make the data meaningful as it will be further analysed with help of OLAP technology that works with multidimensional queries as it will gather large databases that can be used for complex analysis. In the case of OLTP as it can handle large quantities of transactions to maximize the accuracy and of the data.

Image 1 Business Intelligence

Source: towardsdatascience.com

2.3.1 How Business intelligence works

The traditional methods will take a lot of your time and human effort to process and easily the info but with BI it can help to scale back time to create business decisions with help BI tools because it can process the info much faster way [Yen and Wang, 2006]. it'll produce results or insights with help of business analytics and supporting tools and also the data are often pulled from the data warehouse because it was migrated from multiple sources using ETL (extract, transform, load) tools to tug data from multiple sources into data warehouses to process it further.

Generally using intelligence tools we can query the info from the warehouse and later can visualize with help of BI tools supported business interests and even it is

often manipulated further supported kind of data. there are new technologies to ingest the data but still, many organizations are using data warehoused traditionally by most organizations.

Image 2 Process of BI tool
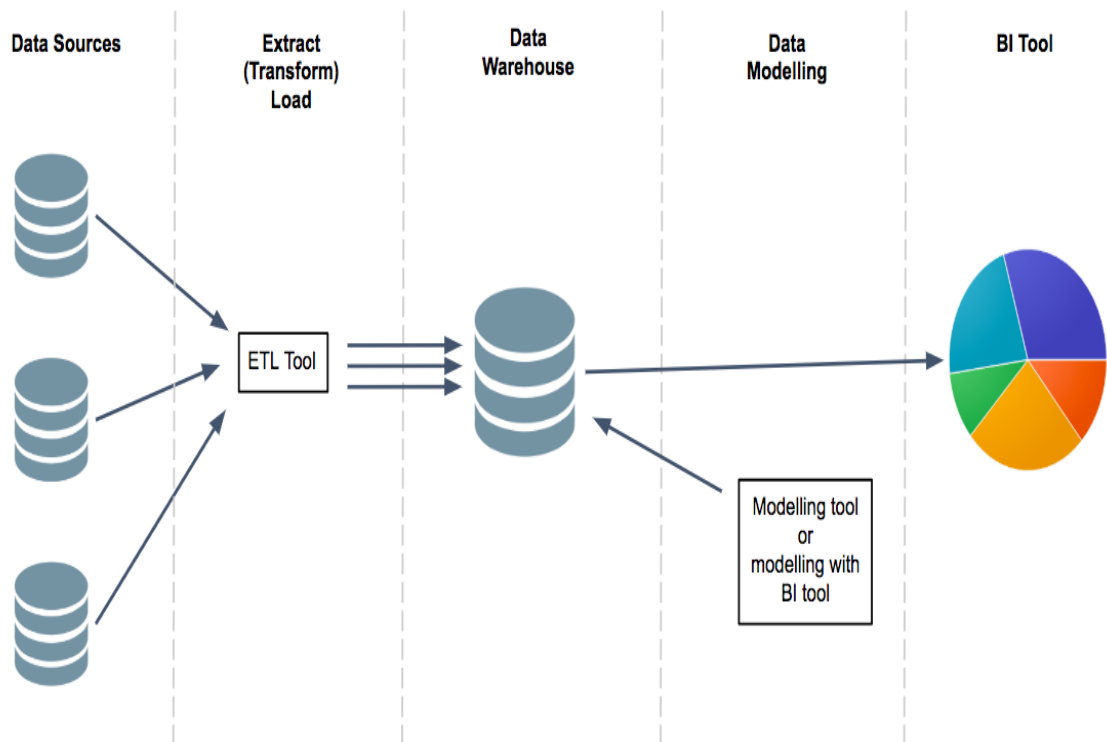
Source: medium.com

ETL

ETL stands for extract, transform, and cargo. it's generally will extract the
information from different sources so we will perform a transformation to create it
usable and can load into data warehouses, where we can analyse the information with help
of BI tools and detect the business issues.

Data warehouse

The data warehouse is a data management system, and the main component for
BI. it's used to store data from multiple sources, and we can query data for reporting and
analytics. Data warehouses can store large amounts of information. the info warehouse
always depends on how well it's designed to process data to form better decisions
efficiently. Normally we are using on-premises data warehoused thanks to the character of
security and data sovereignty, but the most problem with them is scaling when required
and it's very complex to keep up. but nowadays slowly data is storing to cloud because
it will be accessed easily and elastic furthermore.

Power BI

Power BI is a business intelligence tool powered by Microsoft because it will
be accustomed load data from many alternative sources like excel, data warehouses,
databases etc. and it can provide a visualization with insights to form business decisions.
Using Power BI, you'll be able to help to form reports and dashboards which may
be customized to support the necessities. The advantage of using Power BI is that we will}
be able to fetch style of data from multiple places and it is customized
completely supported our needs as Microsoft keeps adding new features.

The Power BI is a security measure for keeping sensitive information protected
and its control over data travel. it's many options to run on multiple platforms like Android,
iOS and Windows-powered devices. Moreover, with help of built-in machine learning can
analyse and find trends to create forecasts and predictions.

2.3.2 Advantages of BI

1. Productivity: Using BI tools can save such a lot of time for an organization as the employees can specialise in productivity rather than managing data manually even, I am automated with help of tools to process incoming data to form business decisions.

2. Customer experience: Customers are an integral part of every company because they can improve their experience with an organization can identify issues with help of tools by analysing past data what exactly customers predict. this could help customers' visits because the organization can address those issues way earlier.

3. Intuitive dashboards: It can help the user with help of tools as dashboards are easier to know reports without having technical knowledge and even it is often customized further as there are plenty of obtainable dashboards depending on the way it is required.

4. Business decisions: Having faster reporting capabilities provides required business decisions because it helps the managers to forecast the sales up to this point so as have meetings with clients if we don't do this the information is often outdated by meetings because it takes a lot of your time to process with traditional way.

2.3.3 Disadvantages of BI

1. Cost: It is often expensive for a few organizations as they have to set up all tools to figure out data, especially on-premises, but the cloud can solve this for tiny organizations.

2. Implementation: it's tough for implementing because it requires knowledge of all the tools and data, as an organization must train employees to urge hands-on implementation.

3. Time: As BI can make it easier to run processes but it takes a lot of your time to implement a data warehouse, but it always depends on the dimensions of the warehouse and organization as resources must be pooled.

2.4. Predictive Modelling

In data processing, predictive modelling is the process of making a model predict an outcome. To do so, the past must be analysed, and, with this information, the model must be constructed and validated. to make this model, there must be predictors, which are variable factors that are expected to influence future behaviour. as an example, when predicting churn these factors may be the number of calls to the operator assistance or the quantity of dropped calls regarding a specific client. If the result is qualitative, it's called classification and if the result is quantitative, it's named regression. Descriptive modelling or clustering is the assignment of observations into clusters so observations within the same cluster are similar. Finally, association rules can find interesting associations amongst observations. the subsequent sections will describe these techniques well and identify specific types for everyone.

2.4.1 Regression

Regression could be a data processing function that predicts a numeric value. It will be accustomed to modelling the relationship between one or more independent variables and dependent variables [Zaki and Meira, 2013].
For example, We can use it to predict children's height given their age and weight. Target values must be known before using a regression model on a dataset. Using the identical example as above: to predict children's height, there must exist data during a period of your time regarding their age, height, weight, and others. Height is then considered the target variable, and the remaining attributes would be the source.
During the model training process, a regression model predicts the target variable. This value is assessed as a function of the predictors. The multiple iterations of this step through all the initial data compose a model. Later, this model may be applied to a special dataset with unknown target values.

$$y_i = F(x_i, p) + e_i$$

Equation 1

22

Below is the function to calculate the continuous target variable Y_i. Here x_i is the predictors, p is the parameter and e_i is the error.

2.4.2 Classification

Classification is a processing task of predicting the results of a target variable by building a model supported by predictors, predictors or attributes that will be categorical or numerical attributes [Zaki and Wong, 2003]. The goal of classification is to predict the maximum amount as an accurate result supported by inputs. to predict this, the algorithm processes a training set containing several attributes and their target outcome (prediction attribute). It then tries to look at relationships between these attributes that cause predicting the result. Next, a replacement set must be taken into consideration: the prediction set. this is often composed of the latest data set with similar attributes because the training set needs for the prediction target. The model produces a prediction supported by this new data, and an accuracy % defines how "good" the algorithm is. A higher % means a good model.

Classification algorithms are often described into four main groups:

1. Frequency Table
• ZeroR
• OneR
• Naive Bayesian
• Decision Tree

2. Covariance Matrix
• Linear Discriminant Analysis
• Logistic Regression (LR)

3. Similarity Functions
• K Nearest Neighbours (KNN)

4. Others

• Artificial Neural Network (ANN)

• Support Vector Machine (SVM)

Table 1 Confusion Matrix

		Predicted	
		Non-Churn	Churn
Actual	Positive	TP	FN
	Negative	FP	TN

Table 1 is showing the confusion matrix, which contains the data about the actual value and predicted value calculated by the classification model. to test the developed model this method, must apply [Brownlee J (2014)].

True positives (TP): The target attribute value is yes that the client did not leave the services, and he will stay.

True negatives (TN): The target attribute value is no, and that they left the services means the user churned.

False positives (FP): The target attribute value is yes, but the client left the services. (called "Type I error.")
False negatives (FN): The target attribute value is no, but the user stayed within the services. (called "Type II error.")

Below factors, we can calculate by a confusion matrix.

Accuracy: Accuracy can be calculated by a total number of true positive and negative values divided by all factors. Below is the formula of accuracy [Sharma P (2019)]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Recall: Recall will show how much the model has a good prediction. If the recall value is high, it means perdition is good. Below is the formula of recall [Sharma P (2019)]:

$$Recall = \frac{TP}{TP + FN} \text{ or } \frac{True\ Positive}{Actual\ Results}$$

Precision: Precision can be calculated by a total number of true positives divided by both positive factors. Below is the formula of precision [Sharma P (2019)]:

$$Precision = \frac{TP}{TP + FP} \text{ or } \frac{True\ Positive}{Predictive\ Results}$$

F-score: F-score can be calculated by recall and precision. Below is the formula of the F-score [Sharma P (2019)]:

$$F\text{-}score = \frac{2 * Recall * Precision}{Recall + Precision}$$

Specificity: Specificity is known as Churn Rate also. Below is the formula for specificity [Sharma P (2019)]:

$$Specificity = \frac{TN}{TN + FP}$$

Equation 6

Image 3 Confusion Matrix explanation

Source: Sharma P (2019)

Image 4 Confusion Matrix result

PREDICTIVE VALUES

POSITIVE (1)   NEGATIVE (0)

	POSITIVE (1)	NEGATIVE (0)	
POSITIVE (1)	TP = 3	FN = 1	4
NEGATIVE (0)	FP = 2	TN = 4	6
	5	5	

ACTUAL VALUES

RECALL

PRECISION

Source: Sharma P (2019)
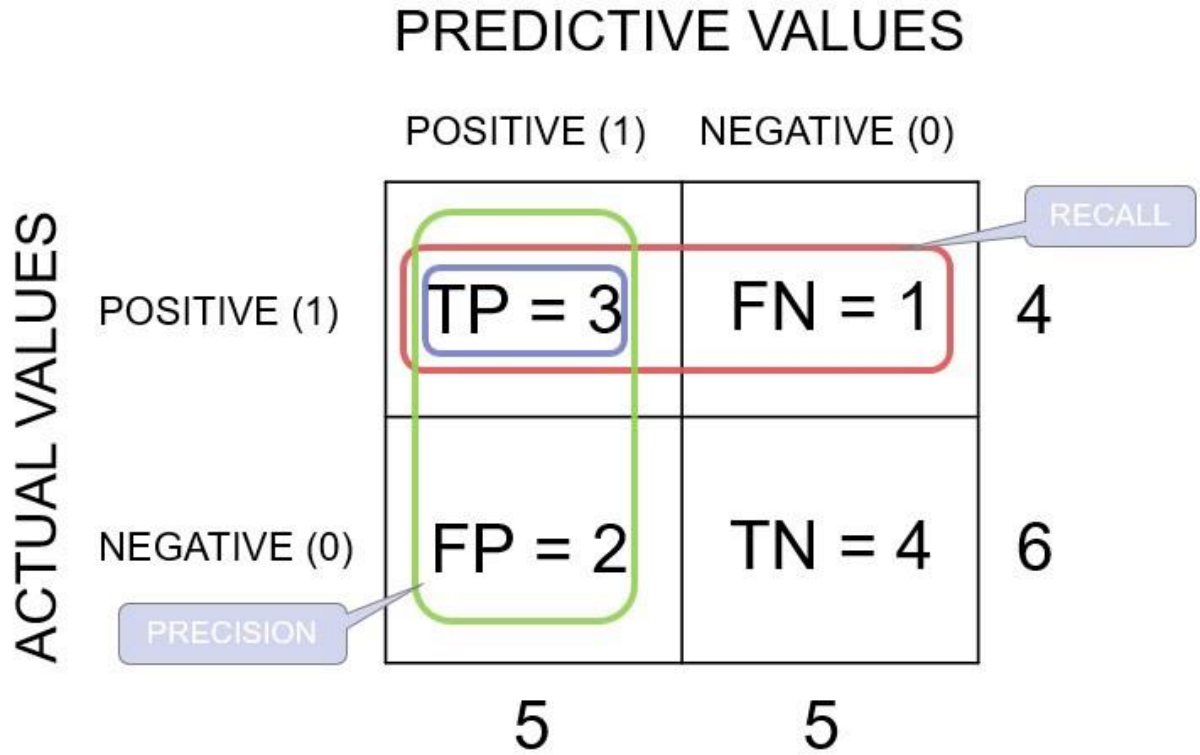
Table 2 Classification values calculated from confusion Matrix result

	Formula	Result			
		TP=3	TN=4	FP=2	FN=1
Accuracy	(TP+TN)/(TP+TN+FP+FN)	(3+4) / (3+4+2+1) = 0.70			
Recall	TP/(TP+FN)	3 / (3+1) = 0.75			
Precision	TP/(TP+FP)	3 / (3+2) = 0.60			
F-score	(2*Recall*Precision)/(Recall+Precision)	(2*0.75*0.60) / (0.75+0.60) = 0.67			
Specificity	TN/(TN+FP)	4 / (4+2) = 0.67			

3. Tools and libraries used

This chapter described tools and libraries which is used to predict the churn data. businesses.

3.1. Python:

Python is a powerful scripting language and is extremely useful for solving statistical problems involving machine learning algorithms. Its various utility functions help in pre-processing. Processing is fast, and it's supported on most platforms. Integration with C++ and other image libraries is incredibly easy, and with in-built functions and libraries store and manipulate data of every kind. It provides the pandas and NumPy framework which helps in the manipulation of knowledge as per our needs. a decent feature set will be created using the NumPy arrays which may have n-dimensional data.

3.2. Anaconda:

Anaconda is an open-source tool. it's the simplest way to add Python /R. it's available for all the software packages like Linux and Windows. Its many features

 • Easy to download Python / R package.
• Manage environments with Conda and manage libraries.
• Data performance with Dask, NumPy, pandas, and Numba and analysis with scalability.
• Show results with Matplotlib, Datashader, Bokeh, and Holoviews.
• Scikit-learn, TensorFlow, and Theano are useful for developing and training machine learning and deep learning algorithm.

3.3. Jupyter Notebook:

Jupyter Notebook is an integrated development environment. we will integrate with python very easily, and every one of the libraries will be employed in our implementation. it's interactive, although some complex computations require time to finish. Plots and pictures are displayed instantly. Most of the libraries like Dlib, OpenCV, Scikit-learn are used easily. I used this to perform my churn prediction.

3.4. NumPy:

It is one of the highest libraries to support analysis, especially it's used for numerical data. the basic package for computing numbers with Python is this. It contains many things:

• a strong N-dimensional array object.
• For broadcasting functions, we can use.
• Tools available for integrating C/C++ and FORTRAN language.
• Useful linear in Algebra, Fourier transforms, and random number problems.

NumPy also can be used with an N-dimensional array. this can be quickly integrated with an outsized style of database.

3.5. Pandas:

Pandas is a python library that will be used for data analysis, and it's one of the highest libraries because it has active communities for contribution. I used this library to analyse the data, check the info, describe the data and groping the records based on the requirement.

3.6. Matplotlib:

Matplotlib is useful to visualize the data in various formats like a bar, pie, bar chart, and so on. I used this library to analyse the data, check the distribution of data, see the feature importance, and so on.

3.7.   Scikit-learn:

Scikit-learn is used for machine learning. The use of this library is very high. In every part, we can use it in classification, perdition, dimension reduction, and so on. For Churn analysis I used this for LR, KNN, SVM, to declare the training and testing set and so on.

4. Dataset

The models that predict churn are based on knowledge regarding the company's clients, monthly and total charges, contract size, and extra services they are using. The information is stored in a CSV file and the CSV data is called a dataset.

This chapter aims to describe the source and columns of information about the data. The data contained information about demographic info, account info, activate different services, and left customer or not information.

4.1. Data

The data set I taken from Kaggle and its IBM sample data [BlastChar (n.d.)]. This is the link (https://www.kaggle.com/datasets/blastchar/telco-customer-churn). The dataset is in CSV and I used the pandas' library to read this data.

4.2. Data Description

The below column is representing the unique id of the customer:

- CustomerId

The below columns are representing the demographic info about customers. Like gender, age, partners information, and dependent information if they have:

- gender
- senior citizen
- Partner
- Dependents

The below columns are representing the account information for a customer. Like how long they are using services, how they use to do payments, and how much they will pay monthly and total:

- Tenure
- Contract
- payment method
- PaperlessBilling,
- MonthlyCharges
- Total charges

The below columns are representing the services used by customers.

- PhoneService
- MultipleLines
- InternetService
- OnlineSecurity
- OnlineBackup
- DeviceProtection
- TechSupport
- StreamingTV
- StreamingMovies

The below column is representing the information who left the services last month:

- Churn

5. Methodology

This chapter aims to describe all the analyses performed on data. Like: pre-processing, cleaning the data, analysis of the data, transformation steps performed on the data, and lastly predicting the churn & important features based on different models and matrices. I checked the model performance and accuracy.

To predict the churn, I have used python and machine learning libraries. And 7 steps are used to analysis and prediction.

- Problem Description
- Reading the data
- Analysis the data
- Visualize the data
- Analysis the Feature
- Train-Test split
- Model Evolution
- Model Selection, Prediction and Assessment

5.1. Problem Description

This dataset contains the information of customers of a Telecom Company.

Based on this dataset, we need to predict whether will customer churns the plan or not. i.e: renew the plan or not.

From the introduction, the main goal is to predict if an individual customer will stay or leave, In the other words, the customer will churn or not. To know this answer I used machine learning models and trained the model with an 80 – 20 ratio and predict customers will churn or not.

To predict the ratio of Churn and non-churn the main question thing to identify the important features/columns in the dataset. I have used a matrix to know the important features then check the model accuracy and performance.

5.2.   Reading the data

For reading the data, I performed the below steps:

- Download the data from the URL: https://www.kaggle.com/blastchar/telco-customer-churn to my local system. Data was in CSV format.

- Setup the anaconda with a new environment variable and used the Jupyter notebook.

- Installed the required library. Like NumPy, pandas, scikit-learn
- Import the required library.

- Read CSV file into a data frame using pandas.

Installed the required library:

```
# installing the required library
!conda install --yes scikit-learn
```

```
Collecting package metadata (current_repodata.json): ...working... done
Solving environment: ...working... done

## Package Plan ##

  environment location: C:\Users\SweetyWadhwa\anaconda3\envs\DA

  added / updated specs:
    - scikit-learn


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    scikit-learn-1.0.2         |   py39hf11a4ad_1        5.1 MB
    ------------------------------------------------------------
                                           Total:        5.1 MB

The following NEW packages will be INSTALLED:

  joblib              pkgs/main/noarch::joblib-1.1.0-pyhd3eb1b0_0
  scikit-learn        pkgs/main/win-64::scikit-learn-1.0.2-py39hf11a4ad_1
  threadpoolctl       pkgs/main/noarch::threadpoolctl-2.2.0-pyh0d69192_0



Downloading and Extracting Packages

scikit-learn-1.0.2   | 5.1 MB    |                   |   0%
scikit-learn-1.0.2   | 5.1 MB    |                   |   0%
scikit-learn-1.0.2   | 5.1 MB    | 6                 |   7%
scikit-learn-1.0.2   | 5.1 MB    | #4                |  15%
scikit-learn-1.0.2   | 5.1 MB    | ##1               |  22%
scikit-learn-1.0.2   | 5.1 MB    | ######5           |  65%
scikit-learn-1.0.2   | 5.1 MB    | ########3         |  84%
scikit-learn-1.0.2   | 5.1 MB    | #########9        |  99%
scikit-learn-1.0.2   | 5.1 MB    | ########## | 100%
Preparing transaction: ...working... done
Verifying transaction: ...working... done
Executing transaction: ...working...

    Windows 64-bit packages of scikit-learn can be accelerated using scikit-learn-intelex.
    More details are available here: https://intel.github.io/scikit-learn-intelex

    For example:

        $ conda install scikit-learn-intelex
        $ python -m sklearnex my_application.py

done
```

Import the required library:

```
# importing required library
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn  as sns
from sklearn.model_selection import train_test_split
```

Read CSV file into a data frame using pandas:

```
# Read CSV file into DataFrame df
df = pd.read_csv('data/Telco-Customer-Churn.csv')
```

5.3.   Analysis the data

After reading the data, performed many steps to analysis the data. The main goal for this
step understanding the datatype of attributes, pre-processing the data, changing the
datatype, visualizing the data, and cleaning the data.

5.1.1   Understanding the data

The first step to understanding the data means to know the information about attributes
numeric or categorical or continuous. I used the panda library to understand the structure
of attributes.

```
# display the first 5 records
df.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	Tech
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

The above result is showing top 5 rows of data with all columns.

```
# shape of data
df.shape
```
]: (7043, 21)

With the above result, we can see the data is containing 7043 rows and 21 columns.

```
▶ # Information about data like name, count of not null data, datatype
  df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

- With the data above result, we can see the data is in the panda's data frame.

- The index value is starting from 0 and ends at 7042.

- The total columns are 21. Below is the columns Explanation.

Data have only 3 Numerical attributes and those are:

➢ Tenure

➢ MonthlyCharges

➢ SeniorCitizen

- TotalCharges is an object but its stores the float value so I will change the datatype of this.

- All Other Variables are Categorical.

- The target Variable is Churn (Whether the customer churned or not (Yes or No)).

```
#Analysis the numberic attributes
df.describe()
```

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

With the above result, we can check the mean, standard derivation, min, maximum value of numeric attributes. I noticed min value of tenure is 0. This means some columns is containing null values.

count - The count is showing the not-empty row count of each column.

mean - The average (mean) value is showing what is the mean or average value of columns.

std - The standard deviation is calculating standard derivation.

min - The min is showing what is the minimum value of the column.

25% - The 25% percentile.

50% - The 50% percentile.

75% - The 75% percentile.

max - The max is showing what is the maximum value of the column.

With min, we can see the minimum MonthlyCharges is 18 and the maximum is 118. Here we can see the MonthlyCharges minimum amount paid by the customer is 18 and the monthly maximum amount is 118.

5.3.2   Visualize the distribution of the dataset

Figure 2 Distribution of dataset.
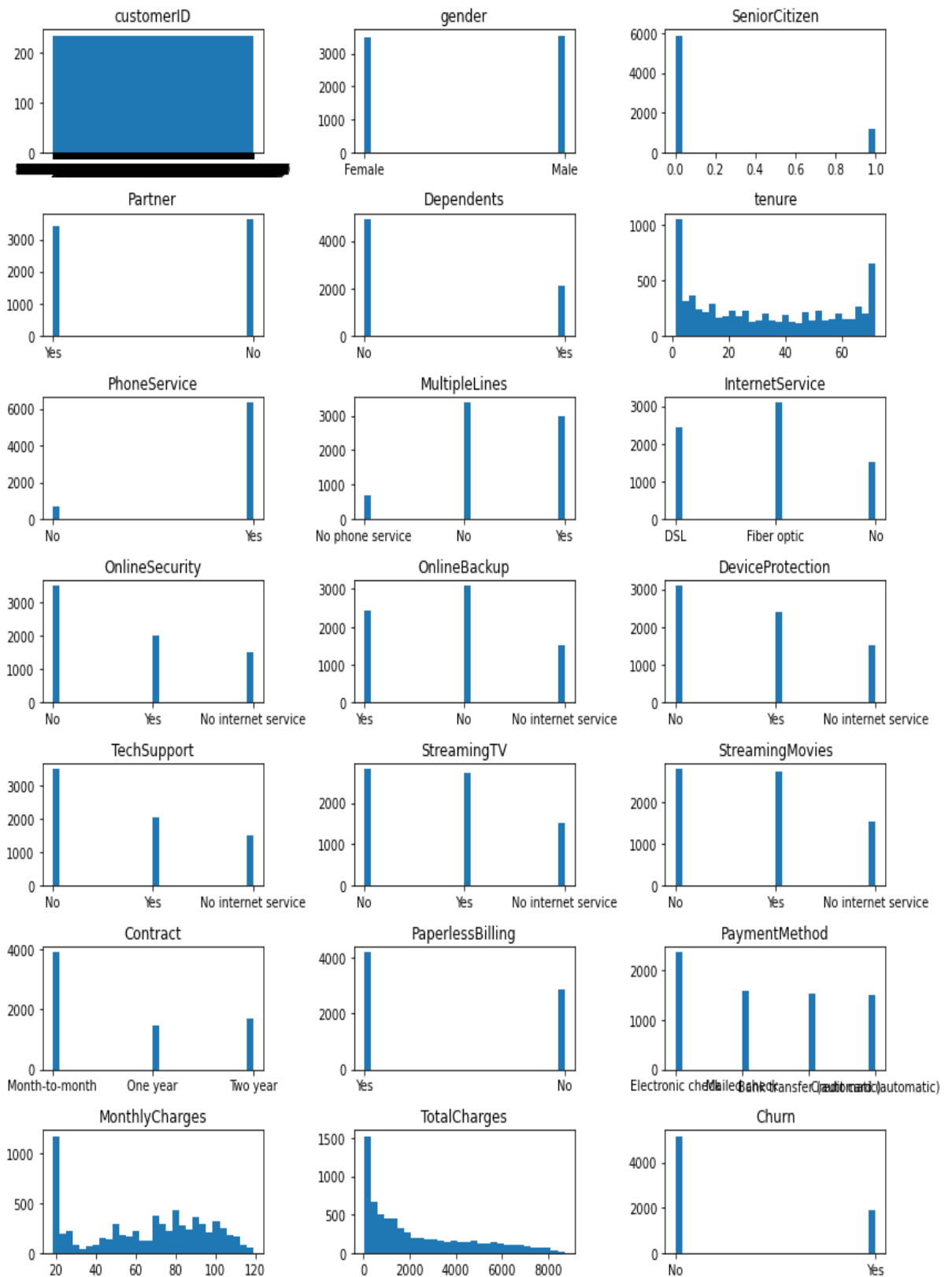
In Figure 2, we can how the data is distributed. We can check if any outlier is there. In my dataset, I can't see any outlier. Below is the description of dataset attributes and data.

Demographic info about customers:

- CustomerId: Unique id of customer.

- Gender: Attribute is containing two values male or female. In Figure 2 I can see for both almost ratio is the same.

- SeniorCitizen: Attribute is containing two values 1(if a customer is a senior citizen) or 0 (not a senior citizen). In Figure 2 I can see more customer is not a senior citizen.

- Partner: Attribute is containing 2 values yes (if they have a partner) or no (in case of no partner). In Figure 2 I can see for almost ratio is the same for partner and without partner.

- Dependents: Attribute is containing two values yes (in case of any dependent) or no (no dependent). In Figure 2 I can see for almost customer is independent.

Account information for a customer:

- Tenure: Attribute is a continuous attribute. It's showing the number of months the customer is using services. In Figure 2, we can see data distribution for tenure is between 0 to 10 and < 60 is more.

- Contract: Attribute is containing 3 values about the contract distribution. Month-to-month, one year, two year and with Figure 2 I can say clearly more customers has a month-to-month contract.

- PaymentMethod: This attribute has 4 values. Electronic check, Mailed check, bank transfer & credit card transfer. It's showing how the customer is used to paying charges and based on Figure 2 I can say clearly more customer is paying with an electronic check.

- PaperlessBilling: This attribute is storing 2 values yes (in case of paperless billing or no (no-paperless billing) and with Figure 2 I can say more customers has paperless billing.

- MonthlyCharges: This attribute is a continuous attribute and stores the monthly paid charges by the customer and with Figure 2 I can see most of the customer is paying >20 monthly charges.

- TotalCharges: This attribute is showing total charges.

Services used by customers.

- PhoneService: This attribute is containing 2 values No (in case a customer is not using phone service) or Yes (if they are using). In Figure 2, I can see most of the customers are using these services.

- MultipleLines: This attribute is containing 3 values No (in case a customer doesn't have multi lines, but they are using phone services) or Yes (if they have) or No phone services (in case they are not using phone services). In Figure 2, I can see most of the customers don't have multiple lines. But in other case only less customer is there who don't have phone lines.

- InternetService: This attribute is showing which kind of internet services they have. It's containing 3 values DSL, Fiber optics, No). In Figure 2, I can see most of customer is using Fiber optics.

- OnlineSecurity: This attribute is storing 3 values yes (in case a customer has online security), No (in case don't have it), or No internet services (in case they are not using the internet). In Figure 2, I can see most of the customer is not using online services.

- OnlineBackup: This attribute is storing 3 values yes (in case a customer has an online backup), No (in case don't have it), or No internet services (in case they are not using the internet). In Figure 2, I can see most of the customer is not using online backup.

- DeviceProtection: This attribute is storing 3 values yes (in case a customer has device protection), No (in case don't have it), or No internet services (in case they are not using the internet). In Figure 2, I can see most of the customer is not using device protection.

- TechSupport: This attribute is storing 3 values yes (in case a customer has tech support), No (in case don't have it), or No internet services (in case they are not using the internet). In Figure 2, I can see most of the customer is not using tech support.

- StreamingTV: This attribute is storing 3 values yes (in case a customer has streaming TV), No (in case don't use it), or No internet services (in case they are not using the internet). In Figure 2, I can see the ratio is almost the same on who is using it or not.

- StreamingMovies; This attribute is storing 3 values yes (in case a customer has a streaming movie), No (in case don't use it), or No internet services (in case they are not using the internet). In Figure 2, I can see the ratio is almost the same on who is using it or not.

Target attribute:

- Churn: Churn is the target variable that I am going to analysis and predict. It is containing 2 values Yes (in case the customer has not left), or No (customer churn)

5.3.3 Pre-processing the data

In data pre-processing I am going to change the datatype of TotalCharges.

```python
# TotalCharges is in String Format, but it should be float
df["TotalCharges"] = pd.to_numeric(df["TotalCharges"],errors='coerce')
```

In the above I changed the type of TotalCharges from object to float.

```
df.info() # After changing type of TotalCharges checking the data info again

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7032 non-null   float64
 20  Churn             7043 non-null   object
dtypes: float64(2), int64(2), object(17)
```

In the above result, I am checking the data type of total charges. Here I can see the type changed. And I can see some rows are null in TotalCharges.

```
df.describe()
```

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000	7032.000000
mean	0.162147	32.371149	64.761692	2283.300441
std	0.368612	24.559481	30.090047	2266.771362
min	0.000000	0.000000	18.250000	18.800000
25%	0.000000	9.000000	35.500000	401.450000
50%	0.000000	29.000000	70.350000	1397.475000
75%	0.000000	55.000000	89.850000	3794.737500
max	1.000000	72.000000	118.750000	8684.800000

Checking again this matrix, Now I can see I have 4 numeric features. And some rows is null for TotalCharges.

```
df.duplicated().sum() # checking duplicate values if any
]: 0
```

In the above, I am checking any duplicate value is present or not. Now result is 0, so no duplicate is there.

5.3.4 Cleaning data

In this section, I am cleaning the data. Dropping null values and not useful fields.

```
df = df.dropna() # dropping null values. Few values are missing that's why in describe function count of TotalCharges is
               #7032 and minimum value of tenure is 0. So its not useful to fill the TotalCharges with some number.
```

dropping null values. Few values are missing that's why in describe function count of TotalCharges is 7032 and the minimum value of tenure is 0. So it's not useful to fill the TotalCharges with some number.

```
df.describe()
```

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7032.000000	7032.000000	7032.000000	7032.000000
mean	0.162400	32.421786	64.798208	2283.300441
std	0.368844	24.545260	30.085974	2266.771362
min	0.000000	1.000000	18.250000	18.800000
25%	0.000000	9.000000	35.587500	401.450000
50%	0.000000	29.000000	70.350000	1397.475000
75%	0.000000	55.000000	89.862500	3794.737500
max	1.000000	72.000000	118.750000	8684.800000

Here, I can analysis the same matrix again. I can see the total number of rows is 7032. And after deleting the null from TotalCharges min value is changed for tenure.

```
#droping customerid
df = df.drop(columns = 'customerID')
```

Above, I am dropping the customerID attribute because it is not required.

5.4.   Visualize the data

In this step, I am using matplotlib library to plot the data.

5.4.1   Counting churn values

```
M Churn = df['Churn'].value_counts() # counting churn values
  Churn
```

```
: No     5163
  Yes    1869
  Name: Churn, dtype: int64
```
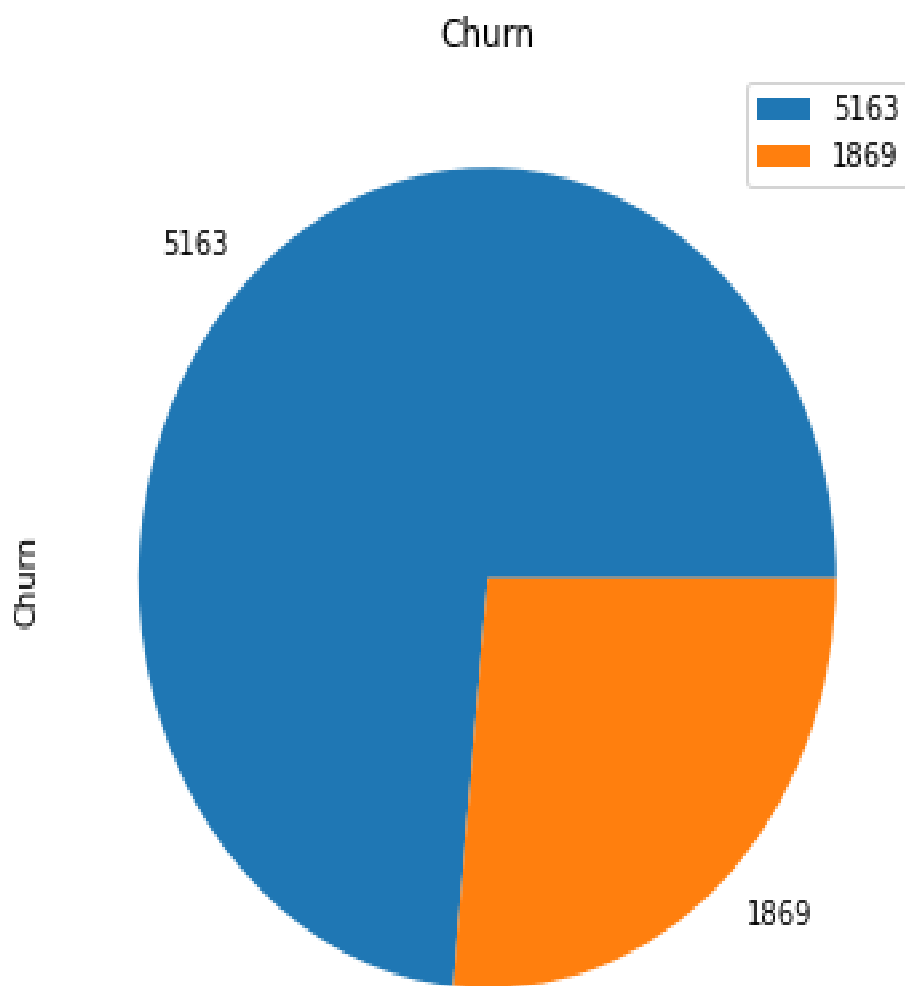
Here, we can see the count of churn and non-churn. No is representing a count of non-churn and yes is a count of churn customers.

non-churn: 5163

churn: 1869

5.4.2   Plotting the pie chart for churn and non churn count

Figure 3 Number of Churn customer.

Churn

Here in

Figure 3, we can see the count of churn and non-churn.

5.4.3   Churn based on Tenure and Monthly Charges

Figure 4 Churn based on Tenure and Monthly Charges

Tenure and Monthly Charges affect on Churn Customers
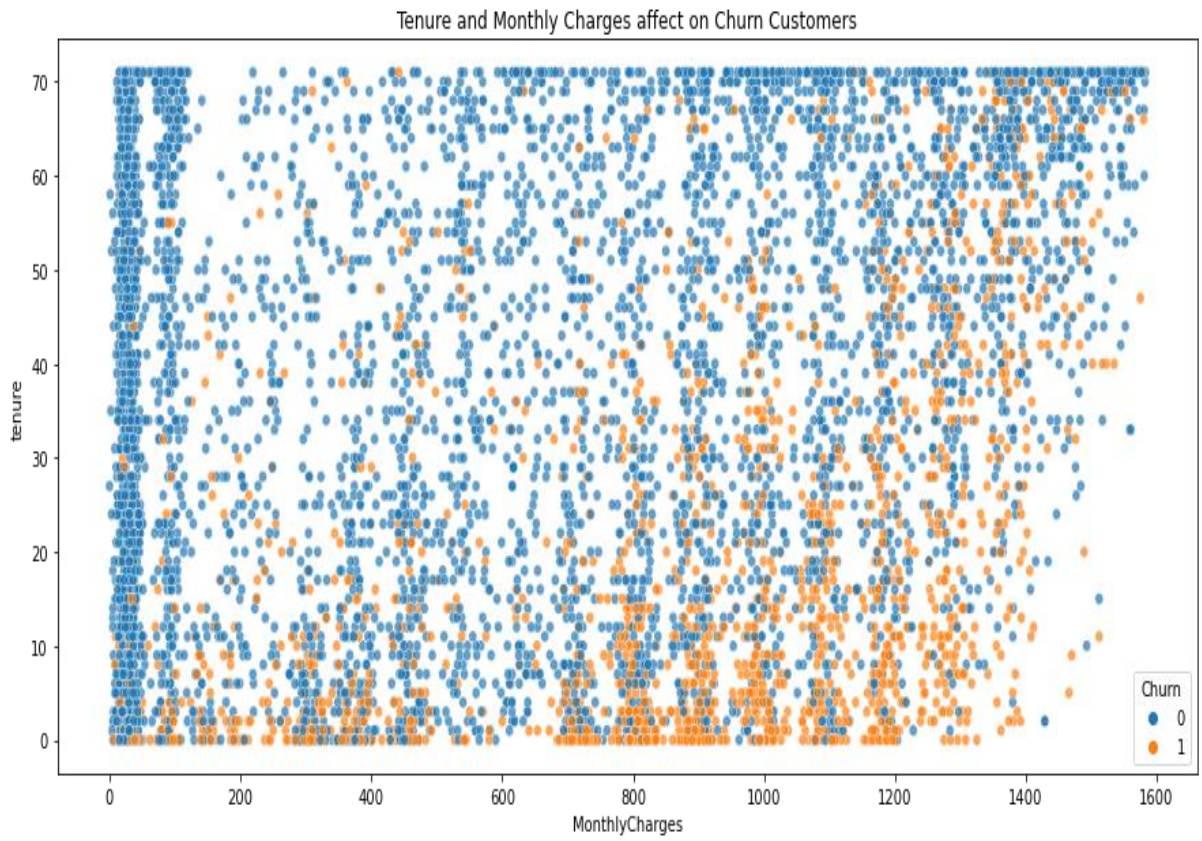
Figure 4 almost churn customer is new customer because tenure is less and they was paying more monthly charges.

5.4.4 Churn based on Tenure and Total Charges
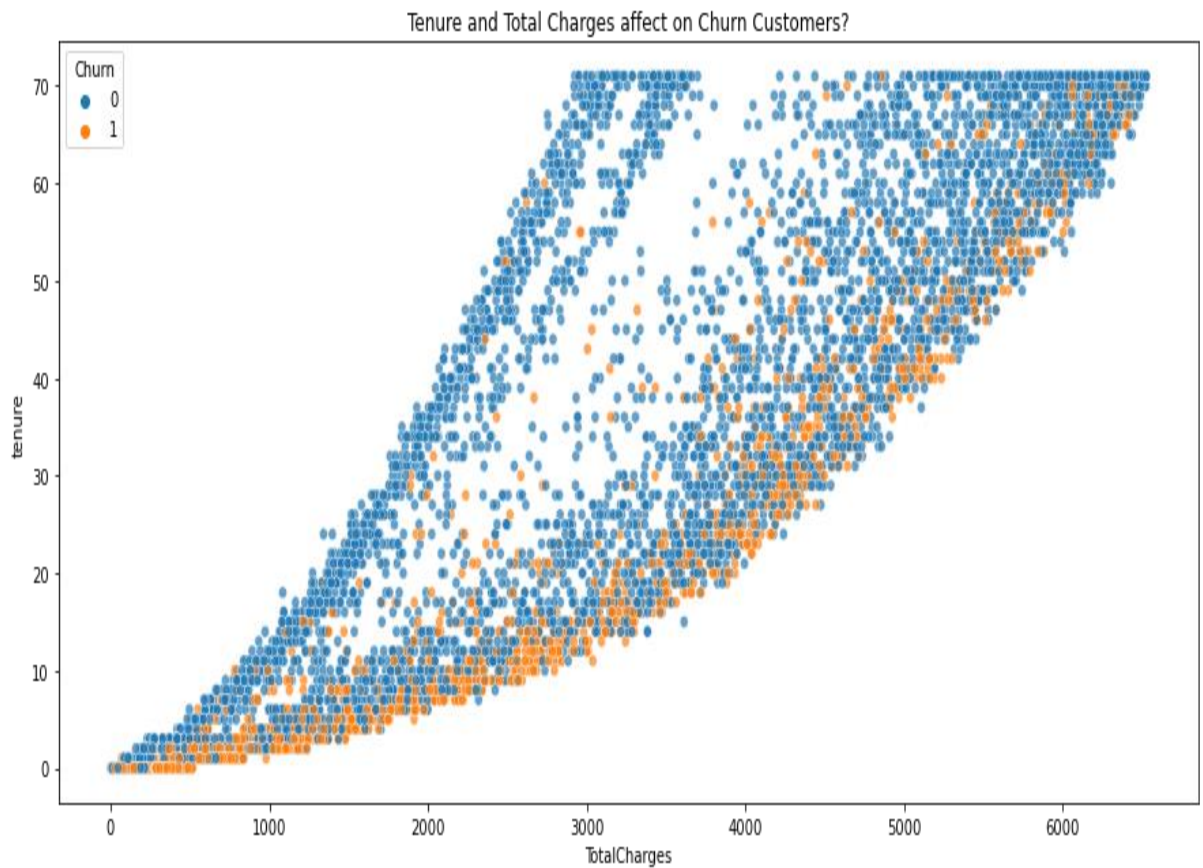
Figure 5 Churn based on Tenure and Total Charges

Figure 5 showing non-churn customer they have less total charges and they are using services from long back. And in case of churn customer they are new customer and because of total charges they left.

To group data by Churn and compute the mean to find out if churners make more Monthly Charge than non-churners

5.4.5 Group data by 'Churn' and compute the mean

```python
# Group data by 'Churn' and compute the mean
print(df.groupby('Churn')['MonthlyCharges','TotalCharges','tenure'].median())
(df.groupby('Churn')['MonthlyCharges','TotalCharges','tenure'].median()).plot(kind = 'box',subplots=True, title = 'Avg value

plt.show()
```

```
C:\Users\SweetyWadhwa\AppData\Local\Temp\ipykernel_19584\2311718574.py:2: FutureWarning: Indexing with multiple keys (implic
itly converted to a tuple of keys) will be deprecated, use a list instead.
  print(df.groupby('Churn')['MonthlyCharges','TotalCharges','tenure'].median())
C:\Users\SweetyWadhwa\AppData\Local\Temp\ipykernel_19584\2311718574.py:3: FutureWarning: Indexing with multiple keys (implic
itly converted to a tuple of keys) will be deprecated, use a list instead.
  (df.groupby('Churn')['MonthlyCharges','TotalCharges','tenure'].median()).plot(kind = 'box',subplots=True, title = 'Avg val
ue of Churn based on Charges and tenure', figsize = (16,6))
```

```
        MonthlyCharges  TotalCharges  tenure
Churn
No              64.45       1683.60    38.0
Yes             79.65        703.55    10.0
```

Figure 6 Avg value of Churn based on Charges and tenure.

Avg value of Churn based on Charges and tenure
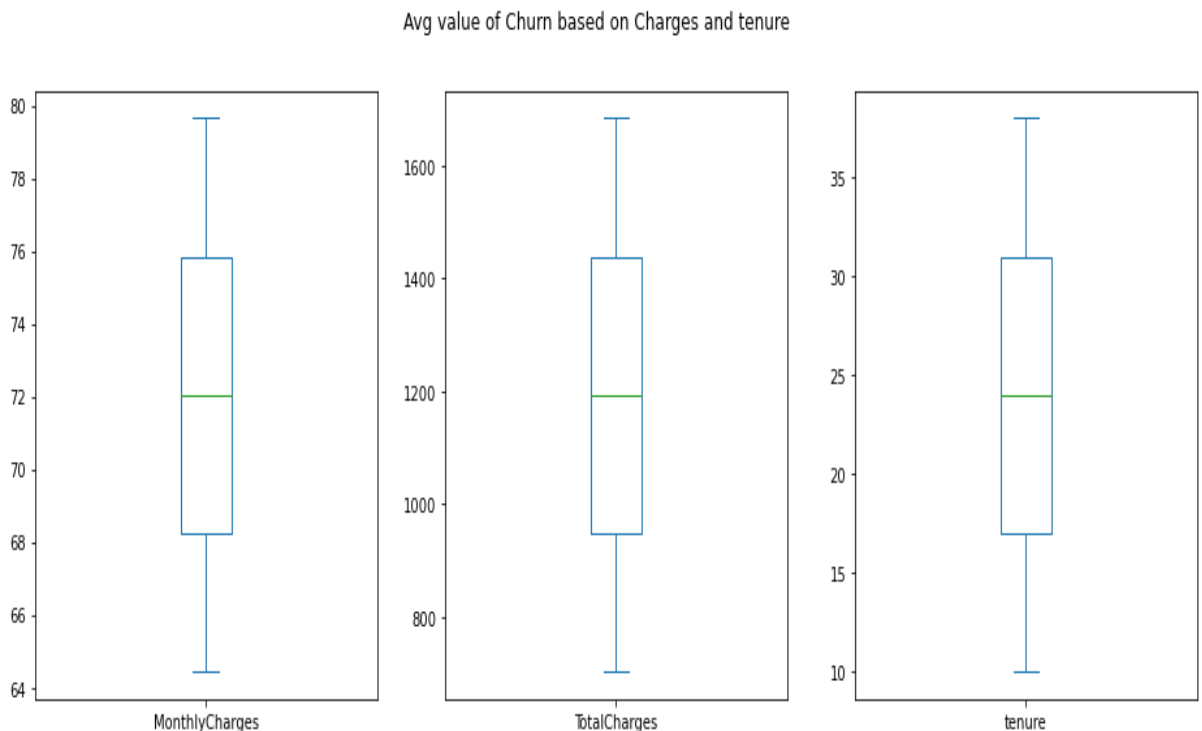
In Figure 6, We can see the churning customers have higher monthly charges with a median of ca. 80 USD and much lower interquartile range compared to that of non-churners (median of ca. 65 USD).

TotalCharges are the result of tenure and MonthlyCharges, which are more insightful on an individual basis.

Churning customers have much lower tenure with a mean of ca. 10 months compared to a median of non-churners of ca. 38 months.

Hence! In above result we can see, churners seem to pay more MonthlyCharges than non-churners.

5.4.6   Counting the customer based on different columns:

Counting Churn data based on SeniorCitizen:

```
# Counting Churn data based on SeniorCitizen
print(pd.crosstab(df.SeniorCitizen,df.Churn,margins=True))

Churn          No   Yes   All
SeniorCitizen
0            4497  1393  5890
1             666   476  1142
All          5163  1869  7032
```

In the above result, we can see 5890 customer was not Senior who was using. and out of 5890, 1393 is churn. and in opposite site 1142 was Senior Citizen and 476 is churned.

Counting Churn data based on gender:

```
# Counting Churn data based on gender
print(pd.crosstab(df.gender,df.Churn,margins=True))
```

```
Churn    No   Yes   All
gender
Female  2544   939  3483
Male    2619   930  3549
All     5163  1869  7032
```

In above result, female, and male for both gender count is approx. same.

5.4.7   % Of the customer based on different columns:

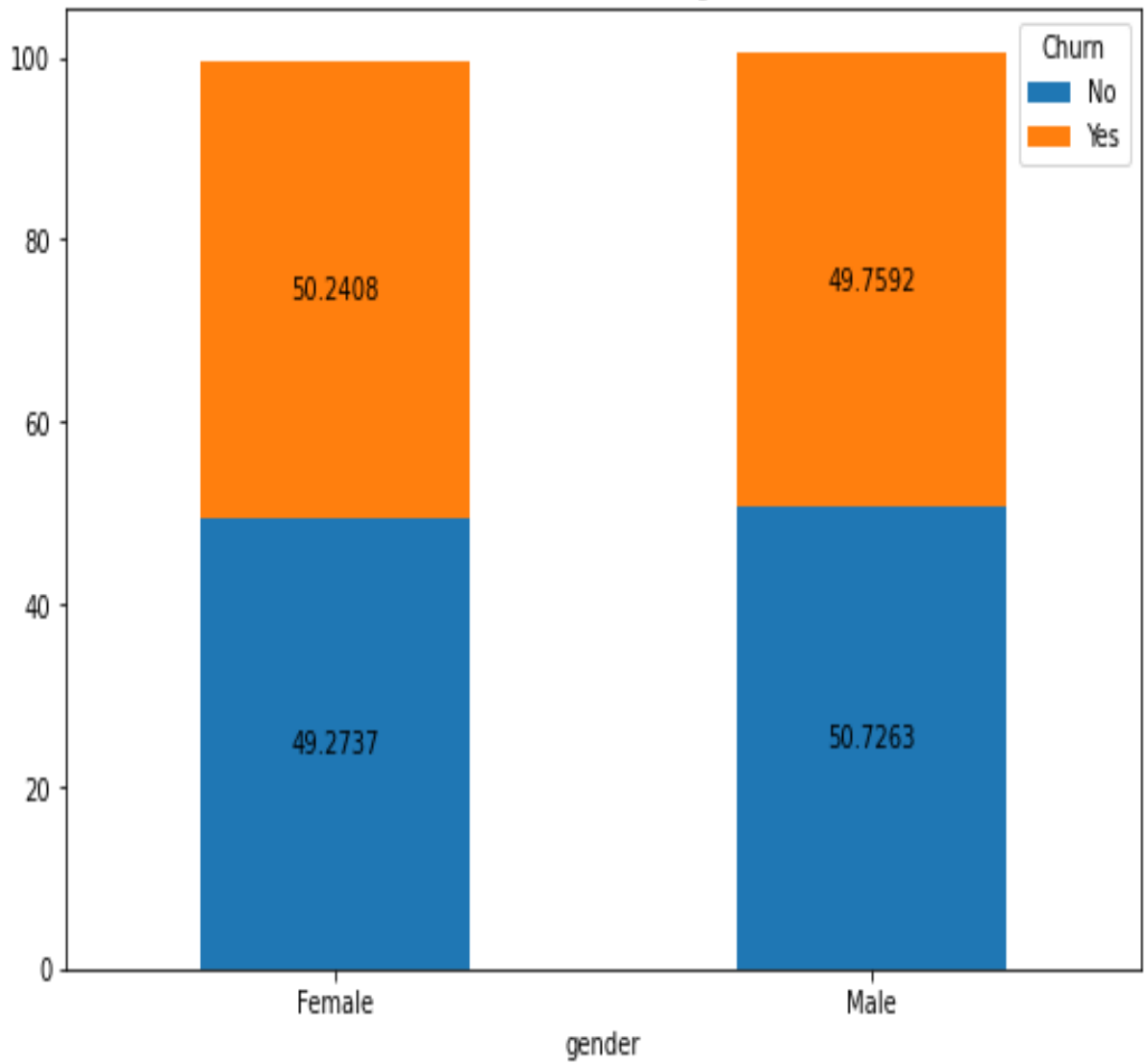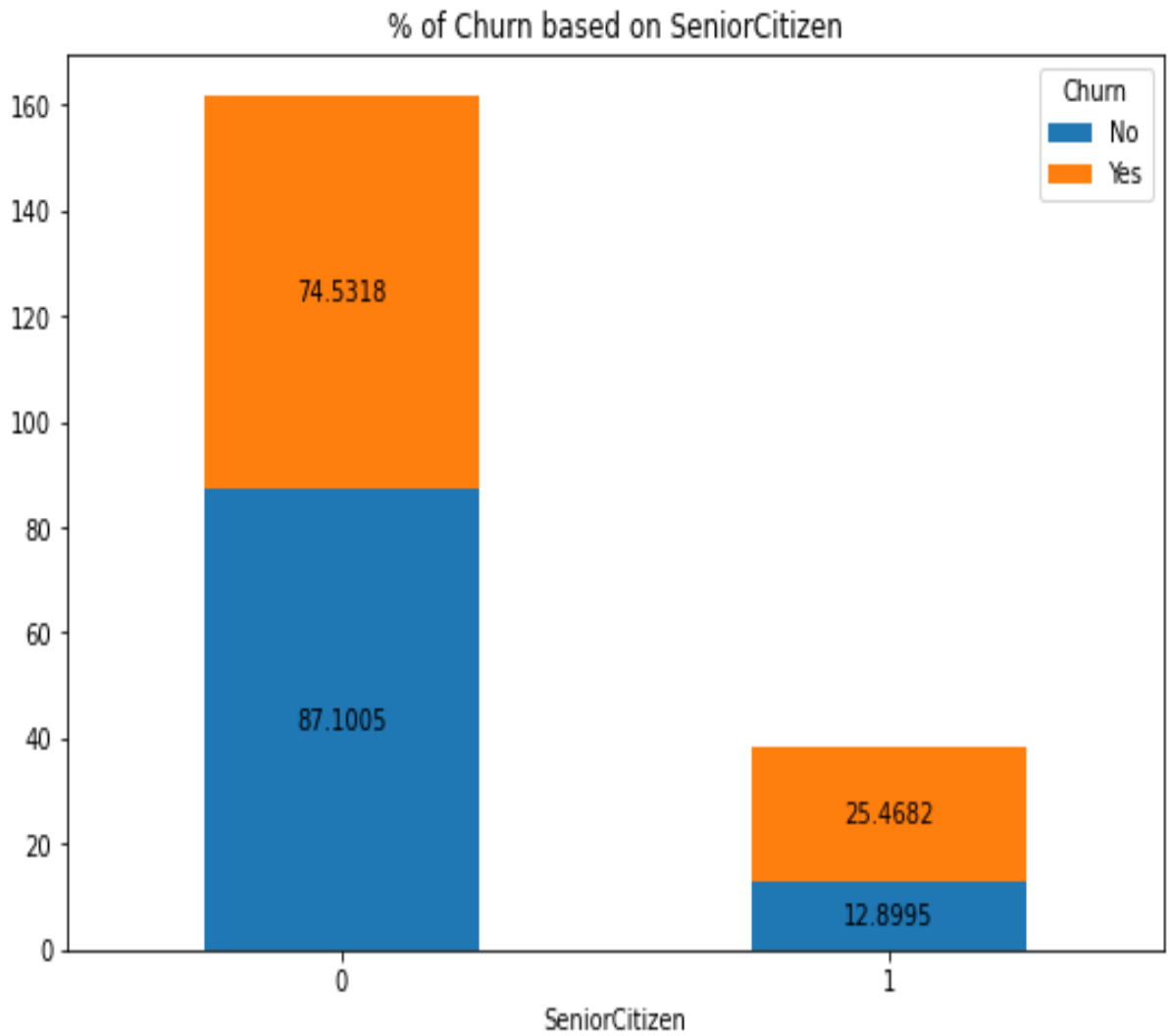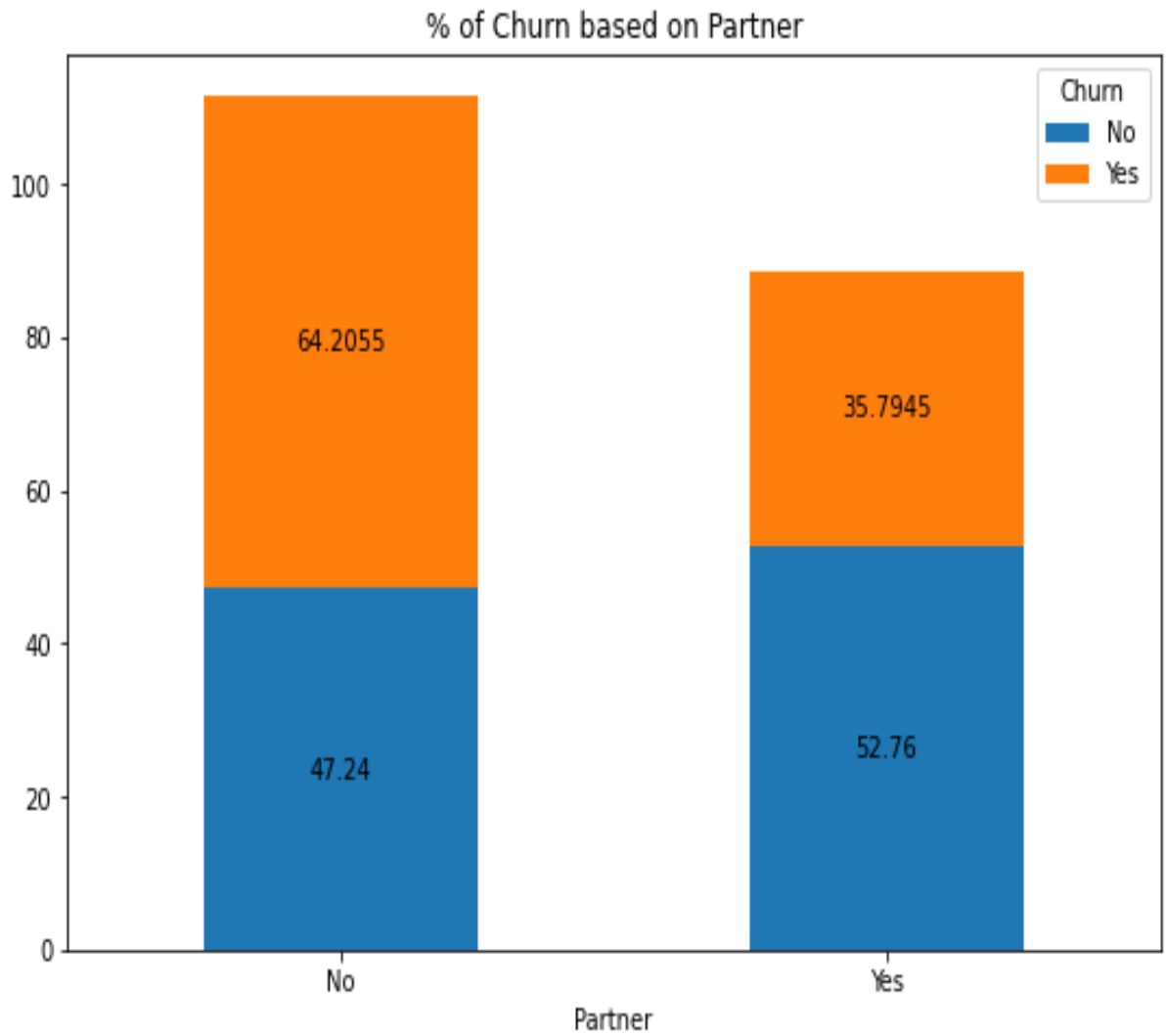Figure 7 **% Of Churn data based on gender**

% of Churn based on gender
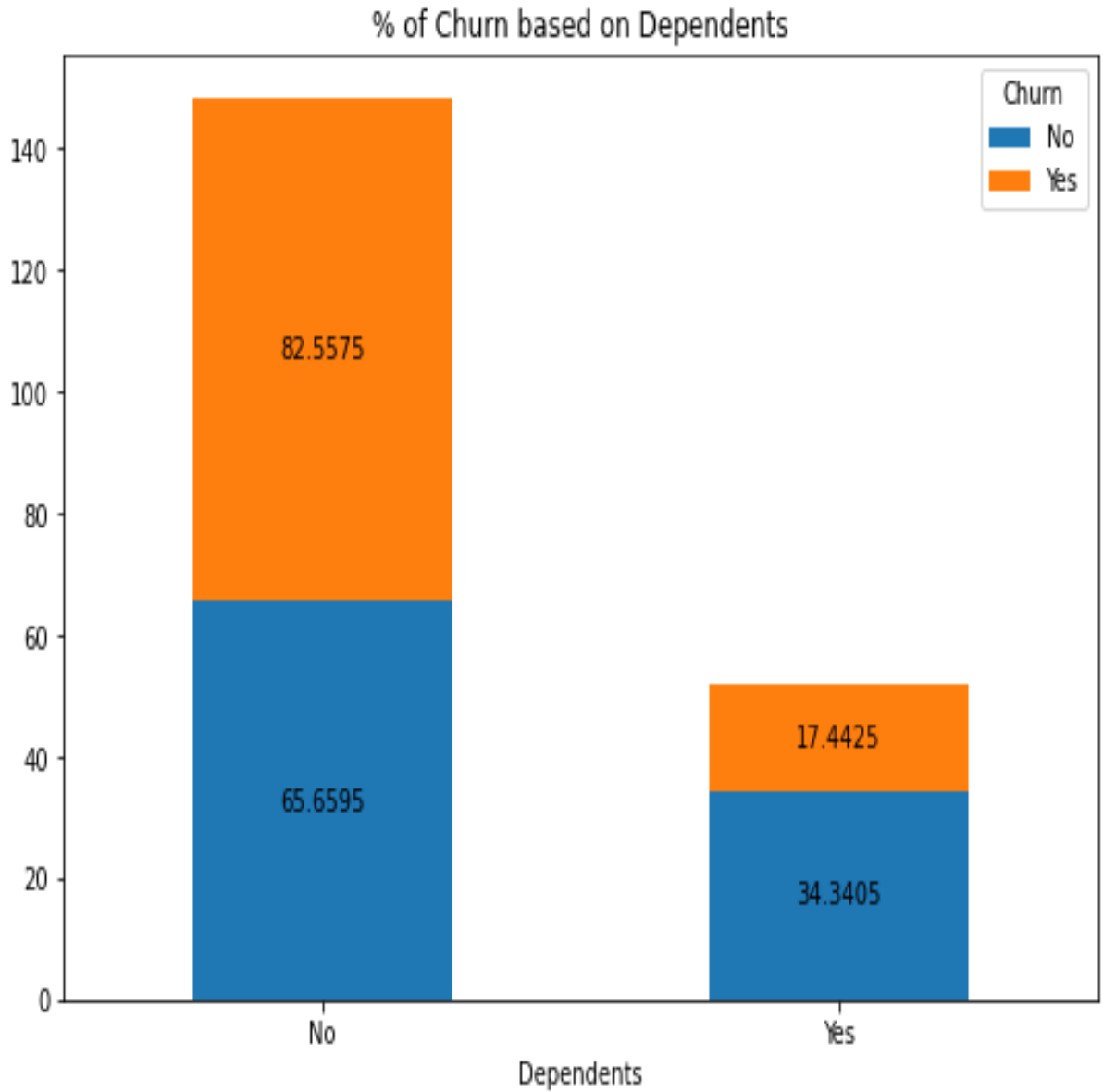
56

Figure 8 % of Churn data based on SeniorCitizen

In the above Figure 8, non-senior citizens churn % is much lower than senior churn %.

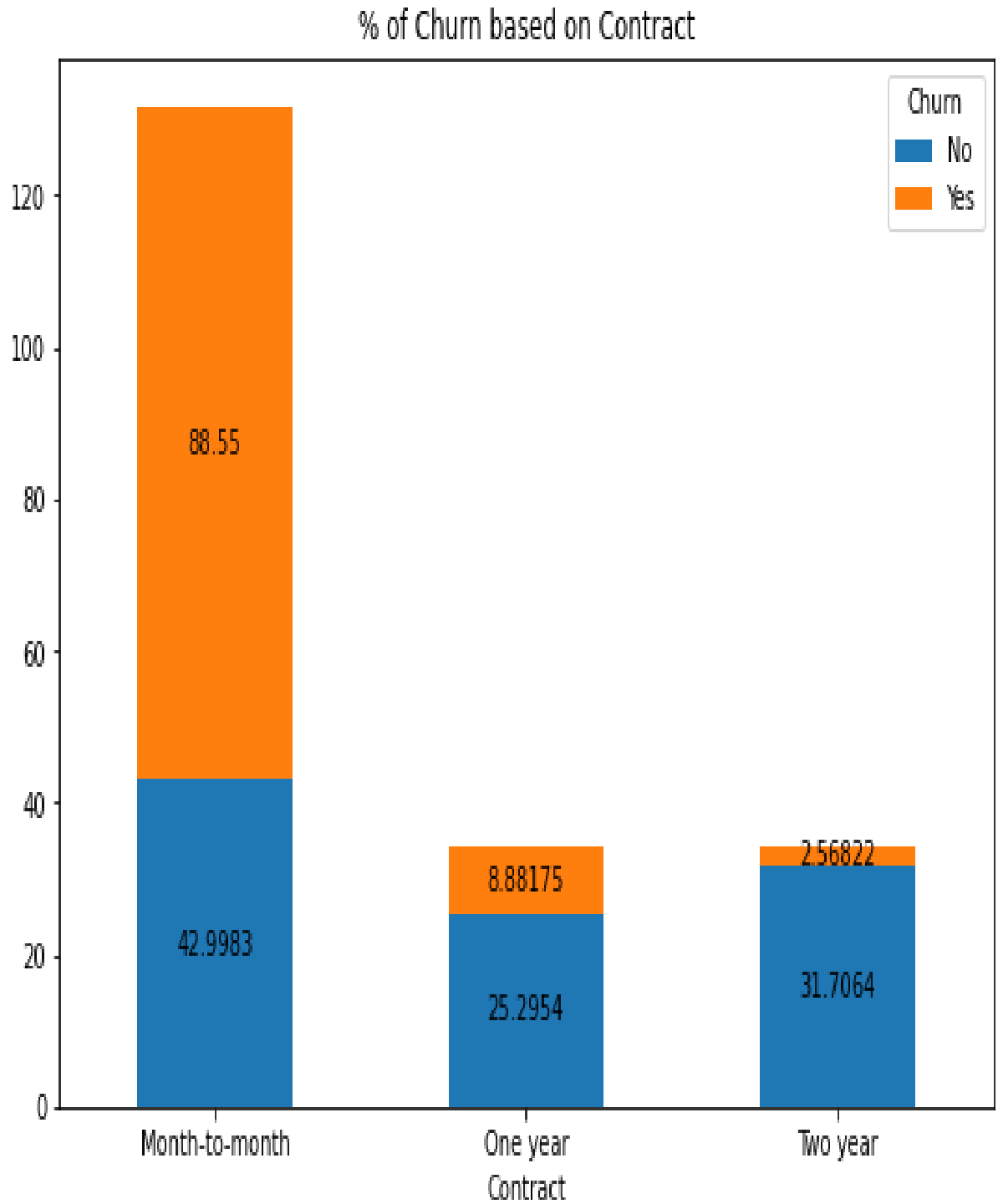Figure 9 % of Churn data based on Partner

% of Churn based on Partner

In above Figure 9, Churn rate for customers without partners is higher than Partner

Figure 10 % of Churn data based on Dependents

% of Churn based on Dependents

In above Figure 10, churn rate for customers without children is very high.

Figure 11 % of Churn data based on Contract

## % of Churn based on Contract

In above Figure 11, Churn % for month-to-month contracts much higher than with remaining contracts, then one year contract is higher than Two year.

Figure 12 % of Churn data based on PaymentMethod

% of Churn based on PaymentMethod
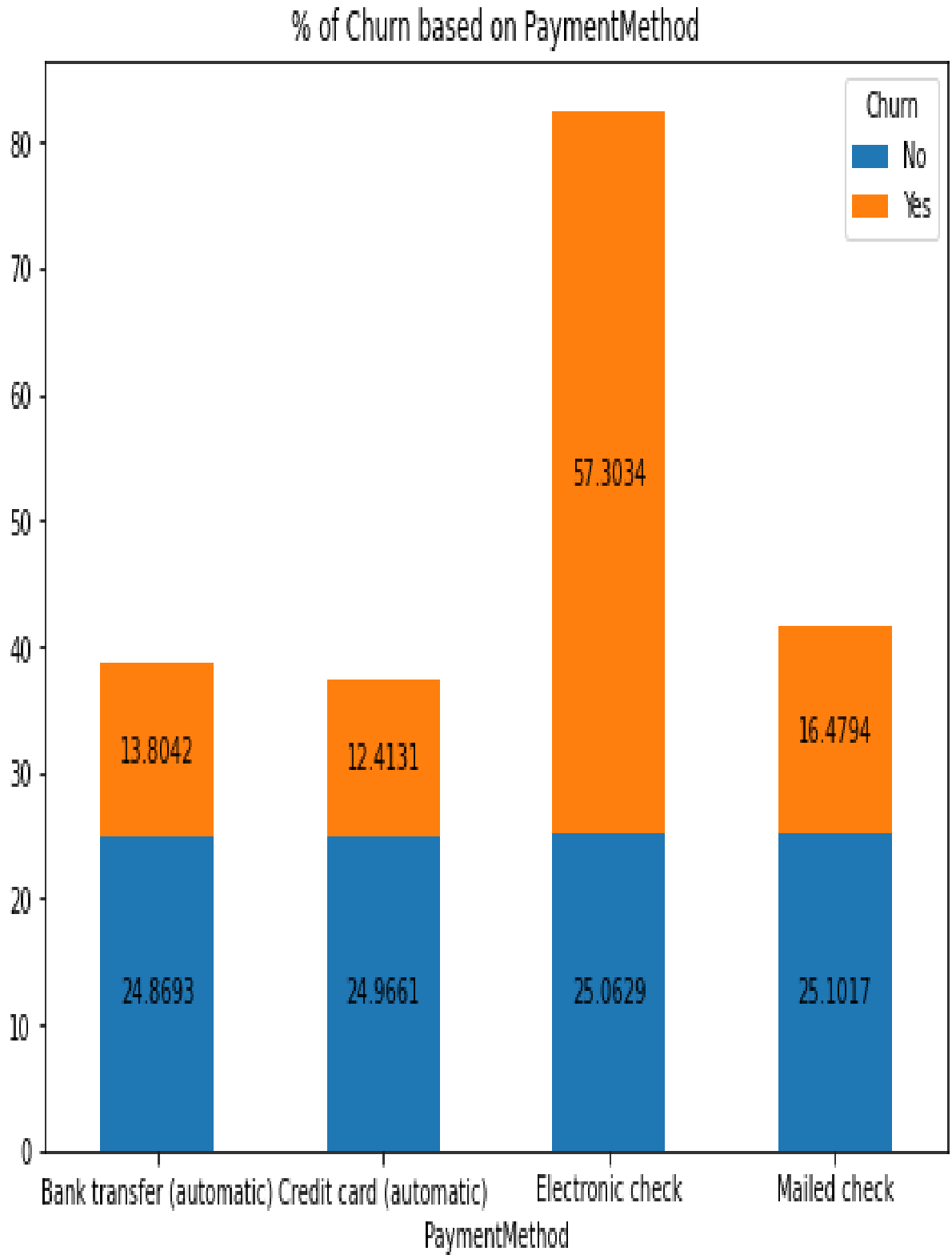
In above Figure 12,electronic check paymentMethod shows much higher churn rate, then mailed check. Bank transfer and credit card is appox same.

Figure 13 % of Churn data based on InternetService

% of Churn based on InternetService
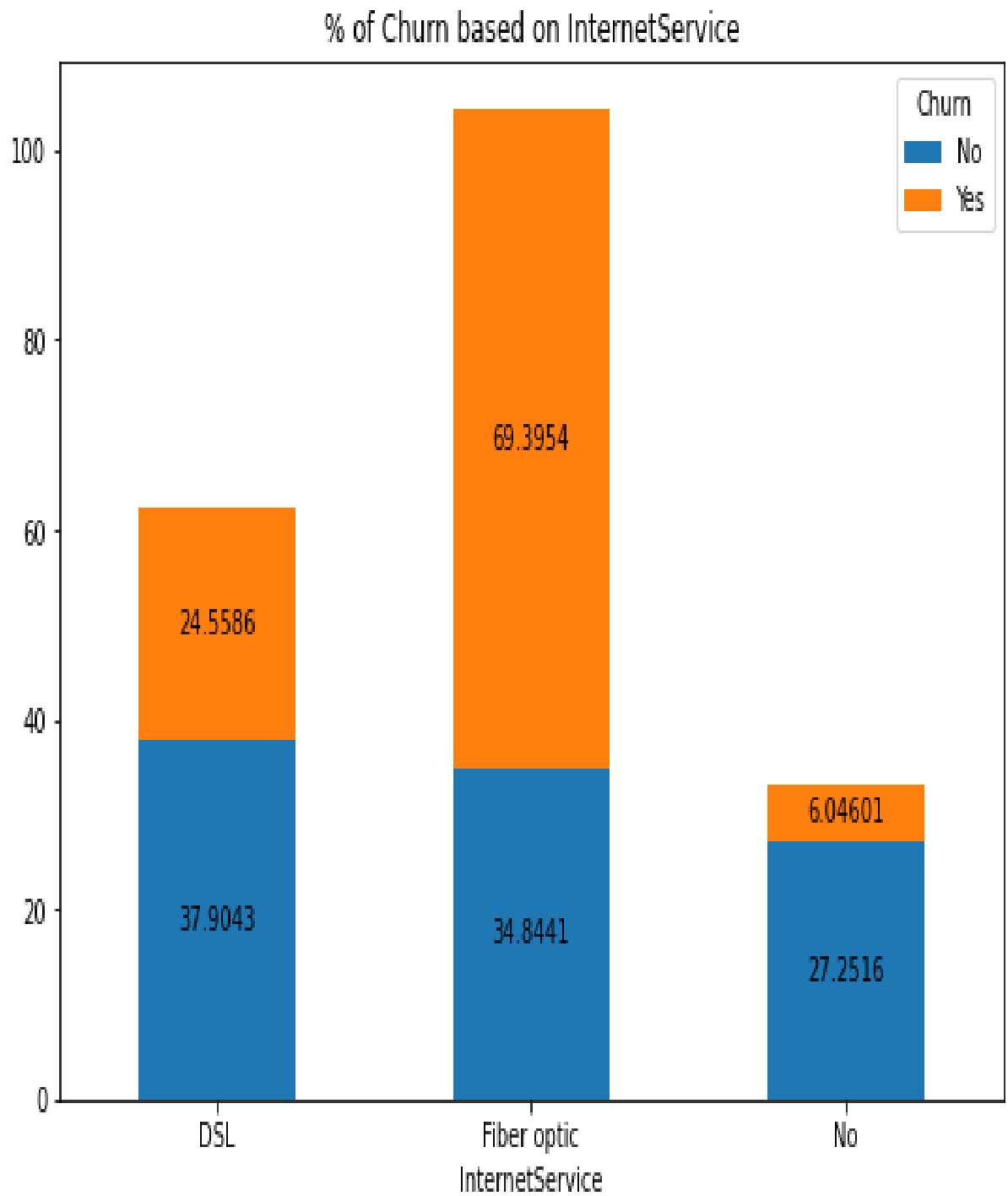
In above Figure 13, Customers with fiber optic InternetService have much higher churn rate.It is approx. 69%.

5.5. Analysis the features

In this step I am converting the categorical attributes into numeric.

```python
def convert_category(df, column_list):
    temp = pd.DataFrame()
    temp = df

    for col in column_list:
        temp[col] = temp[col].astype('category')
        temp[col] =  temp[col].cat.codes

    print('Categorical conversion completed successfully.')
    return temp
```

```python
#Converting categorical Columns to Numberical
column_list = ["gender", "Partner", "Dependents", "tenure", "PhoneService", "MultipleLines", "InternetService", "OnlineSecuri
               "DeviceProtection", "TechSupport", "StreamingTV","StreamingMovies", "Contract", "PaperlessBilling", "PaymentMethc
               "MonthlyCharges", "TotalCharges", "Churn"]
df = convert_category(df, column_list)
```

Categorical conversion completed successfully.

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7032 entries, 0 to 7042
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   gender            7032 non-null   int8
 1   SeniorCitizen     7032 non-null   int64
 2   Partner           7032 non-null   int8
 3   Dependents        7032 non-null   int8
 4   tenure            7032 non-null   int8
 5   PhoneService      7032 non-null   int8
 6   MultipleLines     7032 non-null   int8
 7   InternetService   7032 non-null   int8
 8   OnlineSecurity    7032 non-null   int8
 9   OnlineBackup      7032 non-null   int8
 10  DeviceProtection  7032 non-null   int8
 11  TechSupport       7032 non-null   int8
 12  StreamingTV       7032 non-null   int8
 13  StreamingMovies   7032 non-null   int8
 14  Contract          7032 non-null   int8
 15  PaperlessBilling  7032 non-null   int8
 16  PaymentMethod     7032 non-null   int8
 17  MonthlyCharges    7032 non-null   int16
 18  TotalCharges      7032 non-null   int16
 19  Churn             7032 non-null   int8
dtypes: int16(2), int64(1), int8(17)
memory usage: 254.1 KB
```

In the above result I can see all attributes is storing int value.

Figure 14 Corelation Matrix
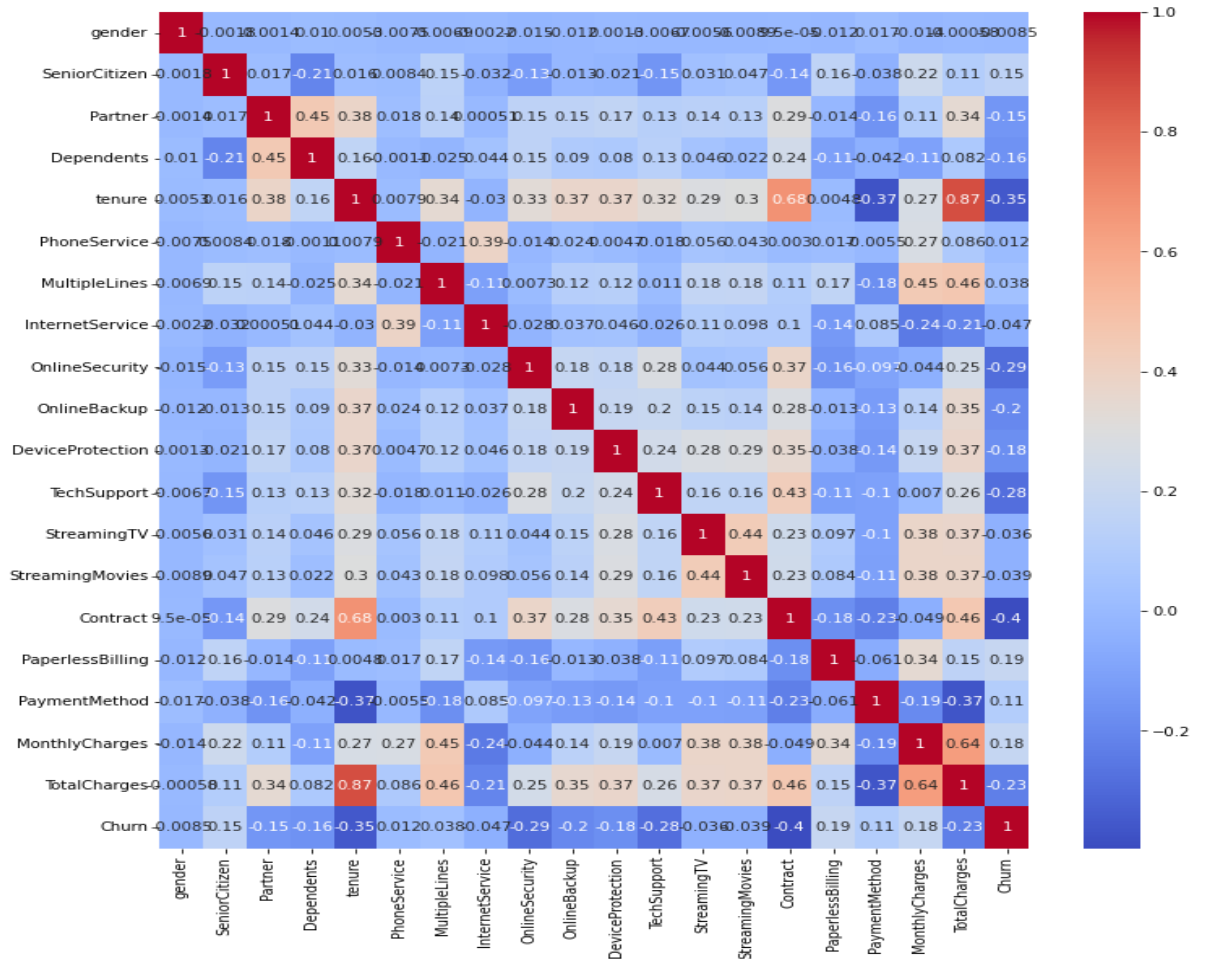
With Figure 14 we can see

- Tenure has very strong relation between contract and TotalCharges.

- TotalCharges has very strong relation between tenure and MonthlyCharges.

- And some services has good relation with MonthlyCharges and TotalCharges. I think so because its paid services.

- Contract has good relation with tenure.

After analysis the data My Hypotheses is Tenure, Monthly Charges, TotalCharges, Contract and some extra services is the reason of churn.

Tenure: Customer is new customer.

MonthlyCharges: Customer is paying more monthly charges.

TotalCharges: Customer is paying more total charges.

Contract: Who as month-to-month contract.

Extra services: Who is using some extra services.

5.6.   Train-Test split

In this step I am splitting the model.

```
In [49]:    #To Create Training and Test sets
            X = df.drop("Churn", axis = 1)
            y = df.Churn

            # Splitting the dataset into the Training and Test sets

            X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size = 0.20,
                                                    random_state = 0)
```

```
In [50]:    y_test.shape
```

```
Out[50]:    (1407,)
```

```
In [51]:    y_train.shape
```

```
Out[51]:    (5625,)
```

```
In [52]:    X_train.shape
```

```
Out[52]:    (5625, 19)
```

```
In [53]:    X_test.shape
```

```
Out[53]:    (1407, 19)
```

```
In [54]:    # Feature Scaling

            sc = StandardScaler()
            X_train = sc.fit_transform(X_train)
            X_test = sc.transform(X_test)
```

5.7.   Model Evaluation

5.7.1.   Random Forest:

```
In [68]:    # Instantiate the classifier
            clf = RandomForestClassifier()

            # Fit to the training data
            clf.fit(X_train, y_train)
```

Out[68]: RandomForestClassifier()

```
In [69]:    # Predict the labels for the test set
            y_pred = clf.predict(X_test)
            feature_imp = clf.feature_importances_
            # Compute accuracy and Evaluating Model Performance
            accuracy_score(y_test, y_pred)
```

Out[69]: 0.7846481876332623

```
In [70]:    def imp_features(feature_imp,X):

                print("Important Feature", feature_imp)
                columns = X.columns
                index = np.arange(len(columns))
                font_size = 4
                fontsize_tittle = 25

                plt.figure(figsize=(30, 9))
                plt.bar(index, feature_imp, width=0.5,
                        alpha=0.4,
                        color='r',
                        label='')

                plt.xlabel('Columns', fontsize=font_size)
                plt.ylabel('Feature Importance', fontsize=font_size)
                plt.title('Feature Importance for each column', fontsize=fontsize_tittle)
                plt.xticks(index, columns)
                plt.show()
```
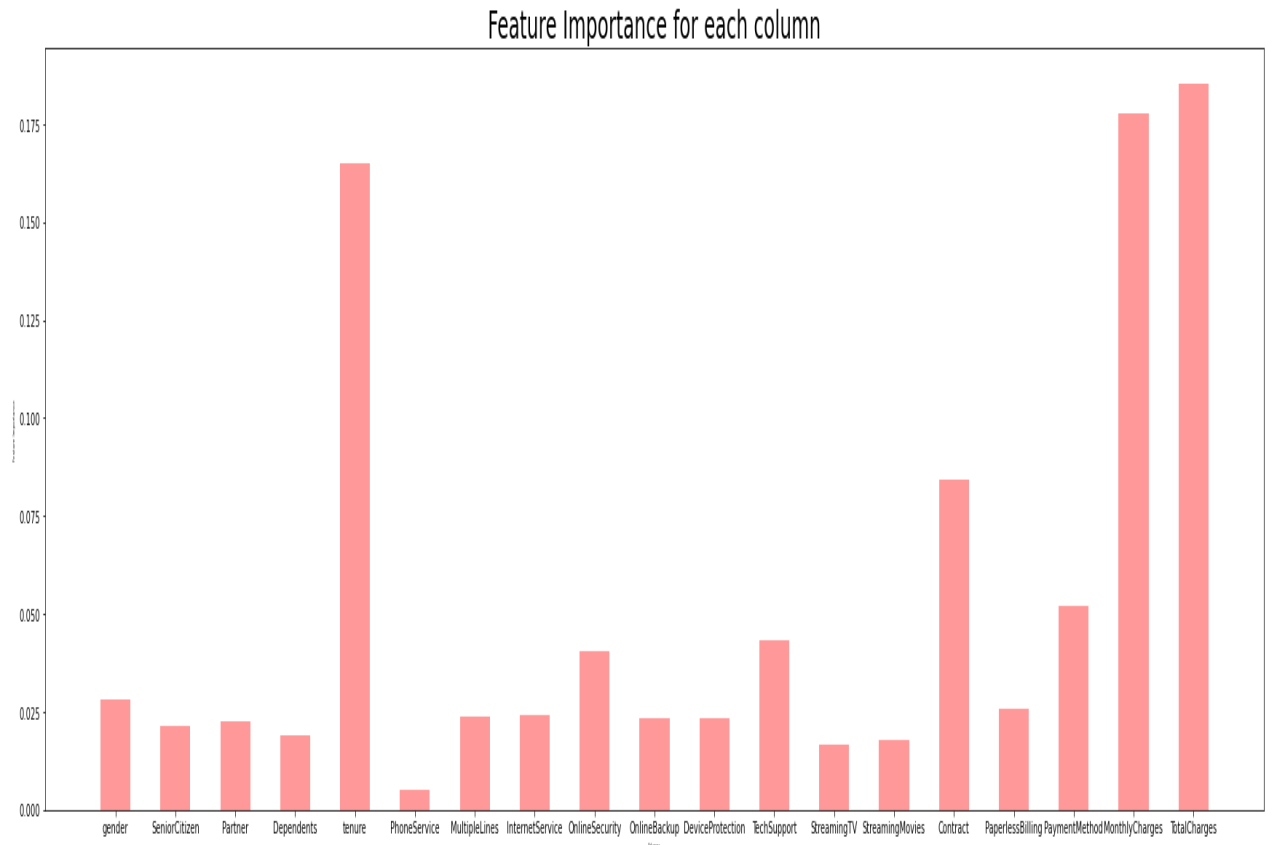
```
In [71]:    # Plotting important features
            imp_features(feature_imp,X)
```

```
Important Feature [0.02815625 0.02153421 0.02267984 0.01893587 0.16505999 0.00509303
 0.02379745 0.02423744 0.04069277 0.02352202 0.02339923 0.04340849
 0.01679282 0.01769545 0.08428872 0.02568571 0.05186035 0.17775159
 0.18540879]
```

Feature Importance for each column

5.7.2. K Nearest Neighbour:

```
In [77]:  # Instantiate the classifier

          clf= KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2 )
          # Fit to the training data
          clf.fit(X_train, y_train)
```

Out[77]: KNeighborsClassifier()

```
In [79]:  #Predicting the test set result
          y_pred= clf.predict(X_test)
          # Compute accuracy and Evaluating Model Performance

          accuracy_score(y_test, y_pred)
```

Out[79]: 0.7739872068230277

5.7.3. Logistic regression:

```python
# Instantiate the classifier
clf = LogisticRegression(max_iter=100)

# Fit to the training data
clf.fit(X_train, y_train)
```

31]: LogisticRegression()

```python
#Predicting the test set result
y_pred= clf.predict(X_test)
# Compute accuracy and Evaluating Model Performance

accuracy_score(y_test, y_pred)
```

33]: 0.8088130774697939

```python
print(print(clf.coef_))
a = np.std(X_train, 0)*clf.coef_
importance = clf.coef_[0]
print(dict(zip(X.columns,clf.coef_[0])))
#importance is a list so you can plot it.
feat_importances = pd.Series(importance)
feat_importances.nlargest(20).plot(kind='bar',title = 'Feature Importance')
print(df.columns)
```

```
[[ 0.01375962  0.1109247   0.0415871  -0.10147455  0.15687149 -0.24607486
   0.09905989  0.06686366 -0.18325583 -0.12255493 -0.01361051 -0.20981523
  -0.02194922  0.02286693 -0.6939176   0.19419214  0.05720168  1.21110646
  -1.15352679]]
None
{'gender': 0.013759618187033017, 'SeniorCitizen': 0.110924701956664214, 'Partner': 0.041587101169156135, 'Dependents': -0.101
47455055216396, 'tenure': 0.15687149443498216, 'PhoneService': -0.24607485722565312, 'MultipleLines': 0.09905988668569321,
'InternetService': 0.06686366042996802, 'OnlineSecurity': -0.18325582962700335, 'OnlineBackup': -0.12255493463510078, 'Devic
eProtection': -0.013610514096386252, 'TechSupport': -0.20981522647231815, 'StreamingTV': -0.021949216518502027, 'StreamingMo
vies': 0.022866925303581993, 'Contract': -0.693917602347338, 'PaperlessBilling': 0.19419213617608802, 'PaymentMethod': 0.057
20168039168878, 'MonthlyCharges': 1.2111064595819767, 'TotalCharges': -1.1535267911237752}
Index(['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure',
       'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
       'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
       'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod',
       'MonthlyCharges', 'TotalCharges', 'Churn'],
      dtype='object')
```
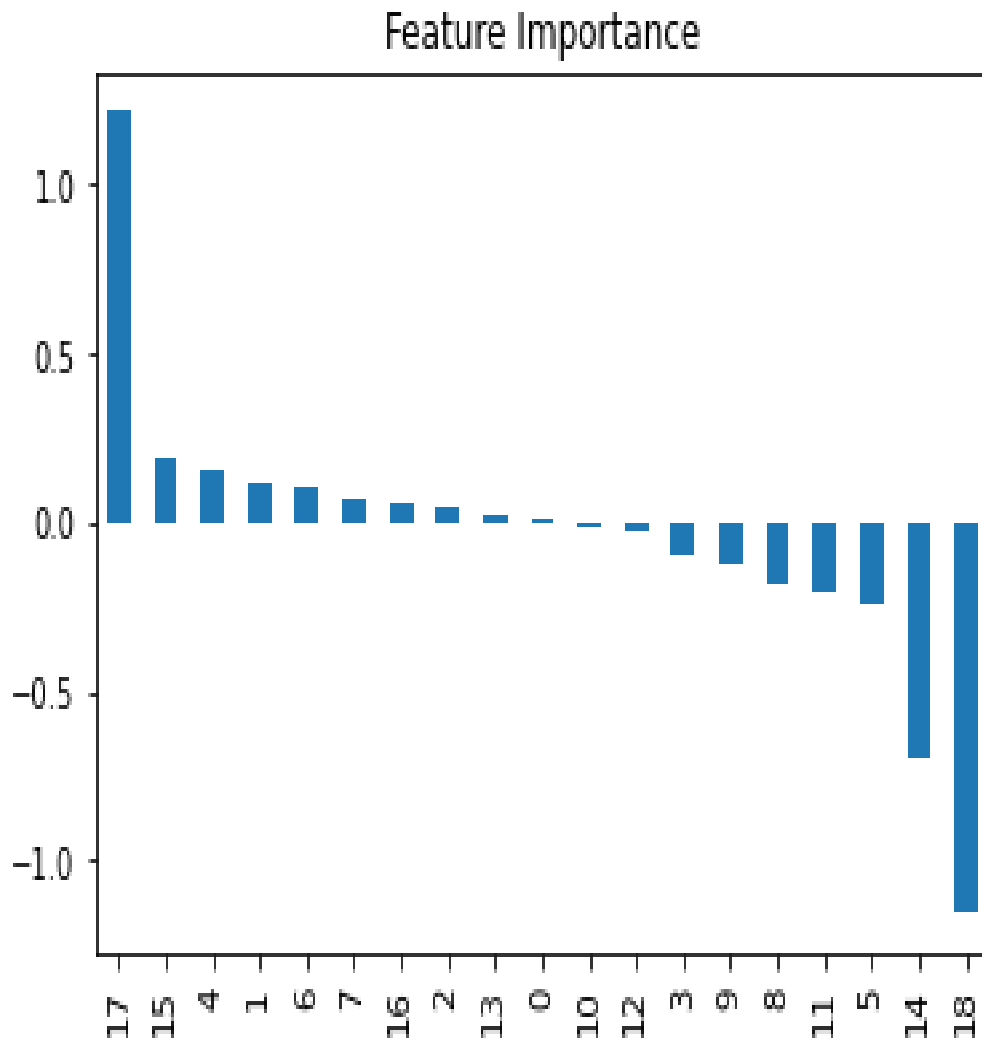
Figure 15 Feature Importance

Feature Importance


```
In [86]: ▶ # Plotting important features
         imp_features(importance,X)

         Important Feature [ 0.01375962  0.1109247   0.0415871  -0.10147455  0.15687149 -0.24607486
           0.09905989  0.06686366 -0.18325583 -0.12255493 -0.01361051 -0.20981523
          -0.02194922  0.02286693 -0.6939176   0.19419214  0.05720168  1.21110646
          -1.15352679]
```

Figure 16 Feature Importance for each columns

Feature Importance for each column
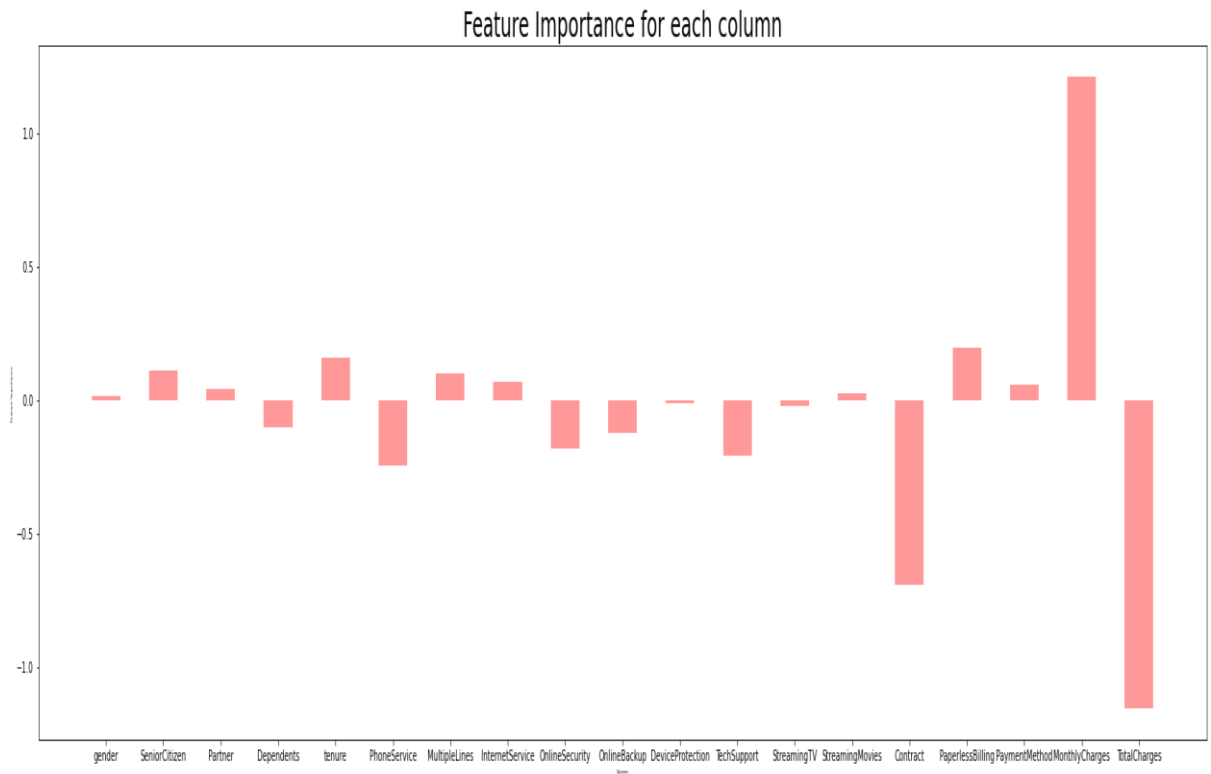
5.7.4. SVM:

SVM

```
# Instantiate the classifier
clf = SVC(gamma='auto')

# Fit to the training data
clf.fit(X_train, y_train)
```

`37]:` SVC(gamma='auto')

```
#Predicting the test set result
y_pred= clf.predict(X_test)
# Compute accuracy and Evaluating Model Performance

accuracy_score(y_test, y_pred)
```

`90]:` 0.8081023454157783

```
print(dict(zip(X.columns,clf.coef_[0])))
f_importances(clf.coef_[0], X)
imp_features(clf.coef_[0],X)
```

{'gender': 0.004631665715920996, 'SeniorCitizen': 0.07867872094195383, 'Partner': 0.05320838691658725, 'Dependents': -0.05637387764913937, 'tenure': 0.4175820885358853, 'PhoneService': -0.21164136313236442, 'MultipleLines': 0.09123894365584917, 'InternetService': 0.1477018575910649, 'OnlineSecurity': -0.14857961103628758, 'OnlineBackup': -0.10591119628260914, 'DeviceProtection': -0.03227659323783039, 'TechSupport': -0.1609654594463883, 'StreamingTV': -0.04777973339016217, 'StreamingMovies': -0.01349782560186740, 'Contract': -0.11356018387452083, 'PaperlessBilling': 0.1376944916795717, 'PaymentMethod': 0.007052023099649807, 'MonthlyCharges': 1.072535576715565, 'TotalCharges': -1.3729059424733456}

Feature Importance for each column
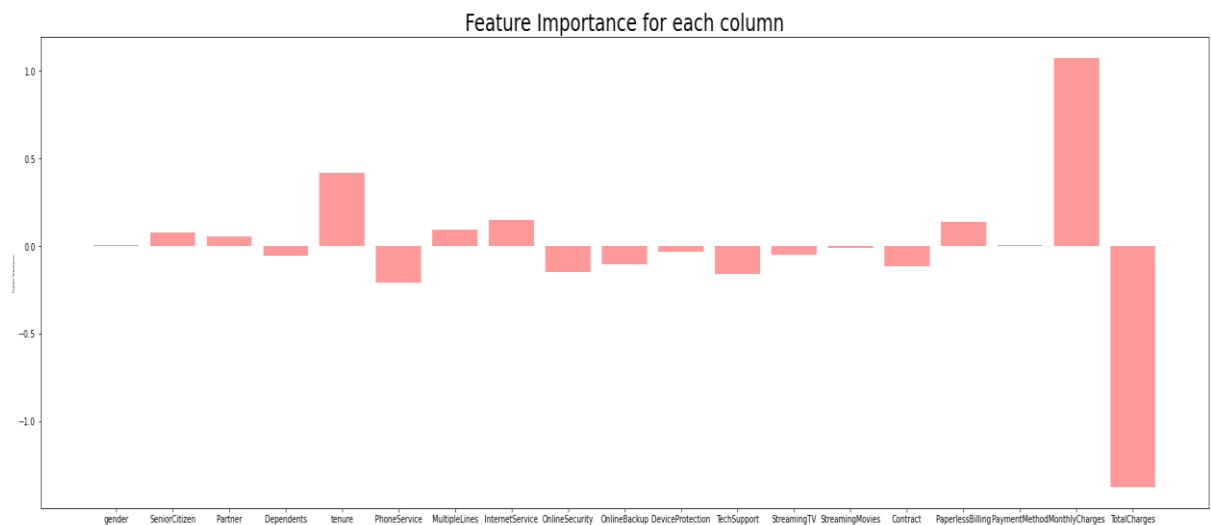
5.8.   Model selection Prediction and Assessment

In this step I am generating confusion matrix.

5.8.1. Random Forest

```
Confusion Matrix [[923 115]
 [188 181]]
 Accuracy = TP+TN/(TP+TN+FP+FN)=  0.7846481876332623


 Missclassification = 1-Accuracy=  0.21535181236673773


 Sensitivity/Recall or True Positive rate = TP/(TP+FN)=  0.4905149051
490515


 Specificity or True Negative Rate = TN/(TN+FP) =  0.8892100192678227


 Precision/Positive Predictiv value = TP/(TP+FP) =  0.611486486486486
5


 Negative predicted value = TN/(TN+FN) =  0.8307830783078308


 Positive Likelihood Ratio = Sensitivity/(1-Specificity) =  4.4274301
87345351


 Negative Likelihood Ratio = (1-sensitivity)/specificity =  0.5729637
361379031
```
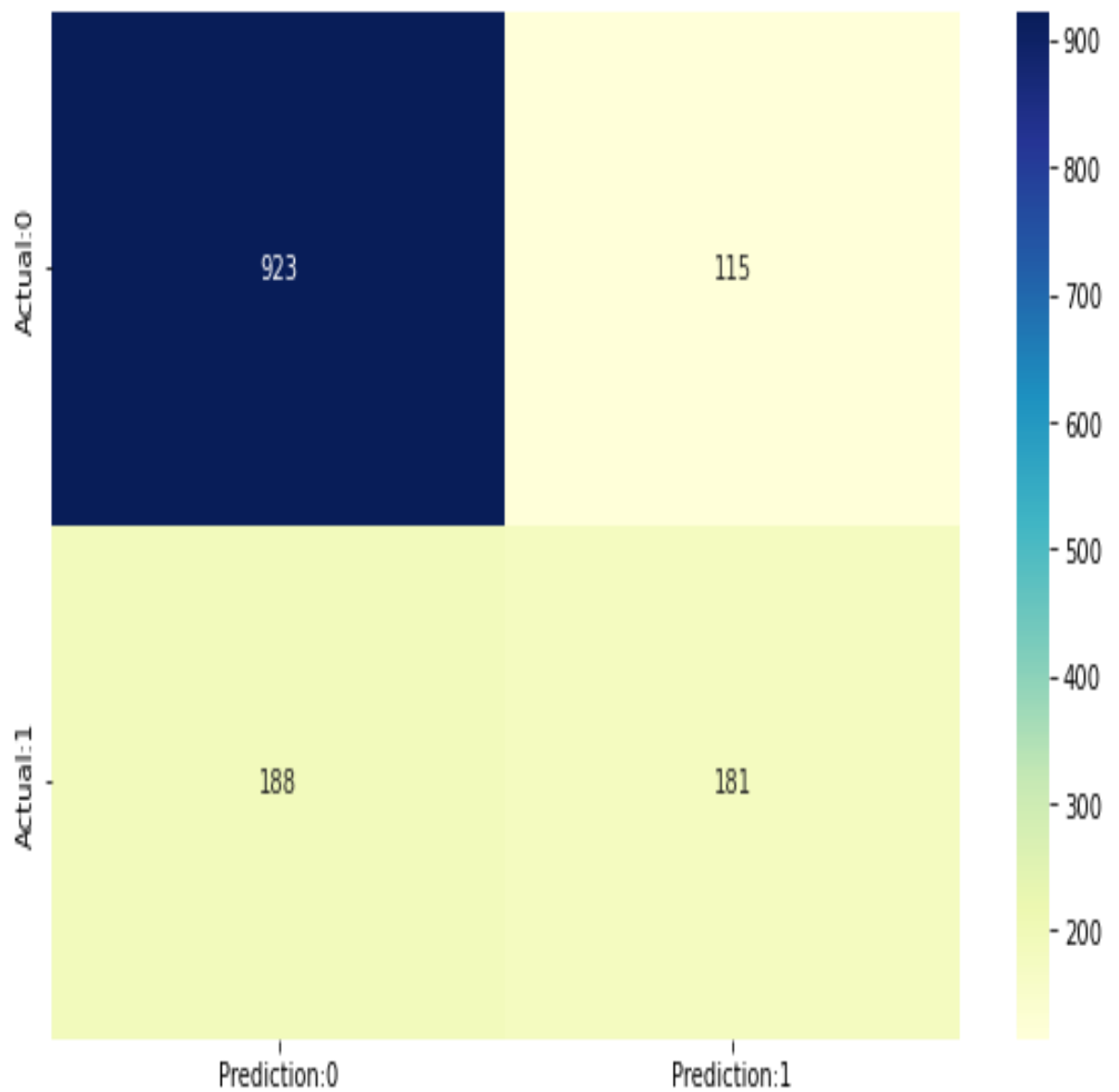
5.8.2. Logistic regresstion

```
Confusion Matrix [[942  96]
 [173 196]]
 Accuracy = TP+TN/(TP+TN+FP+FN)=  0.8088130774697939


 Missclassification = 1-Accuracy=  0.19118692253020608


 Sensitivity/Recall or True Positive rate = TP/(TP+FN)=  0.5311653116
531165
```
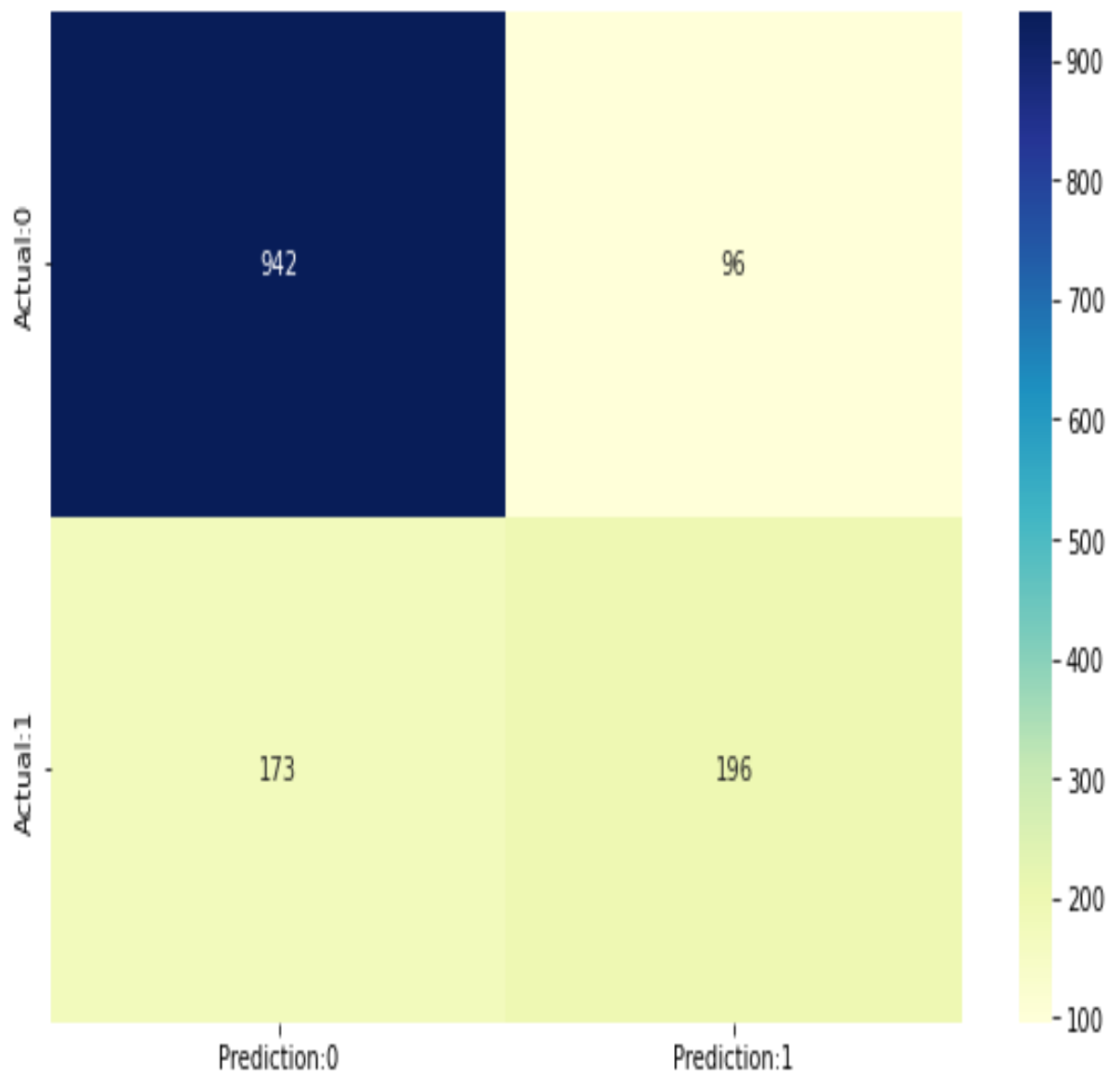
Specificity or True Negative Rate = TN/(TN+FP) = 0.9075144508670521

Precision/Positive Predictiv value = TP/(TP+FP) = 0.671232876712328
8

Negative predicted value = TN/(TN+FN) = 0.8448430493273542

Positive Likelihood Ratio = Sensitivity/(1-Specificity) = 5.7432249
32249325

Negative Likelihood Ratio = (1-sensitivity)/specificity = 0.5166140
196433812

5.8.3. K Nearest Neighbore

```
Confusion Matrix [[898 140]
 [178 191]]
 Accuracy = TP+TN/(TP+TN+FP+FN)=  0.7739872068230277


 Missclassification = 1-Accuracy=  0.22601279317697232


 Sensitivity/Recall or True Positive rate = TP/(TP+FN)=  0.5176151761
517616


 Specificity or True Negative Rate = TN/(TN+FP) =  0.8651252408477842
```
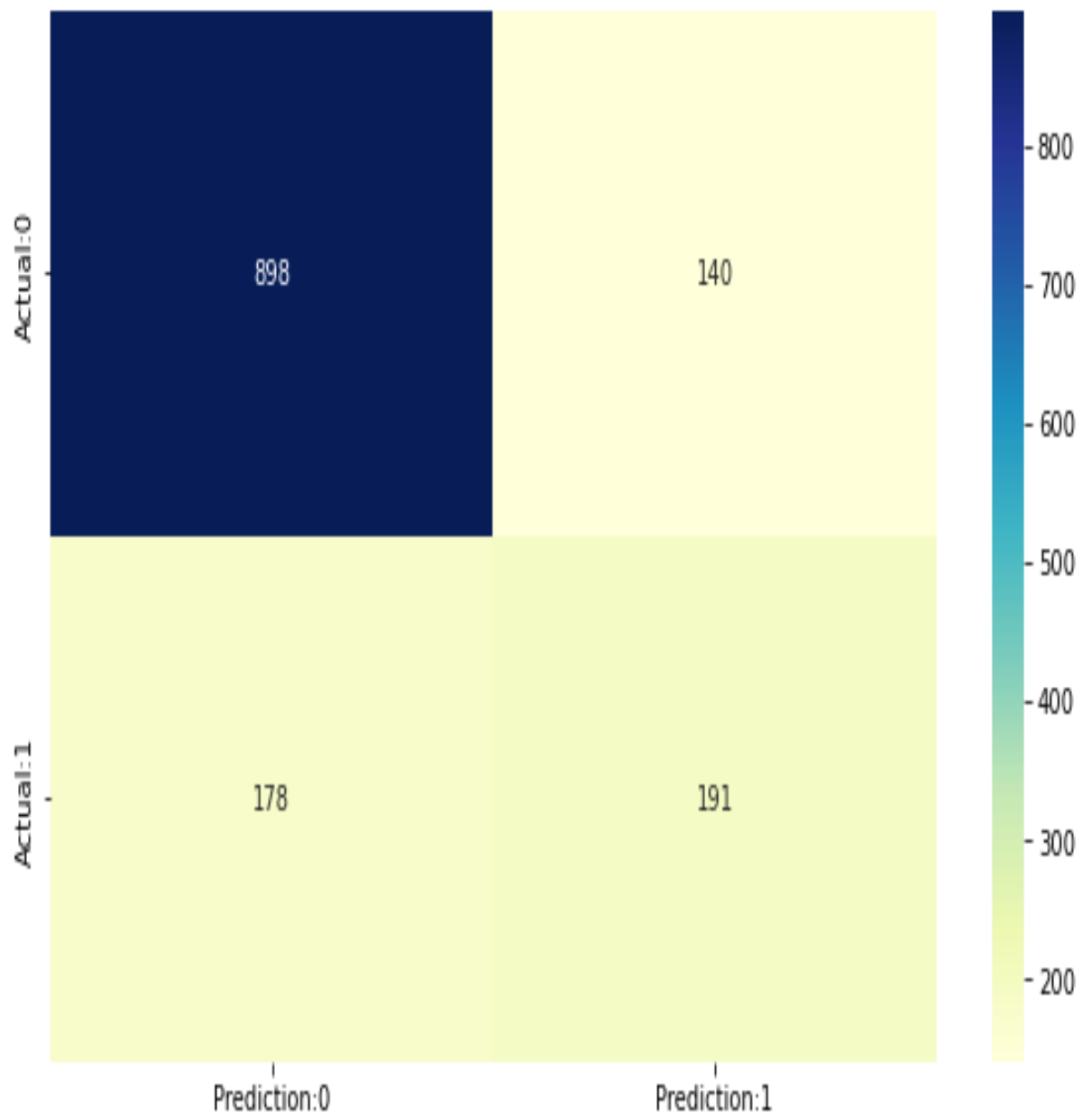
Precision/Positive Predictiv value = TP/(TP+FP) = 0.577039274924471
3

Negative predicted value = TN/(TN+FN) = 0.8345724907063197

Positive Likelihood Ratio = Sensitivity/(1-Specificity) = 3.8377468
06039489

Negative Likelihood Ratio = (1-sensitivity)/specificity = 0.5575895
84804534

5.8.4. SVM

```
Confusion Matrix [[962   76]
 [194 175]]
 Accuracy = TP+TN/(TP+TN+FP+FN)=  0.8081023454157783
```

Missclassification = 1-Accuracy= 0.1918976545842217

Sensitivity/Recall or True Positive rate = TP/(TP+FN)= 0.4742547425
474255

Specificity or True Negative Rate = TN/(TN+FP) = 0.9267822736030829

Precision/Positive Predictiv value = TP/(TP+FP) = 0.697211155378486
1

Negative predicted value = TN/(TN+FN) = 0.8321799307958477

Positive Likelihood Ratio = Sensitivity/(1-Specificity) = 6.4773213
521608906

Negative Likelihood Ratio = (1-sensitivity)/specificity = 0.5672802
258168111
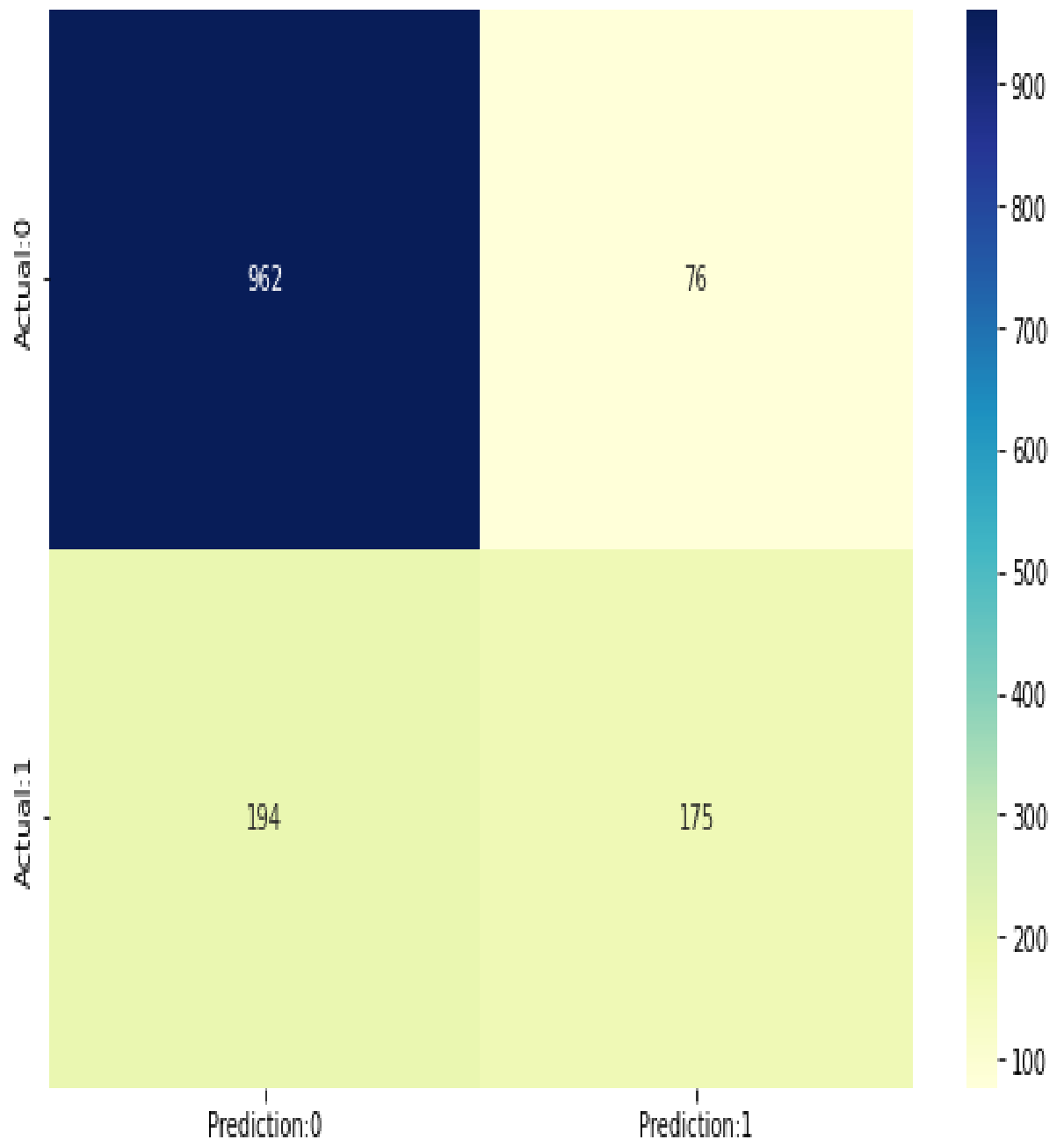
6. **Conclusion**

After analysis all the model results, I can see Random Forest has supported all the hypotheses. And model accuracy is good, after splitting the data between 75% and 25%.

Tenure: Supported by all models.

MonthlyCharges: Supported by all models.

TotalCharges: Supported by Random Forest.

Contract: Supported by Random Forest.

Extra services: Partially supported by all models.

So conclusion is customer who joined services recently they are paying more monthly and Total Charges. That's why they are leaving.

And who has contract month-to-month means small contract they are not happy that's why they left.

7. References

https://www.ibm.com/topics/business-intelligence

(BI Brief History, 2022)

[Retana et al., 2016] Retana, G., Forman, C., and Wu, D. (2016). Proactive customer education, customer retention, and demand for technology support: Evidence from a field experiment. Manufacturing and Service Operations Management, 18(1):34–50

[Balle et al., 2013] Balle, B., Casas, B., Catarineu, A., Gavaldà, R., and Manzano-Macho, D. (2013). The architecture of a churn prediction system based on stream mining. Frontiers in Artificial Intelligence and Applications, 256:157–166.

[Olaleke et al., 2014] Olaleke, O., Borishade, T., Adeniyi, S., and Omolade, O. (2014). Empirical 32 analysis of marketing mix strategy and student loyalty in education marketing. Mediterranean Journal of Social Sciences, 5(23):616–625.

[Yen and Wang, 2006] Yen, C. and Wang, H.-y. (2006). Applying data mining to telecom churn. 31:515–524.

[Zaki and Meira, 2013] Zaki, M. J. and Meira, M. J. (2013). Data Mining and Analysis: Fundamental Concepts and Algorithms.

Brownlee J (2014) Classification Accuracy is Not Enough: More Performance Measures You Can Use. Available at: https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/ (accessed 31 March 2022).

Sharma P (2019) Decoding the Confusion Matrix. Available at:
https://towardsdatascience.com/decoding-the-confusion-matrix-bb4801decbb (accessed 31
March 2022).

BlastChar (n.d.) Telco Customer Churn. Available at:
https://www.kaggle.com/datasets/blastchar/telco-customer-churn (accessed 31 March
2022).

scikit-learn (n.d.) 1.4. Support Vector Machines. Available at: https://scikit-
learn.org/stable/modules/svm.html (accessed 31 March 2022).

8. Appendix

```python
# installing the required library
get_ipython().system('conda install --yes scikit-learn')
get_ipython().system('conda install --yes nbconvert')


# In[2]:


# importing required library
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn  as sns
from sklearn.model_selection import train_test_split


# In[49]:


#from sklearn.pipeline import Pipeline
#from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
#from sklearn.pipeline import FeatureUnion
#from sklearn.base import BaseEstimator, TransformerMixin
from sklearn.linear_model import LogisticRegression
#from sklearn.metrics import classification_report
#from sklearn.metrics import roc_auc_score
#from sklearn.model_selection import StratifiedKFold
#from sklearn.model_selection import cross_val_score
from sklearn.metrics import confusion_matrix
```

```python
#from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier
#from sklearn.ensemble import GradientBoostingClassifier
from sklearn.svm import SVC


# In[4]:


# Read CSV file into DataFrame df
df = pd.read_csv('data/Telco-Customer-Churn.csv')


# In[5]:


# display the first 5 records
df.head()


# # 2. Data Analysis
#
# Let's explore and understand the data and perform Data Cleansing. The goal here is to
spot any missing values, and understand the variables. We should also fix wrong datatypes,
remove NaN values, check for duplicates. We should also add new columns where we are
going to convert yes/no values to 1 and 0.


# In[6]:


# Printing columns of the data
df.columns


# In[7]:
```

```python
# shape of data
df.shape
```

```
# Here we can see data is containing 7032 rows and 21 colums
```

```
# In[8]:
```

```python
# Information about data like name, count of not null data, datatype
df.info()
```

```
# Columns Explanation
#
# We have only 3 Numerical Variables and those are:
#
# 1) Tenure(Number of months the customer has stayed with the company)
#
# 2) MonthlyCharges(The monthly amount charged to the customer)
#
# 3) SeniorCitizen(if Customer is senior citizen then 1 else 0)
#
# 4) TotalCharges(The total amount charged to the customer)
#
# TotalCharges is srting but its stores the float value so we will change the datatype of this.
#
# All Other Variables are Categorical.
#
# Our Target Variable is Churn(Whether the customer churned or not (Yes or No)).
```

```
# In[9]:
```

```
# TotalCharges is in String Format, but it should be float
```

df["TotalCharges"] = pd.to_numeric(df["TotalCharges"],errors='coerce')

In[10]:

df.info() # After changing type of TotalCharges checking the data info again

Now we can see the type of TotalCharges changed from object to float64

In[11]:

#Analysis the numberic attributes
df.describe()

The above function is showing mathmatical calculation of numberic data.
#
count - The count of not-empty values.
mean - The average (mean) value.
std - The standard deviation.
min - The minimum value.
25% - The 25% percentile.
50% - The 50% percentile.
75% - The 75% percentile.
max - The maximum value.
#
with count we can see few data is null for TotalCharges.
#
mean is showing what is the average for columns.
#
With min we can see minimum MonthlyCharges is 18 and maximum is 118. Minimum is
showing lowest value and maximum is showing

higher value.

In[12]:

df = df.dropna() # dropping null values. Few values are missing that's why in describe function count of TotalCharges is
#7032 and minimum value of tenure is 0. So its not useful to fill the TotalCharges with some number.

In[ ]:

In[13]:

df.describe()

After dropping null values now no null values are there and minimum value of tenure is 1.

In[14]:

df.duplicated().sum() # checking duplicate values if any

Exploring Data Visualizations : To understand how variables are distributed.

In[15]:

```python
# Visualize the distribution of dataset
def distr(dist_df):
    fig = plt.subplots(figsize=(12, 13))

    for i, col in enumerate(dist_df.columns):
        plt.subplot(7, 3, i + 1)
        plt.hist(dist_df[col].values, bins=30)
        plt.title(col)
        plt.tight_layout()


# In[16]:


distr(df)


# Description about the Features:
#
# CustomerId – Customer ID
#
# Gender – Male or Female
#
# customerSeniorCitizen – Whether the customer is a senior citizen or not (1, 0)
#
# Partner – Whether the customer has a partner or not (Yes, No)
#
# Dependents – Whether the customer has dependents or not (Yes, No)
#
# Tenure – Number of months the customer has stayed with the company
#
# PhoneService – Whether the customer has a phone service or not (Yes, No)
#
```

MultipleLines – Whether the customer has multiple lines or not (Yes, No, No phone service)

#

InternetService – Customer's internet service provider (DSL, Fiber optic, No)

#

OnlineSecurity – Whether the customer has online security or not (Yes, No, No internet service)

#

OnlineBackup – Whether the customer has online backup or not (Yes, No, No internet service)

#

DeviceProtection – Whether the customer has device protection or not (Yes, No, No internet service)

#

TechSupport – Whether the customer has tech support or not (Yes, No, No internet service)

#

StreamingTv – Whether the customer has streaming TV or not (Yes, No, No internet service)

#

StreamingMovies – Whether the customer has streaming movies or not (Yes, No, No internet service)

#

Contract – The contract term of the customer (Month-to-month, One year, Two year)

#

PaperlessBilling – Whether the customer has paperless billing or not (Yes, No)

#

PaymentMethod – The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))

#

MonthlyCharges – The monthly charge amount

#

TotalCharges – The total amount charged to the customer

```python
#
# Churn – Whether the customer churned or not (Yes or No).(Need to predict)


# In[17]:


Churn = df['Churn'].value_counts() # counting churn values
Churn


# Here we can see count of churn and non-churn. 0 is representing count of non-churn and
# 1 is count of churn customer.
#
#    non-churn : 5163
#    churn    : 1869


# In[18]:


# plotting the pie chart for churn and non churn count
Churn.plot(kind = 'pie', title = 'Churn', labels = Churn, figsize = (6,6))
plt.legend()
plt.show()


# # To group data by Churn and compute the mean to find out if churners make more
# Monthly Charge than non-churners:


# In[19]:


# Group data by 'Churn' and compute the mean
print(df.groupby('Churn')['MonthlyCharges','TotalCharges','tenure'].median())
(df.groupby('Churn')['MonthlyCharges','TotalCharges','tenure'].median()).plot(kind = 'box'
```

,subplots=True, title = 'Avg value of Churn based on Charges and tenure', figsize = (16,6))

plt.show()

Churning customers have higher monthly charges with a median of ca. 80 USD and much lower interquartile range compared to
that of non-churners (median of ca. 65 USD).
#
TotalCharges are the result of tenure and MonthlyCharges, which are more insightful on an individual basis.
#
Churning customers have much lower tenure with a mean of ca. 10 months compared to a median of non-churners of ca. 38 months.
#
Hence! In above result we can see, churners seem to pay more MonthlyCharges than non-churners.

#

Counting Churn customer based on different columns:

In[20]:

# Counting Churn data based on SeniorCitizen
print(pd.crosstab(df.SeniorCitizen,df.Churn,margins=True))

In above result, we can see 5890 customer was not Senior who was using. and out of 5890, 1393 is churn.
and in opposite site 1142 was SeniorCitien and 476 is churned.

```python
# In[21]:


# Counting Churn data based on SeniorCitizen
print(pd.crosstab(df.gender,df.Churn,margins=True))


# In above result, female and male for both gender count is approx same.


# In[22]:


def bar(x,y):
    result = (pd.crosstab(df[x], df[y], normalize='columns')*100).plot(kind="bar",
                            title= '% of Churn based on '+ x ,stacked=True, rot=0,figsize =
(9,6))
    for c in result.containers:

        # set the bar label
        result.bar_label(c, label_type='center')


# In[23]:


# % of Churn data based on gender
bar('gender','Churn')


# In[24]:


# % of Churn data based on SeniorCitizen
bar('SeniorCitizen','Churn')
```

In above result, non-senior citizens churn % is much lower than senior churn %.

In[25]:

# % of Churn data based on Partner
bar('Partner','Churn')

In above result, Churn rate for customers without partners is higher than Partner

In[26]:

# % of Churn data based on Dependents
bar('Dependents','Churn')

In above result, churn rate for customers without children is very high.
In[27]:

# % of Churn data based on Contract
bar('Contract','Churn')

In above result, Churn % for month-to-month contracts much higher than with remaining contracts, then one year contract is higher than Two year

In[28]:

# % of Churn data based on PaymentMethod
bar('PaymentMethod','Churn')

In above result,electronic check paymentMethod shows much higher churn rate, then mailed check. Bank transfer and credit card is appox same.

In[29]:

# % of Churn data based on InternetService
bar('InternetService','Churn')

In above result, Customers with fiber optic InternetService have much higher churn rate.It is approx. 69%

In[30]:

Add new col "ExtraService" by summing up the number of remaining services.
df['ExtraService'] = (df[['OnlineSecurity', 'DeviceProtection', 'StreamingMovies', 'TechSupport',
                'StreamingTV', 'OnlineBackup']] == 'Yes').sum(axis=1)

In[31]:

# Counting Churn data based on ExtraService
result = pd.crosstab(df.ExtraService,df.Churn).plot(kind="bar",
                        title= 'Count of Churn based on ExtraService ' ,stacked=True,
rot=0,figsize = (9,6))
for c in result.containers:

    # set the bar label
    result.bar_label(c, label_type='center')

In above result, Customer who is using less / no extra services they have higher churn rate as comapre who is using more than 3 extra services.

Cleaning data

In[32]:

```
#droping customerid
df = df.drop(columns = 'customerID')
```

In[33]:

```
def label_encoding(features, df):
    for i in features:
        df[i] = df[i].map({'Yes': 1, 'No': 0})
    return
```

In[34]:

```
#Converting categorical Columns to Numberical using label encoding

label_encoding(['Partner', 'Dependents', 'Churn', 'PhoneService', 'PaperlessBilling'], df)
df['gender'] = df['gender'].map({'Female': 1, 'Male': 0})
df
```

In[35]:

```
# One-Hot-Encoding for identified columns.
OHE_feature = ['MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup',
'DeviceProtection',
```

'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract', 'PaymentMethod', 'ExtraService']

```python
df = pd.get_dummies(df, columns=OHE_feature)
```

In[36]:

```python
df.info()
```

In[37]:

```python
# Min-Max-Scaling for identified columns.
from sklearn.preprocessing import MinMaxScaler

features_mms = ['tenure', 'MonthlyCharges', 'TotalCharges']
df_features_mms = pd.DataFrame(df, columns=features_mms)
df_remaining_features = df.drop(columns=features_mms)

mms = MinMaxScaler()
rescaled_features = mms.fit_transform(df_features_mms)

df_rescaled_features = pd.DataFrame(rescaled_features, columns=features_mms,
index=df_remaining_features.index)

df = pd.concat([df_remaining_features, df_rescaled_features], axis=1)
```

In[38]:

```python
#Corelation between all col
```

```
plt.figure(figsize = (30,30))
corr_matrix = df.corr() # Corelation Matrix
sns.heatmap(corr_matrix, cmap="coolwarm")
plt.show()
```

In[39]:

```
#Corelation with churn
df.corr()['Churn'].sort_values(ascending=False)
```

In[40]:

```
#To Create Training and Test sets
X = df.drop("Churn", axis = 1)
y = df.Churn
```

Splitting the dataset into the Training and Test sets

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
                              test_size = 0.25,
                              random_state = 0)
```

In[41]:

```
y_test.shape
```

In[42]:

```python
y_train.shape


# In[43]:


X_train.shape


# In[44]:


X_test.shape


# In[45]:


# Feature Scaling

sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)


# # Random Forest

# In[52]:


# Instantiate the classifier
from sklearn.metrics import accuracy_score
clf = RandomForestClassifier()

# Fit to the training data
clf.fit(X_train, y_train)
```

```python
# In[53]:


# Predict the labels for the test set
y_pred = clf.predict(X_test)
feature_imp = clf.feature_importances_
# Compute accuracy and Evaluating Model Performance
accuracy_score(y_test, y_pred)


# In[ ]:


def imp_features(feature_imp,X):

    print("Important Feature", feature_imp)
    columns = X.columns
    index = np.arange(len(columns))
    font_size = 1
    fontsize_tittle = 25

    plt.figure(figsize=(100, 17))
    plt.bar(index, sorted(feature_imp), width=0.5,
            alpha=0.4,
            color='r',
            label='')

    plt.xlabel('Columns', fontsize=font_size)
    plt.ylabel('Feature Importance', fontsize=font_size)
    plt.title('Feature Importance for each column', fontsize=fontsize_tittle)
    plt.xticks(index, columns)
    plt.show()
```

```
# In[ ]:


# Plotting important features
imp_features(feature_imp,X)


# In[ ]:


#Confusion Matrix
def c_matrix(y_test, y_pred):
    cm = confusion_matrix(y_test, y_pred)
    print("Confusion Matrix" , cm)
    conf_matrix =
pd.DataFrame(data=cm,columns=['Prediction:0','Prediction:1'],index=['Actual:0','Actual:1'
])
    plt.figure(figsize=(10,6))
    sns.heatmap(conf_matrix,annot=True,fmt='d',cmap='YlGnBu')


    #Other classification statistics
    TN=cm[0,0] #True Negative
    TP=cm[1,1] #True Positive
    FN=cm[1,0] #False Negative
    FP=cm[0,1] #False Positive
    sensitivity = TP/float(TP+FN)
    specificity = TN/float(TN+FP)

    print(' Accuracy = TP+TN/(TP+TN+FP+FN)= ', (TP+TN)/float(TP+TN+FP+FN) ,
'\n\n',
        'Missclassification = 1-Accuracy= ', 1- ((TP+TN)/float(TP+TN+FP+FN)),'\n\n',
        'Sensitivity/Recall or True Positive rate = TP/(TP+FN)= ', TP/float(TP+FN), '\n\n',
        'Specificity or True Negative Rate = TN/(TN+FP) = ', TN/float(TN+FP), '\n\n',
        'Precision/Positive Predictiv value = TP/(TP+FP) = ', TP/float(TP+FP), '\n\n',
```

'Negative predicted value = TN/(TN+FN) = ', TN/float(TN+FN), '\n\n',

'Positive Likelihood Ratio = Sensitivity/(1-Specificity) = ',sensitivity/float(1-specificity), '\n\n',

'Negative Likelihood Ratio = (1-sensitivity)/specificity = ',float(1-sensitivity)/specificity)

In[ ]:

c_matrix(y_test, y_pred)

K Nearest Neighbors

In[ ]:

Instantiate the classifier

```
clf= KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2 )
# Fit to the training data
clf.fit(x_train, y_train)
```

In[ ]:

#Predicting the test set result

```
y_pred= clf.predict(x_test)
# Compute accuracy and Evaluating Model Performance
```

accuracy_score(y_test, y_pred)

```python
# In[ ]:


#Confusion Matrix
c_matrix(y_test, y_pred)


# # Logistic Regression


# In[54]:


# Instantiate the classifier
clf = LogisticRegression(max_iter=1000)


# Fit to the training data
clf.fit(X_train, y_train)
print(print(clf.coef_))
a = np.std(X_train, 0)*clf.coef_
importance = clf.coef_[0]
print(dict(zip(X.columns,clf.coef_[0])))
#importance is a list so you can plot it.
feat_importances = pd.Series(importance)
feat_importances.nlargest(20).plot(kind='bar',title = 'Feature Importance')
print(df.columns)


# In[55]:


# Plotting important features
imp_features(importance,X)


# In[ ]:
```

```python
# Predict the labels for the test set
y_pred = clf.predict(X_test)


# Compute accuracy and Evaluating Model Performance
accuracy_score(y_test, y_pred)


# In[ ]:




# In[ ]:



#Confusion Matrix
c_matrix(y_test, y_pred)


# # SVM


# In[ ]:



# Instantiate the classifier
clf = SVC(gamma='auto')


# Fit to the training data
clf.fit(X_train, y_train)


# In[ ]:
```

```python
#Predicting the test set result
y_pred= clf.predict(x_test)
# Compute accuracy and Evaluating Model Performance


accuracy_score(y_test, y_pred)


# In[ ]:


#Confusion Matrix
c_matrix(y_test, y_pred)
```