

PREDICTION OF BANK CUSTOMER CHURN USING MACHINE LEARNING TECHNIQUE

Submitted in partial fulfillment of the requirements
for the award of
Bachelor of Engineering degree in Computer Science and Engineering
by
Gujjalapati Yamini (38110178)
Dasam Meenakshi (38110678)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119 ,MARCH – 2022



DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Gujjalapati Yamini (38110178)** and **Dasam Meenakshi (38110678)** who carried out the project entitled “**PREDICTION OF BANK CUSTOMER CHURN USING MACHINE LEARNING TECHNIQUE**” under my supervision from **December 2021 to May 2022**.

Internal Guide

Dr. B.U. Anu Barathi M.E., Ph.D.

Head of the Department

Dr. L.Lakshmanan M.E., Ph.D, Dr.S. Vigneshwari, M.E.,Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, Gujjalapati Yamini and Dasam Meenakshi hereby declare that the Project Report entitled “**PREDICTION OF BANK CUSTOMER CHURN USING MACHINE LEARNING TECHNIQUE**” done by us under the guidance of **Dr.**

B.U.Anu Barathi M.E,Ph.D..., is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering 2018-2022.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

G.Yamini

D. Meenaakshi

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E.,Ph.D., Dean**, School of Computing, **Dr.L.Lakshmanan M.E., Ph.D** and **Dr. S. Vigneshwari, M.E., Ph.D.,** Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and a deep sense of gratitude to my Project Guide **Dr.B.U. Anu Barathi M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project

Abstract

Now a -days there are a lot of service providers available in every business. There is no shortage of customers in any options. Mainly, in the banking sector when people want to keep their money safely they have a lot of options. As a result, customer churn and loyalty of customers have become a major problem for most banks. In this paper, a method that predicts customer churn in banking using Machine learning with ANN. This research promotes the exploration of the likelihood of churning by customer loyalty.

The logistic regression,random forest,decision tree and naive Bayes Machine Learning algorithms are used in this study. Keras and TensorFlow are ANN concepts that are also used in this study. This study is done on a dataset called churn modeling. The dataset was collected from Kaggle. The results are compared to find an appropriate model with higher accuracy. As a result, the Random Forest algorithm achieved higher accuracy than other algorithms. And accuracy was nearly 87%. The least accuracy was achieved by the Decision tree algorithm and it was 78.59% accuracy.

The number of service providers is increasing very rapidly in every business. These days, there is no shortage of options for customers in the banking sector when choosing where to put their money. As a result, customer churn and engagement have become one of the top issues for most banks. In this project, a method to predict the customer churn in a Bank, using machine learning techniques, which is a branch of artificial intelligence, is proposed. The research promotes the exploration

of the likelihood of churn by analyzing customer behavior. Customer Churn has become a major problem in all industries including the banking industry and banks have always tried to track customer interaction so that they can detect the customers who are likely to leave the bank

TABLE OF CONTENT

CHAPTER.NO	TITLE	PAGE NO
	ABSTRACT TABLE OF CONTENT LIST OF FIGURES LIST OF SYMBOLS	5 6 7 7
01	INTRODUCTION	11
02	LITERATURE SURVEY	15
03	METHODOLOGY 3.1 OBJECTIVE 3.2 LIST OF MODULES 3.3 SYSTEM ARCHITECTURE	18 20 21 23
04	RESULT AND DISCUSSION PERFORMANCE ANALYSIS 4.1 FEATURES 4.2 CODE	25 48 61

05	CONCLUSION AND FUTURE WORK	85
----	----------------------------	----

LIST OF FIGURES

S.NO	TITLE	PAGE.NO
01	SYSTEM ARCHITECTURE	22
02	WORKFLOW DIAGRAM	23
03	ER - DIAGRAM	25
04	MODULE DIAGRAM	29

1. INTRODUCTION

Churning means a customer who leaves one company and transfers to another company. It is not only a loss in income but also other negative effects on the operations and also mainly Customer Relation Management is very important for banking when the company considers it as they try to establish long-term relationships with customers and also it will lead to increase their customer base. The service provider's challenges are found in the behavior of the customer and their expectations. In the current generation, people are mostly educated compared to previous generations. So, the current generation of people is expecting more policies and their diverse demand for connectivity and innovation. This advanced knowledge is leading to changes in purchase behavior. This is a big challenge for current service providers to think innovatively to reach their expectations.

Private sectors need to recognize customers Liu and Shih strengthen this argument in their paper by indicating that increasing pressures on companies to develop new and innovative ideas in marketing, to meet customer expectations and increase loyalty and retention. For Customers, it is very easy to transfer their relations from one bank to another bank. Some customers might be keeping their relationship status null that means they will keep their account status inactive. By keeping this account inactive it might be the customer transferring their relationship with another bank. There are different types of customers are in the bank. Farmers are one of the major customers to the banks they will expect fewer monthly chargers as they were financially low. Businessperson, are also one of the major and important customers because a lot of transactions with huge amount is done by them only usually. These customers will expect better service quality. One of the most important categories was Middle-class customers, mostly in every bank these peoples are more than the type of customers. These people will expect fewer monthly charges, better service quality, and new policies.

So, maintaining different types of customers is not that easy. They need to consider customers and their needs to resolve these challenges delivering reliable service on 5 time and within budget to customers. While maintaining a good working partnership with them is another significant challenge for them. If they failed to resolve these

challenges this may cause churning. Recruiting a new customer is more expensive and harder than keeping already customers. Customers holding on the other hand is usually more expensive because they have already gained the confidence and loyalty of present customers. So, the need for a system that can predict customer churn effectively in the early stages is very important for any banking. This paper aims at a framework that can predict the customer churning banking sectors using some machine learning algorithms with ANN.

Existing System:

Predicting player behaviour and customer churn is one of the central and most common challenges in game analytics. A crucial stage in developing customer churn prediction model is feature engineering. In the mobile gaming field, features are commonly constructed from the raw behavioural telemetry data which leads to challenges related to the establishment of meaningful features and comprehensible feature frameworks. This research proposes an extended Recency, Frequency, and Monetary value (RFM) feature framework for churn prediction in the mobile gaming field by incorporating features related to user Lifetime, Intensity and Rewards (RFMLIR). The proposed framework is verified by exploring behavioral differences between churners and non-churners within the established framework for different churn definitions and definition groups by applying robust exploratory methods and developing univariate and multivariate churn prediction models. Although feature importance varies among churn definitions, long term frequency feature stands out as the most important feature. The top five most important features distinguished by the multivariable churn prediction models include long and short term frequency features, monetary, intensity and lifetime

Disadvantages:

- They done only Feature Engineering and analysis, no a practical model
- No predictive AI model is build.

MACHINE LEARNING

Machine learning is to predict the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves the use of specialized algorithms. It feeds the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data. Machine learning can be roughly separated into three categories. There is supervised learning, unsupervised learning and reinforcement learning. Supervised learning programs are both given the input data and the corresponding labeling to learn data has to be labeled by a human being beforehand. Unsupervised learning has no labels. It provided the learning algorithm. This algorithm has to figure out the clustering of the input data. Finally, Reinforcement learning dynamically interacts with its environment and it receives positive or negative feedback to improve its performance.

Data scientists use many different kinds of machine learning algorithms to discover patterns in python that lead to actionable insights. At a high level, these different algorithms can be classified into two groups based on the way they “learn” about data to make predictions: supervised and unsupervised learning. Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function from input variables(X) to discrete output variables(y). In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observations. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, biometric identification, document classification etc.



Process Of Machine Learning

supervised machine learning is the majority of practical machine learning using supervised learning. Supervised learning is where you have input variables (X) and an output variable (y) and use an algorithm to learn the mapping function from the input to the output is $y = f(X)$. The goal is to approximate the mapping function so well that when you have new input data (X) that you can predict the output variables (y) for that data. Techniques of Supervised Machine Learning algorithms include **logistic regression**, **multi-class classification**, **Decision Trees** and **support vector machines etc.** Supervised learning requires that the data used to train the algorithm is already labeled with correct answers. Supervised learning problems can be further grouped into **Classification** problems. This problem has as goal the construction of a succinct model that can predict the value of the dependent attribute from the attribute variables. The difference between the two tasks is the fact that the dependent attribute is numerical for categorical for classification. A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes. A classification problem is when the output variable is a category, such as “red” or “blue”.

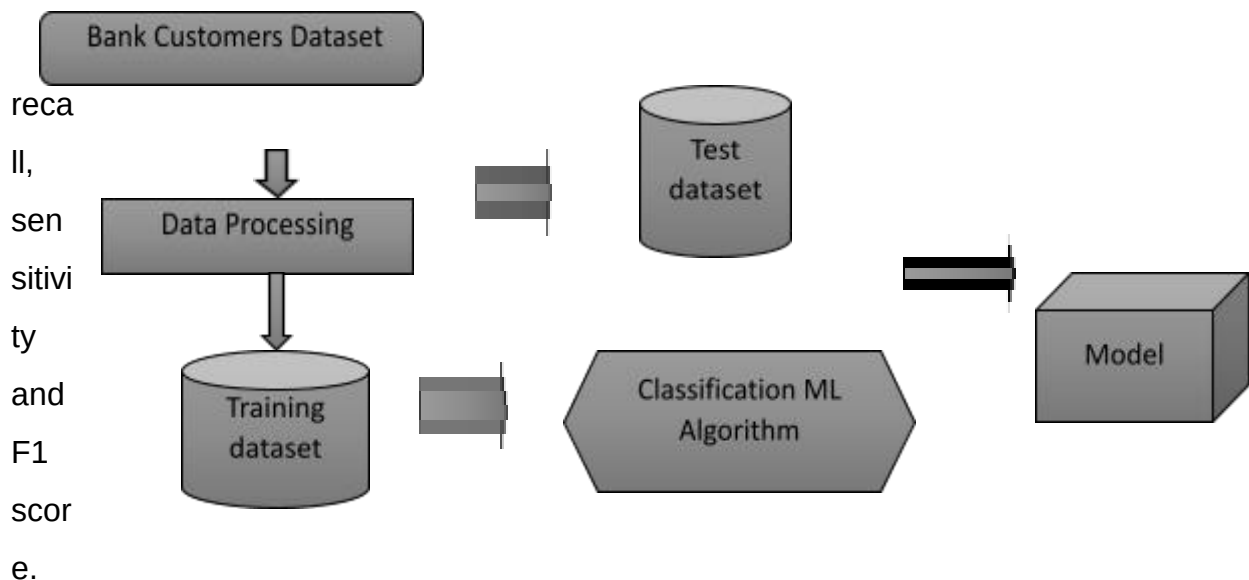
Preparing the Dataset :

This dataset contains 10000 records of features extracted from Bank Customer Data, which were then classified into 2 classes:

- Exit
- Not Exit

Proposed System:

The proposed method is to build a BANK CUSTOMER CHURN Prediction using Machine learning Technique. We are going to develop a AI based model, we need data to train our model. We can use BANK CUSTOMER Dataset in order to train the model. To use this dataset, we need to understand what the intents that we are going to train are. An intent is the intention of the user interacting with a predictive model or the intention behind each Data that the Model receives from a particular user. According to the domain that you are developing an AI solution, these intents may vary from one solution to another. The strategy is to define different intents and make training samples for those intents and train your AI model with those training sample data as model training data and intents as model training categories. The model is build using the process of vectorisation where the vectors made to understand the data. To use different Algorithm we can get a better AI model and best accuracy. After building a model we evaluate the model using different metrics like confusion metrics, precision ,



Architecture of Proposed model

2.Literature survey

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is a secondary source and discusses published information in a particular subject area and sometimes information in a particular subject area within a certain time period. Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

A summary is a recap of important information about the source, but a synthesis is a reorganization, reshuffling of information. It might give a new interpretation of old material or combine new with old interpretations or it might trace the intellectual progression of the field, including major debates. Depending on the situation, the literature review may evaluate the sources and advise the reader on the most pertinent or relevant of them

A comparison of machine learning techniques for customer churn prediction

Praveen Asthana 2018 We present a comparative study on the most popular machine learning methods applied to the challenging problem of customer churning prediction in the telecommunications industry. In the first phase of our experiments, all models were applied and evaluated using cross-validation on a popular, public domain dataset. In the second phase, the performance improvement offered by boosting was studied. In order to determine the most efficient parameter combinations we performed a series of Monte Carlo simulations for each method and for a wide range of parameters. Our results demonstrate clear superiority of the boosted versions of the models against the plain (non-boosted) versions. The best overall classifier was the SVM-POLY using AdaBoost with accuracy of almost 97% and F-measure over 84%. Customer Churn Analysis in Banking Sector G. Jignesh Chowdary¹ , Suganya. G ² , Premalatha. M³2019 The role of ICT in the banking sector is a crucial part of the development of nations. The development of the banking sector mostly depends on its valuable customers. So, customer churn

analysis is needed to determine customers whether they are at risk of leaving or worth retaining. From an organizational point of view, gaining new customers is usually more difficult or more expensive than retaining existing customers. So, customer churn prediction has been popular in the banking industry. By reducing customer churn or attrition, the commercial banks gain not only more profits but also enhancing core competitiveness among the competitors. Although many researchers proposed many single prediction models and some hybrid models, accuracy is still weak and computation time of some algorithms is still increased. In this research, the churn prediction model of classifying bank customers is built by using the hybrid model of k-means and Support Vector Machine data mining methods on bank customer churn dataset to overcome the instability and limitations of single prediction model and predict churn trend of high value users..

Developing a prediction model for customer churn from electronic banking services using data mining Abbas Keramati , Hajar Ghaneei and Seyed Mohammad Mirmohammadi. 2016 Given the importance of customers as the most valuable assets of organizations, customer retention seems to be an essential, basic requirement for any organization. Banks are no exception to this rule. The competitive atmosphere within which electronic banking services are provided by different banks increases the necessity of customer retention. Methods: Being based on existing information technologies which allow one to collect data from organizations' databases, data mining introduces a powerful tool for the extraction of knowledge from huge amounts of data. In this research, the decision tree technique was applied to build a model incorporating this knowledge. Results: The results represent the characteristics of churned customers. Conclusions: Bank managers can identify churners in future using the results of the decision tree. They should be provide some strategies for customers whose features are getting more likely to churner's features.

A Critical Examination of Different Models for Customer Churn Prediction using Data Mining Seema, Gaurav Gupta 2019 Due to competition between online retailers, the need for providing improved customer service has grown rapidly. In addition to reduction in sales due to loss of customers, more investments are needed to be done to attract new customers. Companies now are working continuously to improve

their perceived quality by way of giving timely and quality service to their customers. Customer churn has become one of the primary challenges that many firms are facing nowadays. Several churn prediction models and techniques are proposed previously in literature to predict customer churn in areas such as finance, telecom, banking etc. Researchers are also working on customer churn prediction in e-commerce using data mining and machine learning techniques. In this paper, a comprehensive review of various models to predict customer churn in e-commerce data mining and machine learning techniques has been presented. A critical review of recent research papers in the field of customer churn prediction in e-commerce using data mining has been done. Thereafter, important inferences and research gaps after studying the literature are presented. Finally, the research significance and concluding remarks are described in the end.

bank customer retention prediction and customer ranking based on deep neural networks dr a.p.jagadeesan ph.d 2020 Retention of customers is a major concern in any industry. Customer churn is an important metric that gives the hard truth about the retention percentage of customers. A detailed study about the existing models for predicting the customer churn is made and a new model based on Artificial Neural Network is proposed to find the customer churn in banking domain. The proposed model is compared with the existing machine learning models. Logistic regression, Decision Tree and random forest mechanisms are the baseline models that are used for comparison, the performance metrics that were compared are accuracy, precision, recall and F1 score. It has been observed that the artificial neural network model performs better than the logistic regression model and decision tree model. But when the results are compared with the random forest model considerable difference is not noted. The proposed model differs from the existing models in a way that it can rank the customers in the order in which they would leave the organization.

3.Methodology

This section explains the various works that have been done in order to predict the customer churn. It includes machine learning models. In addition to the conventional data used for predicting the customer churn, the authors have added data from the various sources. It includes the conversation of the customers through

phone, the websites and products the customer has viewed, interactive voice data and other financial data. Binary Classification model is used for predicting the customer churn. Though a good improvement is noticed with this model, the data that has been used in this is not commonly available at all times. Churn prediction is a binary classification problem; the authors specified that from the studies it has been observed that there is no proper means of measuring the certainty of the classifier that has been employed for churn prediction. It has also been observed that the accuracy of the classifiers differs for different zones of the dataset.

Project Goals

□ Exploration data analysis of variable identification

- Loading the given dataset
- Import required libraries packages
- Analyze the general properties
- Find duplicate and missing values
- Checking unique and count values

□ Uni-variate data analysis

- Rename, add data and drop the data
- To specify data type

□ Exploratory data analysis of bi-variate and multivariate

- Plot diagram of pairplot, heatmap, bar chart and Histogram

□ Method of Outlier detection with feature engineering

- Pre-processing the given dataset
- Splitting the test and training dataset
- Comparing the Decision tree and Logistic regression model and random forest etc.

□ Comparing algorithm to predict the result

- Based on the best accuracy

Objectives

The goal is to develop a machine learning model for Bank Churn Prediction, to potentially replace the updatable supervised machine learning classification models by predicting results in the form of best accuracy by comparing supervised algorithms.

Scope of the Project

Here the scope of the project is that integration of Bank support with computer-based records could reduce, enhance Bank safety, decrease the customer churn, and improve Bank Customer Support. This suggestion is promising as data modeling and analysis tools, e.g., data mining, have the potential to generate a knowledge-rich environment which can help to significantly improve the quality of Bank support.

Feasibility study:

Data Wrangling

In this section of the report will load in the data, check for cleanliness, and then trim and clean the given dataset for analysis. Make sure that the document steps carefully and justify cleaning decisions.

Data collection The data set collected for predicting given data is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using Different algorithms on the Training set and based on the test result accuracy, Test set prediction is done.

Preprocessing

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data needs to be preprocessed so as to improve

the efficiency of the algorithm. The outliers have to be removed and also variable conversion needs to be done.

Building the classification model

The prediction of Heart attack, A high accuracy prediction model is effective because of the following reasons: It provides better results in classification problems.

It is strong in preprocessing outliers, irrelevant variables, and a mix of continuous, categorical and discrete variables.

It produces out of bag estimate error which has proven to be unbiased in many tests and it is relatively easy to tune with.

List of Modules: □

Data Pre-
processing

Data Analysis of Visualization

- Comparing Algorithm with prediction in the form of best accuracy result
- Deployment Using Flask

Project Requirements

Requirements are the basic constraints that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed. 1. Functional requirements

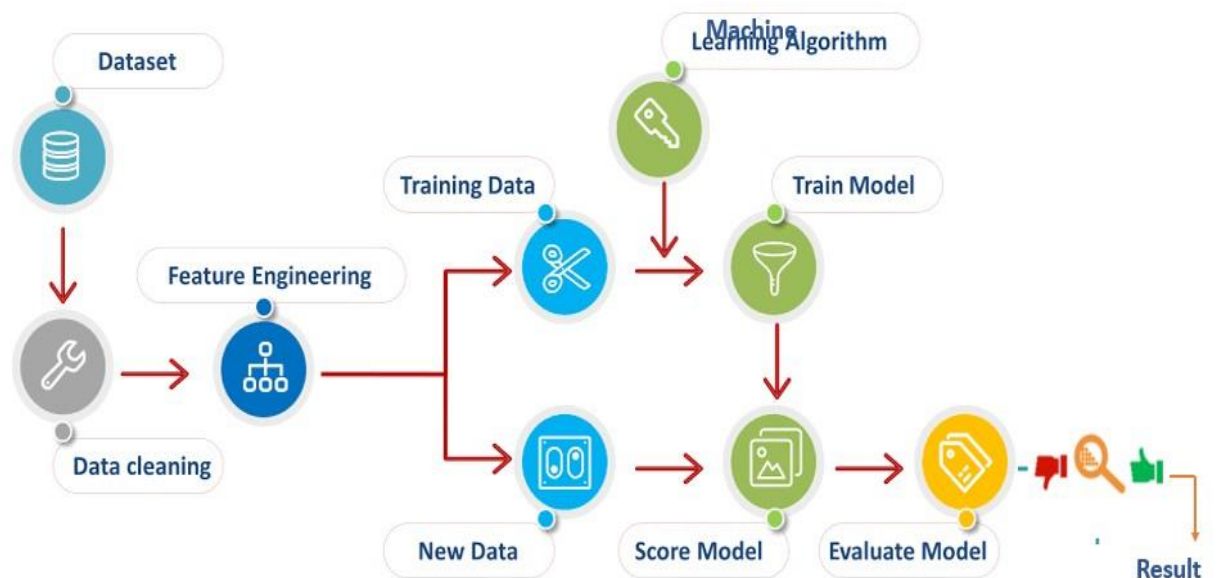
2. Non-Functional requirements

3. Environment requirementsA. Hardware requirements

B. software requirements

Functional requirements:

The software requirements specification is a technical specification of



requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details follow the special libraries like sk-learn, pandas, numpy, matplotlib and seaborn.

Non-Functional Requirements:

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction the result

Environmental Requirements:

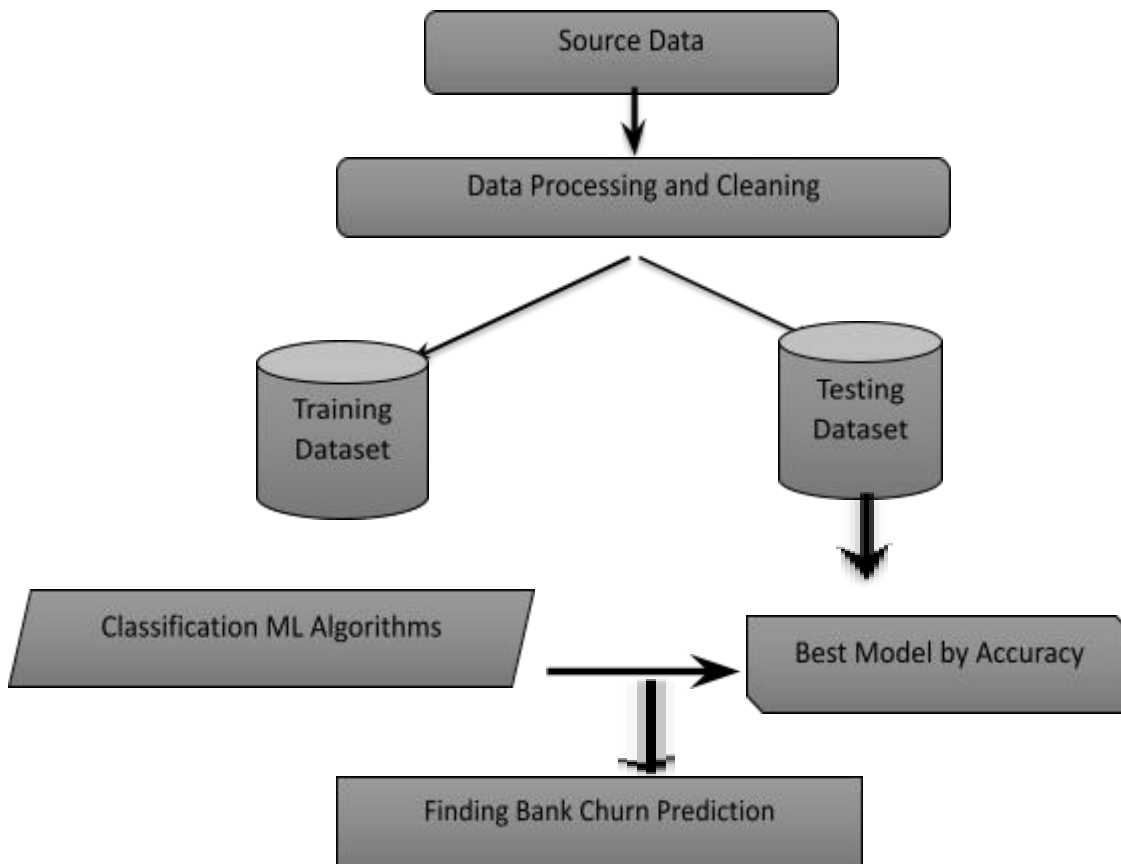
1. Software Requirements :

Operating System : Windows

Tool : Anaconda with Jupyter Notebook

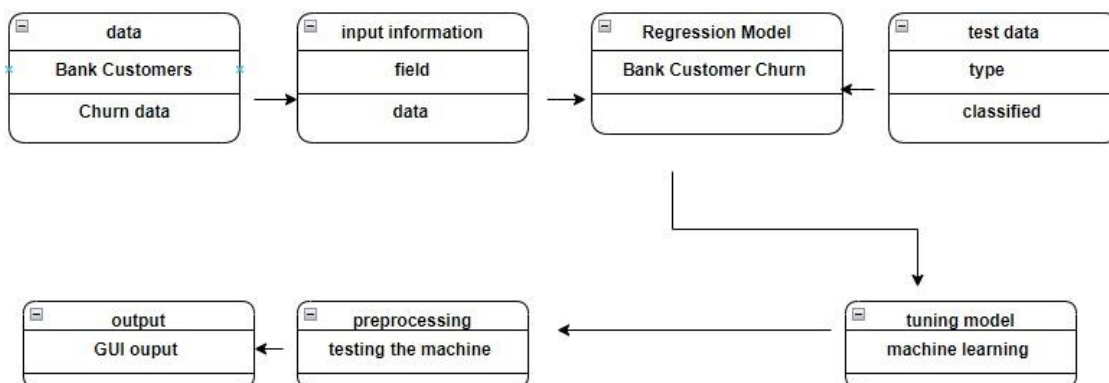
SYSTEM ARCHITECTURE

Workflow diagram



Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So, it can say that uses cases are nothing but the system functionalities written in an organized manner.

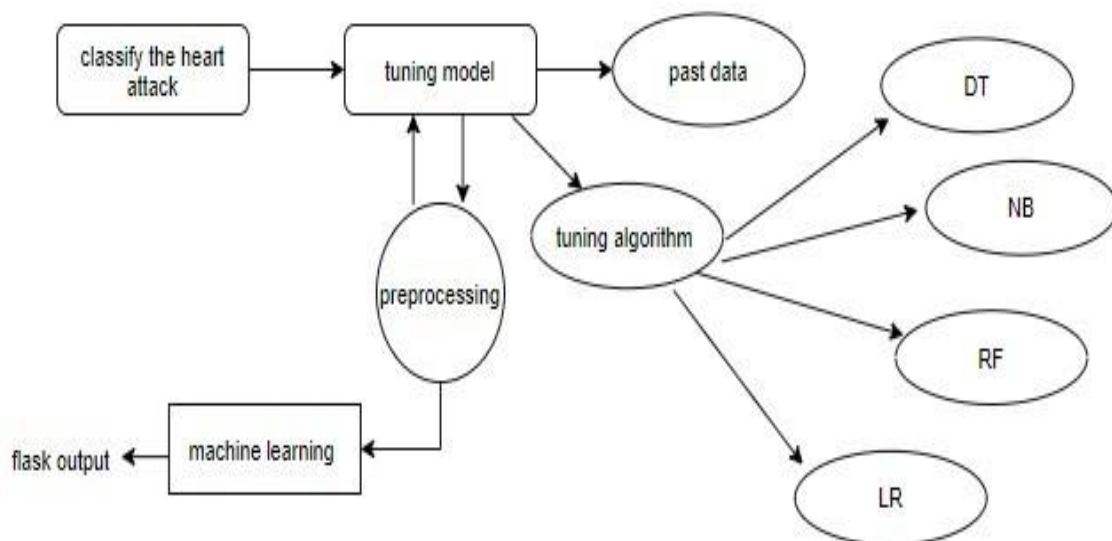
Class Diagram



Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams

represent the whole system. The name of the class diagram should be meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance Responsibility (attributes and methods) of each class should be clearly identified for each class minimum number of properties should be specified and because unnecessary properties will make the diagram complicated. Use notes whenever required to describe some aspect of the diagram and at the end of the drawing it should be understandable to the developer/coder. Finally, before making the final version, the diagram should be drawn on plain paper and reworked as many times as possible to make it correct.

Entity Relationship Diagram (ERD)



An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and be used as the foundation for a relational database. Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a

relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

4.RESULT AND DISCUSSION:

Data Pre-processing

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset

while	tuning			model				hyper			parameters.		
RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10
5	6	15574012	Chu	645	Spain	Male	44	8	113755.78	2	1	0	149756.71
6	7	15592531	Bartlett	822	France	Male	50	7	0.00	2	1	1	10062.80
7	8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.88
8	9	15792365	He	501	France	Male	44	4	142051.07	2	0	1	74940.50
9	10	15592389	H?	684	France	Male	27	2	134603.88	1	1	1	71725.73

The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers use this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model.

A number of different **data cleaning** tasks using Python's Pandas library and specifically, it focus on probably the biggest data cleaning task, **missing values** and it able to

more quickly clean data. It wants to **spend less time cleaning data**, and more time exploring and modeling.

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000
mean	5000.50000	1.569094e+07	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881
std	2886.89568	7.193619e+04	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818
min	1.00000	1.556570e+07	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000
25%	2500.75000	1.562853e+07	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000
50%	5000.50000	1.569074e+07	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000
75%	7500.25000	1.575323e+07	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500
max	10000.00000	1.581569e+07	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It's important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it's important to understand the sources of missing data. Here are some typical reasons why some data is missing

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10000 entries, 0 to 9999
Data columns (total 14 columns):
#   Column              Non-Null Count  Dtype
---  -
0   RowNumber           10000 non-null  int64
1   CustomerId          10000 non-null  int64
2   Surname             10000 non-null  object
3   CreditScore         10000 non-null  int64
4   Geography           10000 non-null  object
5   Gender              10000 non-null  object
6   Age                 10000 non-null  int64
7   Tenure              10000 non-null  int64
8   Balance             10000 non-null  float64
9   NumOfProducts       10000 non-null  int64
10  HasCrCard           10000 non-null  int64
11  IsActiveMember      10000 non-null  int64
12  EstimatedSalary     10000 non-null  float64
13  Exited              10000 non-null  int64
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

- User forgot to fill in a field.

- Data was lost while transferring manually from a legacy database.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
RowNumber	1.000000	0.004202	0.005840	0.000783	-0.006495	-0.009067	0.007246	0.000599	0.012044	-0.005988	-0.016571
CustomerId	0.004202	1.000000	0.005308	0.009497	-0.014883	-0.012419	0.016972	-0.014025	0.001665	0.015271	-0.006248
CreditScore	0.005840	0.005308	1.000000	-0.003965	0.000842	0.006268	0.012238	-0.005458	0.025651	-0.001384	-0.027094
Age	0.000783	0.009497	-0.003965	1.000000	-0.009997	0.028308	-0.030680	-0.011721	0.085472	-0.007201	0.285323
Tenure	-0.006495	-0.014883	0.000842	-0.009997	1.000000	-0.012254	0.013444	0.022583	-0.028362	0.007784	-0.014001
Balance	-0.009067	-0.012419	0.006268	0.028308	-0.012254	1.000000	-0.304180	-0.014858	-0.010084	0.012797	0.118533
NumOfProducts	0.007246	0.016972	0.012238	-0.030680	0.013444	-0.304180	1.000000	0.003183	0.009612	0.014204	-0.047820
HasCrCard	0.000599	-0.014025	-0.005458	-0.011721	0.022583	-0.014858	0.003183	1.000000	-0.011866	-0.009933	-0.007138
IsActiveMember	0.012044	0.001665	0.025651	0.085472	-0.028362	-0.010084	0.009612	-0.011866	1.000000	-0.011421	-0.156128
EstimatedSalary	-0.005988	0.015271	-0.001384	-0.007201	0.007784	0.012797	0.014204	-0.009933	-0.011421	1.000000	0.012097
Exited	-0.016571	-0.006248	-0.027094	0.285323	-0.014001	0.118533	-0.047820	-0.007138	-0.156128	0.012097	1.000000

Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:

import libraries for access and functional purpose and read the given dataset

General Properties of Analyzing the given dataset

Display the given dataset in the form of data frame

show columns

shape of the data frame

To describe the data frame

Checking data type and information about dataset

Checking for duplicate data

Checking Missing values of data frame

Checking unique values of data frame

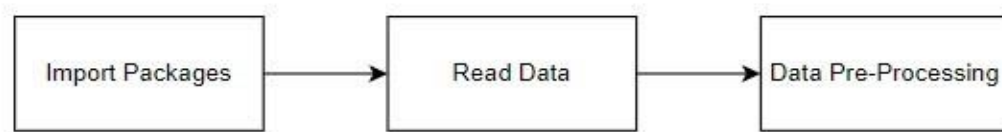
Checking count values of data frame

Rename and drop the given data frame

To specify the type of values

To create extra columns

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data output :

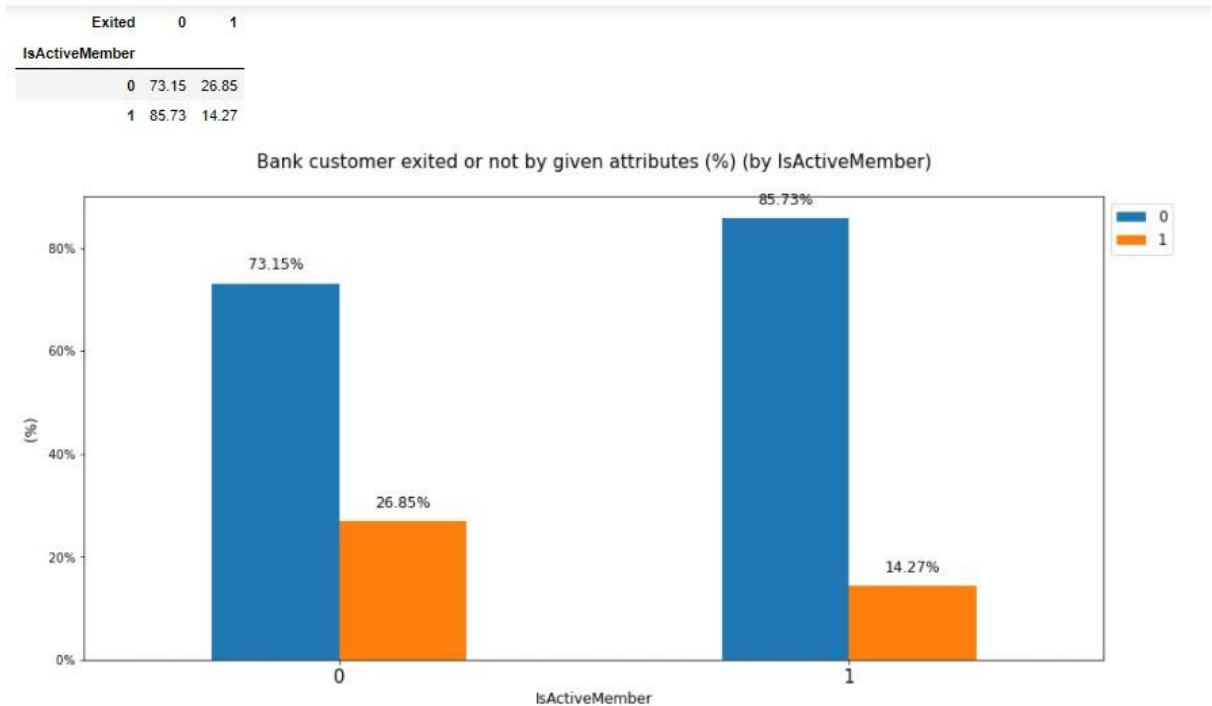
removing noisy data

Data Validation/ Cleaning/Preparing Process

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

Exploration data analysis of visualization

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding.

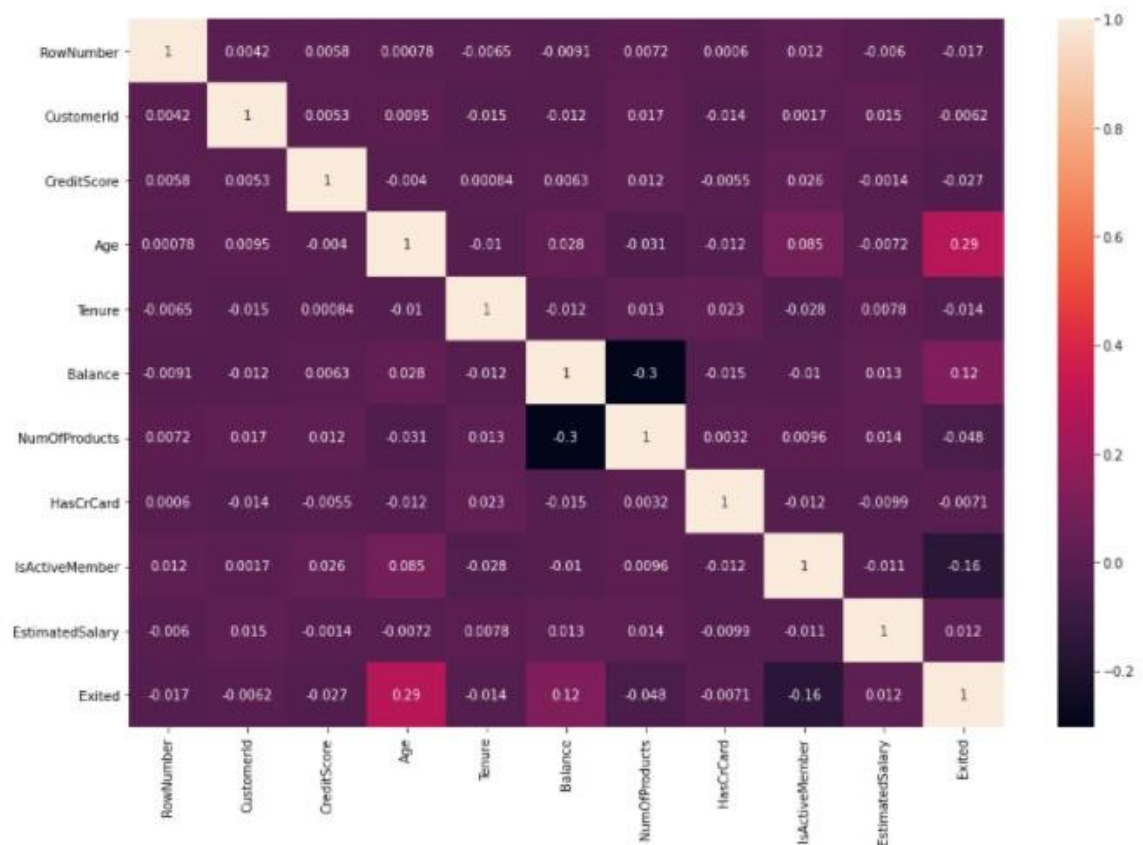
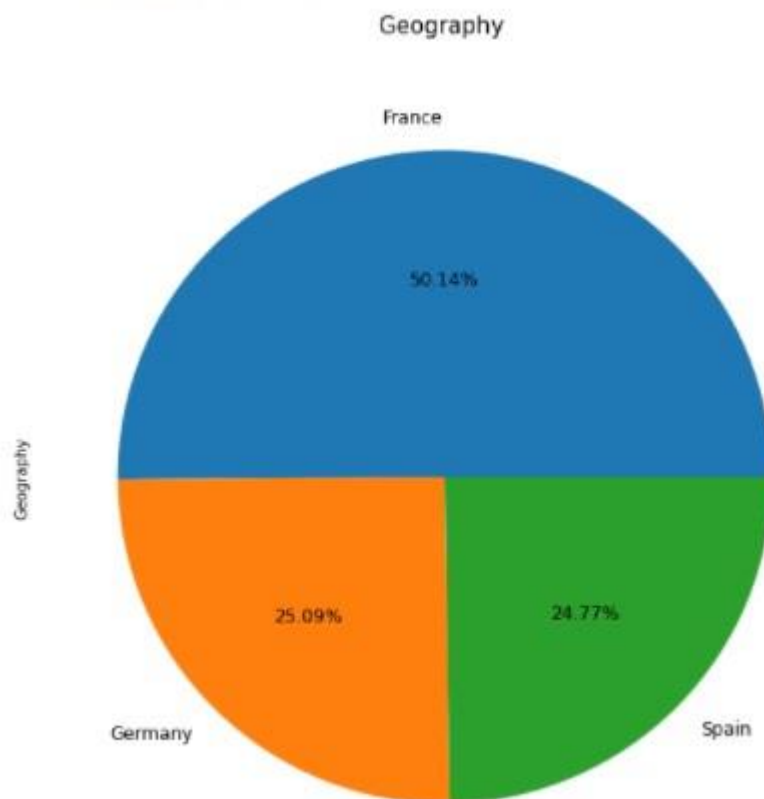


This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

```

France    50.14
Germany   25.09
Spain     24.77
Name: Geography, dtype: float64

```

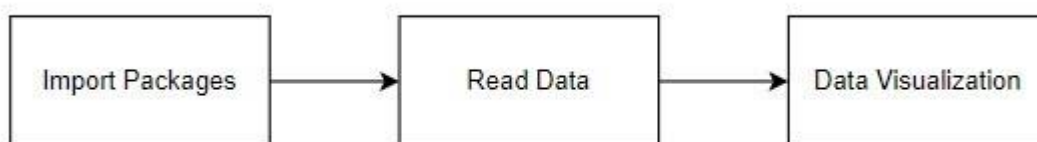


Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

How to chart time series data with line plots and categorical quantities with bar charts.

How to summarize data distributions with histograms and box plots.

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data output :

visualized data

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set. And another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in a given dataset.

False Positives (FP): A person who will pay is predicted as a defaulter. When the actual class is no and the predicted class is yes. E.g. if the actual class says this passenger did not survive but the predicted class tells you that this passenger will survive.

False Negatives (FN): A person who default predicted as payer. When the actual class is yes but the predicted class is no. E.g. if the actual class value indicates that this passenger survived and the predicted class tells you that passenger will die.

True Positives (TP): A person who will not pay is predicted as a defaulter. These are the correctly predicted positive values which means that the value of the actual class is yes and the value of predicted class is also yes. E.g. if the actual class value indicates that this passenger survived and the predicted class tells you the same thing.

True Negatives (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. If the actual class says this passenger did not survive and the predicted class tells you the same thing.

Comparing Algorithm with prediction in the form of best accuracy result

It is important to compare the performance of multiple different machine learning algorithms consistently and to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When having a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

In the example below 4 different algorithms are compared:

Logistic Regression

Random Forest

Decision Tree Classifier

Naive Bayes

The K-fold cross validation procedure is used to evaluate each algorithm, importantly configured with the same random seed to ensure that the same splits to the training data are performed and that each algorithm is evaluated in precisely the same way. Before comparing algorithms, build a Machine Learning Model using Scikit-Learn libraries. In this library package, we have to do preprocessing, linear model with logistic regression method, cross validating by KFold method, ensemble with random forest method and tree with decision tree classifier. Additionally, splitting the train set and test set. To predict the result by comparing accuracy.

Prediction result by accuracy:

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. It needs the output of the algorithm to be classified variable data. Higher accuracy predicting the result is a logistic regression model by comparing the best accuracy.

$$\text{True Positive Rate(TPR)} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{False Positive rate(FPR)} = \text{FP} / (\text{FP} + \text{TN})$$

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy calculation:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positives and false negatives are almost the same.

Precision: The proportion of positive predictions that are actually correct.

$$\text{Precision} = TP / (TP + FP)$$

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that are labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

$$\text{Recall} = TP / (TP + FN)$$

Recall(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If

the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

General Formula:

F- Measure = $2TP / (2TP + FP + FN)$ **F1-**

Score Formula:

F1 Score = $2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$

ALGORITHM AND TECHNIQUES

Algorithm Explanation

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observations. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, biometric identification, document classification etc. In Supervised Learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data.

Used Python Packages:

sklearn:

- In python, sklearn is a machine learning package which includes a lot of ML algorithms.
- Here, we are using some of its modules like `train_test_split`, `DecisionTreeClassifier` or `Logistic Regression` and `accuracy_score`.

NumPy:

- It is a numeric python module which provides fast maths functions for calculations.

- It is used to read data in numpy arrays and for manipulation purposes.

Pandas:

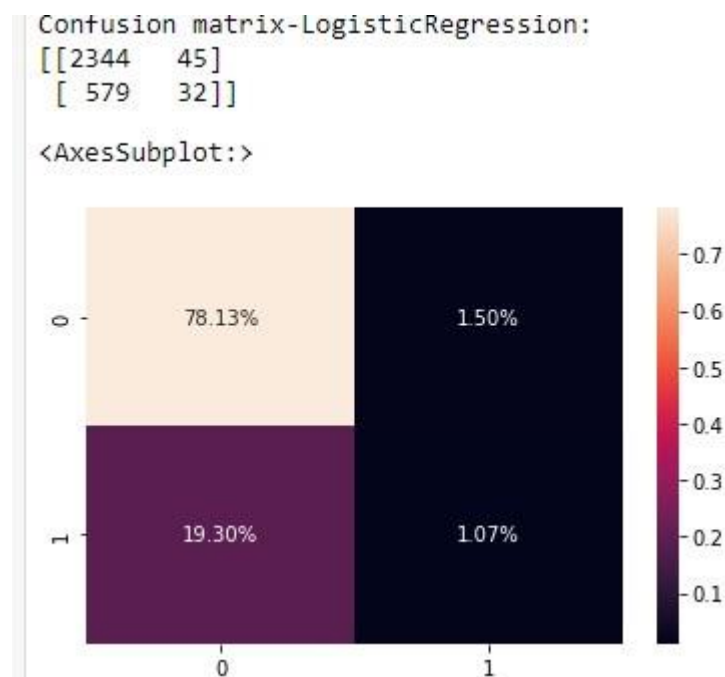
- Used to read and write different files.
- Data manipulation can be done easily with data frames.

Matplotlib:

- Data visualization is a useful way to help with identifying the patterns from a given dataset.
- Data manipulation can be done easily with data frames.

Logistic Regression

It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).



In other words, the logistic regression model predicts $P(Y=1)$ as a function of X .

Logistic regression Assumptions:

Binary logistic regression requires the dependent variable to be binary.

For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.

Only the meaningful variables should be included.

The independent variables should be independent of each other. That is, the model should have little.

The independent variables are linearly related to the log odds.

Logistic regression requires quite large sample sizes.

Classification report of Logistic Regression Results:

	precision	recall	f1-score	support
0	0.80	0.98	0.88	2389
1	0.42	0.05	0.09	611
accuracy			0.79	3000
macro avg	0.61	0.52	0.49	3000
weighted avg	0.72	0.79	0.72	3000

Confusion Matrix result of Logistic Regression is:

```
[[2344  45]
 [ 579  32]]
```

Sensitivity : 0.981163666806195

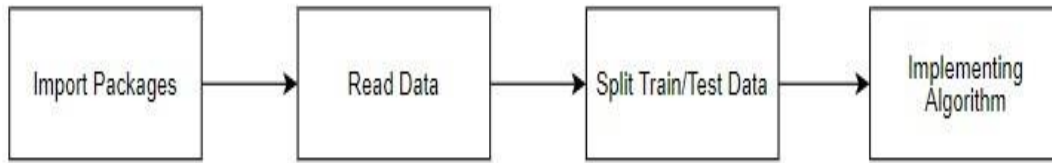
Specificity : 0.05237315875613748

Cross validation test results of accuracy:

```
[0.7915 0.791 0.793 0.789 0.7885]
```

Accuracy result of Logistic Regression is: 79.06

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data output :

getting accuracy

Random Forest Classifier

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over-fitting to their training set. Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or the same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision *trees*, resulting in a *forest of trees*, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

Classification report of Random Forest Classifier Results:

	precision	recall	f1-score	support
0	0.87	0.97	0.92	2389
1	0.78	0.44	0.56	611
accuracy			0.86	3000
macro avg	0.82	0.70	0.74	3000
weighted avg	0.85	0.86	0.84	3000

Confusion Matrix result of Random Forest Classifier is:

```
[[2312  77]
 [ 345 266]]
```

Sensitivity : 0.9677689409794893

Specificity : 0.4353518821603928

Cross validation test results of accuracy:

```
[0.8605 0.869 0.858 0.8605 0.8565]
```

Accuracy result of Random Forest Classifier is: 86.09

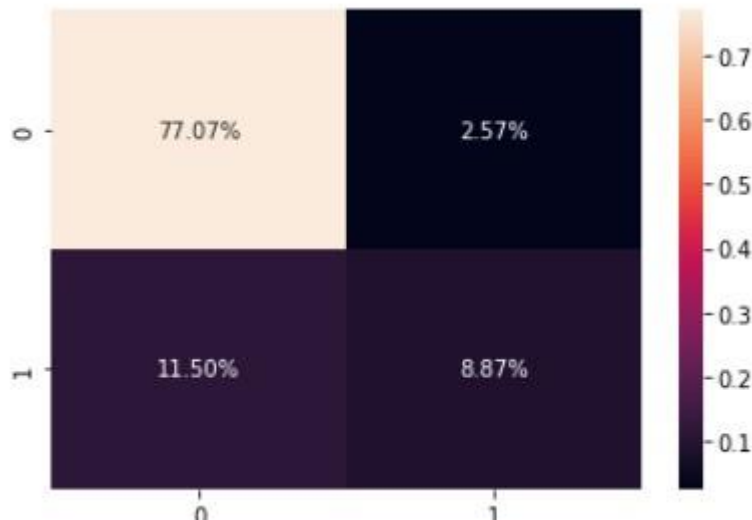
The following are the basic steps involved in performing the random forest algorithm:

Pick N random records from the dataset.

Build a decision tree based on these N records.

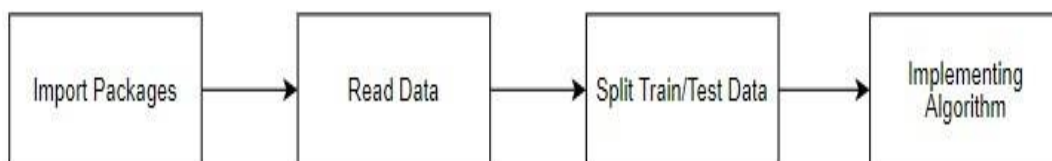
Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

```
Confusion matrix-RandomForestClassifier:
[[2312  77]
 [ 345 266]]
<AxesSubplot:>
```



In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data output :

getting accuracy

Decision Tree Classifier

It is one of the most powerful and popular algorithms. Decision-tree algorithms fall under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. Assumptions of Decision tree:

```
Classification report of Decision Tree Classifier Results:
```

	precision	recall	f1-score	support
0	0.87	0.87	0.87	2389
1	0.50	0.49	0.49	611
accuracy			0.79	3000
macro avg	0.68	0.68	0.68	3000
weighted avg	0.79	0.79	0.79	3000

```
Confusion Matrix result of Decision Tree Classifier is:
```

```
[[2084 305]
 [ 312 299]]
```

```
Sensitivity : 0.872331519464211
```

```
Specificity : 0.48936170212765956
```

```
Cross validation test results of accuracy:
```

```
[0.7815 0.8025 0.786 0.785 0.789 ]
```

```
Accuracy result of Decision Tree Classifier is: 78.88000000000001
```

At the beginning, we consider the whole training set as the root.

Attributes are assumed to be categorical for information gain, attributes are assumed to be continuous.

On the basis of attribute values records are distributed recursively.

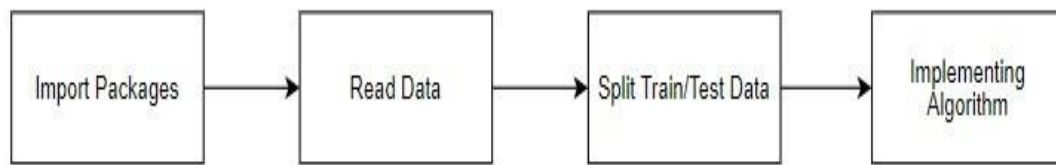
We use statistical methods for ordering attributes as root or internal node.



Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. Decision trees build classification or regression models in the form of a tree structure. It utilizes an if-then rule set which is mutually exclusive and exhaustive for classification. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covered by the rules are removed.

This process is continued on the training set until meeting a termination condition. It is constructed in a top-down recursive divide-and-conquer manner. All the attributes should be categorical. Otherwise, they should be discretized in advance. Attributes in the top of the tree have more impact in the classification and they are identified using the information gain concept. A decision tree can be easily over-fitted generating too many branches and may reflect anomalies due to noise or outliers.

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data output :

getting accuracy

Naive Bayes algorithm:

The Naive Bayes algorithm is an intuitive method that uses the probabilities of each attribute belonging to each class to make a prediction. It is the supervised learning approach you would come up with if you wanted to model a predictive modeling problem probabilistically.

Naive Bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. This is a strong assumption but results in a fast and effective method.

The probability of a class value given a value of an attribute is called the conditional probability. By multiplying the conditional probabilities together for each attribute for a given class value, we have a probability of a data instance belonging to that class. To make a prediction we can calculate probabilities of the instance belonging to each class and select the class value with the highest probability.

Classification report of Naive Bayes Results:

	precision	recall	f1-score	support
0	0.80	0.97	0.88	2389
1	0.40	0.08	0.13	611
accuracy			0.79	3000
macro avg	0.60	0.52	0.50	3000
weighted avg	0.72	0.79	0.73	3000

Confusion Matrix result of Naive Bayes is:

```
[[2320  69]
 [ 565  46]]
```

Sensitivity : 0.9711176224361657

Specificity : 0.07528641571194762

Cross validation test results of accuracy:

```
[0.782 0.7815 0.788 0.785 0.7815]
```

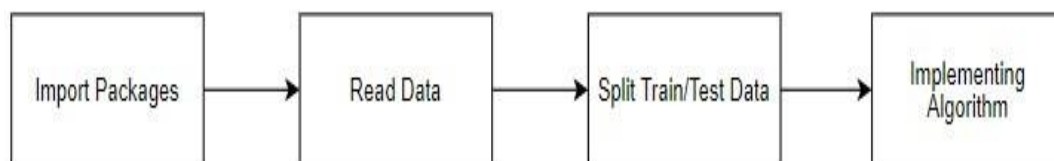
Accuracy result of Naive Bayes is: 78.36

Naive Bayes is a statistical classification technique based on Bayes Theorem. It is one of the simplest supervised learning algorithms. Naive Bayes classifier is the fast, accurate and reliable algorithm. Naive Bayes classifiers have high accuracy and speed on large datasets.

Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features. For example, a loan applicant is desirable or not depending on his/her income, previous loan and transaction history, age, and location. Even if these features are interdependent, these features are still considered independently. This assumption simplifies computation, and that's why it is considered naive. This assumption is called class conditional independence.



MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data output :

getting accuracy

Deployment

Flask (Web Framework) :

Flask is a micro web framework written in Python.

It is classified as a micro-framework because it does not require particular tools or

libraries.

BANK CUSTOMER CHURN

CreditScore
CreditScore

Balance
Balance

Geography
FRANCE

Gender
FEMALE

Age
Age

Tenure
Tenure

NumOfProducts
one

HasCreditCard
has credit card

IsActive
Not Active

EstimatedSalary
EstimatedSalary

Predict

It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

However, Flask supports extensions that can add application features as if they were implemented in Flask itself.

Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

Flask was created by Armin Ronacher of Pocoo, an international group of Python enthusiasts formed in 2004. According to Ronacher, the idea was originally an April Fool's joke that was popular enough to make into a serious application. The name is a play on the earlier Bottle framework.

When Ronacher and Georg Brand created a bulletin board system written in Python, the Pocoo projects Werkzeug and Jinja were developed.

In April 2016, the Pocoo team was disbanded and development of Flask and related libraries passed to the newly formed Pallets project.

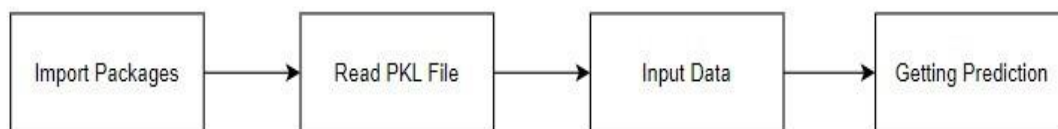
Flask has become popular among Python enthusiasts. As of October 2020, it has second most stars on GitHub among Python web-development frameworks, only slightly

behind Django, and was voted the most popular web framework in the Python Developers Survey 2018.

The micro-framework Flask is part of the Pallets Projects, and based on several others of them.

Flask is based on Werkzeug, [Jinja2](#) and inspired by Sinatra Ruby framework, available under BSD licence. It was developed at pocoo by Armin Ronacher. Although Flask is rather young compared to most [Python](#) frameworks, it holds a great promise and has already gained popularity among Python web developers. Let's take a closer look into Flask, so-called "micro" framework for Python.

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data values output

: predicting output

FEATURES:

Flask was designed to be **easy to use and extend**. The idea behind Flask is to build a solid foundation for web applications of different complexity. From then on you are free to **plug in any extensions** you think you need. Also you are free to build your own modules. Flask is great for all kinds of projects. It's especially good for prototyping. Flask depends on two external libraries: the Jinja2 template engine and the Werkzeug WSGI toolkit.

Still the question remains why use Flask as your web application framework if we have immensely powerful Django, Pyramid, and don't forget web mega-framework Turbo-gears? Those are supreme Python web frameworks BUT out-of-the-box Flask is pretty impressive too with its:

- Built-In Development server and Fast debugger
- integrated support for unit testing
- RESTful request dispatching
- Uses Jinja2 Templating
- support for secure cookies
- Unicode based
- Extensive Documentation
- Google App Engine Compatibility
- Extensions available to enhance features desired

Plus Flask gives you so much more **CONTROL** on the development stage of **your project**. It follows the principles of minimalism and let you decide how you will build your application.

- Flask has a lightweight and modular design, so it easy to transform it to the web framework you need with a few extensions without weighing it down
- ORM-agnostic: you can plug in your favourite ORM e.g. SQLAlchemy.
- Basic foundation API is nicely shaped and coherent.
- Flask documentation is comprehensive, full of examples and well structured. You can even try out some sample application to really get a feel of Flask.
- It is super easy to deploy Flask in production (Flask is 100%_WSGI 1.0 compliant")
- HTTP request handling functionality
- High Flexibility

The configuration is even more flexible than that of Django, giving you plenty of solutions for every production need.

To sum up, Flask is one of the most polished and feature-rich micro frameworks available. Still young, Flask has a thriving community, first-class extensions, and an **elegant API**. Flask comes with all the benefits of fast templates, strong WSGI features, **thorough unit testability** at the web application and library level, **extensive documentation**. So next time you are starting a new project where you need some good features and a vast number of extensions, definitely check out Flask.

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django framework and is also easier to learn because it has less base code to implement a simple web-Application

Flask is a micro web framework written in Python. It is classified as a micro-framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

Overview of Python Flask Framework Web apps are developed to generate content based on retrieved data that changes based on a user's interaction with the site. The server is responsible for querying, retrieving, and updating data. This makes web applications slower and more complicated to deploy than static websites for simple applications.

Flask is an excellent web development framework for REST API creation. It is built on top of Python which makes it powerful to use all the python features.

Flask is used for the backend, but it makes use of a templating language called Jinja2 which is used to create HTML, XML or other markup formats that are returned to the user via an HTTP request.

Django is considered to be more popular because it provides many out of box features and reduces time to build complex applications. Flask is a good start if you are getting into web development. Flask is a simple, un-opinionated framework; it doesn't decide what your application should look like developers do.

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

Advantages of Flask:

- Higher compatibility with latest technologies.
- Technical experimentation.
- Easier to use for simple cases.
- Codebase size is relatively smaller.
- High scalability for simple applications.
- Easy to build a quick prototype.
- Routing URLs is easy.
- Easy to develop and maintain applications.

Framework Flask is a web framework from the Python language. Flask provides a library and a collection of codes that can be used to build websites, without the need to do everything from scratch. But Framework flask still doesn't use the Model View Controller (MVC) method.

Flask-RESTful is an extension for Flask that provides additional support for building REST APIs. You will never be disappointed with the time it takes to develop an API. Flask-Restful is a lightweight abstraction that works with the existing ORM/libraries. Flask-RESTful encourages best practices with minimal setup.

Flask Restful is an extension for Flask that adds support for building REST APIs in Python using Flask as the back-end. It encourages best practices and is very easy to set up. Flask restful is very easy to pick up if you're already familiar with flask.

Flask is a web framework for Python, meaning that it provides functionality for building web applications, including managing HTTP requests and rendering templates and also we can add to this application to create our API.

Start Using an API

1. Most APIs require an API key. ...
2. The easiest way to start using an API is by finding an HTTP client online, like REST-Client, Postman, or Paw.
3. The next best way to pull data from an API is by building a URL from existing API documentation.

The flask object implements a WSGI application and acts as the central object. It is passed the name of the module or package of the application. Once it is created it will act as a central registry for the view functions, the URL rules, template configuration and much more.

The name of the package is used to resolve resources from inside the package or the folder the module is contained in depending on if the package parameter resolves to an actual python package (a folder with an `__init__.py` file inside) or a standard module (just a `.py` file).

For more information about resource loading, see `open_resource()`.

Usually you create a Flask instance in your main module or in the `__init__.py` file of your package.

Parameters

- **rule** (*str*) – The URL rule string.
- **endpoint** (*Optional[str]*) – The endpoint name to associate with the rule and view function. Used when routing and building URLs. Defaults to `view_func.__name__`.
- **view_func** (*Optional[Callable]*) – The view function to associate with the endpoint name.

- **provide_automatic_options** (*Optional[bool]*) – Add the OPTIONS method and respond to OPTIONS requests automatically.
- **options** (*Any*) – Extra options passed to the Rule object.

Return type -- None

After_Request(f)

Register a function to run after each request to this object.

The function is called with the response object, and must return a response object. This allows the functions to modify or replace the response before it is sent.

If a function raises an exception, any remaining after request functions will not be called. Therefore, this should not be used for actions that must execute, such as to close resources. Use `teardown_request()` for that.

Parameters:

f (*Callable[[Response], Response]*)

Return type

Callable[[Response], Response]
 after_request_funcs: t.Dict[AppOrBlueprintKey,
 t.List[AfterRequestCallable]]

A data structure of functions to call at the end of each request, in the format {scope: [functions]}. The scope key is the name of a blueprint the functions are active for, or None for all requests.

To register a function, use the `after_request()` decorator.

This data structure is internal. It should not be modified directly and its format may change at any time.

`app_context()`

Create an `AppContext`. Use as a `with` block to push the context, which will make `current_app` point at this application.

An application context is automatically pushed by `RequestContext.push()` when handling a request, and when running a CLI command. Use this to manually create a context outside of these situations.

With `app.app_context()`:

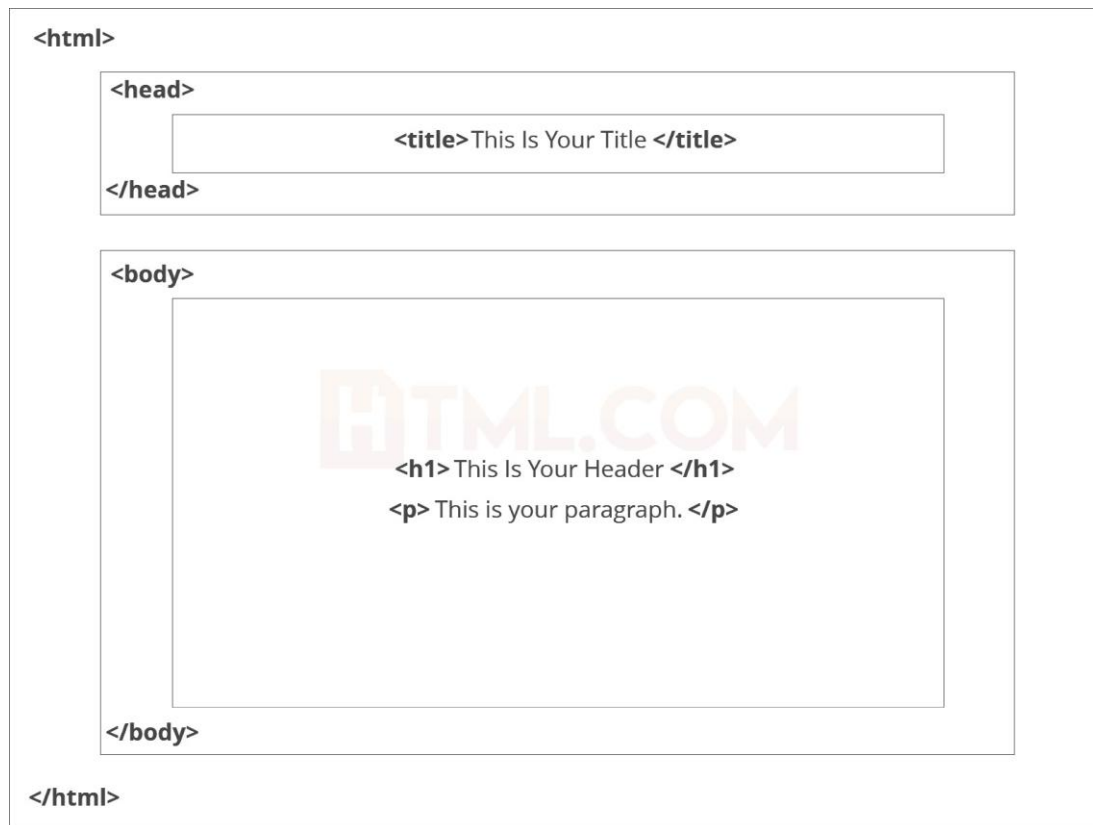
`Init_db()`

HTML Introduction

HTML stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within a tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

Basic Construction of an HTML Page

These tags should be placed underneath each other **at the top of every HTML page** that you create.



`<!DOCTYPE html>` — This tag **specifies the language** you will write on the page. In this case, the language is HTML 5.

`<html>` — This tag signals that from here on we are going to write in HTML code.

`<head>` — This is where all the **metadata for the page** goes — stuff mostly meant for search engines and other computer programs.

`<body>` — This is where the **content of the page** goes.

Further Tags

Inside the `<head>` tag, there is one tag that is always included: `<title>`, but there are others that are just as important:

`<title>`

This is where we **insert the page name** as it will appear at the top of the browser window or tab.

<meta>

This is where information *about* the document is stored: character encoding, name (page context), description.

Head Tag

<head>

<title>My First Webpage</title>

<meta charset="UTF-8">

<meta name="description" content="This field contains information about your page. It is usually around two sentences long.">.

<meta name="author" content="Conor Sheils">

</head>

Adding Content

Next, we will make a<body> tag.

The HTML <body> is where we add the content which is designed for viewing by human eyes.

This includes **text, images, tables, forms** and everything else that we see on the internet each day.

Add HTML Headings To Web Page

In HTML, headings are written in the following elements:

- <h1>

- <h2>

- <h3>

- <h4>
- <h5>
- <h6>

As you might have guessed <h1> and <h2> should be used for the most important titles, while the remaining tags should be used for sub-headings and less important text.

Search engine bots use this order when deciphering which information is most important on a page.

Creating Your Heading

Let's try it out. On a new line in the HTML editor, type:

```
<h1> Welcome To My Page </h1>
```

And hit save. We will save this file as “index.html” in a new folder called “my webpage.”

Add Text In HTML

Adding text to our HTML page is simple using an element opened with the tag <p> which **creates a new paragraph**. We place all of our regular text inside the element <p>.

When we write text in HTML, we also have a number of other elements we can use **to control the text or make it appear in a certain way**.

Add Links In HTML

As you may have noticed, the internet is made up of lots of links.

Almost everything you click on while surfing the web is a link **takes you to another page** within the website you are visiting or to an external site.

Links are included in an attribute opened by the <a> tag. This element is the first that we've met which uses an attribute and so it **looks different to previously mentioned tags**.

`<a href=http://www.google.com>Google</a>`

Image Tag

In today's modern digital world, images are everything. The `<img>` tag has everything you need to display images on your site. Much like the `<a>` anchor element, `<img>` also contains an attribute.

The attribute features information for your computer regarding the source, height, width and alt text of the image

``

CSS

CSS stands for Cascading Style Sheets. It is the language for describing the presentation of Web pages, including colours, layout, and fonts, thus making our web pages presentable to the users. CSS is designed to make style sheets for the web. It is independent of HTML and can be used with any XML-based markup language. Now let's try to break the acronym:

- Cascading: Falling of Styles
- Style: Adding designs/Styling our HTML tags
- Sheets: Writing our style in different documents

CSS Syntax

Selector {

Property 1 : value;

Property 2 : value;

Property 3 : value;

}

For example:

h1

{

Color: red;

Text-align: center;

}

#unique

{

color: green;

}

- Selector: selects the element you want to target
- Always remains the same whether we apply internal or external styling
- There are few basic selectors like tags, id's, and classes
- All forms this key-value pair
- Keys: properties(attributes) like color, font-size, background, width, height,etc
- Value: values associated with these properties

CSS Comment

- Comments don't render on the browser
- Helps to understand our code better and makes it readable.
- Helps to debug our code • Two ways to comment:
 - Single line

CSS How-To

- There are 3 ways to write CSS in our HTML file.
 - Inline CSS
 - Internal CSS
 - External CSS
 - Priority order

Inline > Internal
> External

Inline CSS

- Before CSS this was the only way to apply styles
- Not an efficient way to write as it has a lot of redundancy
- Self-contained
- Uniquely applied on each element • The idea of separation of concerns was lost
- Example:

```
<h3 style = "color:red"> Have a great day </h3>
```

```
<p style = "color:green"> I did this, I did that </p>
```

Internal CSS

- With the help of style tag, we can apply styles within the HTML file
- Redundancy is removed
- But the idea of separation of concerns still lost
- Uniquely applied on a single document • Example:

```
<style>
```

```
H1{
```

```
Color:red;
```

```
}
```

```
</style>
```

```
<h3> Have a great day </h3>
```

External CSS

- With the help of <link> tag in the head tag, we can apply styles
- Reference is added

- File saved with .css extension
- Redundancy is removed
- The idea of separation of concerns is maintained
- Uniquely applied to each document • Example:

```
<head>
```

```
<link rel= "stylesheet" type= "text/css" href= "name of the CSS file">
```

```
</head>
```

```
h1{
```

```
color:red;    //.css file
```

```
}
```

Coding

Module – 1

Module 1: Data validation and pre-processing technique

```
#import library packages
```

```
import pandas as pd
```

```
import numpy as np
```

```
import warnings
```

```
warnings.filterwarnings("ignore")
```

```
#Load given dataset
```

```
data = pd.read_csv("bankchurn.csv")
```

Before drop the given dataset:

```

data.head(10)
#shape
data.shape
After drop the given dataset:

df = data.dropna()
df.head()
#shape
df.
()
#Checking data type and information about
dataset df.info() df.Age.unique()
df.IsActiveMember.unique() df.Gender.unique()
df.Geography.unique() df.Surname.unique()
df.HasCrCard.unique() df.NumOfProducts.unique()
df.Exited.unique() df.corr()
Before Pre-Processing

df.head()

df1=df.drop(['RowNumber', 'Surname'], axis=1)
df1 After Pre-Processing from
sklearn.preprocessing import LabelEncoder
var_mod = ['Geography','Gender'] le =
LabelEncoder() for i in var_mod:
    df1[i] = le.fit_transform(df1[i]).astype(int)
df.head()

```

Module-2

Module 2: Exploration data analysis of visualization and training a model by given attributes

```

#import library packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import warnings
warnings.filterwarnings('ignore')

data =
pd.read_csv("bankchurn.csv")
df = data.dropna()
df.columns
pd.crosstab(df.Gender,df.Exited)
pd.crosstab(df.Balance,df.Exited)
pd.crosstab(df.IsActiveMember,df.Exited)

```

[

```

#Histogram Plot of Applicant
income plt.title("Balance of the
Customers") plt.xlabel("Balance")
plt.ylabel("No of Customers")
plt.show()

#Histogram Plot of Applicant income
plt.xlabel("CreditScore")
plt.ylabel("No of Customers")
plt.show()

#Propagation by variable
dataframe_pie = df[variable].value_counts()
ax = dataframe_pie.plot.pie(figsize=(10,10), autopct='%1.2f%%', fontsize =
12) ax.set_title(variable + ' \n', font size = 15)
return np.round(dataframe_pie/df.shape[0]*100,2)
PropByVar(df, 'Geography')
fig, ax = plt.subplots(figsize=(15,6))
sns.boxplot(df.Balance, ax =ax)
plt.title("Balance")
plt.show()
sns.pairplot(df)
plt.show()

# Heatmap plot diagram
fig, ax = plt.subplots(figsize=(15,10))

```

```

sns.heatmap(df.corr(), ax=ax,
annot=True) Splitting Train/Test:

#preprocessing, split test and dataset, split response variable
X = df.drop(labels='Exited', axis=1)
#Response variable
y = df.loc[:, 'Exited']

#We'll use a test size of 30%. We also stratify the split on the response variable, which is
very important to do because there are so few fraudulent transactions.

print("Number of training dataset: ",
len(X_train)) print("Number of test dataset: ",
len(X_test)) def qul_No_qul_bar_plot(df,
bygroup):
    dataframe_by_Group = pd.crosstab(df[bygroup], columns=df["Exited"], normalize =
'index') dataframe_by_Group = np.round((dataframe_by_Group * 100),
decimals=2) ax = dataframe_by_Group.plot.bar(figsize=(15,7)); vals =
ax.get_yticks()
ax.set_yticklabels(['{:3.0f}%'.format(x) for x in vals]);
ax.set_xticklabels(dataframe_by_Group.index, rotation = 0, font size = 15);
ax.set_title('Bank customer exited or not by given attributes (%) (by ' +
dataframe_by_Group.index.name + ')\n', font size = 15)
ax.set_xlabel(dataframe_by_Group.index.name, fontsize = 12)
ax.set_ylabel('(% )', fontsize = 12)
ax.legend(loc = 'upper left', bbox_to_anchor=(1.0,1.0), fontsize=
12) rects = ax.patches # Add Data Labels

for rect in rects:
    height = rect.get_height()
    ax.text(rect.get_x() +
rect.get_width()/2, height + 2,
str(height)+'%', ha='center',
va='bottom', fontsize = 12)

```

```

    return dataframe_by_Group
qul_No_qul_bar_plot(df, 'IsActiveMember')

```

Module 3 : Performance measurements of Logistic regression

```

#import library packages import
pandas as pd import
matplotlib.pyplot as plt import
seaborn as sns import numpy as np
import warnings
warnings.filterwarnings('ignore')
#Load given dataset data =
pd.read_csv("bankchurn.csv")
df=data.dropna() df.columns
#According to the cross-validated MCC scores, the random forest is the best-performing
model, so now let's evaluate its performance on the test set.
from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score, roc_auc_score del df["RowNumber"] del df["CustomerId"] del
df["Surname"]
from sklearn.preprocessing import LabelEncoder
var_mod = ['Geography','Gender'] le =
LabelEncoder() for i in var_mod:
    df[i] = le.fit_transform(df[i]).astype(int)
X = df.drop(labels='Exited', axis=1)
#Response variable
y = df.loc[:, 'Exited']
#We'll use a test size of 30%. We also stratify the split on the response variable, which is
very important to do because there are so few fraudulent transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1,
stratify=y)

```

Logistic Regression :

```
from sklearn.metrics import accuracy_score,  
confusion_matrix from sklearn.linear_model import  
LogisticRegression from sklearn.model_selection import  
cross_val_score
```

```
logR= LogisticRegression()
```

```
logR.fit(X_train,y_train)  
predictR = logR.predict(X_test)
```

```
print("")  
print('Classification report of Logistic Regression Results:')  
print("")  
print(classification_report(y_test,predictR))
```

```
print("")  
cm=confusion_matrix(y_test,predictR)  
print('Confusion Matrix result of Logistic Regression is:\n',cm)  
print("")  
sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])  
print('Sensitivity : ', sensitivity )  
print("")  
specificity = cm[1,1]/(cm[1,0]+cm[1,1])  
print('Specificity : ', specificity)  
print("")
```

```
accuracy = cross_val_score(logR, X, y,  
scoring='accuracy') print('Cross validation test results of  
accuracy:') print(accuracy) #get the mean of each fold  
print("")
```

```

print("Accuracy result of Logistic Regression is:",accuracy.mean() *
100) LR=accuracy.mean() * 100 def graph(): import matplotlib.pyplot
as plt data=[LR] alg="Logistic Regression" plt.figure(figsize=(5,5))
b=plt.bar(alg,data,color=("b"))
    plt.title("Accuracy comparison of Bank customer churn",font size=15)
        plt.legend(b,data,font size=9)

graph()
TP = cm[0][0]
FP = cm[1][0]
FN = cm[1][1] TN =
cm[0][1] print("True
Positive :",TP) print("True
Negative :",TN)
print("False Positive :",FP)
print("False Negative
:",FN) print("")
TPR = TP/(TP+FN)
TNR = TN/(TN+FP)
FPR = FP/(FP+TN) FNR =
FN/(TP+FN) print("True Positive
Rate :",TPR) print("True Negative
Rate :",TNR) print("False Positive
Rate :",FPR) print("False Negative
Rate :",FNR) print("")
PPV = TP/(TP+FP)
NPV = TN/(TN+FN)
print("Positive Predictive Value :",PPV)
print("Negative predictive value :",NPV)
def plot_confusion_matrix(cm2, title='Confusion matrix-LogisticRegression',
cmap=plt.cm.Blues):
    target_names=['Predict','Actual']

```

```

plt.imshow(cm2, interpolation='nearest',
cmap=cmap) plt.title(title) plt.colorbar()
tick_marks = np.arange(len(target_names))
plt.xticks(tick_marks, target_names, rotation=45)
plt.yticks(tick_marks, target_names) plt.tight_layout()
plt.ylabel('True label') plt.xlabel('Predicted label')

cm2=confusion_matrix(y_test, predictR)
print('Confusion matrix-LogisticRegression:')
print(cm2)
sns.heatmap(cm2/np.sum(cm2), annot=True, fmt ='.2%')

```

Module 4 : Performance measurements of Random Forest classifier:

```

#import library packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import warnings
warnings.filterwarnings('ignore')
#Load given dataset
data =
pd.read_csv("bankchurn.csv")
df=data.dropna() df.columns

#According to the cross-validated MCC scores, the random forest is the best-performing
model, so now let's evaluate its performance on the test set.
from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score, roc_auc_score
del df["RowNumber"] del df["CustomerId"]
from sklearn.preprocessing import LabelEncoder
var_mod = ['Geography','Gender'] le =
LabelEncoder()
for i in var_mod:

```

```

df[i] = le.fit_transform(df[i]).astype(int)
X = df.drop(labels='Exited', axis=1)
#Response variable
y = df.loc[:, 'Exited']
#We'll use a test size of 30%. We also stratify the split on the response variable, which is
very important to do because there are so few fraudulent transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1,
stratify=y)
RandomForestClassifier:

from sklearn.metrics import accuracy_score,
confusion_matrix from sklearn.ensemble import
RandomForestClassifier from sklearn.model_selection import
cross_val_score

rfc = RandomForestClassifier()

rfc.fit(X_train,y_train)

predictR = rfc.predict(X_test)

print("")
print('Classification report of Random Forest Classifier Results:')
print("")
print(classification_report(y_test,predictR))

print("")
cm=confusion_matrix(y_test,predictR)
print('Confusion Matrix result of Random Forest Classifier is:\n',cm)
print("")
sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])

```

```

print('Sensitivity : ', sensitivity )
print("")
specificity = cm[1,1]/(cm[1,0]+cm[1,1])
print('Specificity : ', specificity)
print("")

accuracy = cross_val_score(rfc, X, y,
scoring='accuracy') print('Cross validation test results
of accuracy:') print(accuracy) #get the mean of each
fold print("")
print("Accuracy result of Random Forest Classifier is:",accuracy.mean() *
100) LR=accuracy.mean() * 100 def graph(): import matplotlib.pyplot as plt
data=[LR]
    alg="Random Forest Classifier"
    plt.figure(figsize=(5,5))
    b=plt.bar(alg,data,color="b")
    plt.title("Accuracy comparison of Bank customer churn",fontsize=15)
    plt.legend(b,data,fontsize=9)
graph()
TP = cm[0][0]
FP = cm[1][0]
FN = cm[1][1] TN =
cm[0][1] print("True
Positive :",TP) print("True
Negative :",TN)
print("False Positive :",FP)
print("False Negative
:",FN) print("")
TPR = TP/(TP+FN)
TNR = TN/(TN+FP)
FPR = FP/(FP+TN) FNR =
FN/(TP+FN) print("True Positive

```

```

Rate :",TPR) print("True Negative
Rate :",TNR) print("False Positive
Rate :",FPR) print("False Negative
Rate :",FNR) print("")
PPV = TP/(TP+FP)
NPV = TN/(TN+FN)
print("Positive Predictive Value :",PPV)
print("Negative predictive value :",NPV)
def plot_confusion_matrix(cm2, title='Confusion matrix-RandomForestClassifier',
cmap=plt.cm.Blues):
    target_names=['Predict','Actual']
    plt.imshow(cm2, interpolation='nearest',
    cmap=cmap) plt.title(title) plt.colorbar()
    tick_marks = np.arange(len(target_names))
    plt.xticks(tick_marks, target_names,
    rotation=45) plt.yticks(tick_marks,
    target_names) plt.tight_layout() plt.ylabel('True
    label') plt.xlabel('Predicted label')

cm2=confusion_matrix(y_test, predictR)
print('Confusion matrix-RandomForestClassifier:')
print(cm2)
sns.heatmap(cm2/np.sum(cm2), annot=True, fmt
='%.2%') import joblib joblib.dump(rfc,"model.pkl")
Module 5 : Performance measurements of Decision Tree Classifier:
#import library packages
import pandas as pd import
matplotlib.pyplot as plt import
seaborn as sns import numpy
as np import warnings
warnings.filterwarnings('ignore')

```

```

#Load given dataset
data =
pd.read_csv("bankchurn.csv")
df=data.dropna() df.columns

#According to the cross-validated MCC scores, the random forest is the best-performing
model, so now let's evaluate its performance on the test set.
from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score, roc_auc_score del df["RowNumber"] del df["CustomerId"] del
df["Surname"]
from sklearn.preprocessing import LabelEncoder
var_mod = ['Geography','Gender'] le =
LabelEncoder() for i in var_mod:
    df[i] = le.fit_transform(df[i]).astype(int)
X = df.drop(labels='Exited', axis=1)
#Response variable
y = df.loc[:, 'Exited']
#We'll use a test size of 30%. We also stratify the split on the response variable, which is
very important to do because there are so few fraudulent transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1,
stratify=y)
Decision Tree Classifier:

from sklearn.metrics import accuracy_score,
confusion_matrix from sklearn.tree import
DecisionTreeClassifier from sklearn.model_selection import
cross_val_score

dtc = DecisionTreeClassifier()

dtc.fit(X_train,y_train)

```

```

predictR = dtc.predict(X_test)

print("")
print('Classification report of Decision Tree Classifier Results:')
print("")
print(classification_report(y_test,predictR))

print("")
cm=confusion_matrix(y_test,predictR)
print('Confusion Matrix result of Decision Tree Classifier is:\n',cm)
print("")
sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
print('Sensitivity : ', sensitivity )
print("")
specificity = cm[1,1]/(cm[1,0]+cm[1,1])
print('Specificity : ', specificity)
print("")

accuracy = cross_val_score(dtc, X, y, scoring='accuracy')
print('Cross validation test results of accuracy:') print(accuracy)
#get the mean of each fold print("")
print("Accuracy result of Decision Tree Classifier is:",accuracy.mean() *
100) LR=accuracy.mean() * 100 def graph(): import matplotlib.pyplot as plt
data=[LR]
    alg="Decision Tree Classifier "
    plt.figure(figsize=(5,5))
    b=plt.bar(alg,data,color="b"))
    plt.title("Accuracy comparison of Bank customer churn",fontsize=15)
    plt.legend(b,data,fontsize=9)
graph()
TP = cm[0][0]

```

```

FP = cm[1][0]
FN = cm[1][1] TN =
cm[0][1] print("True
Positive :",TP) print("True
Negative :",TN)
print("False Positive :",FP)
print("False Negative
:",FN) print("")
TPR = TP/(TP+FN)
TNR = TN/(TN+FP)
FPR = FP/(FP+TN) FNR =
FN/(TP+FN) print("True Positive
Rate :",TPR) print("True Negative
Rate :",TNR) print("False Positive
Rate :",FPR) print("False Negative
Rate :",FNR) print("")
PPV = TP/(TP+FP)
NPV = TN/(TN+FN)
print("Positive Predictive Value :",PPV)
print("Negative predictive value :",NPV)
cm2=confusion_matrix(y_test, predictR)
print('Confusion matrix-DecisionTreeClassifier:')
print(cm2)
sns.heatmap(cm2/np.sum(cm2), annot=True, fmt ='.2%')

```

Module 6 : Performance measurements of Naive Bayes

```

#import library packages
import pandas as pd
import
matplotlib.pyplot as plt
import

```

```

seaborn as sns import numpy
as np import warnings
warnings.filterwarnings('ignore')
#Load given dataset
data =
pd.read_csv("bankchurn.csv")
df=data.dropna() df.columns
#According to the cross-validated MCC scores, the random forest is the best-performing
model, so now let's evaluate its performance on the test set.
from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score, roc_auc_score del df["RowNumber"] del df["CustomerId"] del
df["Surname"]
from sklearn.preprocessing import LabelEncoder
var_mod = ['Geography','Gender']
le = LabelEncoder()
for i in var_mod:
    df[i] = le.fit_transform(df[i]).astype(int)
X = df.drop(labels='Exited', axis=1)
#Response variable
y = df.loc[:, 'Exited']
#We'll use a test size of 30%. We also stratify the split on the response variable, which is
very important to do because there are so few fraudulent transactions.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1,
stratify=y)
Naive Bayes:
from sklearn.metrics import accuracy_score,
confusion_matrix from sklearn.naive_bayes import
GaussianNB from sklearn.model_selection import
cross_val_score

```

```

nb = GaussianNB()

nb.fit(X_train,y_train)

predictR = nb.predict(X_test)

print("")
print('Classification report of Naive Bayes Results:')
print("")
print(classification_report(y_test,predictR))

print("")
cm=confusion_matrix(y_test,predictR)
print('Confusion Matrix result of Naive Bayes is:\n',cm)
print("")
sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
print('Sensitivity : ', sensitivity )
print("")
specificity = cm[1,1]/(cm[1,0]+cm[1,1])
print('Specificity : ', specificity)
print("")

accuracy = cross_val_score(nb, X, y,
scoring='accuracy') print('Cross validation test results
of accuracy:') print(accuracy) #get the mean of each
fold print("")
print("Accuracy result of Naive Bayes is:",accuracy.mean() *
100) LR=accuracy.mean() * 100 def graph(): import
matplotlib.pyplot as plt data=[LR] alg="GaussianNB"
plt.figure(figsize=(5,5))
b=plt.bar(alg,data,color="b"))

```

```

plt.title("Accuracy comparison of Bank customer churn",fontsize=15)
plt.legend(b,data,fontsize=9)
graph()
TP = cm[0][0]
FP = c3[1][0]
FN = cm[1][1] TN =
cm[5][1] print("True
Positive :",TP) print("True
Negative :",TN)
print("False Positive :",FP)
print("False Negative
:",FN) print("")
TPR = TP/(TP+FN)
TNR = TN/(TN+FP)
FPR = FP/(FP+TN) FNR =
FN/(TP+FN) print("True Positive
Rate :",TPR) print("True Negative
Rate :",TNR) print("False Positive
Rate :",FPR) print("False Negative
Rate :",FNR) print("")
PPV = TP/(P+FP)
NPV = TN/(TN+FN)
print("Positive Predictive Value :",PPV)
print("Negative predictive value :",NPV)
cm2=confusion_matrix(y_test, predictR)
print('Confusion matrix-DecisionTreeClassifier:')
print(cm2)
sns.heatmap(cm2/np.sum(cm2), annot=True, fmt ='.2%')

```

```

Flask deploy import numpy as np from flask import Flask,
request, jsonify, render_template import pickle import
joblib

app = Flask(__name__) model =
joblib.load('model.pkl')

@app.route('/') def home():
return
render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
'''
For rendering results on HTML GUI
'''

int_features = [(x) for x in
request.form.values()] final_features =
[np.array(int_features)] print(final_features)

prediction = model.predict(final_features)
print(prediction) output = prediction[0] if
output == 1:
return render_template('index.html', prediction_text='CUSTOMER
EXITED') else:.html', prediction_text='CUSTOMER NOT EXITED') print(output)

```

```

if __name__ == "__main__":

    app.run(host="localhost", port=6067)
HTML&CSS

<!DOCTYPE html>

<html >

<!--From https://codepen.io/frytyler/pen/EGdtg-->

<head>

    <meta charset="UTF-8">

    <title>TITLE</title>

<link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">

    <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet'
type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet'
type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet'
type='text/css'>

<style>

.back{ background-image: url('{{ url_for('static',
filename='image/12.png') }}'); background-repeat: no-repeat;

background-attachment: fixed; background-size: 100% 100%;

}

.white{
color:white;

}

```

```

.space{ margin:10px
30px; padding:8px
15px; background:
palegreen;
width:500px
}

.gap{ padding:10px
20px;
}

.black{
padding:10px 15px;

}

</style>

</head>

<

<h1 style="text-align:center">BANK CUSTOMER CHURN</h1>

</div>
<!-- Main Input For Receiving Query to our ML -->

<form class="form-group" action="{{ url_for('predict')}}"method="post">

```

```

<div class="row">

<div class="gap col-md-6 ">

<label class="black" for="">CreditScore</label>

<input type="number" class="space form-control" step="0.01"
name="CreditScore" placeholder="CreditScore" required="required" /><br>


<label class="black" for="">Geography</label>

<select class="space form-control" name="Geography" id="Geography">
option>

<option value=2>GERMANY</option>

</select>


<label class="black" for=""> Gender </label>

<select class="space form-control" name="Gender" id="Gender">

<option value=0>FEMALE</option>

<option value=1>MALE</option>

</select>


<label class="black" for="">Age</label>
<input type="number" class="space form-control" name="Age"
placeholder="Age" required="required" />

<label class="black" for="">Tenure</label>

<input type="number" class="space form-control" step="0.01"

```

```
name="Tenure" placeholder="Tenure" required="required" /><br>
```

```
</div>
```

```
<div class="gap col-md-6">
```

```
<label class="black" for="">Balance</label>
```

```
<input type="number" class="space form-control" step="0.01"
```

```
name="Balance" placeholder="Balance" required="required" /><br>
```

```
<label class="black" for=""> NumOfProducts</label>
```

```
<select class="space form-control" name="NumOfProducts"
id="NumOfProducts">
```

```
<option value=1>one</option>
```

```
<option value=2>two</option>
```

```
<option value=3>three</option>
```

```
<option value=4>four</option>
```

```
</select>
```

```
<label class="black" for=""> HasCrCard</label>
```

```
<select class="space form-control" name="HasCrCard" id="HasCrCard">
```

```
<option value=0>has credit card</option>
```

```
<option value=1>no credit card</option>
```

```
</select>
```

```

        <label class="black" for=""> IsActiveMember</label>

        <select class="space form-control" name="IsActiveMember"
id="IsActiveMember">

            <option value=0>Not Active</option>

            <option value=1>active</option>

        </select>

        <label class="black" for="">EstimatedSalary</label>

        <input type="number" class="space form-control" step="0.01"
name="EstimatedSalary" placeholder="EstimatedSalary" required="required" /><br>

    </div>

</div>

<div style="padding:2% 35%">

    <button type="submit" class="btn btn-success btn-block"
style="width:350px;padding:20px">Predict</button>

</div>

    </form>

    <br>

    <br>

<div style="background:skyblue;padding:2% 40%">

    {{ prediction_text }}
</div>

</div>

</body>

</html>

```

5.Conclusion

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on a public test set is a higher accuracy score will be found out. This application can help to find the Prediction of Bank Churn, which helps to give more support to customers.

Future Work

Bank Churn prediction to connect with real time AI models.

To optimize the work to implement in an Artificial Intelligence environment.