A Project report on

Customer Churn Prediction

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

G.LAVANYA	**164G1A0548**
S.BHARGAVI	**164G1A0514**
S.AYESHA BEGUM	**164G1A0508**
K.KRANTHI KUMAR	**164G1A0541**

Under the Guidance of

Mrs. M.SOUMYA, M. Tech
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY
ANANTHAPURAMU
(Accredited by NAAC with 'A' Grade & Accredited by NBA, Affiliated to JNTUA, Approved by AICTE, New Delhi)

2019-2020

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY: ANANTAPUR

(Accredited by NAAC with 'A' Grade, Affiliated to JNUTA, Approved by AICTIE, New Delhi)

Rotarypuram Village, B K Samudram Mandal, Ananthapuramu-515701

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(B.Tech program accredited by NBA)

Certificate

This is to certify that a seminar report entitled CUSTOMER CHURN PREDICTION is the bonafide work carried out by **G.LAVANYA** bearing Roll Number **164G1A0548, S.BHARGAVI** bearing Roll Number **164G1A0514, S.AYESHA BEGUM** bearing Roll Number **164G1A0508, K.KRANTHI KUMAR** bearing Roll Number **164G1A0541** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year2019-2020.

Signature of the Guide

Mrs.M.Soumya,M.Tech.,

Assistant Professor

Head of the Department

Dr.G.K.V.Narashima Reddy, Ph.D,

Professor

Date:

Place: Rotarypuram

EXTERNAL EXAMINAR

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express my gratitude for all of them.

It is with immense pleasure that we would like to express my indebted gratitude to my Guide **Mrs.M.Soumya**, M.Tech., **Assistant Professor**, **Computer Science & Engineering**, who has guided us a lot and encouraged us in every step of the project work. We thank her for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep felt gratitude to **Mr. R. Sandeep Kumar**, M.Tech., **Assistant Professor,** project coordinator valuable guidance and unstinting encouragement enabled us to accomplish our project successfully in time.

We are very much thankful to **Dr. G. K. V. NARASIMHA REDDY**, Ph.D**, Professor and Head of Department, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey my special thanks to **Dr. T. Hitendra Sarma**, Ph.D, **Principal, Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to management for providing excellent facilities.

Finally, we wish to convey our gratitude to our families who fostered all the requirements and facilities that we need.

DECLARATION

We, Ms. G.Lavanya having reg no: 164g1a0548, Ms. S.Bhargavi having reg no: 164g1a0514, Ms. S.Ayesha Begum having reg no: 164g1a0508, Mr. K.Kranthi Kumar having reg no: 164g1a0541 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram, hereby declare that dissertation entitled ”CUSTOMER CHURN PREDICITON” embodies the report of our project work carried out by us during IV year Bachelor of Technology in Mrs. M.Soumya.,M.Tech., Department of CSE, SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY,ANANTAPUR and this work has been submitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

G.LAVANYA	Reg no: 164G1A0548
S.BHARGAVI	Reg no: 164G1A0514
S.AYESHA BEGUM	Reg no: 164G1A0508
K.KRANTHI KUMAR	Reg no: 164G1A0541

LIST OF CONTENTS

ABSTRACT

New Telecom Companies provide various schemes and services to attract various customers to switch from competitor's service to their service because may be customers are not happy with old schemes and services. Therefore the main goal is for those companies to retain their existing customers. Hence it is necessary or those companies to know in advance which customer may switch from their service to their competitor's service. So the goal of our project is to analyze and predict which customers might be going churn or not by using a dataset of customers. with help of these telecom companies to know in advance which customer may switch from their service to their competitor's service. So they can do something for the customers and retain their customers in advance. It will avoid the companies from facing big losses.

Then, we proposed that analysis of different machine learning techniques on a dataset. Classify the customers into churn and non-churn using accurate classifiers and provide the factors behind the churning of customers in the telecom sector. Feature selection is performed by using information gain and correlation attribute ranking filter. By knowing the significant churn factors from customers data, CRM can improve productivity, recommend relevant promotions to the churn customers and excessively improve marketing campaigns of the company.

LIST OF FIGURES

ABBREVIATIONS

CRM	Customer Relationship Management
QOS	Quality of Service
SIEDS	Systems and Information Engineering Design Symposium
SaaS	Software as a service
ROC	Receiving operating characteristics
SRS	Software Requirement Specification
DFD	Data Flow Diagram
GUI	Graphical User Interface
ML	Machine Learning
MATLAB	Matrix Laboratory
UML	Unified Modeling Language
SVM	Support Vector Machine
RBF	Radial Basis Function
KNN	K nearest neighbor
ADABoost	Adaptive boosting
XGboost	Extreme Gradient
LightGBM	Light Gradient Boosting Machine
EDA	Exploratory Data Analysis
TP	True Positives
FP	False Positives
TN	True Negatives
FN	False Negatives
AUC	Area Under the Curve

CHAPTER-1

INTRODUCTION

In the present world, a huge volume of data is being generated by telecom companies at an exceedingly fast rate. There is a range of telecom service providers competing in the market to increase their client share. Customers have multiple options in the form of better and less expensive services. The ultimate goal of telecom companies is to maximize their profit and stay alive in a competitive market place. A customer churn happens when a vast percentage of clients are not satisfied.

It results in service migration of customers who start switching to other service providers. There are many reasons for churning. Unlike postpaid customers, prepaid customers are not bound to a service provider and may churn at any time. Churning also impacts the overall reputation of a company which results in its brand loss. A loyal customer, who generates high revenue for the company, gets rarely affected by the competitor companies. Such customers maximize the profit of a company by referring it to their friends, family members and colleagues. Telecom companies consider policy shifts when the number of customers drops below a certain level which may result in a huge loss of revenue .Churn prediction is vital in the telecom sector as telecom operators have to retain their valuable customers and enhance their Customer Relationship Management (CRM) administration . The most challenging job for CRM is to retain existing customers . Due to the saturated and competitive market, customers have the option to switch to other service providers. Telecom companies have developed procedures to identify and retain their customers as it is less expensive than attracting the new ones.This is due to the cost involved in advertisements, workforce, and concessions which can scale up to almost five to six times than retaining existing customers .

1.1 Project Overview

In the telecom industry, customers are able to choose from multiple service providers and actively switch from one operator to another. In this highly competitive market, the telecommunications industry experiences an average of 15-25% annual churn rate. Given the fact that it costs 5-10 times more to acquire a new customer than to retain an existing one, customer retention has now become even more important than customer acquisition.To reduce customer churn, telecom companies need to predict which customers are at high risk of churn. In this project, you will analyse customer-level data of a leading telecom firm, Information Gain and Correlation Attributes Ranking Filter feature selection techniques and select the top features to form both results. In this study, we proposed a churn prediction model that uses various machine learning algorithms.We identified the factors behind the churning of customers by using the rules proposed churn prediction model is evaluated using information retrieval metrics.

The accuracy is calculated for the churn prediction model using TP rate, FP rate, Precision, Recall, F-measure and ROC area. The objective of the study is to investigate the existing techniques in machine learning and data mining and to propose a model for customer churn predictions, to identify churning factors and to provide retention strategies. From the experiments, we observed that our proposed model performed better in terms of classification of churners by achieving high accuracy.

Our contribution to this study is to propose a churn prediction model. The important features are selected using feature selection techniques such as information gain and correlation attribute ranking filter. We used a number of machine learning techniques for churn and non-churn classification on two large datasets of the telecom sector. We performed customer profiling based on the behaviour of customers into groups.

1.2 Objectives

- Here we are using the past data of the customers which includes all the details of the customer.
- This model is then to identify how much probability of the customer going to churn from the service.
- Our model given that probability rate of a customer and reduced the churn rate

Thus the project includes training a model in such a way that it can be used to give the probability of a customer by using machine learning algorithms. Here we also find factors affecting customers leaving the service and customer profiling using K Means unsupervised algorithm.

CHAPTER 2
LITERATURE SURVEY

2.1 Introduction

1.Machine-Learning Techniques for Customer Retention: A Comparative Study ,Sahar F. Sabbeh

Nowadays, customers have become more interested in the quality of service (QoS) that organizations can provide them. Services provided by different vendors are not highly distinguished which increases competition between organizations to maintain and increase their QoS. Customer Relationship Management systems are used to enable organizations to acquire new customers, establish a continuous relationship with them and increase customer retention for more profitability. CRM systems use machine-learning models to analyze customers’ personal and behavioral data to give organizations a competitive advantage by increasing customer retention rate. Those models can predict customers who are expected to churn and reasons of churn. Predictions are used to design targeted marketing plans and service offers. This paper tries to compare and analyze the performance of different machine-learning techniques that are used for churn prediction problems.

2. Yizhe Ge ; Shan He ; Jingyue Xiong ; Donald E. Brown, "Customer churn analysis for a software-as-a-service company", In the proceedings of Systems and Information Engineering Design Symposium (SIEDS)

SaaS companies generate revenues by charging recurring subscription fees for using their software services. The fast growth of SaaS companies is usually accompanied with huge upfront costs in marketing expenses targeted at their potential customers. Customer retention is a critical issue for SaaS companies because it takes twelve months on average to break-even with the expenses for a single customer. This study describes a methodology for helping SaaS companies manage their customer relationships. We investigated the time-dependent software feature usage data, for example, login numbers and comment numbers, to predict whether a customer would churn within the next three months. Our study compared model performance across classification algorithms. The prediction model yielded the best results for identifying the most important software usage features and for classifying customers as either churn type or non-risky type.

3. IRFAN ULLAH1, BASIT RAZA, AHMAD KAMRAN MALIK , " Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector":

In the telecom sector, a huge volume of data is being generated on a daily basis due to a vast client base. Decision makers and business analysts emphasized that attaining new customers iscostlierthan retaining the existing ones .Business analysts and customer relationship management(CRM)analyzer need to know the reasons for churn customers, as well as, behavior patterns from the existing churn customers' data.This paper proposes a churn prediction model that uses classification,as well as,clustering techniques to identify the churn customers and provides the factors behind the churning of customers in the telecom sector. Feature selection is performed by using information gain and correlation attribute ranking filter. The proposed model first classifies customer data using classification algorithms .

Creating Effective retention policies is an essential task of the CRM to prevent churners. After classification, the proposed model segments the churning customer's data by categorizing the churn customers in groups using cosine similarity to provide group-based retention offers. This paper also identified churn factors that are essential in determining the root causes of churn. By knowing the significant churn factors from customers' data, CRM can improve productivity, recommend relevant promotions to the group of likely churn customers based on similar behavior patterns, and excessively improve marketing campaigns of the company.

2.2 Existing System :

Decision trees, the most popular predictive models, is a tree graph presenting the variables' relationships. Used to solve classification and prediction problems, decision tree models are represented and evaluated in a top-down way.

The two phases to develop decision trees are tree building and tree pruning. Starting from the root node representing a feature to be classified, a decision tree is built. Selecting a feature can be done by evaluating its information gain ratio. The lower level nodes are then constructed in a similar way to the divide and conquer strategy. Improving predictive accuracy and reducing complexity, pruning process is applied on decision trees to produce a smaller tree and guarantee a better generalisation by removing branches containing the largest estimated error rate.The decision about a given case regarding to which of the two classes it belongs is thus made by moving from the root node to all leaves.

2.3 Disadvantages of Existing System

1. Complex decision trees are very hard to be visualised and interpreted.
2. It suffers from the lack of robustness and over-sensitivity to training data sets
3. A small change in the data can cause a large change in the structure of the decision tree causing instability.
4. For a Decision tree sometimes calculation can go far more complex compared to other algorithms.
5. Decision trees often involve higher time to train the model.
6. Decision tree training is relatively expensive as complexity and time taken is more.

2.4 Proposed System

Proposes a churn prediction model that uses classification,as well as,clustering techniques to identify the churn customers and provides the factors behind the churning of customers in the telecom sector. Feature selection is performed by using information gain and correlation attribute ranking filters.Apply different machine learning techniques on telecom dataset and compare accuracy of different machine learning classify the data with more accurate classifier. After classification, the proposed model segments the churning customer's data by categorizing the churn customers in groups using k-means algorithm to provide group-based retention offers.By knowing the significant churn factors from customers' data, CRM can improve productivity, recommend relevant promotions to the group of likely churn customers based on similar behavior patterns, and excessively improve marketing campaigns of the company. The proposed churn prediction model is evaluated using metrics, such as accuracy, precision, recall, f-measure, and receiving operating characteristics (ROC) area.

CHAPTER 3
FEASIBILITY STUDY

Feasibility analysis is an assessment of the practicality of a proposed plan or method. And it reduces the development risks. The major areas considered in feasibility analysis are as follows.

The feasibility study concerns the considerations made to verify whether the system is to fit to be developed in all terms. Once an idea to develop software is put forward the question that arises first will pertain to the feasibility aspects. It involves developing and understanding of the selected program.

This documentation presents the results of an independent study of the feasibility of completing the training models for churn analytics. It enhances the probability of success by addressing and mitigating factors early on that could affect the project. There are different aspects in the feasibility study.
- Technical Feasibility - Operational Feasibility - Social Feasibility

3.1 Technical Feasibility

The technical capability of the personnel as well as the capability of the available technology should be considered. The technical aspects we have included are suitable for the modern environment and the technological tools which we have used to train and run our model are quite preferable.

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed designs of the system, making it difficult to access issues like performance, costs etc. A number of issues have to be considered while doing a technical analysis. Understand the different technologies involved in the proposed system before commencing the project, we have to be clear about the technologies that are to be required for the development of the system.

This project involves python as a programming language and jupyter notebook as a platform environment for analyzing and manipulating the machine learning algorithms and for easy running of the python code.

3.2 Operational Feasibility

Proposed projects are beneficial only if they can be turned into information systems that will meet the organization operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed.It determines if the human resources are available to operate once it has been executed. It focuses on the complexity of the problem and checks if the given solution will solve the problem. Our solution to train the model for predicting the churn customers and profiling based on their behaviour analysis. The classifiers that are used for implementing the project will sustain and make the problem to be solved to some extent.

Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible. The operations of handling the imbalanced datasets, applying confusion matrix and analyzing feature importance operations have been organized feasibly.

3.3 Social Feasibility

Social feasibility is one of the feasibility studies where the level of acceptance of the people is considered regarding the project. This involves the feasibility of the proposed solution to generate economic and social benefits. As we are dealing with the customer churn data, economically this solution will help telecom companies to retain their existing customers and more benefits since we are trying to detect the churn rate by providing the models to predict future trends. It describes the effect on users from the introduction of the new system considering whether there will be a need for retraining the workforce. It describes how you propose to ensure user cooperation before changes are introduced.

CHAPTER-4

REQUIREMENTS

4.1 Software Requirement Specification

Software Requirement Specification (SRS) is the starting point of the software development activity. It is a complete description of the behaviour of a system which is to be developed. The SRS document enlists all necessary requirements for project development. To derive the requirements we need to have a clear and thorough understanding of the product which is to be developed. This is prepared after detailed communication with the project team and the customer. A SRS is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform. An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real world situations

Characteristics of SRS:

Correct – An SRS is correct if, and only if, every requirement stated therein is one that the software shall meet. Traceability makes this procedure easier and less prone to error.

Unambiguous – An SRS is unambiguous if, and only if, every requirement stated therein has only one interpretation. As a minimum, this requires that each characteristic of the final product be described using a single unique term.

Verifiable – It is verifiable if there exists some finite cost-effective process with which a person or machine checks whether a software product meets requirements.

Consistent – Consistency refers to internal consistency. If an SRS does not agree with some higher level document, such as a system requirements specification, then it is not correct. An SRS is internally consistent if, and only if, no subset of individual requirements described in its conflict.

Modifiable – SRS is said to be modifiable if its structure and style are such that any changes to the requirements can be made easily, completely and consistently while retaining the structure and style.

Traceable – SRS is said to be traceable if the origin of each of its requirements is clear and it facilitates the referencing of each requirement in future enhancement.

Stability – SRS is ranked for importance or stability if each requirement in it has an identifier to indicate either the importance or stability of that particular requirement.

4.1.1 User Requirements

The software requirements specification is produced at the culmination of the analysis task. The function and performance allocated to the software as a part of system engineering and refined by establishing a complex information description, detailed functional and behavioural description, and indication of performance requirements and design constraints, appropriate validation criteria and other data pertinent to requirements. The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigation from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

4.2 Hardware Requirements

RAM : 4 or 8 GB

Hard disk : 1 Tb

4.3 Software Requirements

Operating Systems – Windows, Linux, Mac

Platform – Jupyter Notebook 5.7.4

Navigator – Anaconda Navigator 1.9.6

Programming in - Python

4.4 Software Model

The system development life cycle concept applies to a range of hardware and software configurations. The stages on the cycle include analysis, design, development, implementation, testing, documentation and evaluation. In the case of machine learning and data science related concepts, agile methodologies have become the go-to approach.

A central element of these methodologies is iterative development. In agile, solutions and requirements are constantly evolving based on new understandings, input and feedback. In agile, a feature driven development framework organizes software development around making processes based on features. The majority of the effort on an feature driven development project is comprised of few steps:

- Design by Feature
- Build by Feature
- Plan by Feature

In the below we have a data flow diagram which represent the how the can process the data and give the output

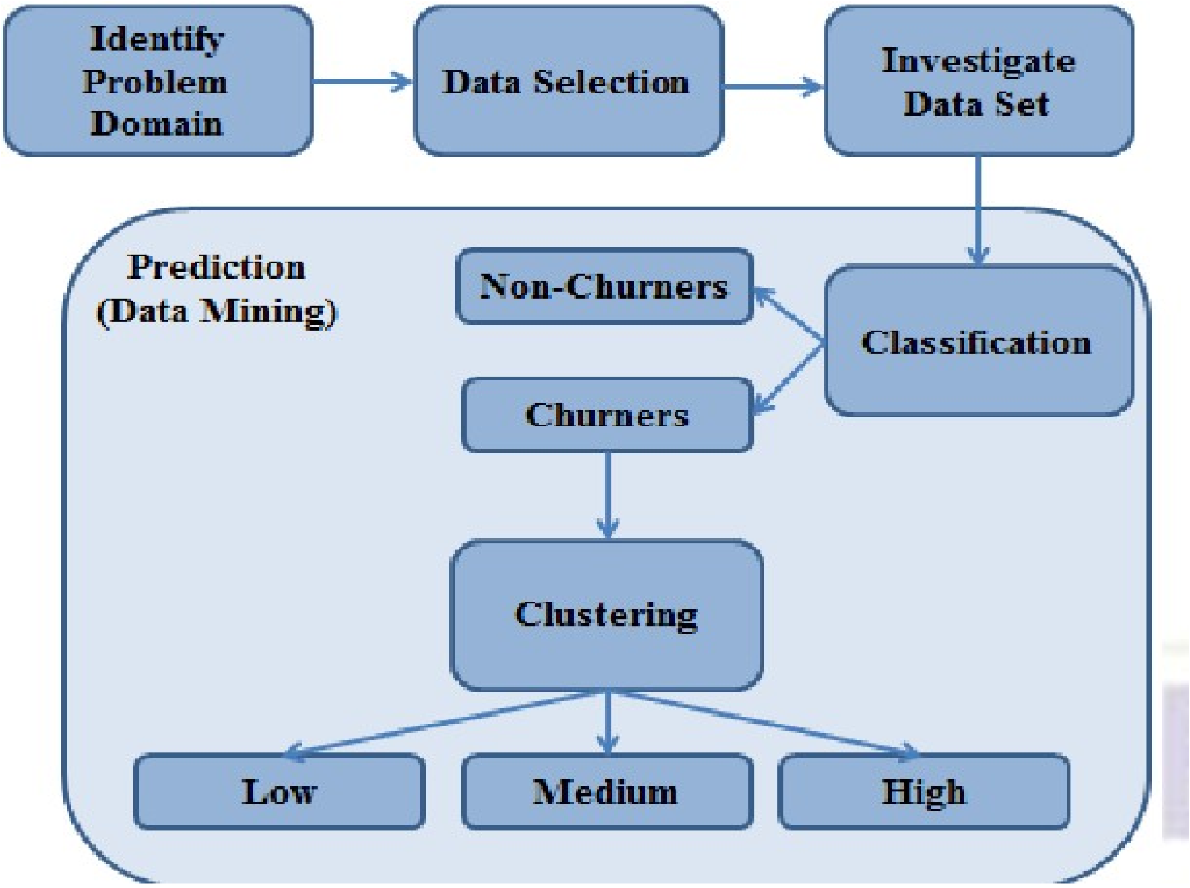


Figure 4.1 : Data Flow diagram

CHAPTER - 5

TECHNOLOGY

5.1 Language Used

The programming language that was used in this fraud detection project is Python. The implementation of source code was done through python. Python is an interpreted, interactive, object-oriented programming language which is suitable for implementing machine learning algorithms in an easier way.

5.1.1 Applications of Python

Here, we are specifying application areas where python can be applied.

- Web Applications
- Desktop GUI Applications
- Software Development
- Scientific and Numeric
- Business Applications
- Console Based Application
- Audio or video based Applications
- 3D CAD Applications
- Enterprise Applications
- Applications for Images

5.1.2 Features of Python

Python which is a developer-friendly and high level programming language provides some of the features.

- Easy to Learn and Use
- Expressive Language
- Interpreted Language
- Cross-platform Language
- Free and Open Source
- Object-Oriented Language
- Extensible
- Large Standard Library

- GUI Programming Support
- Integrated

5.2 Machine Learning Algorithms

Machine Learning plays a key role in many scientific disciplines and its applications are part of our daily life. It is used for example to filter spam email, for weather prediction, in medical diagnosis, product recommendation, face detection, fraud detection, etc. Machine Learning studies the problem of learning, which can be defined as the problem of acquiring knowledge through experience.

This process typically involves observing a phenomenon and constructing a hypothesis on that phenomenon that will allow one to make predictions or, more in general, to take rational actions. For computers, the experience or the phenomenon to learn is given by the data, hence we can define ML as the process of extracting knowledge from data.

Machine learning is closely related to the fields of Statistics, Pattern Recognition and Data Mining. At the same time, it emerges as a subfield of computer science and gives special attention to the algorithmic part of the knowledge extraction process. In summary, the focus of ML is on algorithms that are able to learn automatically the patterns hidden in the data.

5.2.1 Supervised Learning

This project is about supervised learning, where ML algorithms are trained on some annotated data (the training set) to build predictive models, or learners, which will enable us to predict the output of new unseen observations. It is called supervised because the learning process is done under the supervision of an output variable, in contrast with unsupervised learning where the response variable is not available.

Supervised learning assumes the availability of labeled samples, i.e. observations annotated with their output, which can be used to train a learner. In the training set we can distinguish between input features and an output variable that is assumed to be dependent on the inputs.

The output, or response variable, defines the class of observations and the input features are the set of variables that have some influence on the output and are used to predict the value of the response variable. Depending on the type of output variable we can distinguish between two types of supervised task:

- classification (Logistic Regression, Random Forest)
- regression

The first assumes a categorical output, while the latter a continuous one. Fraud detection belongs to the first type since observations are transactions that can be either genuine or fraudulent, while in other problems such as stock price predictions the response is a continuous variable.

On the other hand, in both classification and regression tasks, input features can include both quantitative and qualitative variables. In a classification problem an algorithm is assessed on its overall accuracy to predict the correct classes of new unseen observation.

Supervised learning as the name indicates a presence of supervisor as teacher. Basically supervised learning is a learning in which we teach or train the machine using data which is well labelled that means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples (data) so that a supervised learning algorithm analyses the training data (set of training examples) and produces a correct outcome from labelled data.

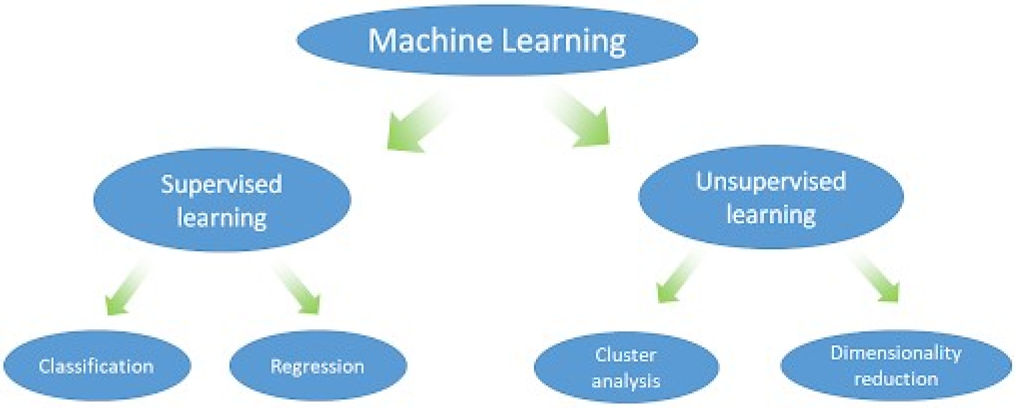


Figure 5.1 Machine Learning Classification

For instance, suppose you are given a basket filled with different kinds of fruits. Now the first step is to train the machine with all different fruits one by one like this:

- If the shape of the object is a long curving cylinder having color Green-Yellow then it will be labeled as Banana.
- If the shape of the object is rounded and depressed at top having color Red then it will be labelled as Apple.

Since machines have already learnt the things from previous data and this time they have to use it wisely. It will first classify the fruit with its shape and color, and would confirm the fruit name as banana and put it in Banana category. Thus the machine learns the things from training data (basket containing fruits) and then apply the knowledge to test data(new fruit).

5.2.2 Unsupervised Learning

Unsupervised learning is the training of machines using information that is neither classified nor labelled and allowing the algorithm to act on that information without guidance. Here the task of the machine is to group unsorted information according to similarities, patterns and differences without any prior training of data.

Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore machines are restricted to find the hidden structure in unlabeled data by our-self. For instance, suppose it is given an image of both dogs and cats which have not ever been seen. This machine has no idea about the features of dogs and cats so we can’t categorize it in dogs and cats. But it can categorize them according to their similarities, patterns and differences i.e., we can easily categorize the above picture into two parts. First may contain all pictures having dogs in it and second part may contain all pictures having cats in it.

5.3 Libraries Used

Python is increasingly being used as a scientific language. Matrix and vector manipulations are extremely important for scientific computations. Both NumPy and Pandas have emerged to be essential libraries for any scientific computation, including machine learning, in python due to their intuitive syntax and high-performance matrix computation capabilities.

Figure 5.2: Python Libraries

5.3.1. Numpy

NumPy stands for ‘Numerical Python’ or ‘Numeric Python’. It is an open source module of Python which provides fast mathematical computation on arrays and matrices. Since arrays and matrices are an essential part of the Machine Learning ecosystem, NumPy along with Machine Learning modules like Scikit-learn, Pandas, Matplotlib, TensorFlow, etc. complete the Python Machine Learning Ecosystem.

NumPy provides the essential multi-dimensional array-oriented computing functionalities designed for high-level mathematical functions and scientific computation. Numpy can be imported into the notebook using import numpy as np

NumPy’s main object is the homogeneous multidimensional array. It is a table with same type elements, i.e., integers or strings or characters (homogeneous), usually integers. In NumPy, dimensions are called axes. The number of axes is called the rank.

Some of the important attributes of a NumPy object are:

- Ndim: displays the dimension of the array
- Shape: returns a tuple of integers indicating the size of the array
- Size: returns the total number of elements in the NumPy array
- Dtype: returns the type of elements in the array, i.e., int64, character
- Itemsize: returns the size in bytes of each item
- Reshape: Reshapes the NumPy array

In python, a vector can be represented in many ways, the simplest being a regular python list of numbers. Since Machine Learning requires lots of scientific calculations, it is much better to use NumPy's array, which provides a lot of convenient and optimized implementations of essential mathematical operations on vectors.

Vectorized operations perform faster than matrix manipulation operations performed using loops in python. For example, to carry out a 100 * 100 matrix multiplication, vector operations using NumPy are two orders of magnitude faster than performing it using loops.

5.3.2 Pandas

Similar to NumPy, Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table objects called Dataframe. It is like a spreadsheet with column names and row labels.

Hence, with 2d tables, pandas is capable of providing many additional functionalities like creating pivot tables, computing columns based on other columns and plotting graphs. Pandas can be imported into Python using:

import pandas as pd

Pandas Series object is created using the pd.series function. Each row is provided with an index and by defaults is assigned numerical values starting from 0. Like NumPy, Pandas also provide the basic mathematical functionalities like addition, subtraction and conditional operations and broadcasting.

Pandas dataframe object represents a spreadsheet with cell values, column names, and row index labels. Dataframe can be visualized as dictionaries of Series. Dataframe rows and columns are simple and intuitive to access. Pandas also provide SQL-like functionality to filter, sort rows based on conditions.

- Matplotlib is a 2d plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments. Matplotlib can be used in Python scripts, Python and IPython shell, Jupyter Notebook, web application servers and GUI toolkits. Matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Majority of plotting commands in pyplot have MATLAB analogs with similar arguments.

- Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

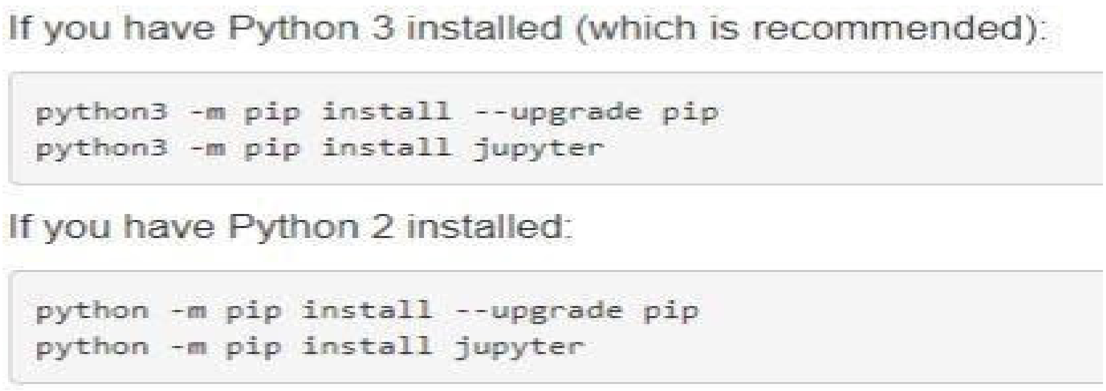
Keys Features

- Seaborn is a statistical plotting library
- It has beautiful default styles
- It also is designed to work very well with Pandas data frame objects.
- Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machines, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

5.4 Jupyter Notebook Installation

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

The installation of Jupyter notebook can be done in two ways. One is with Anaconda Navigator and the other is using commands in python.



If you have Python 3 installed (which is recommended):

```
python3 -m pip install --upgrade pip
python3 -m pip install jupyter
```

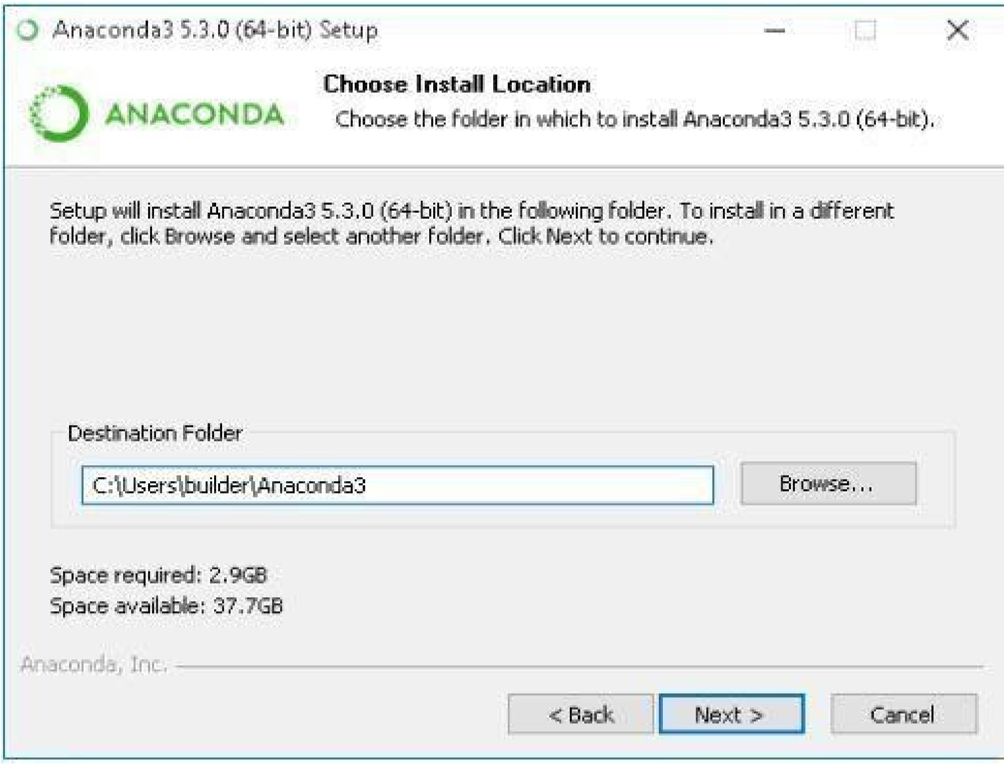
If you have Python 2 installed:

```
python -m pip install --upgrade pip
python -m pip install jupyter
```

Figure 5.3 : Jupyter Installation Commands

The other way of installing the Jupyter notebook is by using Anaconda Navigator. Anaconda Navigator is a desktop graphical user interface included in Anaconda that allows you to launch applications and easily manage packages, environments and channels without the need to use command line commands. The following are some of the steps of the installation process.

- Download the Anaconda installer.
- Double click the installer to launch. [To prevent permission errors, do not launch the installer from the Favorites folder and if any encounter issues during installation occurs, temporarily disable anti-virus software during install then re-enable it after the installation concludes].
- Click Next.
- Read the licensing terms and click “I Agree”.
- Select the destination folder to install Anaconda and click the Next button.[Do not install as administrator unless admin privileges are required].



Choose Install Location

Choose the folder in which to install Anaconda3 5.3.0 (64-bit).

Setup will install Anaconda3 5.3.0 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

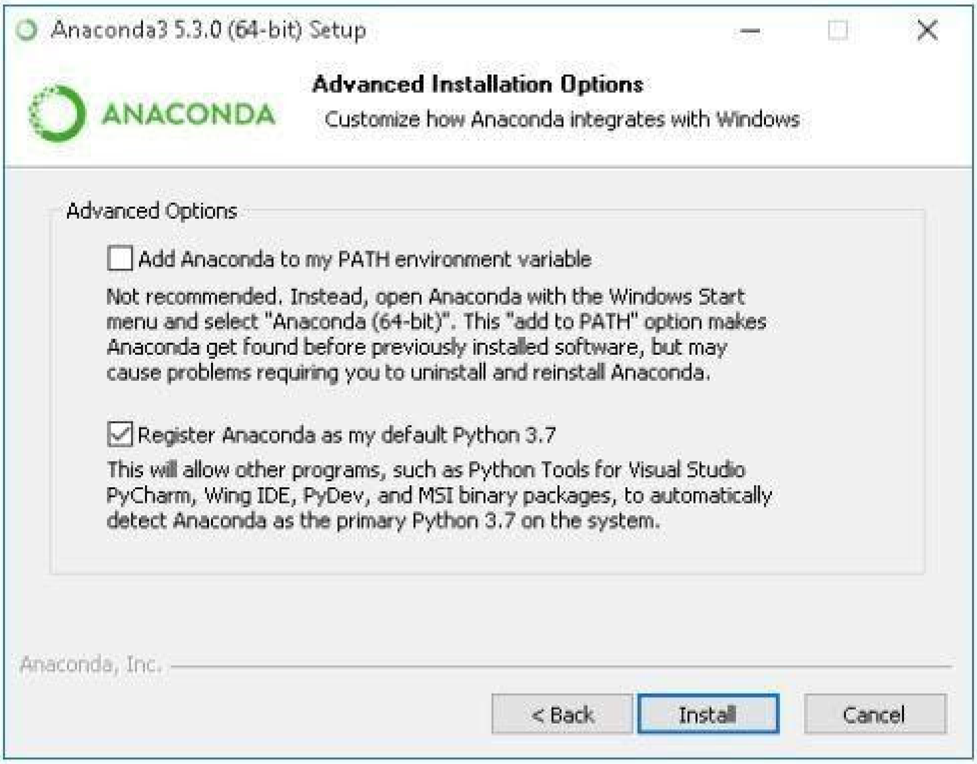
Destination Folder: C:\Users\builder\Anaconda3

Space required: 2.9GB
Space available: 37.7GB

Figure 5.4 : Anaconda Installation

- Choose whether to add Anaconda to the PATH environment variable. It is recommended not adding Anaconda to the PATH environment variable, since this can interfere with other software. Instead, use

Anaconda software by opening Anaconda Navigator or prompt from the start menu.



Anaconda3 5.3.0 (64-bit) Setup

Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

□ Add Anaconda to my PATH environment variable

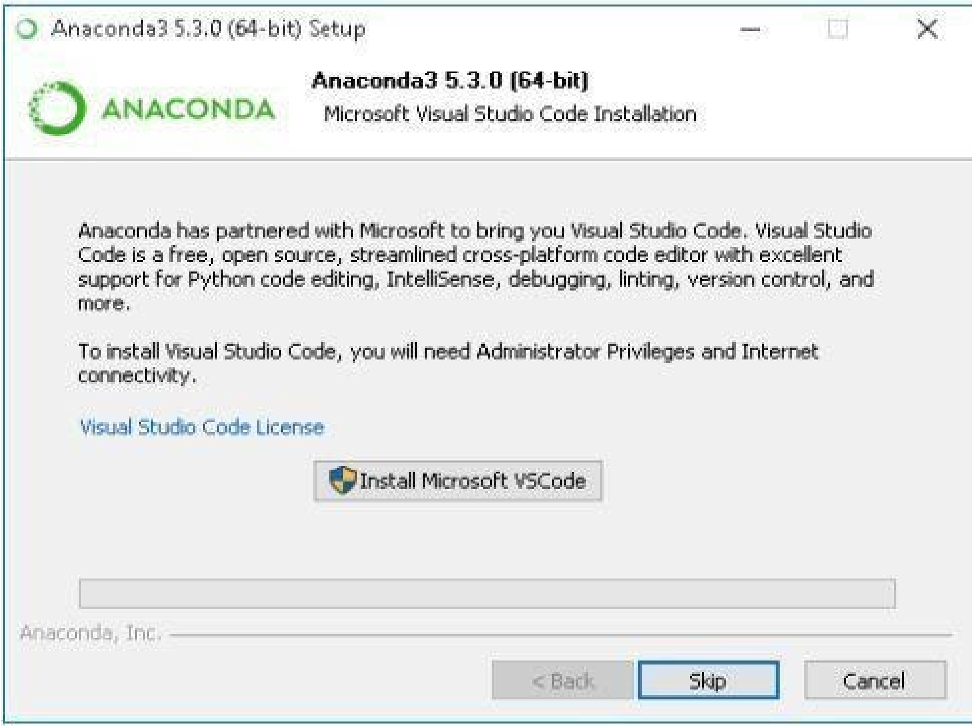
Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☑ Register Anaconda as my default Python 3.7

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.7 on the system.

Figure 5.5: Path Environment Variable

- Choose whether to register Anaconda as default Python. Unless planning on installing and running multiple versions of Anaconda or multiple versions of python, accept the default and leave the box checked.



Anaconda3 5.3.0 (64-bit) Setup

Anaconda3 5.3.0 (64-bit)

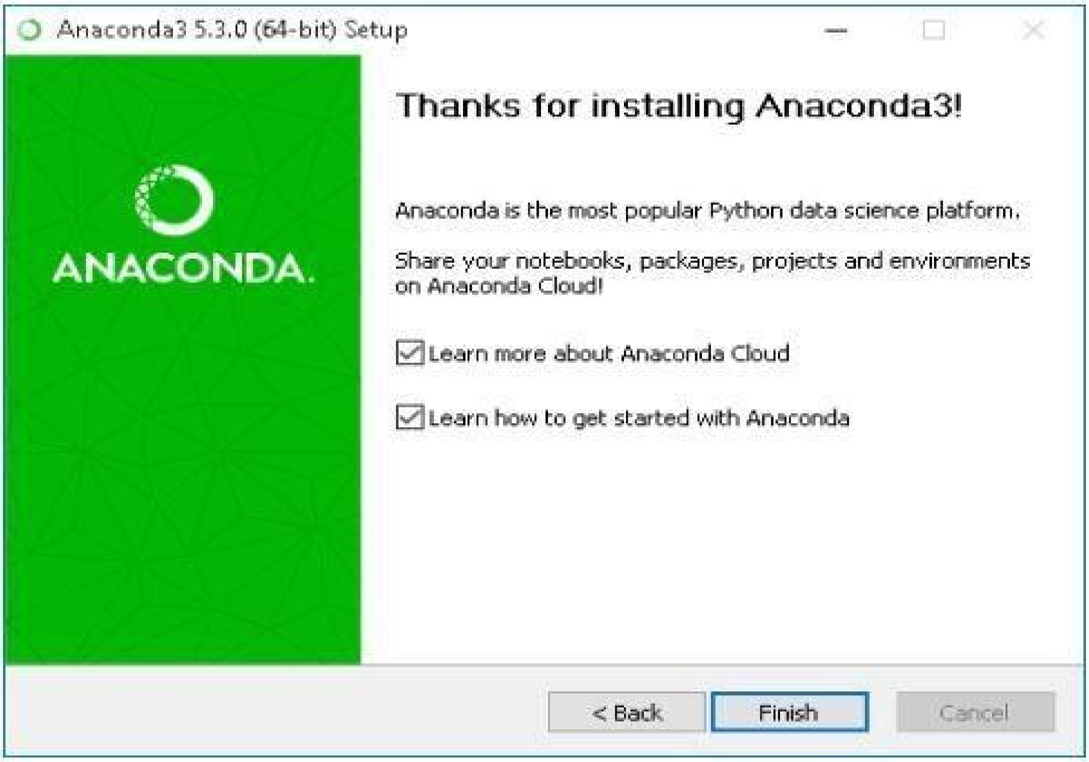
Microsoft Visual Studio Code Installation

Anaconda has partnered with Microsoft to bring you Visual Studio Code. Visual Studio Code is a free, open source, streamlined cross-platform code editor with excellent support for Python code editing, IntelliSense, debugging, linting, version control, and more.

To install Visual Studio Code, you will need Administrator Privileges and Internet connectivity.

Figure 5.6 : Installation SetUp

After successful installation, “Thanks for installing Anaconda “dialog box isdisplayed.



Anaconda3 5.3.0 (64-bit) Setup

Thanks for installing Anaconda3!

Anaconda is the most popular Python data science platform.

Share your notebooks, packages, projects and environments on Anaconda Cloud!

☑ Learn more about Anaconda Cloud

☑ Learn how to get started with Anaconda

Figure 5.7 : Finish Dialog Box

- If it is needed to study more details about Anaconda, use the boxes shown in the above figure. After completing the installation, verify it from the start menu in the homepage.

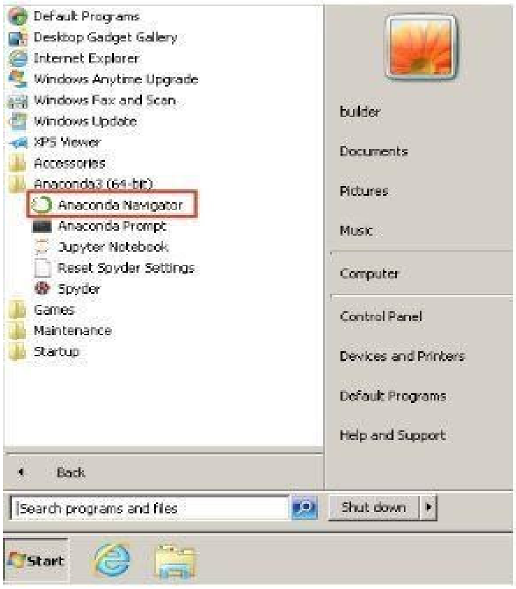


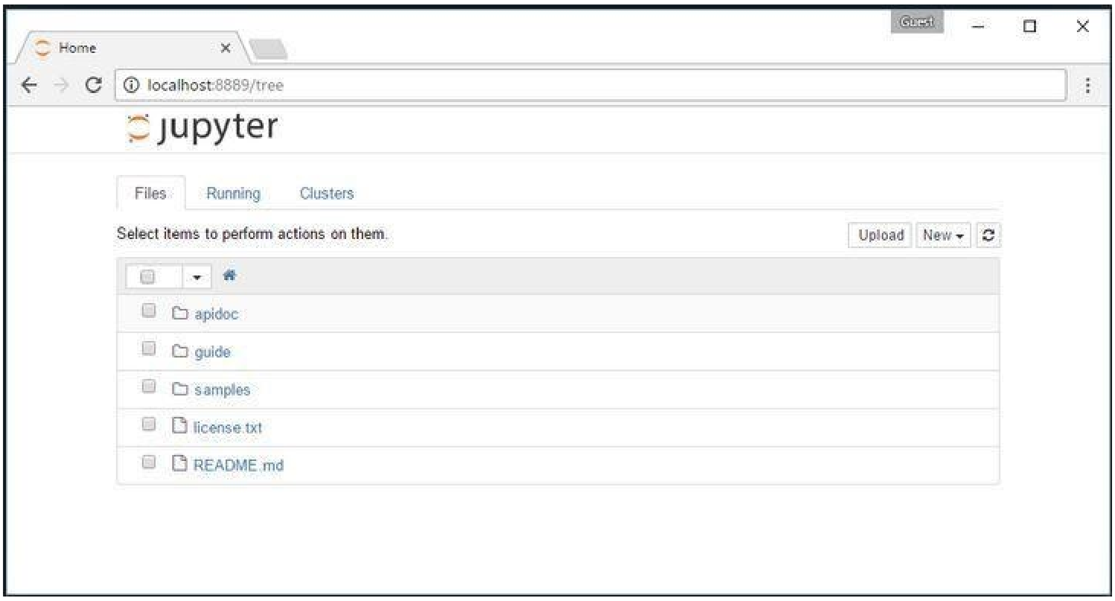
Figure 5.8 : Home Page

- The below image can be seen after clicking.

Figure 5.9 : Launching Anaconda

- Now Anaconda Navigator is open.

After clicking the launch option of Jupyter notebook, the home page of it is opened through localhost in the browser.



localhost:8889/tree

Select items to perform actions on them.

Figure 5.10 : Home Page Of Jupyter NoteBook

Thus Jupyter notebook can be successfully installed.

CHAPTER-6
DESIGN

6.1 UML Introduction

unified modeling language allows the software engineer to express an analysis model using the modeling The notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective.

UML is specifically constructed through two different domains, they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the systems.
- UML Design modeling, this focuses on the behavioral modeling, implementation modeling and environmental model views.

6.2 Usage of UML in Project

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time to the market. These techniques include component technology, visual programming, patterns and frameworks. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The UML was designed to respond to these needs.

Simply, systems design refers to the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements which can be done easily through UML diagrams.

6.3 Data Flow Diagram

Data flow diagram will describe the flow of the project in a clear manner. The data flow diagrams (DFD) are used for modeling the requirements and it focuses on flow of the data but not order of the data.DFD can be represented in the form of levels.

Level-0

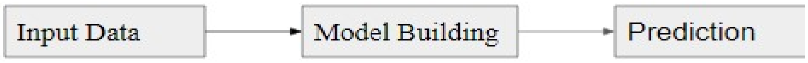


Fig 6.1: DFD Level-0

In this level, there will be the highest abstraction of data. Only source, process and output are provided at this level. Datasets can be obtained as:

- Download from various websites
- Generate own datasets
- Consider random data points
- Use previous datasets from different projects

Level-1

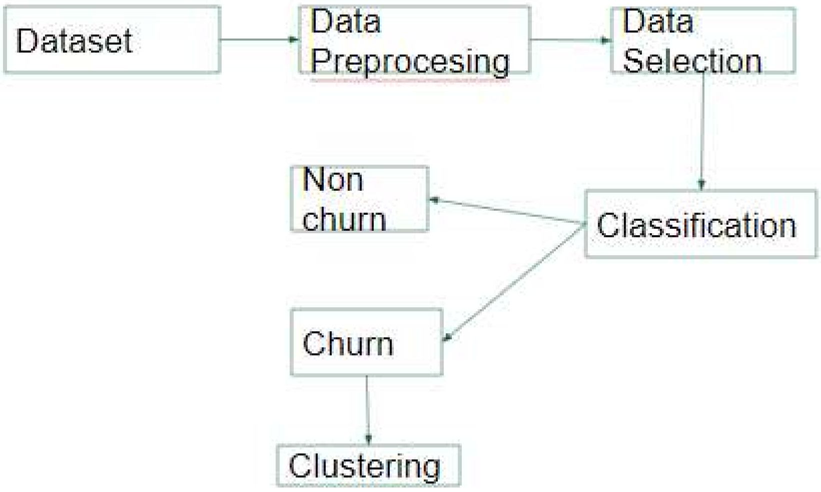


Figure 6.2: DFD Level-1

In this level, the flow diagram will explain a more detailed view of the project. In which it consist of step by step process

Initially obtain the datasets from any websites or generate own datasets which is a bit harder task. After obtaining the datasets, perform data transformation to it in such a way that there shouldn’t be any integration problem or any redundancy issue.

Now, the actual task begins. By using machine learning techniques, apply the classification algorithms which come under supervised machine learning. Select the classifiers that are to be applied to the project and apply them to the obtained dataset. Applying the classifiers to the dataset actually means that you need to train the model with the classifiers and test the data so that the model will be fit.Thus after the model is completed, it provides some metrics that describe the performance of each classifier that we have used above. Through those metrics, we can compare and select the best classifier that performs well only for the given input dataset. The best classifier can be varied when the dataset is changed. It is not a generalized best classifier, but specific to the given dataset.

The next step is to clean the datasets from duplicates and redundancy. Cleaning each and every tuple will eliminate the above issues. Now transform the data points by normalizing them to a certain scaling range.

Since the existing systems provided the model with any one of the classifiers, our project is to consider at least more than one classifier so that we will provide better comparison with those classifiers for the given dataset.Thus UML diagrams will clearly depict the scope and steps of the process of the project. These techniques include component technology, visual programming, patterns and frameworks.

Steps involved in Design

1. Loading the dataset
2. Exploratory data Analysis
    a. Treating null and missing values
    b. Treating Correlated variables
3. Feature Selection
4. Data Splitting in to train and test datasets
5. Fitting data on different algorithms
6. Classification report on each algorithm
7. Clustering on churn data finding behaviour analysis
8. Each step has its own specific reason and plays a prominent role in building up a model of the project.

CHAPTER 7

IMPLEMENTATION

7.1 Data Exploration

I will use the Telecom Customer Churn dataset, which is available at http://www.kaggle.com. The dataset provides 7043 customers information in 21 columns.We have both numerical and categorical type of information in this dataset. I am planning to use 25% of data for testing purposes and 75% of data for training purposes.

The data set includes information about:

- Customers who left within the last month — the column is called Churn
- 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

Input Data:

CustomerID: customerID

gender: Customer gender (female, male)

SeniorCitizen: 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: Customers 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)

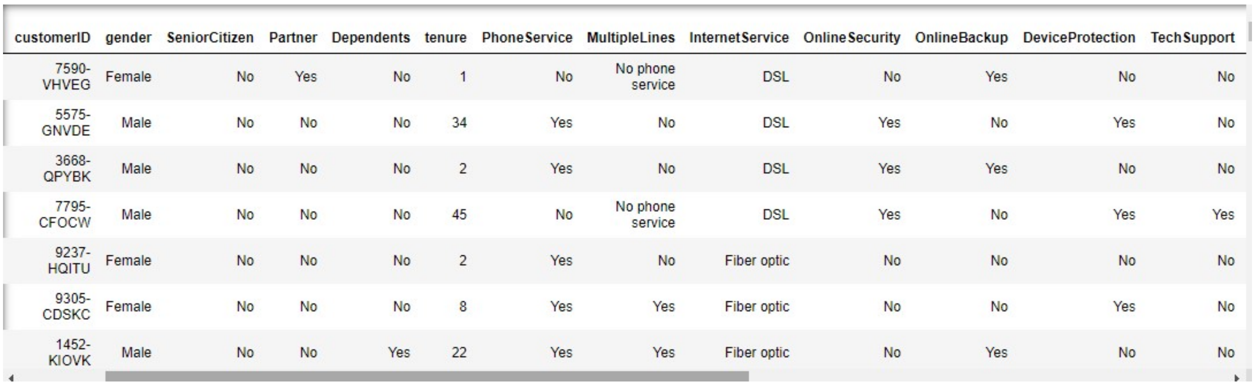
PaymentMethod: The customers payment method (Electronic check, Mailed check, Bank transfer(automatic), Credit card (automatic)

MonthlyCharges: The amount charged to the customer monthly

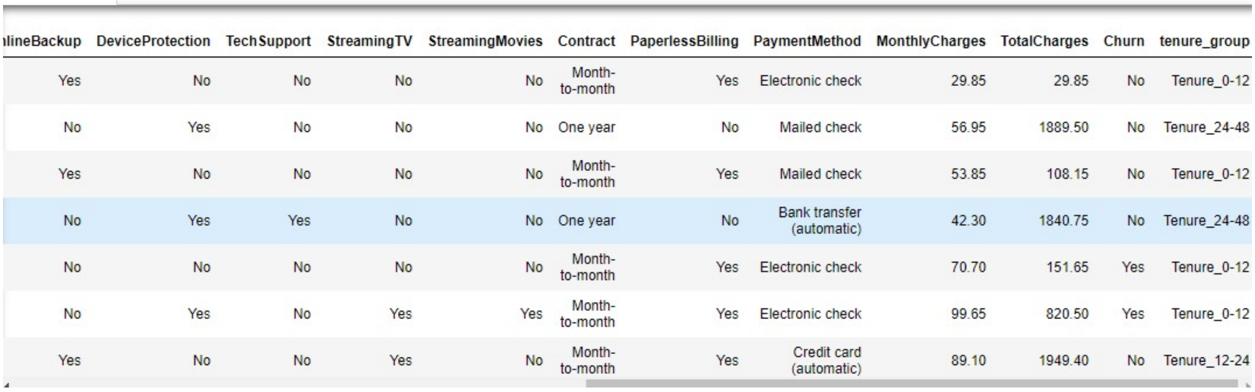
TotalCharges: The total amount charged to the customer

Output Data:

Churn: Yes for the customers that left the company and No for the customers that stayed with company



customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport
7590-VHVEG	Female	No	Yes	No	1	No	No phone service	DSL	No	Yes	No	No
5575-GNVDE	Male	No	No	No	34	Yes	No	DSL	Yes	No	Yes	No
3668-QPYBK	Male	No	No	No	2	Yes	No	DSL	Yes	Yes	No	No
7795-CFOCW	Male	No	No	No	45	No	No phone service	DSL	Yes	No	Yes	Yes
9237-HQITU	Female	No	No	No	2	Yes	No	Fiber optic	No	No	No	No
9305-CDSKC	Female	No	No	No	8	Yes	Yes	Fiber optic	No	No	Yes	No
1452-KIOVK	Male	No	No	Yes	22	Yes	Yes	Fiber optic	No	Yes	No	No



lineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn	tenure_group
Yes	No	No	No	No	Month-to-month	Yes	Electronic check	29.85	29.85	No	Tenure_0-12
No	Yes	No	No	No	One year	No	Mailed check	56.95	1889.50	No	Tenure_24-48
Yes	No	No	No	No	Month-to-month	Yes	Mailed check	53.85	108.15	No	Tenure_0-12
No	Yes	Yes	No	No	One year	No	Bank transfer (automatic)	42.30	1840.75	No	Tenure_24-48
No	No	No	No	No	Month-to-month	Yes	Electronic check	70.70	151.65	Yes	Tenure_0-12
No	Yes	No	Yes	Yes	Month-to-month	Yes	Electronic check	99.65	820.50	Yes	Tenure_0-12
Yes	No	No	Yes	No	Month-to-month	Yes	Credit card (automatic)	89.10	1949.40	No	Tenure_12-24

Figure 7.1: Telecom customer data

7.2.DataPreprocessing:

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put it in a formatted way. So for this, we use data preprocessing tasks.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

Handling Missing data:

The next step of data preprocessing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.

Ways to handle missing data:

There are mainly two ways to handle missing data, which are:

By deleting the row:

The first way is used to commonly deal with null values. In this way, we just delete the specific row or column which consists of null values. But this way is not so efficient and removing data may lead to loss of information which will not give the accurate output.

By calculating the mean:

In this way, we will calculate the mean of that column or row which contains any missing value and will put it in the place of missing value. This strategy is useful for the features which have numeric data such as age, salary, year, etc.

Encoding Categorical data:

Categorical data is data which has some categories such as, in our dataset; there are two categorical variables, **device Protection**, and **contract** .

Since the machine learning model completely works on mathematics and numbers, but if our dataset would have a categorical variable, then it may create trouble while building the model. So it is necessary to encode these categorical variables into numbers.

Firstly, we will convert the categorical data into numbers. So to do this, we will use the **LabelEncoder()** class from the preprocessing library.we have imported the LabelEncoder class of **sklearn library.** This class has successfully encoded the variables into digits.

Dummy Variables:

Dummy variables are those variables which have values 0 or 1. The 1 value gives the presence of that variable in a particular column, and rest variables become 0. With dummy encoding, we will have a number of columns equal to the number of categories.

In our dataset, we have columns which are having more than one unique value .If column have 3 categories so it will produce three columns having 0 and 1 values. For Dummy Encoding, we will use the fit_transform class of the preprocessing library.

Feature Scaling:

Feature scaling is the final step of data preprocessing in machine learning. It is a technique to standardize the independent variables of the dataset in a specific range. In feature scaling, we put our variables in the same range and in the same scale so that no variable dominates the other variable.

For feature scaling, we will import **StandardScaler** class of **sklearn.preprocessing** library as:

```
from sklearn.preprocessing import StandardScaler
```

7.3 Feature Selection:

Feature selection is a crucial step for selecting the relevant features from a dataset based on domain knowledge. A number of techniques exist in the literature for feature selection in the context of churn predictions.In This study, we used Information Gain and Correlation Attributes Ranking Filter techniques for feature selection.

Information gain:

Assess the dependency of the independent variable in predicting the target variable. In other words, it determines the ability of the independent feature to predict the target variable.

Correlation Attribute:

Correlation is a well-known similarity measure between two features. If two features are linearly dependent, then their correlation coefficient is ±1. If the features are uncorrelated, the correlation coefficient is 0. The association between the features is found out by using the correlation method. There are two broad categories that can be used to measure the correlation between two random variables. One is based on classical linear correlation and the other is based on information theory. Out of these two, the most familiar measure is linear correlation coefficient

7.4 Modelling

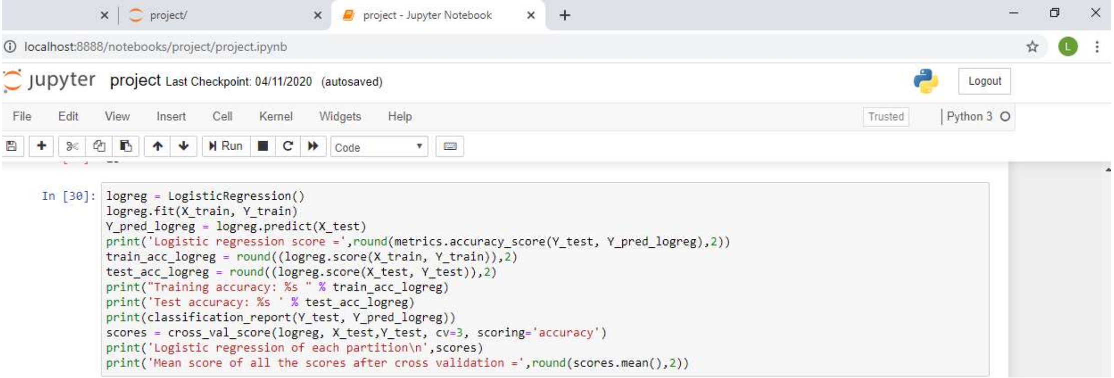
In this phase, the pre-processed data obtained was used to build the machine learning model to predict customer churn. The main objective of the research was to build supervised machine learning models to do a comparative study and to find the best model for prediction. From the best predictive model we find factors behind the customer leaving the services of the company. On the churn customers data we implement the clustering to find the common behavior analysis.We used different classification algoritms in this project. They are:

- Logistic Regression
- Random Forest Classifier
- Support Vector Machine
- K-nearest neighbour
- Gradient Boost
- Ada Boost
- XGboost
- LightGBM

7.4.1 Logistic Regression:

Firstly, the Logistic Regression model was built. It is the most preferred algorithm for modelling binary dependent variables. It is a type of probability statistical classification model mainly used for classification problems. The technique can work well with a different combination of variables and can help in predicting the customer churn with higher accuracy. The predictive power of the variables can be calculated.

It is a statistical model in which the curve is fitted to the dataset. This technique is useful when the target variable is dichotomous. It is a predictive analysis algorithsm based on the concept of probability.


project/
project - Jupyter Notebook
localhost:8888/notebooks/project/project.ipynb
jupyter
project Last Checkpoint: 04/11/2020 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
In [30]: logreg = LogisticRegression()
         logreg.fit(X_train, Y_train)
         Y_pred_logreg = logreg.predict(X_test)
         print('Logistic regression score =',round(metrics.accuracy_score(Y_test, Y_pred_logreg),2))
         train_acc_logreg = round((logreg.score(X_train, Y_train)),2)
         test_acc_logreg = round((logreg.score(X_test, Y_test)),2)
         print("Training accuracy: %s " % train_acc_logreg)
         print('Test accuracy: %s ' % test_acc_logreg)
         print(classification_report(Y_test, Y_pred_logreg))
         scores = cross_val_score(logreg, X_test,Y_test, cv=3, scoring='accuracy')
         print('Logistic regression of each partition\n',scores)
         print('Mean score of all the scores after cross validation =',round(scores.mean(),2))
```

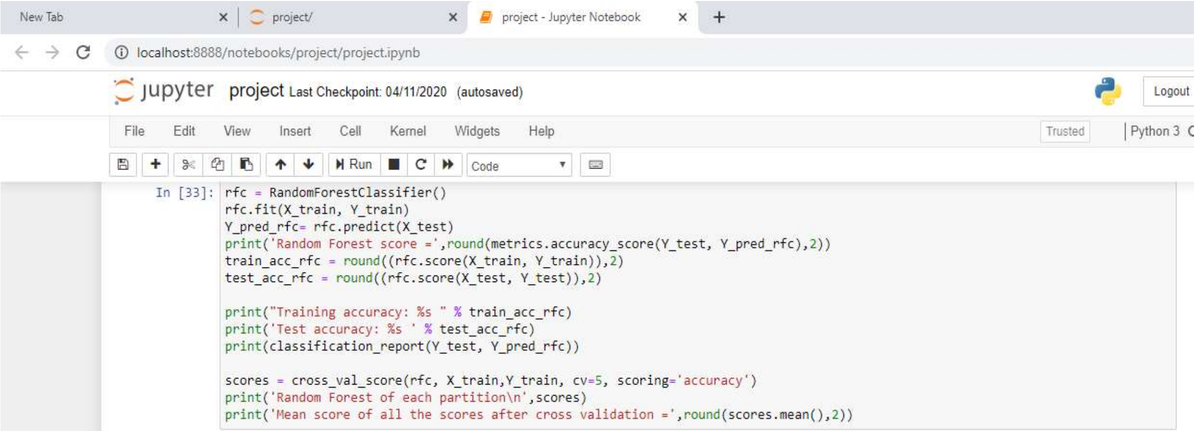
Figure 7.2: Logistic Regression

The Logistic Regression uses a complex cost function which is defined as 'Sigmoid Function' or also known as the 'logistic function'. The sigmoid function was used to map the predictions to probabilities. It limits the output and returns a probability score between 0 and 1. Logistic Regression was one of the popular algorithms for classification problems.

7.4.2Random Forest Classifier:

The Random Forest machine learning technique was chosen for predicting customer churn. It is a combination of multiple decision trees. It is an ensemble (a group of Decision Trees) learning methods for classification, regression problems and uses the bagging technique to generate the results. In ensemble learning, a group of weak learners come together to form a strong learner. Bagging also known as Bootstrap Aggregation was used to reduce the variance of the Decision Tree. It is an ensemble method in machine learning which is used to combine the predictions from multiple machine learning algorithms together to generate accurate results. The default hyperparameters of Random Forest give good results and it is great at avoiding overfitting. In Random forest, the most common output in all the decision trees was selected as the predicted class as the result.

It operates by constructing a multitude of decision trees at training time and outputting the class as the mode of the class or mean prediction of the individual trees. Random Forest was used for customer churn prediction in this research as it was quite fast and can deal with unbalanced and missing data as well.


New Tab
project/
project - Jupyter Notebook
localhost:8888/notebooks/project/project.ipynb
jupyter
project Last Checkpoint: 04/11/2020 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
In [33]: rfc = RandomForestClassifier()
         rfc.fit(X_train, Y_train)
         Y_pred_rfc= rfc.predict(X_test)
         print('Random Forest score =',round(metrics.accuracy_score(Y_test, Y_pred_rfc),2))
         train_acc_rfc = round((rfc.score(X_train, Y_train)),2)
         test_acc_rfc = round((rfc.score(X_test, Y_test)),2)

         print("Training accuracy: %s " % train_acc_rfc)
         print('Test accuracy: %s ' % test_acc_rfc)
         print(classification_report(Y_test, Y_pred_rfc))

         scores = cross_val_score(rfc, X_train,Y_train, cv=5, scoring='accuracy')
         print('Random Forest of each partition\n',scores)
         print('Mean score of all the scores after cross validation =',round(scores.mean(),2))
```

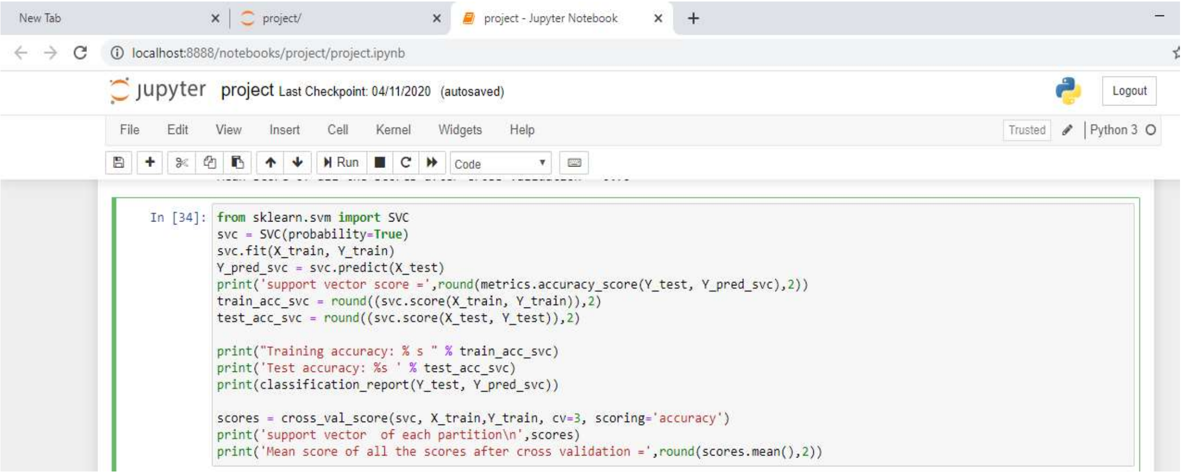
Figure 7.3: Random Forest

7.4.3 Support Vector Machine:

Another machine learning technique that will be evaluated is SVM. SVM is mostly used for classification problems and it builds the hyperplane margin between two classes. This algorithm uses a set of mathematical functions called a kernel which transforms the input data into the required form. Popular kernel choices for SVM were linear, polynomial and Radial Basis Function (RBF). Here in this research linear kernel was used based on the previous research on customer churn for linearly separable data.

Support Vector Machine (SVM) is a supervised machine learning model with associated learning algorithms that analyze the data for classification or regression problems. This algorithm works on the kernel function. The data is transformed based on the kernel function and an optimal boundary is set between the possible outputs.

Support Vector Machine(SVM) to determine the customer churn prediction in telecommunication customer data. SVM solved the nonlinearity, high dimension, and local minimization problems in customer churn prediction. As per the existing research SVM can work well with financial customer dataset also .



New Tab
project/
project - Jupyter Notebook
localhost:8888/notebooks/project/project.ipynb
jupyter
project Last Checkpoint: 04/11/2020 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
In [34]: from sklearn.svm import SVC
         svc = SVC(probability=True)
         svc.fit(X_train, Y_train)
         Y_pred_svc = svc.predict(X_test)
         print('support vector score =',round(metrics.accuracy_score(Y_test, Y_pred_svc),2))
         train_acc_svc = round((svc.score(X_train, Y_train)),2)
         test_acc_svc = round((svc.score(X_test, Y_test)),2)

         print("Training accuracy: % s " % train_acc_svc)
         print('Test accuracy: %s ' % test_acc_svc)
         print(classification_report(Y_test, Y_pred_svc))

         scores = cross_val_score(svc, X_train,Y_train, cv=3, scoring='accuracy')
         print('support vector  of each partition\n',scores)
         print('Mean score of all the scores after cross validation =',round(scores.mean(),2))
```

Fig 7.4: Support Vector Machine

7.4.4 K-nearest neighbor:

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

"Birds of a feather flock together."

The KNN Algorithm

1.Load the data

2. Initialize K to your chosen number of neighbors

3. For each example in the data

3.1 Calculate the distance between the query example and the current example from the data.

3.2 Add the distance and the index of the example to an ordered collection

4.Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances

5. Pick the first K entries from the sorted collection

6. Get the labels of the selected K entries

7. If regression, return the mean of the K labels

8. If classification, return the mode of the K labels

KeyPoints:

This algorithm can be used for both Classification and Regression. This method is also known as Lazy Learning method. Unlike other algorithms it never stores patterns from training data. It is time consuming but very accurate.

Maths behind algorithm:

• Euclidean Distance Formula:

o This formula is used when data contains all continuous variables.

Fig- Distance Formula

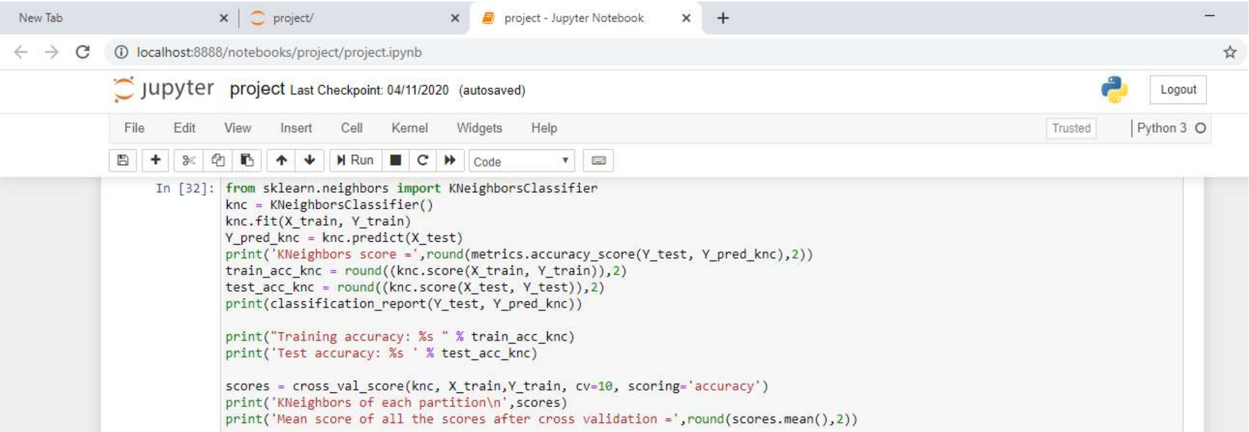
• Manhattan Formula:

o This formula will be used if our data contains both Continuous as well as Categorical variables.

Fig-Manhattan Distance Formula

• Weighted Distance Method:

o Suppose we have 100 TV, now we have good business knowledge and we want to assign weights as per our knowledge. Here a weighted distance method would be used.


New Tab
project/
project - Jupyter Notebook
localhost:8888/notebooks/project/project.ipynb
jupyter project Last Checkpoint: 04/11/2020 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [32]: from sklearn.neighbors import KNeighborsClassifier
         knc = KNeighborsClassifier()
         knc.fit(X_train, Y_train)
         Y_pred_knc = knc.predict(X_test)
         print('KNeighbors score =',round(metrics.accuracy_score(Y_test, Y_pred_knc),2))
         train_acc_knc = round((knc.score(X_train, Y_train)),2)
         test_acc_knc = round((knc.score(X_test, Y_test)),2)
         print(classification_report(Y_test, Y_pred_knc))

         print("Training accuracy: %s " % train_acc_knc)
         print('Test accuracy: %s ' % test_acc_knc)

         scores = cross_val_score(knc, X_train,Y_train, cv=10, scoring='accuracy')
         print('KNeighbors of each partition\n',scores)
         print('Mean score of all the scores after cross validation =',round(scores.mean(),2))
```

Figure 7.5: K nearest neighbor

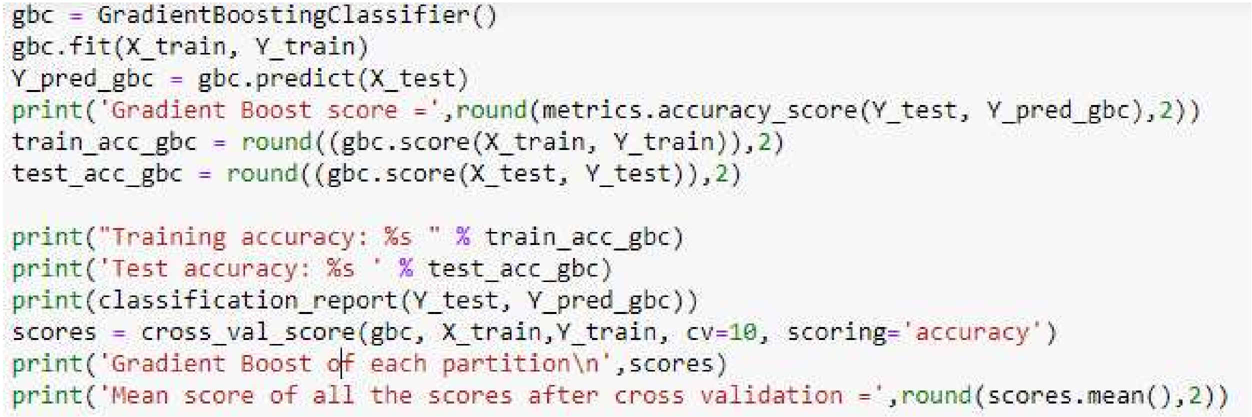
7.4.5 Gradient Boost:

Boosting is a method of converting weak learners into strong learners. In boosting, each new tree is a fit on a modified version of the original data set. The gradient boosting algorithm (gbm) can be most easily explained by first introducing the AdaBoost Algorithm. The AdaBoost Algorithm begins by training a decision tree in which each observation is assigned an equal weight.

After evaluating the first tree, we increase the weights of those observations that are difficult to classify and lower the weights for those that are easy to classify. The second tree is therefore grown on this weighted data. Here, the idea is to improve upon the predictions of the first tree. Our new model is therefore Tree 1 + Tree 2.
We then compute the classification error from this new 2-tree ensemble model and grow a third tree to predict the revised residuals. We repeat this process for a specified number of iterations. Subsequent trees help us to classify observations that are not well classified by the previous trees. Predictions of the final ensemble model is therefore the weighted sum of the predictions made by the previous tree models.

Gradient Boosting trains many models in a gradual, additive and sequential manner. The major difference between AdaBoost and Gradient Boosting Algorithm is how the two algorithms identify the shortcomings of weak learners (eg. decision trees). While the AdaBoost model identifies the shortcomings by using high weight data points, gradient boosting performs the same by using gradients in the loss function (y=ax+b+e ,e needs a special mention as it is the error term).

The loss function is a measure indicating how good are model's coefficients are at fitting the underlying data. A logical understanding of loss function would depend on what we are trying to optimise. For example, if we are trying to predict the sales prices by using a regression, then the loss function would be based off the error between true and predicted house prices. Similarly, if our goal is to classify credit defaults, then the loss function would be a measure of how good our predictive model is at classifying bad loans. One of the biggest motivations of using gradient boosting is that it allows one to optimise a user specified cost function, instead of a loss function that usually offers less control and does not essentially correspond with real world applications.



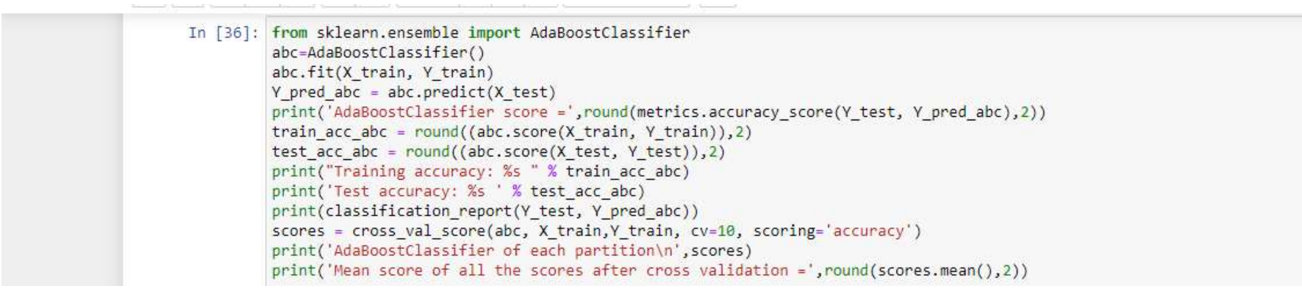
```
gbc = GradientBoostingClassifier()
gbc.fit(X_train, Y_train)
Y_pred_gbc = gbc.predict(X_test)
print('Gradient Boost score =',round(metrics.accuracy_score(Y_test, Y_pred_gbc),2))
train_acc_gbc = round((gbc.score(X_train, Y_train)),2)
test_acc_gbc = round((gbc.score(X_test, Y_test)),2)

print("Training accuracy: %s " % train_acc_gbc)
print('Test accuracy: %s ' % test_acc_gbc)
print(classification_report(Y_test, Y_pred_gbc))
scores = cross_val_score(gbc, X_train,Y_train, cv=10, scoring='accuracy')
print('Gradient Boost of each partition\n',scores)
print('Mean score of all the scores after cross validation =',round(scores.mean(),2))
```

Figure 7.6 : Gradient Boosting Classifier

7.4.6 AdaBoost:

AdaBoost (Adaptive Boosting) : It works on a similar method as discussed above. It fits a sequence of weak learners on different weighted training data. It starts by predicting the original data set and gives equal weight to each observation. If prediction is incorrect using the first learner, then it gives higher weight to observations which have been predicted incorrectly. Being an iterative process, it continues to add learner(s) until a limit is reached in the number of models or accuracy.

Mostly, we use decision stamps with AdaBoost. But, we can use any machine learning algorithms as a base learner if it accepts weight on the training data set. We can use AdaBoost algorithms for both classification and regression problems.



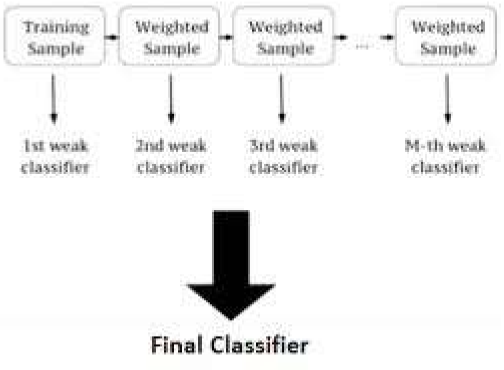
```
In [36]: from sklearn.ensemble import AdaBoostClassifier
         abc=AdaBoostClassifier()
         abc.fit(X_train, Y_train)
         Y_pred_abc = abc.predict(X_test)
         print('AdaBoostClassifier score =',round(metrics.accuracy_score(Y_test, Y_pred_abc),2))
         train_acc_abc = round((abc.score(X_train, Y_train)),2)
         test_acc_abc = round((abc.score(X_test, Y_test)),2)
         print("Training accuracy: %s " % train_acc_abc)
         print('Test accuracy: %s ' % test_acc_abc)
         print(classification_report(Y_test, Y_pred_abc))
         scores = cross_val_score(abc, X_train,Y_train, cv=10, scoring='accuracy')
         print('AdaBoostClassifier of each partition\n',scores)
         print('Mean score of all the scores after cross validation =',round(scores.mean(),2))
```

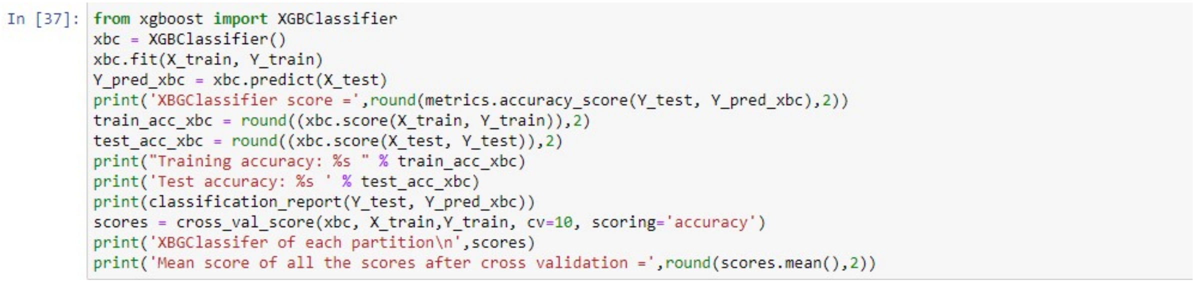
Figure 7.7 : Adaboost Classifier

7.4.7 XGboost:

XGboost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of its leafs that contains a continuous score. XGBoost minimizes a regularized (L1 and L2) objective function that combines a convex loss function (based on the difference between the predicted and target outputs) and a penalty term for model complexity (in other words, the regression tree functions). The training proceeds iteratively, adding new trees that predict the residuals or errors of prior trees that are then combined with previous trees to make the final prediction. It's called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.





```
from xgboost import XGBClassifier
xbc = XGBClassifier()
xbc.fit(X_train, Y_train)
Y_pred_xbc = xbc.predict(X_test)
print('XBGClassifier score =',round(metrics.accuracy_score(Y_test, Y_pred_xbc),2))
train_acc_xbc = round((xbc.score(X_train, Y_train)),2)
test_acc_xbc = round((xbc.score(X_test, Y_test)),2)
print("Training accuracy: %s " % train_acc_xbc)
print('Test accuracy: %s ' % test_acc_xbc)
print(classification_report(Y_test, Y_pred_xbc))
scores = cross_val_score(xbc, X_train,Y_train, cv=10, scoring='accuracy')
print('XBGClassifer of each partition\n',scores)
print('Mean score of all the scores after cross validation =',round(scores.mean(),2))
```

Figure 7.8 : XGboost

7.4.8 LightGBM:

Light GBM is a gradient boosting framework that uses a tree based learning algorithm.How does it differ from other tree based algorithms?

Light GBM grows tree vertically while other algorithms grow trees horizontally meaning that Light GBM grows tree leaf-wise while other algorithms grow level-wise. It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithm can reduce more loss than a level-wise algorithm.Below diagrams explain the implementation of LightGBM and other boosting algorithms.

Explains how LightGBM works

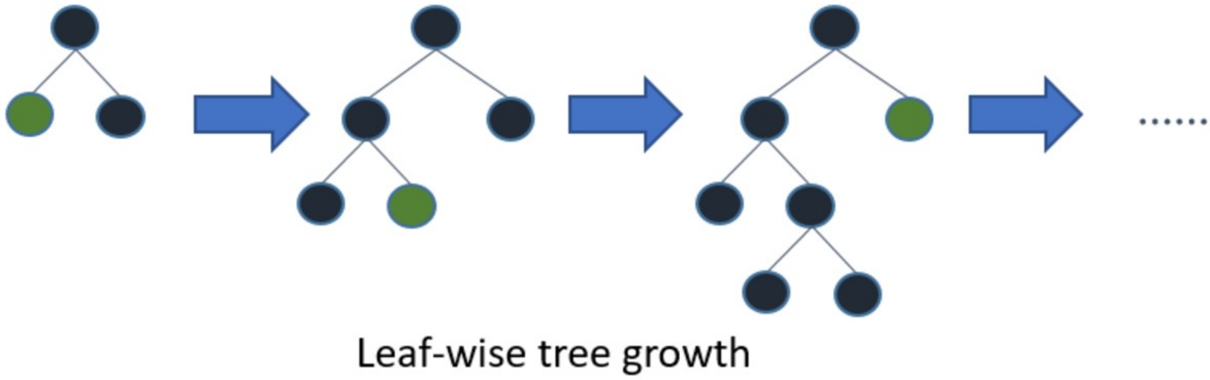


Figure 7.9 : Leaf-wise tree growth

How other boosting algorithm works

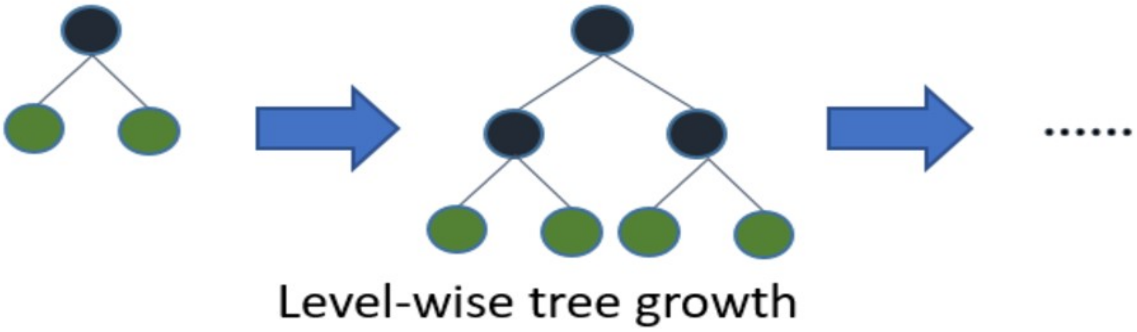
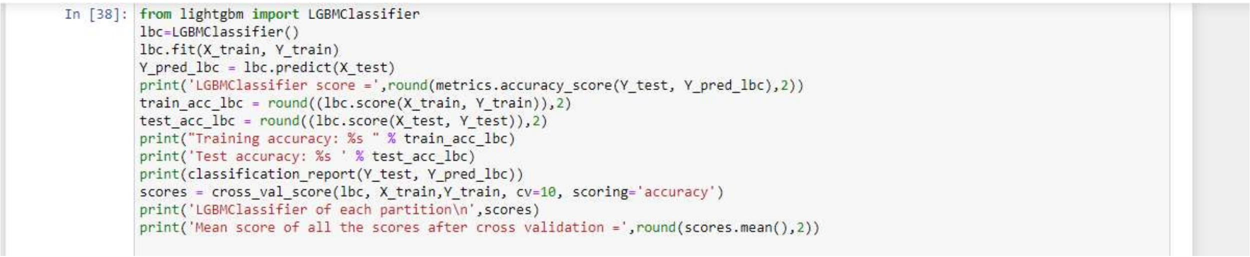


Figure 7.10: Level-wise tree growth

Why is Light GBM gaining extreme popularity:

The size of data is increasing day by day and it is becoming difficult for traditional data science algorithms to give faster results. Light GBM is prefixed as 'Light' because of its high speed. Light GBM can handle the large size of data and takes lower memory to run. Another reason why Light GBM is popular is because it focuses on accuracy of results. LGBM also supports GPU learning and thus data scientists are widely using LGBM for data science application development.



```
In [38]: from lightgbm import LGBMClassifier
         lbc=LGBMClassifier()
         lbc.fit(X_train, Y_train)
         Y_pred_lbc = lbc.predict(X_test)
         print('LGBMClassifier score =',round(metrics.accuracy_score(Y_test, Y_pred_lbc),2))
         train_acc_lbc = round((lbc.score(X_train, Y_train)),2)
         test_acc_lbc = round((lbc.score(X_test, Y_test)),2)
         print("Training accuracy: %s " % train_acc_lbc)
         print('Test accuracy: %s ' % test_acc_lbc)
         print(classification_report(Y_test, Y_pred_lbc))
         scores = cross_val_score(lbc, X_train,Y_train, cv=10, scoring='accuracy')
         print('LGBMClassifier of each partition\n',scores)
         print('Mean score of all the scores after cross validation =',round(scores.mean(),2))
```

Figure 7.11 : LGBMClassifier

7.5 Clustering:

Cluster is the collection of data objects which are similar to one another within the same group (class or category) and are different from the objects in the other clusters.Clustering is an unsupervised learning technique in which there is predefined classes and prior information which defines how the data should be grouped or labeled into separate classes. It could also be considered as an Exploratory Data Analysis (EDA) process which helps us to discover hidden patterns of interest or structure in data. Clustering can also work as a standalone tool to get the insights about the data distribution or as a preprocessing step in other algorithms.

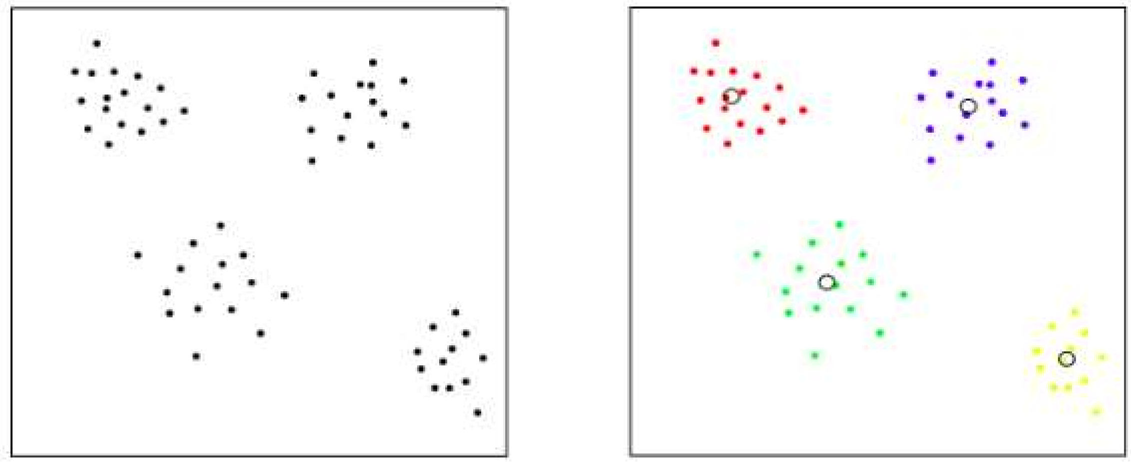


Figure7.12 : Clusters

Clustering allows us to find the hidden relationship between the data points in the dataset.

Examples:

1. In marketing, customers are segmented according to similarities to carry out targeted marketing.

2. Given a collection of text, we need to organize them, according to the content similarities to create a topic hierarchy
3. Detecting distinct kinds of pattern in image data (Image processing). It's effective in biology research for identifying the underlying patterns.

How do we define good Clustering algorithms?

High quality clusters can be created by reducing the distance between the objects in the same cluster known as intra-cluster minimization and increasing the distance with the objects in the other cluster known as inter-cluster maximization.

Intra-cluster minimization: The closer the objects in a cluster, the more likely they belong to the same cluster.

Inter-cluster Maximization: This makes the separation between two clusters. The main goal is to maximize the distance between 2 clusters.

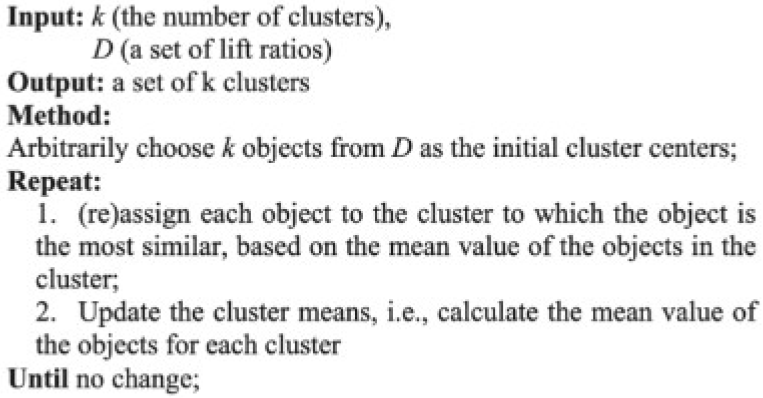
In this project we are going to use **k-means** clustering.

K-Means

K-Means is one of the most popular "clustering" algorithms. K-Means stores k centroids that it uses to define clusters. A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid.

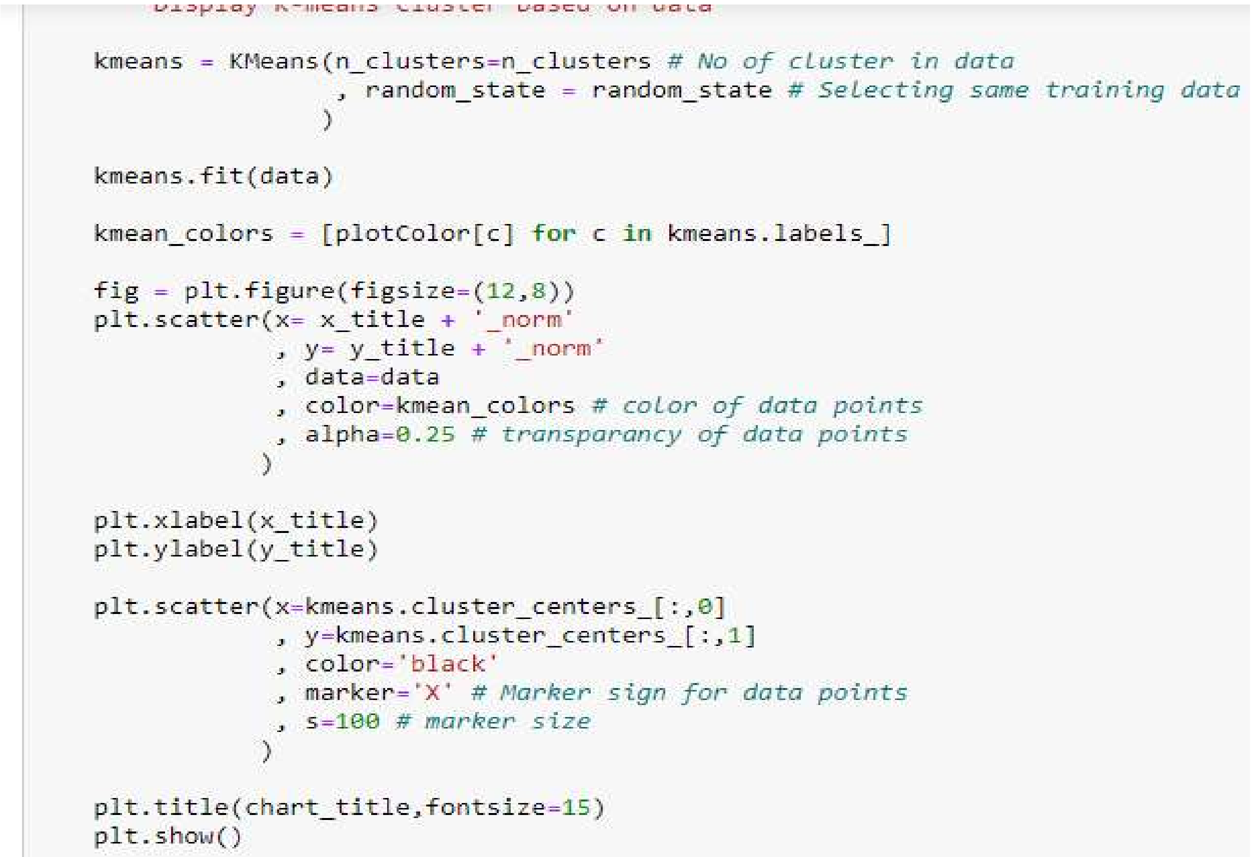
K-Means finds the best centroids by alternating between (1) assigning data points to clusters based on the current centroids (2) choosing centroids (points which are the center of a cluster) based on the current assignment of data points to clusters.

How it works:



```
Input: k (the number of clusters),
       D (a set of lift ratios)
Output: a set of k clusters
Method:
Arbitrarily choose k objects from D as the initial cluster centers;
Repeat:
   1. (re)assign each object to the cluster to which the object is
   the most similar, based on the mean value of the objects in the
   cluster;
   2. Update the cluster means, i.e., calculate the mean value of
   the objects for each cluster
Until no change;
```

The distance metric used to calculate similarity in step 1 is Euclidean distance.



```
        Display K-means cluster based on data

    kmeans = KMeans(n_clusters=n_clusters # No of cluster in data
                    , random_state = random_state # Selecting same training data
                   )

    kmeans.fit(data)

    kmean_colors = [plotColor[c] for c in kmeans.labels_]

    fig = plt.figure(figsize=(12,8))
    plt.scatter(x= x_title + '_norm'
                , y= y_title + '_norm'
                , data=data
                , color=kmean_colors # color of data points
                , alpha=0.25 # transparancy of data points
               )

    plt.xlabel(x_title)
    plt.ylabel(y_title)

    plt.scatter(x=kmeans.cluster_centers_[:,0]
                , y=kmeans.cluster_centers_[:,1]
                , color='black'
                , marker='X' # Marker sign for data points
                , s=100 # marker size
               )

    plt.title(chart_title,fontsize=15)
    plt.show()
```

Figure7.13 : K-Means Clustering

CHAPTER-8

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

8.1 TYPES OF TESTS

8.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centred on the following items: Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

8.1.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.1.5 White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purposeful. It is used to test areas that cannot be reached from a black box level.

8.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the innerworkings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

8.1.7 Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

o All field entries must work properly.

o Pages must be activated from the identified link.

o The entry screen, messages and responses must not be delayed.

Features to be tested

o Verify that the entries are of the correct format

o No duplicate entries should be allowed

o All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level –interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

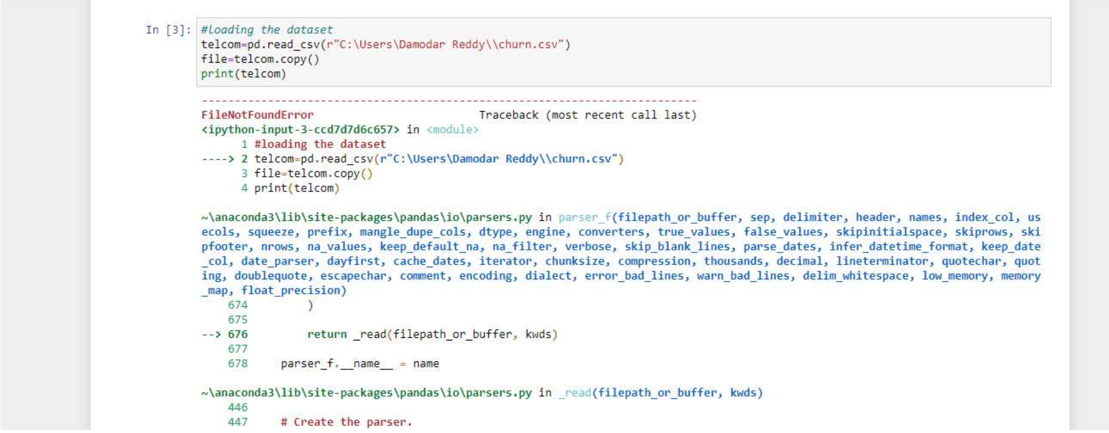
Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

In our project we prefer unit testing because jupyter envirnoment have cells .so,we are executing each cell one by one .Unit testing is done on every individual module as they are completed and become executable. It is confined that the platform we are using is comprised of both the input and output are collaborated at single environment and each executable output can be verified at that point.

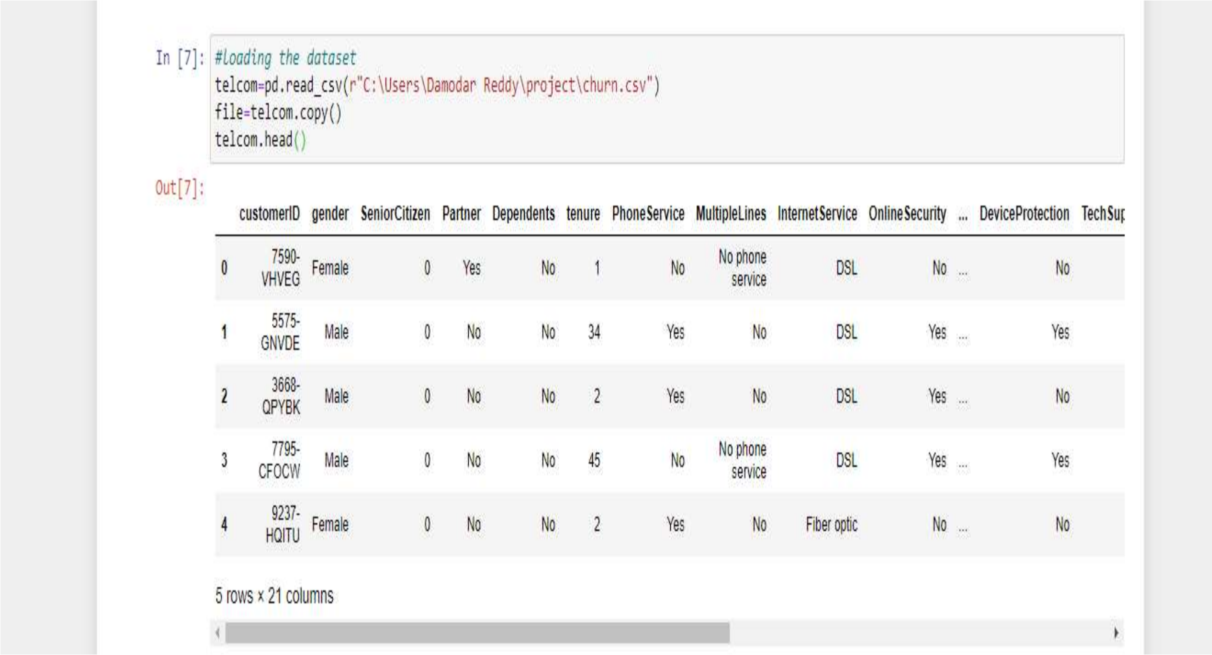
Test Case Result:

While taking the input file from the file directory which is a csv file, an error raised due to file not found in the specified path since the path that have mentioned is not appropriate.



```
In [3]: #loading the dataset
        telcom=pd.read_csv(r"C:\Users\Damodar Reddy\\churn.csv")
        file=telcom.copy()
        print(telcom)

        ---------------------------------------------------------------------------
        FileNotFoundError                         Traceback (most recent call last)
        <ipython-input-3-ccd7d7d6c657> in <module>
              1 #loading the dataset
        ----> 2 telcom=pd.read_csv(r"C:\Users\Damodar Reddy\\churn.csv")
              3 file=telcom.copy()
              4 print(telcom)

        ~\anaconda3\lib\site-packages\pandas\io\parsers.py in parser_f(filepath_or_buffer, sep, delimiter, header, names, index_col, us
        ecols, squeeze, prefix, mangle_dupe_cols, dtype, engine, converters, true_values, false_values, skipinitialspace, skiprows, ski
        pfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date
        _col, date_parser, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quot
        ing, doublequote, escapechar, comment, encoding, dialect, error_bad_lines, warn_bad_lines, delim_whitespace, low_memory, memory
        _map, float_precision)
            674         )
            675 
        --> 676         return _read(filepath_or_buffer, kwds)
            677 
            678     parser_f.__name__ = name

        ~\anaconda3\lib\site-packages\pandas\io\parsers.py in _read(filepath_or_buffer, kwds)
            446 
            447     # Create the parser.
```

The solution for this error is to make sure the file directory correct while entering the path of the file from the desktop.



In [7]:

```
#Loading the dataset
telcom=pd.read_csv(r"C:\Users\Damodar Reddy\project\churn.csv")
file=telcom.copy()
telcom.head()
```

Out[7]:

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

CHAPTER-9

EXECUTION AND RESULTS

9.1 Metrics Used

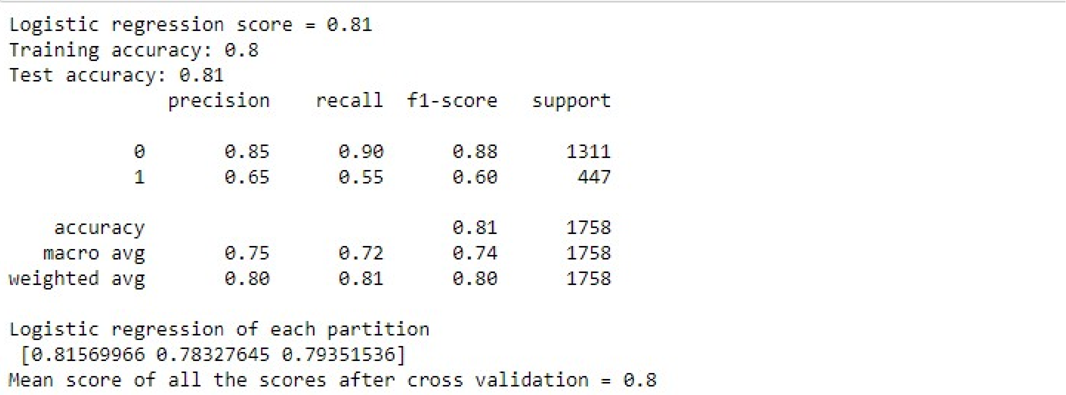
There are few metrics that are used for this project. The metrics that are obtained from the model are:

- Classification Report
- Confusion Matrix
- Recall
- Accuracy
- Precision
- F-measure

9.1.1 Classification Report

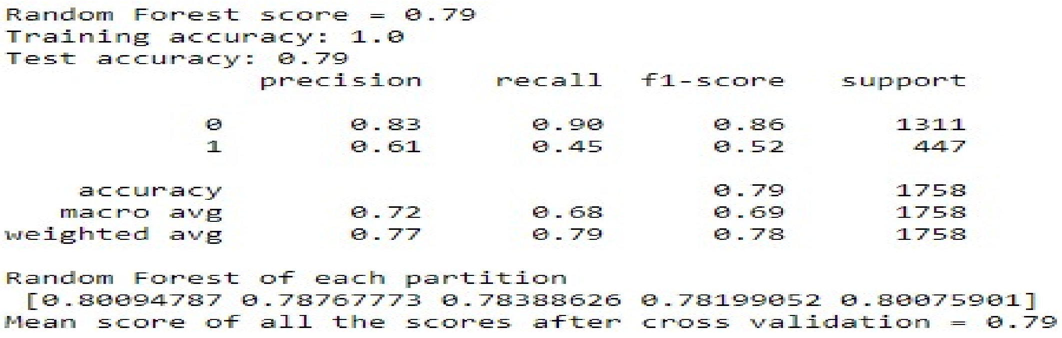
A Classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report as shown below.

Logistic Regression:



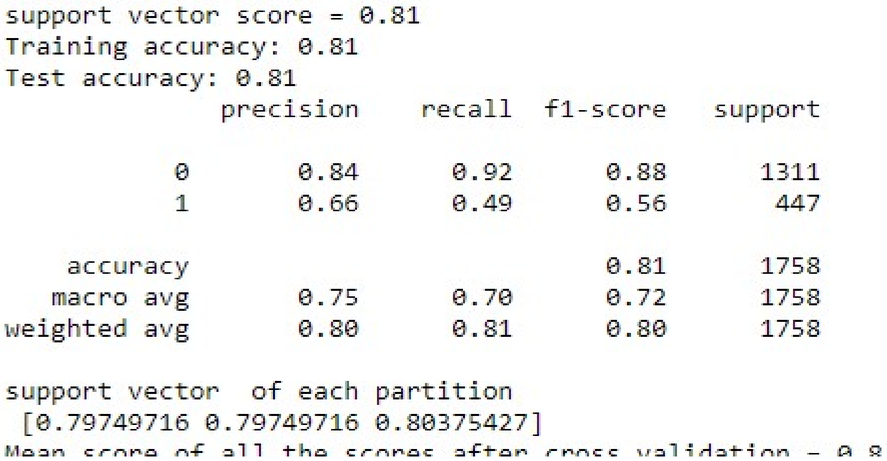
```
Logistic regression score = 0.81
Training accuracy: 0.8
Test accuracy: 0.81
              precision    recall  f1-score   support

           0       0.85      0.90      0.88      1311
           1       0.65      0.55      0.60       447

    accuracy                           0.81      1758
   macro avg       0.75      0.72      0.74      1758
weighted avg       0.80      0.81      0.80      1758

Logistic regression of each partition
 [0.81569966 0.78327645 0.79351536]
Mean score of all the scores after cross validation = 0.8
```

Random Forest Classifier:



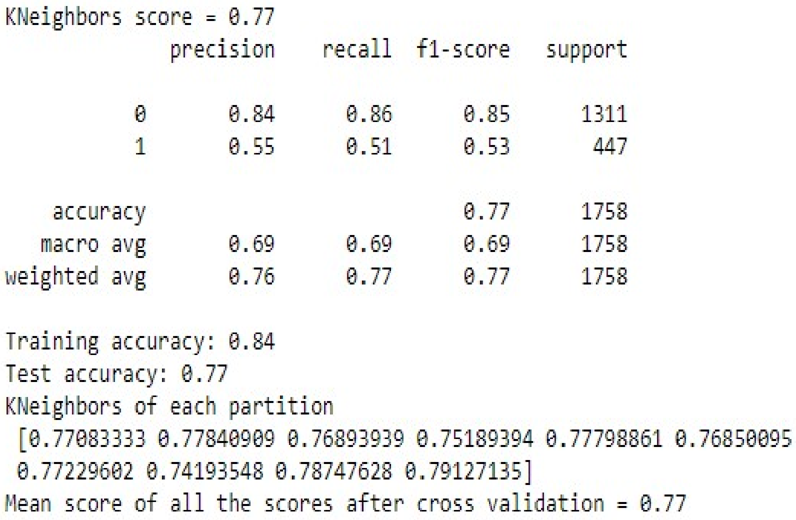
```
Random Forest score = 0.79
Training accuracy: 1.0
Test accuracy: 0.79
              precision    recall  f1-score   support

           0       0.83      0.90      0.86      1311
           1       0.61      0.45      0.52       447

    accuracy                           0.79      1758
   macro avg       0.72      0.68      0.69      1758
weighted avg       0.77      0.79      0.78      1758

Random Forest of each partition
 [0.80094787 0.78767773 0.78388626 0.78199052 0.80075901]
Mean score of all the scores after cross validation = 0.79
```

Support Vector Machine



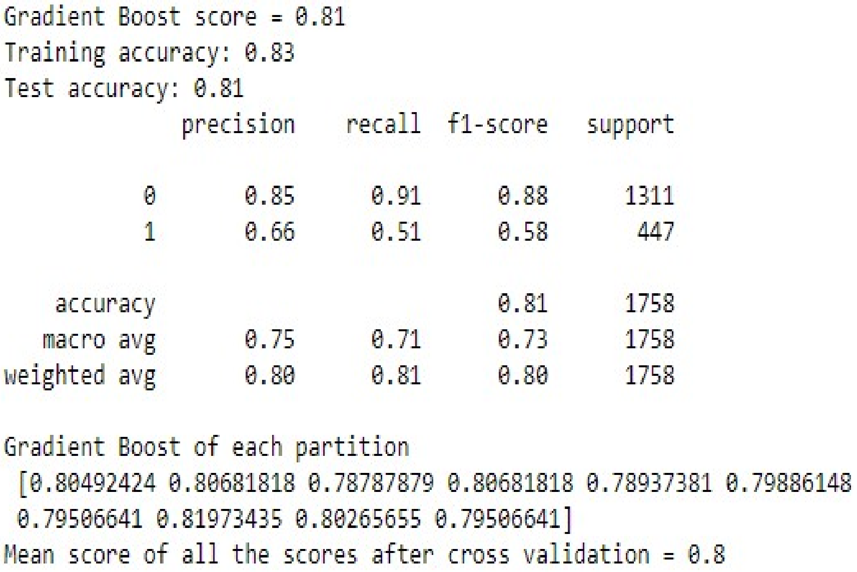
```
support vector score = 0.81
Training accuracy: 0.81
Test accuracy: 0.81
              precision    recall  f1-score   support

           0       0.84      0.92      0.88      1311
           1       0.66      0.49      0.56       447

    accuracy                           0.81      1758
   macro avg       0.75      0.70      0.72      1758
weighted avg       0.80      0.81      0.80      1758

support vector  of each partition
 [0.79749716 0.79749716 0.80375427]
Mean score of all the scores after cross validation = 0.8
```

K nearest neighbor



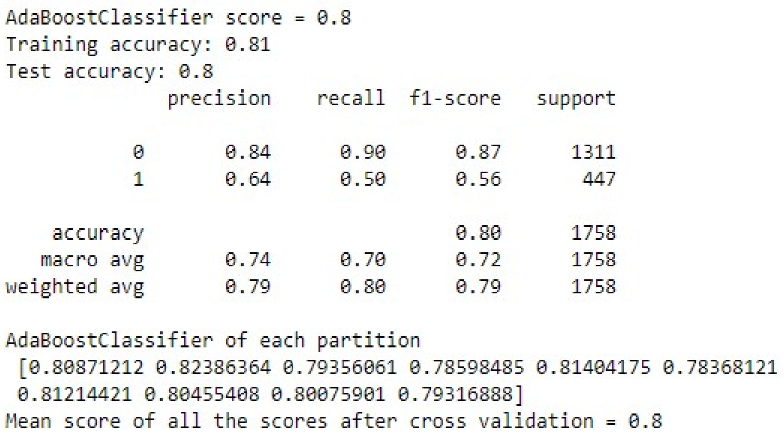
```
KNeighbors score = 0.77
              precision    recall  f1-score   support

           0       0.84      0.86      0.85      1311
           1       0.55      0.51      0.53       447

    accuracy                           0.77      1758
   macro avg       0.69      0.69      0.69      1758
weighted avg       0.76      0.77      0.77      1758

Training accuracy: 0.84
Test accuracy: 0.77
KNeighbors of each partition
 [0.77083333 0.77840909 0.76893939 0.75189394 0.77798861 0.76850095
 0.77229602 0.74193548 0.78747628 0.79127135]
Mean score of all the scores after cross validation = 0.77
```

Gradient Boosting



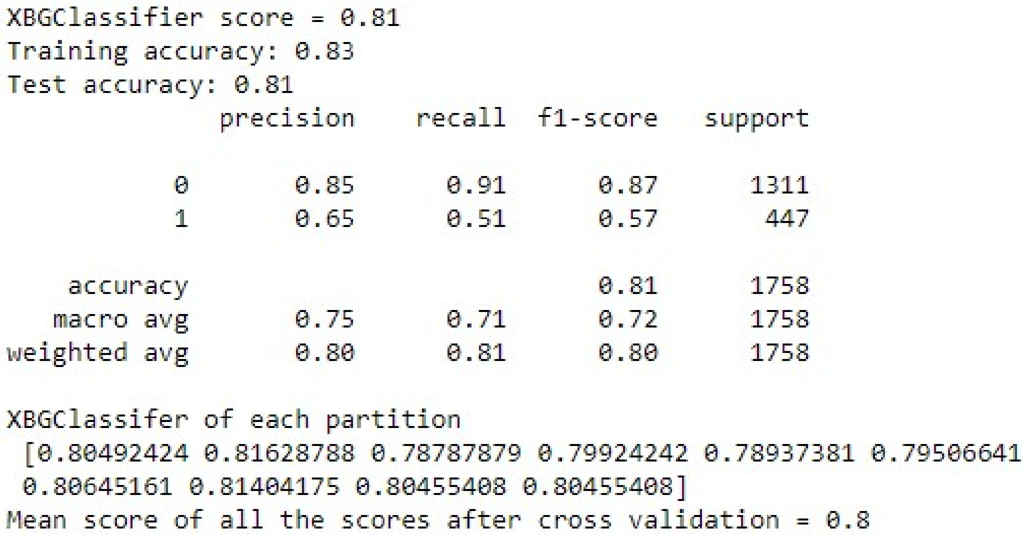
```
Gradient Boost score = 0.81
Training accuracy: 0.83
Test accuracy: 0.81
              precision    recall  f1-score   support

           0       0.85      0.91      0.88      1311
           1       0.66      0.51      0.58       447

    accuracy                           0.81      1758
   macro avg       0.75      0.71      0.73      1758
weighted avg       0.80      0.81      0.80      1758

Gradient Boost of each partition
 [0.80492424 0.80681818 0.78787879 0.80681818 0.78937381 0.79886148
 0.79506641 0.81973435 0.80265655 0.79506641]
Mean score of all the scores after cross validation = 0.8
```

Ada Boosting



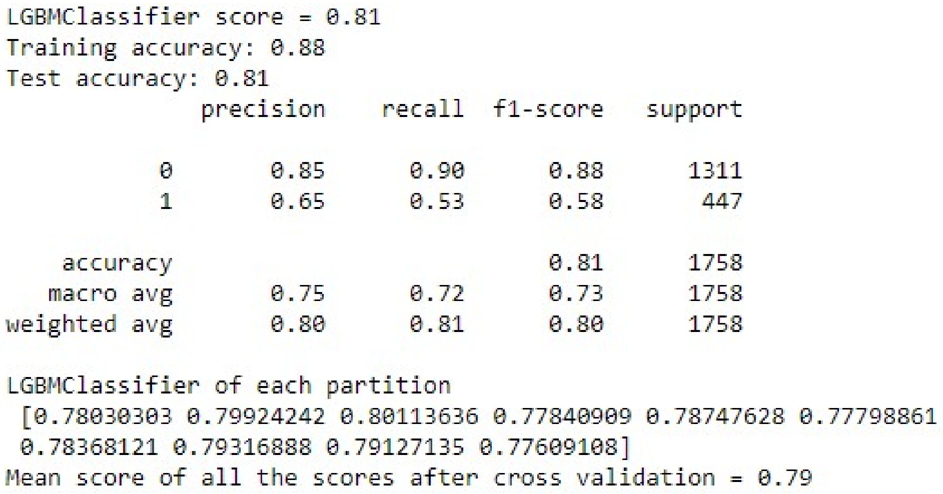
```
AdaBoostClassifier score = 0.8
Training accuracy: 0.81
Test accuracy: 0.8
              precision    recall  f1-score   support

           0       0.84      0.90      0.87      1311
           1       0.64      0.50      0.56       447

    accuracy                           0.80      1758
   macro avg       0.74      0.70      0.72      1758
weighted avg       0.79      0.80      0.79      1758

AdaBoostClassifier of each partition
 [0.80871212 0.82386364 0.79356061 0.78598485 0.81404175 0.78368121
 0.81214421 0.80455408 0.80075901 0.79316888]
Mean score of all the scores after cross validation = 0.8
```

XGboost



```
XBGClassifier score = 0.81
Training accuracy: 0.83
Test accuracy: 0.81
              precision    recall  f1-score   support

           0       0.85      0.91      0.87      1311
           1       0.65      0.51      0.57       447

    accuracy                           0.81      1758
   macro avg       0.75      0.71      0.72      1758
weighted avg       0.80      0.81      0.80      1758

XBGClassifer of each partition
 [0.80492424 0.81628788 0.78787879 0.79924242 0.78937381 0.79506641
 0.80645161 0.81404175 0.80455408 0.80455408]
Mean score of all the scores after cross validation = 0.8
```

LGBMClassifier



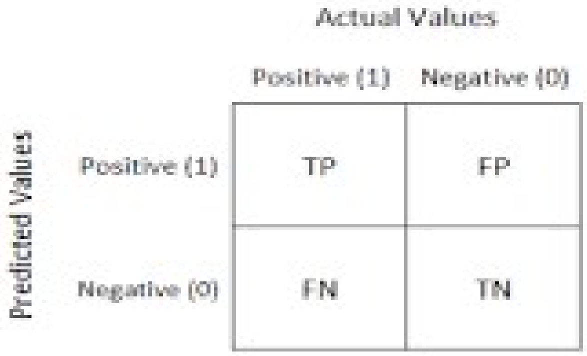
```
LGBMClassifier score = 0.81
Training accuracy: 0.88
Test accuracy: 0.81
              precision    recall  f1-score   support

           0       0.85      0.90      0.88      1311
           1       0.65      0.53      0.58       447

    accuracy                           0.81      1758
   macro avg       0.75      0.72      0.73      1758
weighted avg       0.80      0.81      0.80      1758

LGBMClassifier of each partition
 [0.78030303 0.79924242 0.80113636 0.77840909 0.78747628 0.77798861
 0.78368121 0.79316888 0.79127135 0.77609108]
Mean score of all the scores after cross validation = 0.79
```

9.1.2 Confusion Matrix

It is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.

A confusion matrix is a table that shows the number of instances classified correctly or not in each class. It is a binary classifier. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It describes the performance of the classification model.

True Positive – actually fraud, and predicted as fraud	[TP]
False Positive – actually fraud, but predicted as normal	[FP]
True Negative – actually normal, predicted as normal	[TN]
False Negative – actually normal, but predicted as fraud	[FN]



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

It is extremely useful for measuring Recall, Precision, Specificity, Accuracy and most importantly AUC-ROC Curve. Let’s understand TP, FP, FN, TN in terms of pregnancy analogy.

True Positive:

Interpretation: You predicted positive and it’s true.
You predicted that a woman is pregnant and she actually is.

True Negative:

Interpretation: You predicted negative and it’s true.

You predicted that a man is not pregnant and he actually is not.

False Positive: (Type 1 Error)

Interpretation: You predicted positive and it’s false.

You predicted that a man is pregnant but he actually is not.

False Negative: (Type 2 Error)

Interpretation: You predicted negative and it’s false.

You predicted that a woman is not pregnant but she actually is.

Just Remember, We describe predicted values as Positive and Negative and actual values as True and False.

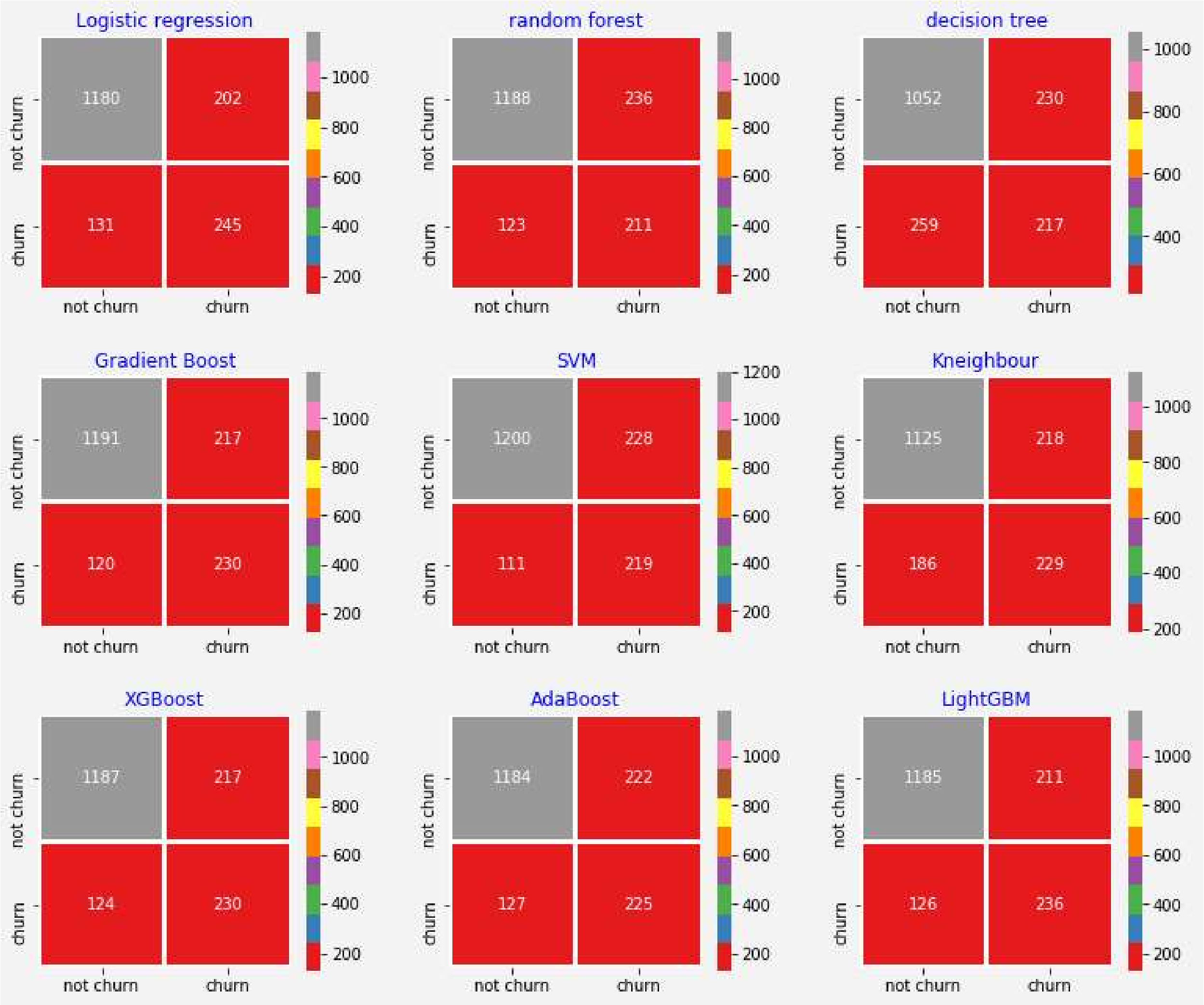


Figure 9.1 : Confusion matrix

Recall

Out of all the positive classes, how much we predicted correctly. It should be high as possible.

Precision

Out of all the positive classes we have predicted correctly, how many are actually positive.

F-measure

It is difficult to compare two models with low precision and high recall or vice versa. So to make them comparable, we use F-Score. F-score helps to measure Recall and Precision at the same time. It uses Harmonic Mean in place of Arithmetic Mean by punishing the extreme values more.

Accuracy

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

It works well only if there are an equal number of samples belonging to each class.

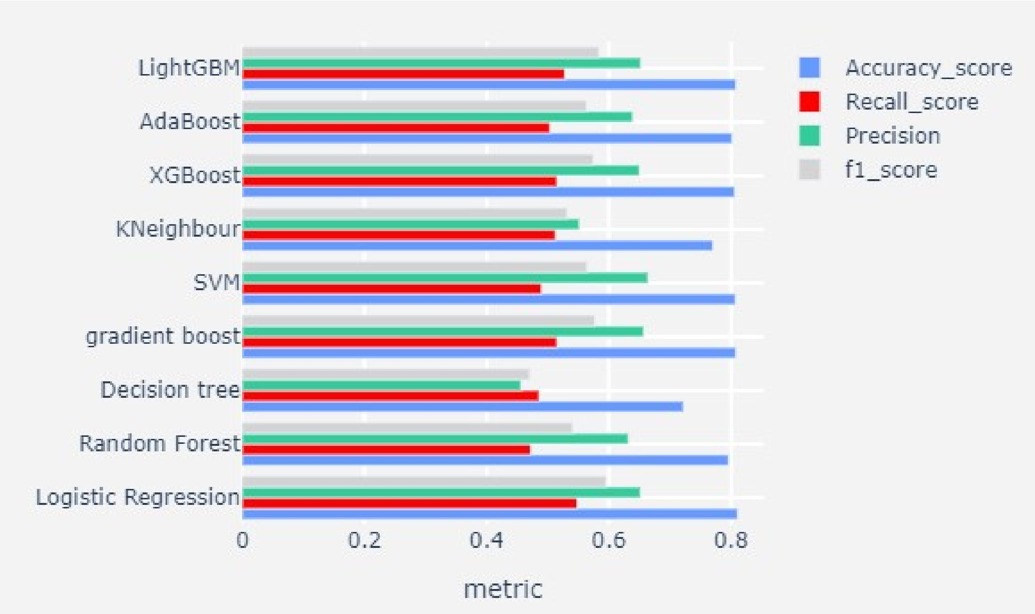


Figure 9.2 : Model Performance

9.1.3 Auc-Roc curve

In Machine Learning, performance measurement is an essential task. So when it comes to a classification problem, we can count on an AUC - ROC Curve. When we need to check or visualize the performance of the multi - class classification problem, we use AUC (**Area Under The Curve**) ROC (**Receiver Operating Characteristics**) curve. It is one of the most important evaluation metrics for checking any classification model's performance. It is also written as AUROC (**Area Under the Receiver Operating Characteristics**).Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s.

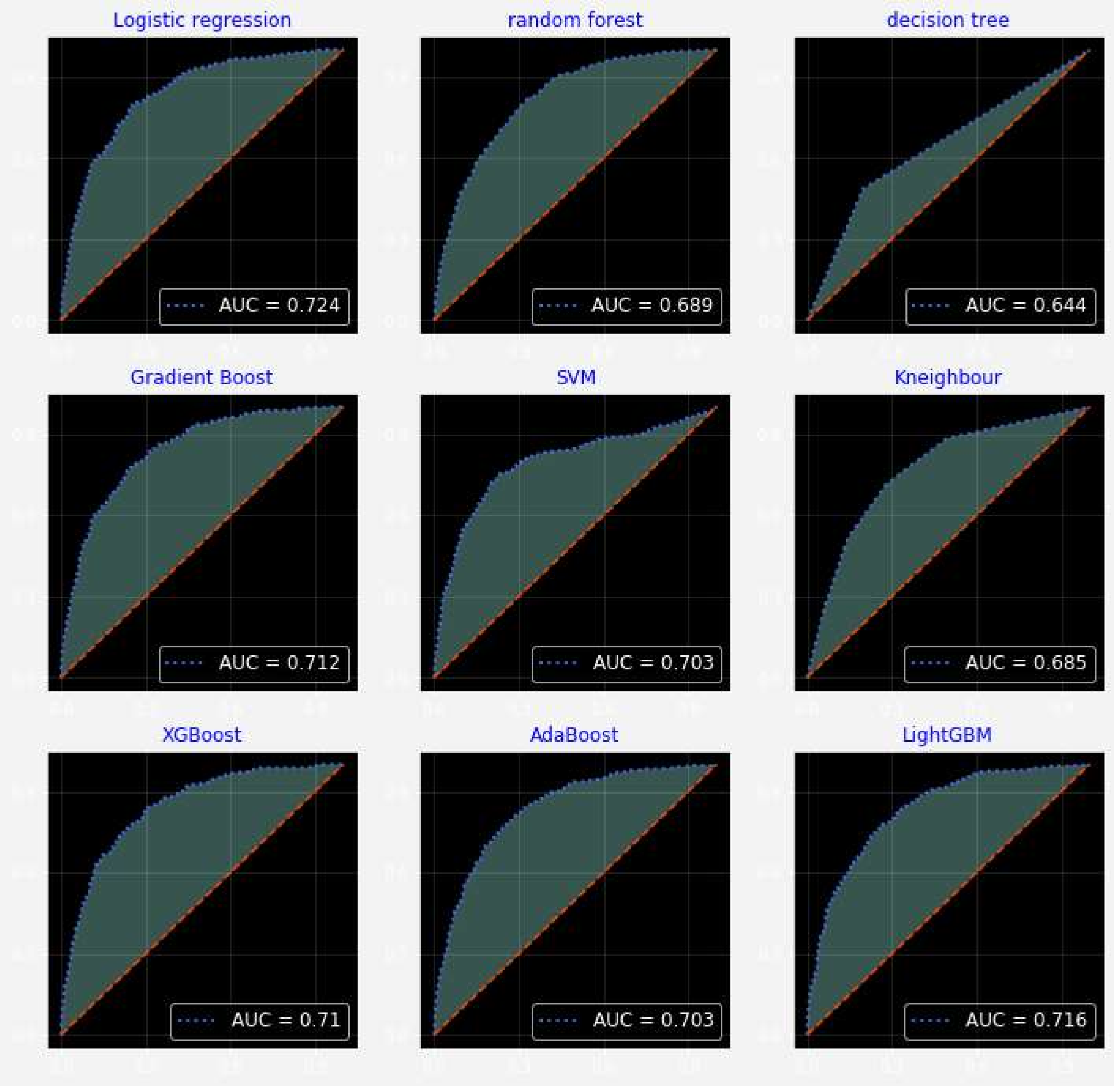


Figure 9.3: AUC curve

9.1.4 Feature Importance:

You can get the **feature importance** of each **feature** of your dataset by using the **feature importance** property of the model. **Feature importance** gives you a score for each **feature** of your data, the higher the score more **important** or relevant is the **feature** towards your output **variable**.

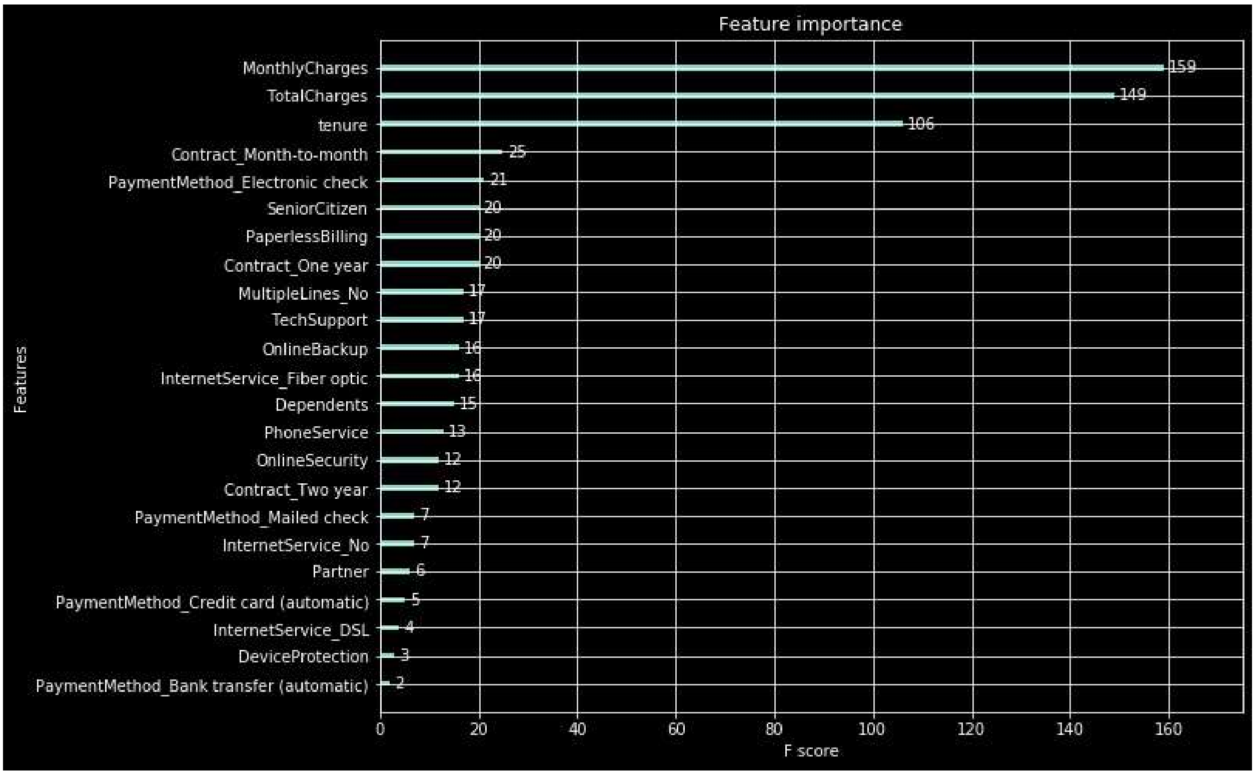


Figure 9.4 : Feature Importance

9.1.5 Clustering

Based on **Demographic information,**

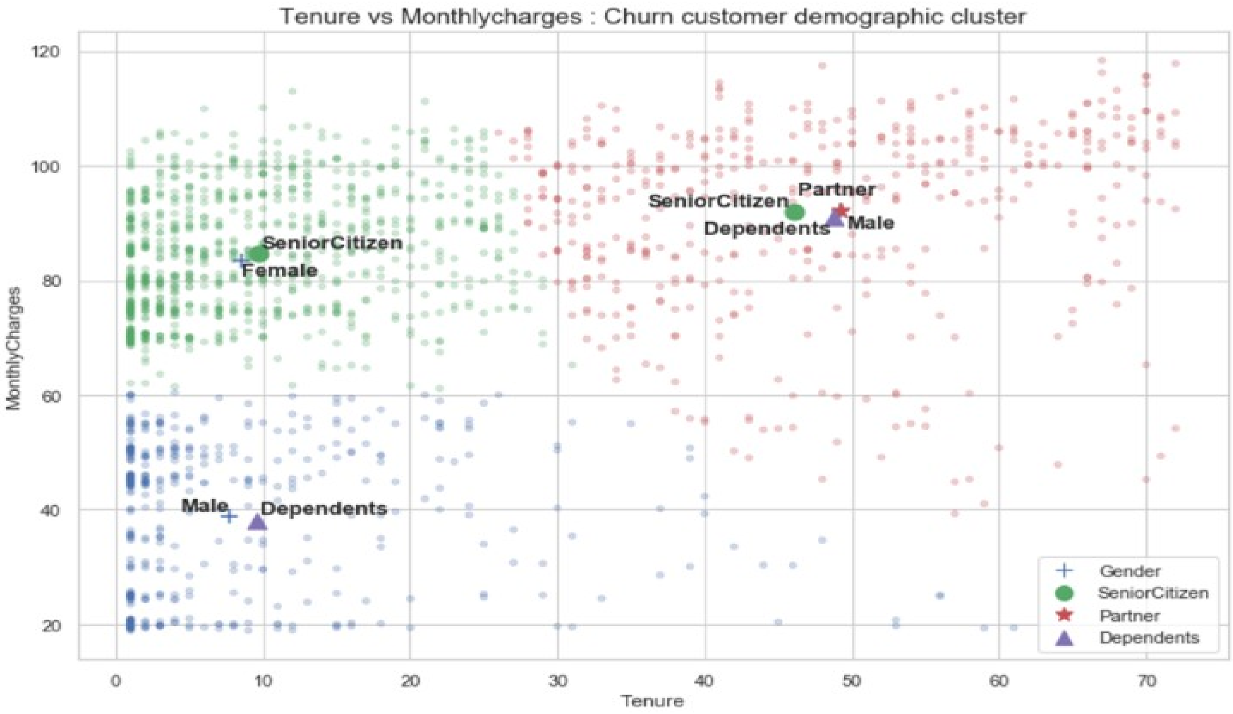


Figure 9.5 : Churn customer demographic cluster

Low Tenure and Low Monthly Charges customers

- Male, Dependents

Low Tenure and High Monthly Charges customers

- Senior citizens, Female

High Tenure and High Monthly Charges customers

- Male, Partner, Dependents and Senior Citizen

Based on **Account information,**

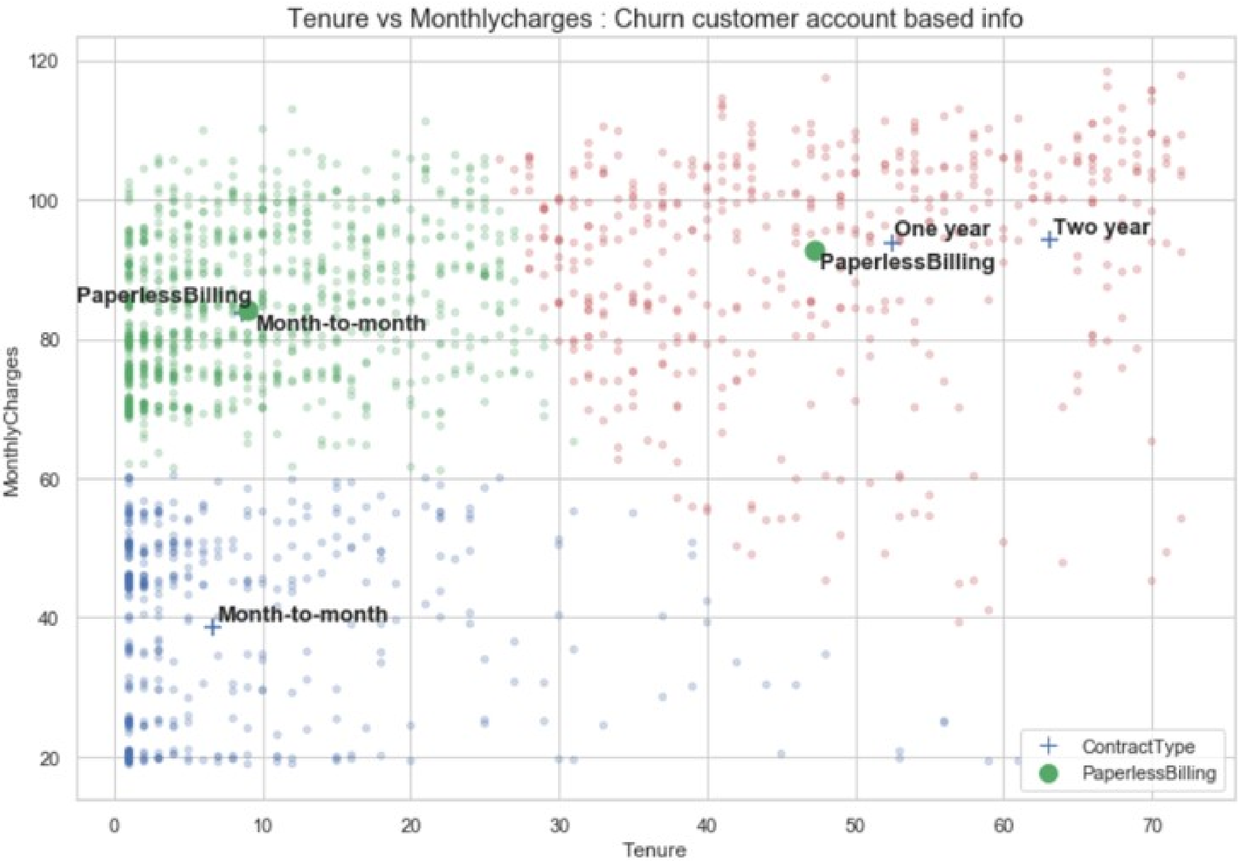


Figure 9.6 : Churn customer account based info

Low Tenure and Low Monthly Charges customers

- Month-to-month contract plan

Low Tenure and High Monthly Charges customers

- Paperless billing, Month-to-month contract plan

High Tenure and High Monthly Charges customers

- Paperless billing, One/Two year contract type

Based on **Usage information**,

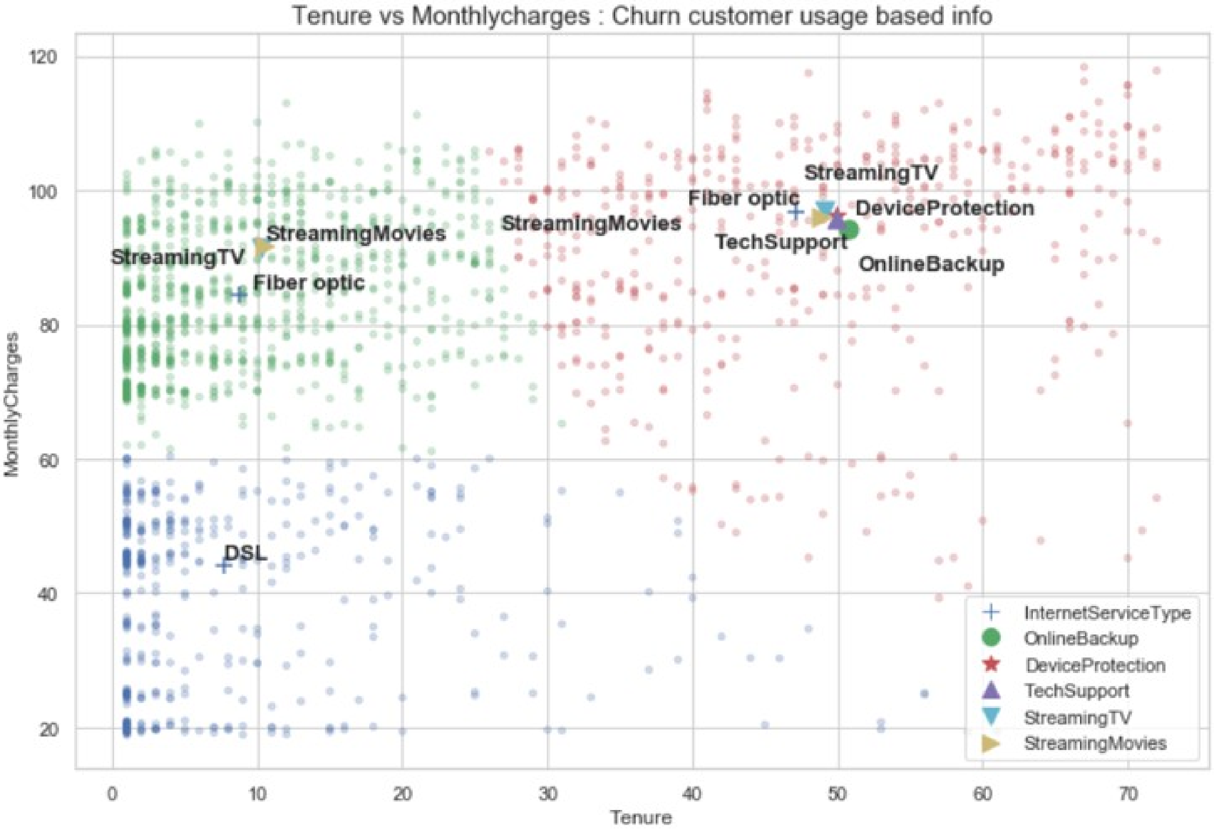


Figure 9.7 : Churn customer usage based info

Low Tenure and Low Monthly Charges customers

- Have DSL internet service

Low Tenure and High Monthly Charges customers

- Have Streaming TV / Streaming Movies, Fiber optic internet service

High Tenure and High Monthly Charges customers

- Online services like Online Backup, Device Protection and Tech Support, Fiber optic internet service, Have Streaming TV / Streaming Movies