

Title Page

CHURN PREDICTION IN BANKING USING ML WITH ANN

Submitted in partial fulfillment of the requirements for the award
of
Bachelor of Engineering degree in Computer Science and Engineering

by
YARNAGU KAVYARSHITHA (38110664)
VENATI SANDHYA (38110632)



**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



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **YARNAGU KAVYARSHITHA(38110664), VENATI SANDHYA (38110632)** who carried out the project entitled “ **CHURN PREDICTION IN BANKING USING ML WITH ANN** ” under my supervision from **DECEMBER 2021** to **MAY 2022**.

Internal Guide

Dr. Malini Deepika, MTech, Ph.D...,

Head of the Department

Submitted for Viva voce Examination held on_____

DECLARATION

I, **YARNAGU KAVYARSHITHA & VENATI SANDHYA** hereby declare that the project report entitled **CHURN PREDICTION IN BANKING USING ML WITH ANN** done by me under the guidance of **Dr. Malini Deepika, MTech, Ph.D...**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE: 09-03-2022

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

**Y. KAVYARSHITHA
V.SANDHYA**

ACKNOWLEDGEMENT

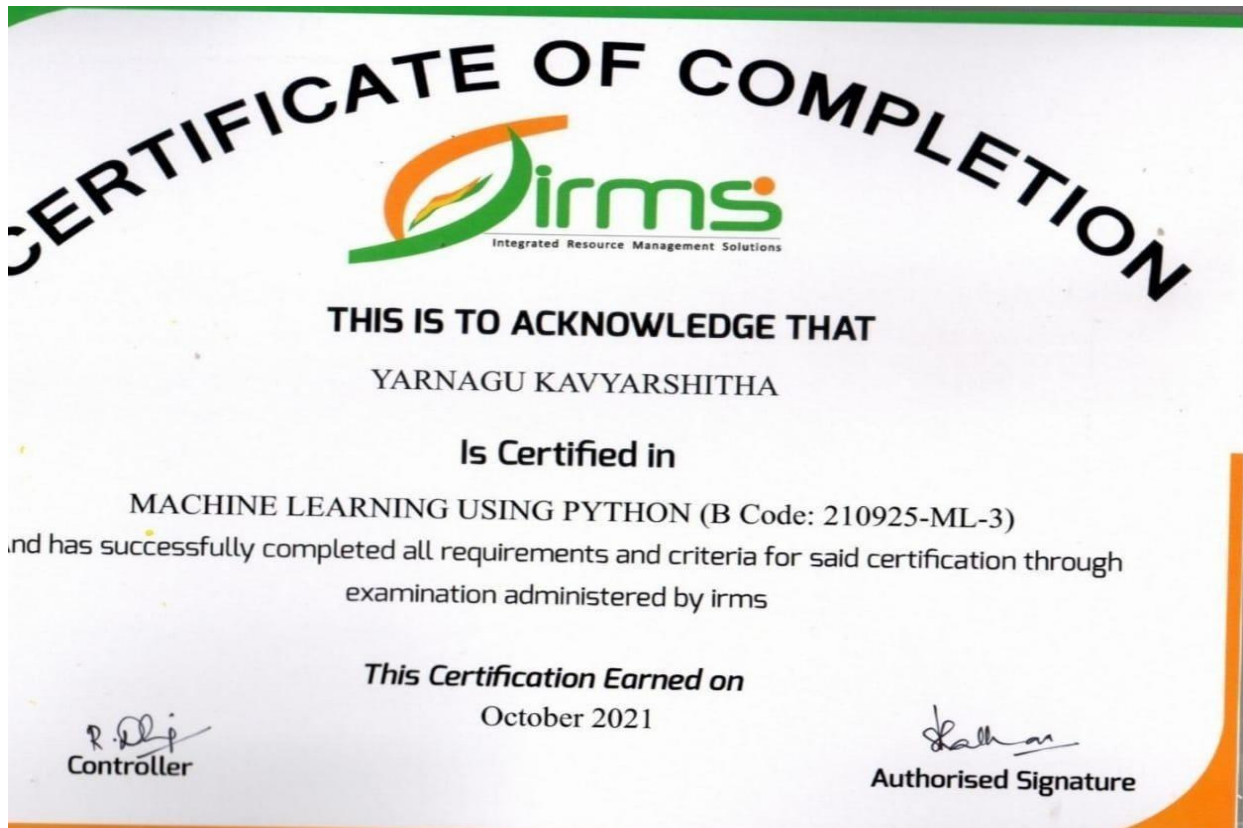
I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D.** and **Dr. L. Lakshmanan, 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 deep sense of gratitude to my Project Guide **Dr. Malini Deepika, MTech, Ph.D...**, for his valuable guidance, suggestions and constant encouragement paved 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.

TRAINING CERTIFICATE



ABSTRACT

Now a -days there are a lot of service providers are available in every business. There is no shortage of customers in any options. Mainly, in the banking sector when 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 Random Forest, SVM, KNN, and Decision Tree 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 78.3% accuracy.

The number of service providers are being increasing very rapidly in every business. In 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 CONTENTS

CHAPTER No.	TITLE	PAGE No
	ABSTRACT	i
	LIST OF FIGURES	ii
	LIST OF TABLES	iii
	LIST OF ABBREVIATIONS	iv
1	INTRODUCTION	1
2.	LITERATURE SURVEY	3
3.	METHODOLOGY	7
	3.1 Module Implementation	6
	3.2 Libraries	10
4.	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	12
5	CONCLUSIONS AND FUTURE WORK	18
	REFERENCES	19
	APPENDIX	20
	A. SOURCE CODE	20
	B. SCREENSHOTS	26
	C. PUBLICATION WITH PLAGARISM REPORT	36

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
3.1	System Architecture	8
4.1	Tenure vs No.of customers	14
4.2	No.of products vs No.of customers	14
4.3	Credit Score vs No.of customers	15
4.4	Estimated Salary vs No.of customers	15
4.5	Age vs No.of customers	16
4.6	Balance vs No.of customers	16
4.7	Gender vs No.of customers	17
4.8	Geography vs No.of customers	17

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
4.1	Classification & their accuracies	12

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
SVM	Support Vector Machine
KNN	K-Nearest Neighbor
RF	Random Forest
DT	Decision Tree

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

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.

2. LITERATURE SURVEY

There are a lot of studies done on customer churning and a lot of research has been done on this concept. One of the studies was done by D-R.Liu &Y.Y.sinh ^[1] on “Integrating ahp and data mining for product recommendation based on customer lifetime value ^[1]” in the year 2005. In this study they mainly focused on improving the quality of recommendation to fulfill customer’s needs is important in most competitive environments. The major advantage in this was although various recommender system has been found, few have labeled the lifetime value of a customer to the firm. They developed a product recommendation methodology that combined group decision making and data mining techniques and the major disadvantage was, it is more fruitful for loyal customers but not done for less loyal customers.

In another study, G.Canning Jr^[2] has done a study on “Do a value analysis of your customer base^[2]” in the year 1982. In this study, he mainly focused on the company’s customer base can yield valuable data on customer sales volumes, profits, growth trends, and technical strength. The main advantage in this research was, there he used the CVA process that can establish this value in a way that leads to an effective marketing program and the major disadvantage was marketing source of a company is its customers the experience accumulated through serving them in the past is of substantial value in serving them in future.

In another study which was done by, P.G.Nieto, E.F.Combarro, J.del coz Dia and E.Montanies ^[3] on "An SVM based regression, model to study the air quality of local

scales in Oviedo urban area ^[3]" in 2013. They focuses on a method of daily air pollution modeling by using support vector machine procedure in Oviedo urban areas at a regional scale. In this study, the main advantage is SVR fit captures the prime idea of statistical learning theory to get a good anguring of the dependence among the main pollution in the city of Oviedo and the main disadvantage was, the methodology used in this study can be applied successfully to other cities with similar or different sources of pollutants, but it is always compulsory to take into account the particular of each location.

In another study which was done by, M.K.Kim, M.C.Park, and D.H.Jeong ^[4] on "The results of customer happiness and switching barriers customer loyalty in Korean mobile telecommunication services ^[4]" in 2004. They mainly focused on, the Korean mobile telecommunication services industry is entering a new transition period. The main advantage of this study was the effects on customer trust of client satisfaction and the switching barrier and the structural relationship between these factors in the Korean mobile telecommunication on the service industry. The main disadvantage is, there are other factors influencing customer loyalty, apart from factors suggested in this study, such as the demographic characteristics of customers and their usage pattern of mobile telecommunication services.

In another study, B.He, Y.shi, Q. Wan, and X.zhao ^[5] has done a study on the Prediction of customer attrition of commercial banks based on the SVM model ^[5]" in 2014. Mainly they focused on Chinese commercial banks that are facing triple huge pressure, including financial churn rate, interest rate marketization and internet finance. In this study, the Main advantage was, commercial bank client churn prediction based on the SVM model and uses random sampling method to improve the SVM model, in view of the sample imbalance characteristics of customer data sets. This method can effectively enhance the prediction accuracy of the selected model and the main disadvantage of this study is, Due to the imbalanced characteristics of the actual commercial bank customer churn dataset. SVM model cannot predict the churners effectively and only general rating criteria cannot measure the predictive power of the model.

In another study that was done by A.Bilal Zoric ^[6] in 2016, on "Predicting customer

churn in banking industry using neural networks ^[6]". In this study, he mainly focused

on the usage of the data mining methods, neural networks, in knowledge discovery from databases in the banking industry. The main advantage was the risk of leaving and analyzing whether those customers are worth retaining. Clients who use more bank services are more loyal, so the bank should focus on those clients who use less than three products and offer them products according to their needs and Main disadvantage of this study was the analysis focused on churn prediction based on only one method and a neural network. They could access other important information that could help banks to get a competitive advantage by using other methods such as segmentation, Decision trees, self-organizing maps.

In another study, A.K.Ahmed, A.Jafar, and K.Aljoumaa ^[7] on "Client churn prediction in telecom company using machine learning algorithms in big data platform ^[7]" in 2019. In this study, they mainly focused on customer churn, it is a major problem and one of the most important concerns for large companies. Due to the effect on the revenue of the companies, especially in the telecom companies. These companies are seeking to develop means to predict potential customers to churn. In this study, the main advantage was, working on a large dataset created by changing big underdone data provided by Syriatel Telecom company. Best results were obtained by applying the XGBOOST algorithm and the main disadvantage of this study was, the method was restricted to customer churn in the telecom industry. This method focuses only on the 2 telecom companies in Syria which are Syria Tel and MTN telecom companies.

In another study which was done by Y.Jiang and C.Li ^[8] on "MRMR based feature selection for classification of cotton foreign matter using hyperspectral imaging ^[8]" in 2015. They mainly focused on different cotton and decreases the monetary value of cotton. Hyperspectral imaging foreign matter. The main advantage of this study was, it explored a new 2 stage approach for hyperspectral imaging optimal wavelength selection. A total of 12 wavelengths in the visible/NR range were selected for cotton FM classification and the main Disadvantage was, It is not focused on the implementation of automatic FM image segmentation and classification method using the selected wavelength.

In another study was done by J. Raj and V. Ananthi^[9] on "Recurrent neural networks

and nonlinear prediction in support vector machines ^[9]” in 2019. This research was mainly focused on the Non-Linear regression estimation issues that are solved by the

successful application of a novel neural network technique termed a support vector machine. Evaluation of recurrent neural networks can assist in pattern recognition of several real-time applications and reduce the pattern mismatch. In this study the main advantage was, this model presents the application of 3 dynamic optimization algorithms namely gravitational search algorithm, Artificial Bee colony algorithm, and differential evolution algorithm to create a hybrid prediction model and the Main disadvantage was depend on comparing more optimization algorithms and extracting their best features for developing a furthermore efficient system. Also, the focus is to be done on application-specific algorithm development.

In another study which was done by Rahul J. Jadhav and Usharani T. Pawar ^[10] on "Churn prediction in Telecommunication using data mining technology ^[10] " in 2019. They mainly focused on an attempt to build a decision support system using Data mining technology for churn prediction in Telecommunication companies where faces a considerable loss of revenue, because some of the customers are at risk of leaving a company. The main advantage of this study was the process of designing such a decision support system through data mining techniques is described. The proposed model is capable of predicting customer churn behavior will in advance and the Main disadvantage of this study was the scope of the model can be widened to include insolvency prediction of telecommunication customers.

3. METHODOLOGY

3.1 Module Implementation : Here totally 5 modules are using in this project they are

3.1.1 **Collecting Bank Churning Data:** The bank churn dataset used in this analysis was collected from Kaggle. This dataset includes information of 10000 bank customers. It contains 14 features, they are

3.1.1.1 *CustomerId:* Id of a customer, It is a unique identification code that is usually got from the respective bank.

3.1.1.2 *Surname:* Name of the customer or the customer's first name.

- 3.1.1.3 *CreditScore*: It is the measure of independent capacity to pay back the borrowed amount. It is a numerical representation of their creditability.
- 3.1.1.4 *Geography*: In this dataset, there 3 geographical areas were mentioned France, Spain, and Germany.
- 3.1.1.5 *Gender*: Gender of a customer. In this dataset, it is mentioned as a string (Female and Male).
- 3.1.1.6 *Age*: Age of the customer. The objective of this is int.
- 3.1.1.7 *Tenure*: Length of time that will be taken by the lender to repay their loan along with the interest.
- 3.1.1.8 *Balance*: Amount of money present in the account either it can be a savings account or salary account.
- 3.1.1.9 *Number of Products*: Various services offered by a bank to the customers. The number of services that a customer is using is mentioned in the dataset and the objective of this is int.
- 3.1.1.10 *Has Cr Card*: Does a customer contain the credit card or not? If yes, it is mentioned as 1, and if no it is mentioned as 0.
- 3.1.1.11 *Is Active Member*: Does a customer is active or not? It is decided by their transactions. In this dataset, it is mentioned as 1 if the customer was active. If the customer was not active it is mentioned as 0.
- 3.1.1.12 *Estimated Salary*: Salary of a customer (in rupees).
- 3.1.1.13 *Exited*: Whether the customer was exited or not. It is also mentioned in binary values. If exited it is mentioned as 1 if not it is mentioned a 0.

3.2.1 Data Scrubbing And Feature Selection:

- 3.2.1.1 *Data Scrubbing*: Removing, correcting, or identifying the errors in the dataset. Which may impact the predictive model.
- 3.2.1.2 *Feature Selection*: To reduce the number of input features. Selecting wanted features and removing the

unwanted features.

By applying, Data scrubbing and feature selection will improve the data's consistency and accuracy. The selection phase is very important it will help us to reduce the training and skip the high-dimensional curse.

3.2.1.3 *Transformation:* Data transformation is the data was available may not be in the right format or may need transformations to make it more useful. By removing null values, unwanted duplicates, incorrect indexing, and incompatible formats. In this dataset Credit Score, Age, Tenure, Balance, the Estimated salary was Big value. By using MinMaxScaler these values normalized or transformed into the range [0,1]. Columns like Row number, Surname, Customer Id are unwanted columns.

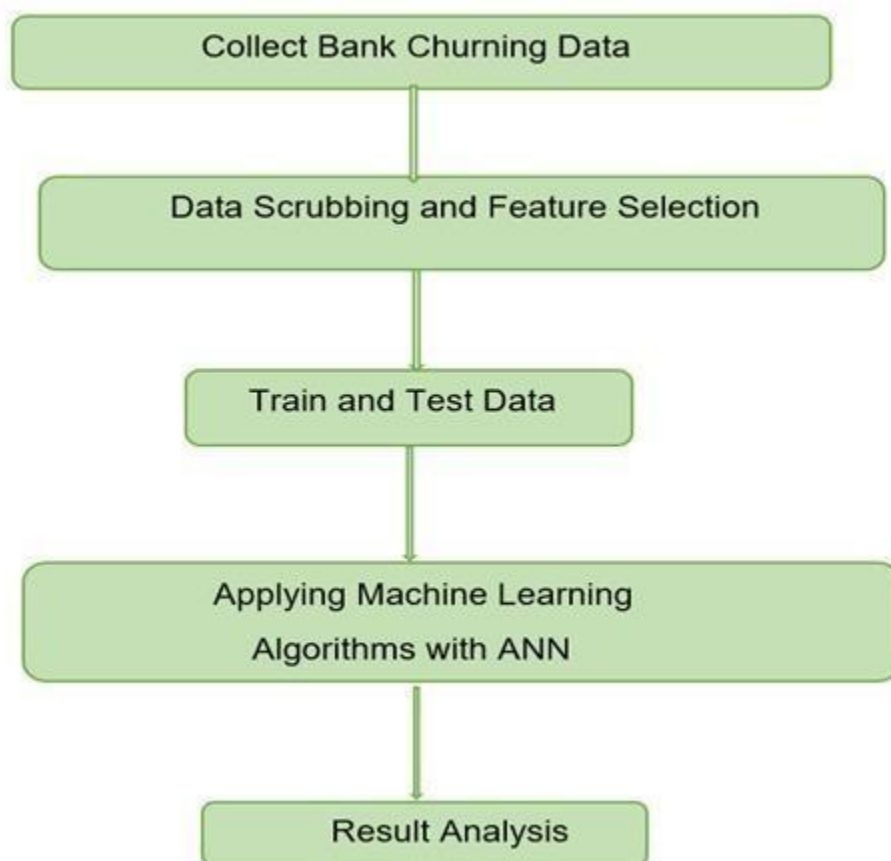


FIG 3.1 System Architecture

3.3.1 Train And Test Data: Training and testing data by applying different Machine learning algorithms like SVM, KNN, Decision Tree, Random Forest, and ANN concepts like Keras and Tensorflow. The dataset is going to be split into two as train and test datasets by using the “train_test_split()” method. It is a quick and simple procedure to perform, the result of which allows us to compare the performance of Machine Learning algorithms for predictive modeling problems. The training dataset was used to fit the machine learning model and a test dataset was used to evaluate and fit the Machine learning model. To control the size of the train and test dataset with parameters train_size and test_size respectively. Either the train_size or test_size should be include in train_test_split() method. Overall the dataset was 1 then the dataset is split into 2 subsets as train and test. Then train_size should be more than the test_size.

3.4.1 Applying ML Algorithms With ANN:

3.4.1.1 *Classifications:* The classifications like SVM, Decision Tree, Random Forest, and KNN were the main algorithms used in this research and also included Keras and TensorFlow.

3.4.1.2 *SVM:* Support vector machine (SVM) is one of the popular supervised learning algorithms. Used for both Regression and Classification problems. It consists of a hyperplane or set of hyperplanes in high or infinite-dimensional space. Support vector classifier SVC() is used to fit the data provide returning the best fit. Hyperplane that divides the data. Support Vector Machine is highly choose by many because it produces high accuracy with less computation. The simple way to implement the SVM algorithm is by applying the scikit learn library and by calling related functions to implement the SVM model. The code reduces too few lines by using the scikit learn library.

3.4.1.3 *Decision Tree:* Decision tree (DT) is one of the supervised learning algorithms used for both regression and classification. Data in the dataset is constant split according to

a certain parameter. This tree can be classified by entities
namely

decision nodes and leaves. The leaves are the outcomes. A simple way to implement the decision tree algorithm is by using SCIKIT learn library and calling related functions to implement the Decision Tree model. Decision Tree classifier is used to fit the data provide returning the best fit.

3.4.1.4 *KNN*: K-Nearest Neighbor is one of the best accurate models because it makes highly accurate models. The best quality of predictions depends on the measure of distance. This algorithm model classifies the points based on `n_neighbors` points that are the most similar to it. It uses the test dataset to make an "Educated guess" on unclassified points that should be classified. The `n_neighbors` are defined as the square root of the number of data's in the dataset. A simple way to implement the KNN algorithm is by using SCIKIT learn library and calling neighbors classifier and also related functions to implement KNN.

3.4.1.5 *Random Forest*: Random Forest is also one of the most popular supervised learning techniques. It is used for both classification and regression. It builds multiple decision trees and merges them to get more accurate. Compare to all algorithms in this model Random Forest gives more accuracy. By using the SCIKIT learn library Random Forest can be implemented easily. By calling Random Forest classifier that will ensemble learning method for regression and calling some related functions to implement Random Forest.

3.4.1.6 *ANN*: Artificial Neural Network uses the processing of brains it is used to model complex patterns and prediction problems. In this model, Tensorflow and Keras are used.

3.4.1.7 *TensorFlow*: It is an Open Source of AI library for various tasks in ML. It provides both high-level and low-level APIs. It is mainly used for prediction, classification, and perception.

high-level APIs. By calling respective functions like loss and optimizer. By using seaborn heatmap to plot rectangular data as a colored matrix to find accuracy by this rectangular data truth vs predictions.

3.2 Libraries :-

3.4.1.8 **eras**: It is a neural network library. It provides only

3.2.1 **Pandas**: It's a Popular python based data analysis toolkit which can be imported using `import pandas as PD`. It represents a diverse range of utilities, ranging from parsing multiple file-formats to converting an entire data table into a NumPy matrix array.

3.2.2 **Numpy**: It is a python library used for working with arrays. It also has functions for working with an array. It also has functions for working in the domain of linear algebra, Fourier transform & matrices.

3.2.3 **Seaborn**: It is a library for making statistical graphics in python. It builds on top of matplotlib & integrates closely with pandas data structures. It helps us to explore & understand our data. Its plotting functions operate on Data frames & arrays containing whole datasets & internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

3.2.4 **TensorFlow**: It is a python library that invokes C++ to construct and execute dataflow graphs. It supports many classification and regression algorithms, and more generally, deep learning and neural networks.

3.2.5 **Matplotlib**: It is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

3.2.6 **Scikit-learn**: It is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction via a consistent interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy, and Matplotlib.

3.2.7 **Keras:** It is a high-level neural networks library that is running on top of TensorFlow, CNTK, and Theano. Using Keras in deep learning allows for easy

and fast prototyping as well as running seamlessly on CPU and GPU. This framework is written in Python code which is easy to debug and allows ease for extensibility.

By using different Machine learning algorithms and ANN, Predicting the customers who are churning, which type of people are churning is the main aim of this research.

4. RESULTS AND DISSCUSSION

When the processing of data is computed, the data has been operational form. The IO features which are after preprocessing are considered for further study. Among these data has been divided into test and train datasets. The training dataset is used 80% out of 100 and the remaining data was the test dataset it is 20% out of 100. After feature selection the result has been recorded there is a minor correction. Occurred before and after the feature selection. The unwanted indexing like Row number, surname, Customer Id was removed before the feature selection. The Main feature selection was done. Indexes like Gender, Geography are done primarily. Fields like Gender, Geography are done as feature selection. Fields like Age, Tenure, Credit Score, Estimated Salary are major feature selections was after primary feature selection.

TABLE 4.1 : CLASSIFICATIONS AND THEIR ACCURACIES

classification	BEFORE			AFTER		
	ACCURACY	PRECISION	RECALL	ACCURACY	PRECISION	RECALL
RANDOM FOREST	86%	0.88	0.96	86.4%	0.87	0.97
ANN	85.3%	0.87	0.95	86%	0.88	0.95
KNN	81.7%	0.83	1.00	81.9%	0.87	0.96
SVM	79.4%	0.79	1.00	84.1%	0.85	0.98
DECISION TREE	78%	0.87	0.86	78.3%	0.88	0.85

This model classifiers like SVM, Random Forest, Decision Tree, KNN, and TensorFlow have recorded the classification report before and after feature selection. But in TensorFlow, it was recorded before applying seaborn and after seaborn.

Random Forest recorded the highest accuracy in before and after feature selection. Before feature selection, the accuracy was 86% but after feature selection, the accuracy was increased by 0.4%. The second highest accuracy was recorded by TensorFlow before it was 85.3% but it was increased by 0.7% and become 86%. The third highest accuracy was recorded by the KNN algorithm before feature selection it was 81.7% after it was increased by 0.2% only. It's not a drastic change. The fourth highest accuracy was recorded by SVM before the feature selection it was 79.4% after the feature selection it was 4.9%. It is a drastic change but least accuracy compared to other classifiers. At last Decision Tree algorithm has given 78% accuracy before but after feature selection, it was increased by 0.3%.

By observing the accuracies that have been recorded by algorithms. Random Forest recorded the highest accuracy after the feature selection 86.4%. ANN recorded the second-highest accuracy, just 0.4% difference between Random Forest and ANN. KNN algorithm recorded the highest accuracy than SVM before the feature selection. But after a drastic change inaccuracy has been recorded it's nearly 5%. So, after feature selection, the changes have been changed SVM occupied the third-highest but KNN replaced SVM. The least accuracy was recorded by the Decision Tree before and after the feature selection.

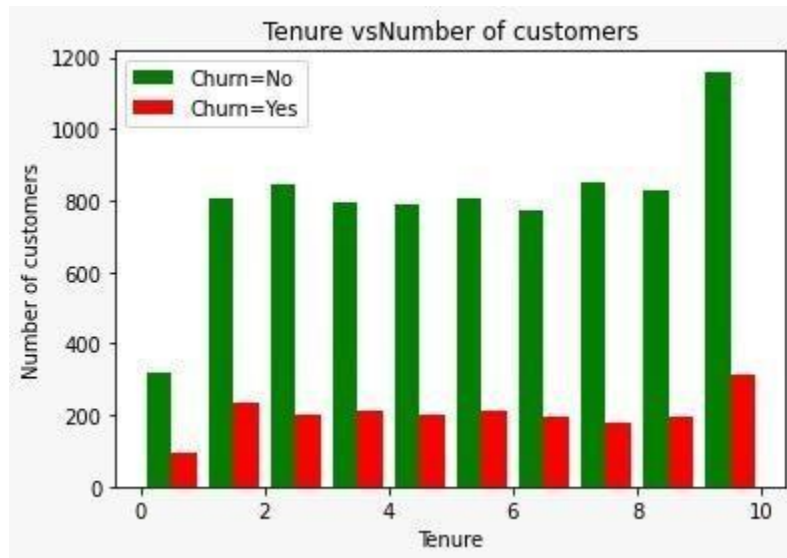


FIG 4.1: TENURE vs NUMBER OF CUSTOMERS

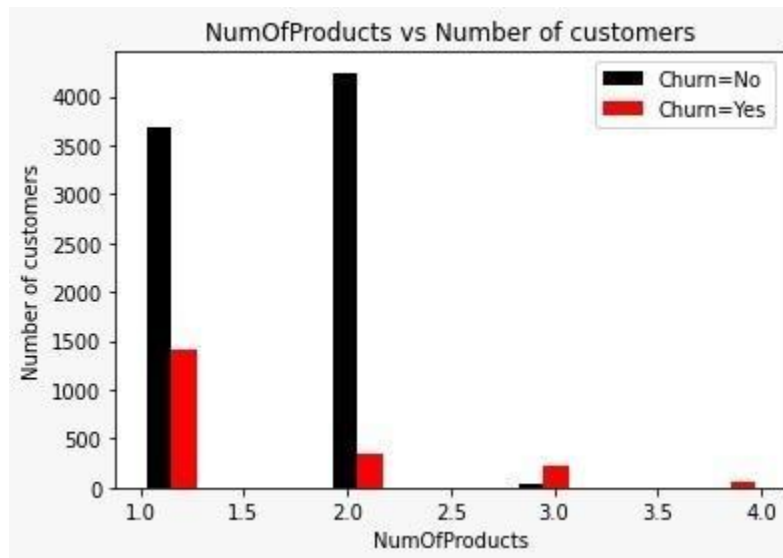


FIG 4.2: NUMBER OF PRODUCTS VS NUMBER OF CUSTOMERS

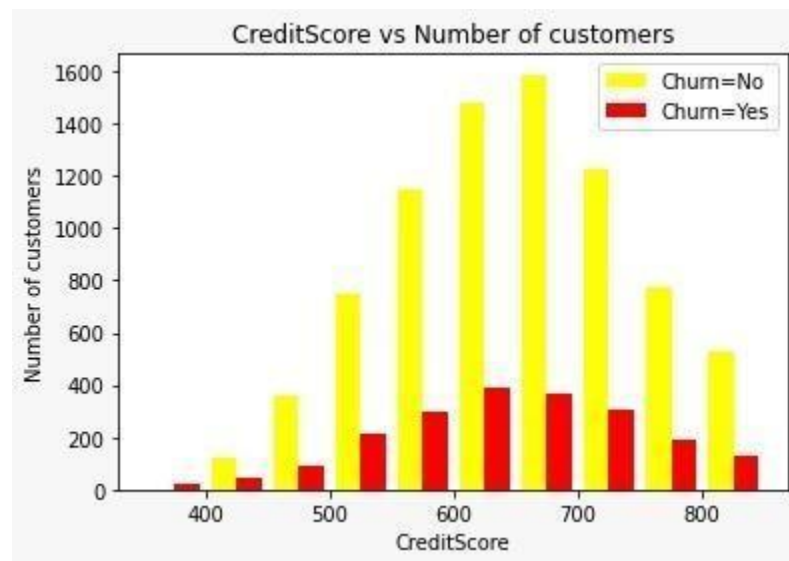


FIG 4.3: CREDIT SCORE VS NUMBER OF CUSTOMERS

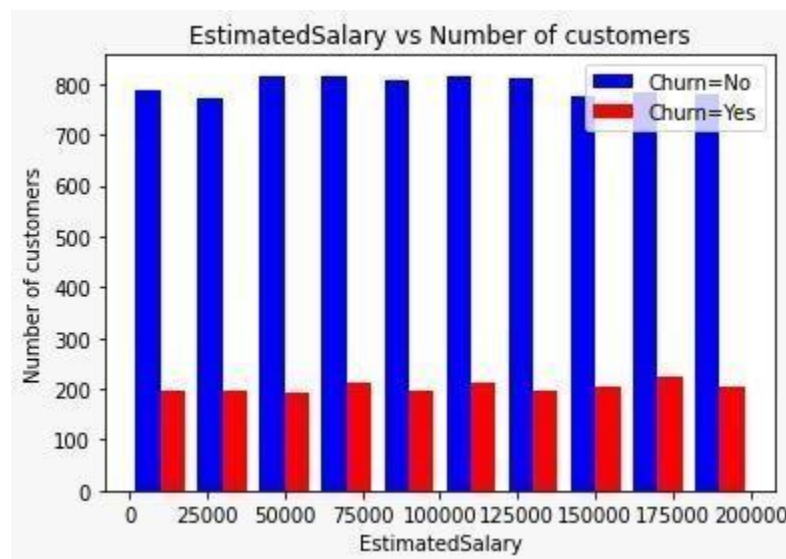


FIG 4.4: ESTIMATED SALARY VS NUMBER OF CUSTOMERS

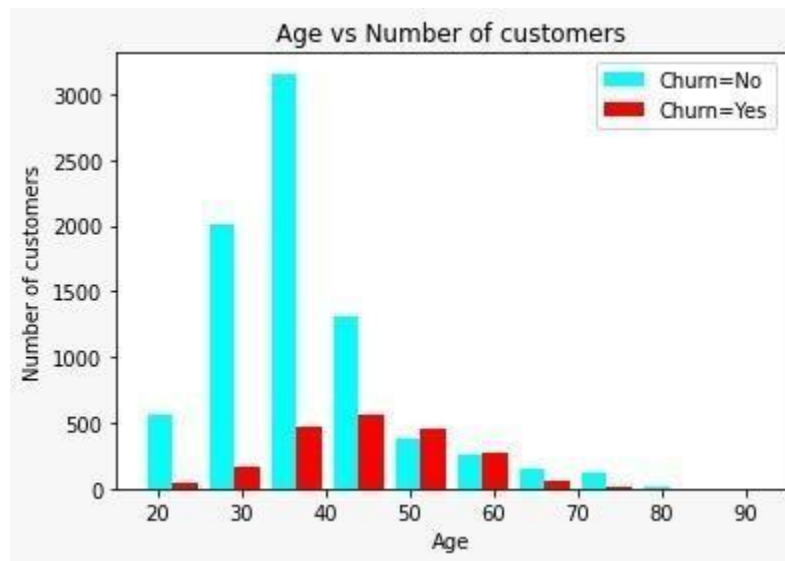


FIG 4.5: AGE VS NUMBER OF CUSTOMERS

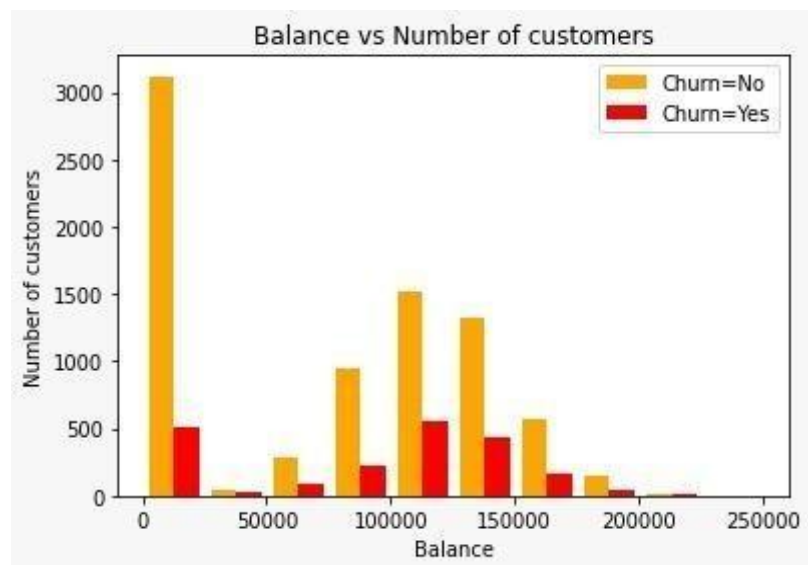


FIG 4.6: BALANCE VS NUMBER OF CUSTOMERS

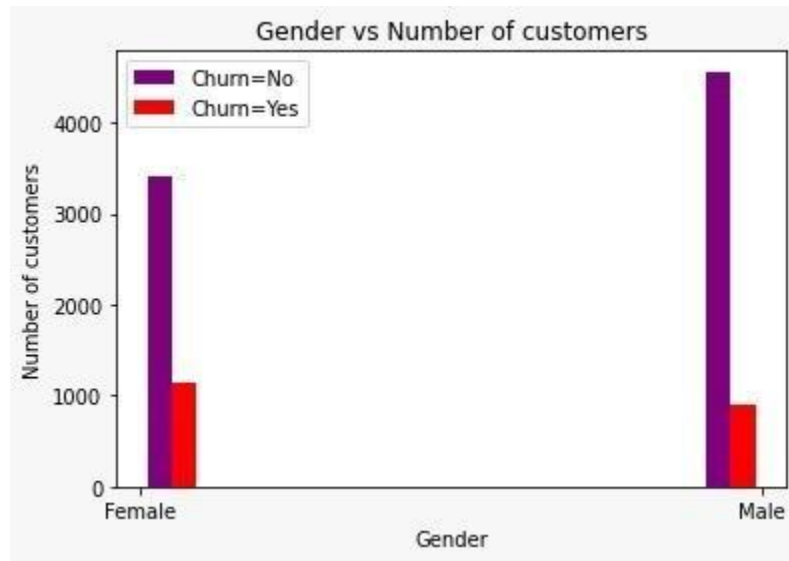


FIG 4.7: GENDER VS NUMBER OF CUSTOMERS

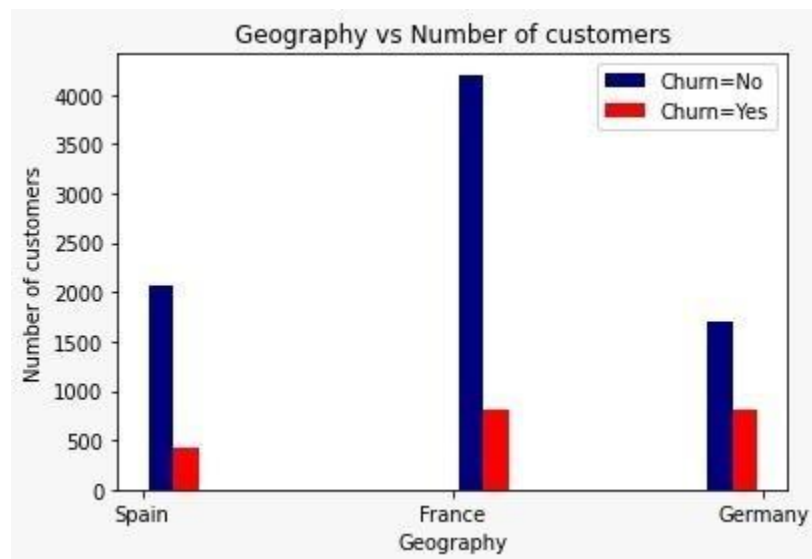


FIG 4.8 : GEOGRAPHY VS NUMBER OF CUSTOMERS

5. CONCLUSION AND FUTURE WORK

From this bank dataset churning has been done for different reasons. By plotting histogram graph between Tenure, Credit Score, Number of Products, Estimated Salary, Age, Balance, Gender, Geography with the number of customers exited and not. By observing the graphs analyzing which type of customers are churning. From the graph between Tenure vs the number of customers who have Tenure as 10 are churning. From the graph between credit score vs the number of customers who have the credit score between 600 to 700 are churning. From the graph between the number of products vs the number of customers observed that the customer who has the number of products as 1 those who are churning.

From the estimated salary vs the number of customers graph the customers who have estimated salaries between 1,75,000 to 2,00,000 are churning. From graph balance vs the number of customers, 1lakhs to 1.5 lakhs had balance customers are churning. From Age, Gender and Geography graphs 40 to 50 aged female and Germany people are churning. This prediction can help bank organizations to take preventions before churning.

Random Forest is the best algorithm than other algorithms for this dataset because of the best accuracy achieved by the Random Forest. The Decision Tree is not well enough for this data to handle. After the Random Forest, ANN is also one of the best algorithms. ANN can also prefer Random Forest. The main aim of this study was to identify the type of people who are mostly churning earlier to prevent the loss and loyalty from customers. Future Work was Predicting the customers churn using a large dataset and working with highly imbalanced data because the real commercial bank dataset was much bigger than this data. One category of people who are churning is the same people who are not churning actually by this dataset.

REFERENCES

- [1]. D.-R. Liu and Y.-Y. Shih, "Integrating PHP and data mining for product recommendation based on customer lifetime value," Information & Management, vol. 42, no. 3, pp. 387–400, 2005.
- [2]. G. Canning Jr, "Do a value analysis of your customer base," Industrial Marketing Management, vol. 11, no. 2, pp. 89–93, 1982.

- [3]. P. G. Nieto, E. F. Combarro, J. del Coz D'íaz, and E. Montañes, "A svm- ´ based regression model to study the air quality at local scale in oviedo urban area (northern spain): A case study," *Applied Mathematics and Computation*, vol. 219, no. 17, pp. 8923–8937, 2013.
- [4]. M.-K. Kim, M.-C. Park, and D.-H. Jeong, "The effects of customer satisfaction and switching barrier on customer loyalty in korean mobile telecommunication services," *Telecommunications policy*, vol. 28, no. 2, pp. 145–159, 2004.
- [5]. B. He, Y. Shi, Q. Wan, and X. Zhao, "Prediction of customer attrition of commercial banks based on svm model," *Procedia Computer Science*, vol. 31, pp. 423–430, 2014.
- [6]. A. Bilal Zoric, "Predicting customer churn in banking industry using ´ neural networks," *Interdisciplinary Description of Complex Systems: INDECS*, vol. 14, no. 2, pp. 116–124, 2016.
- [7]. A. K. Ahmad, A. Jafar, and K. Aljoumaa, "Customer churn prediction in telecom using machine learning in big data platform," *Journal of Big Data*, vol. 6, no. 1, p. 28, 2019.
- [8]. Y. Jiang and C. Li, "mrmmr-based feature selection for classification of cotton foreign matter using hyperspectral imaging," *Computers and Electronics in Agriculture*, vol. 119, pp. 191–200, 2015.
- [9]. J. Raj and V. Ananthi, "Recurrent neural networks and nonlinear prediction in support vector machines," *Journal of Soft Computing Paradigm*, vol. 2019, pp. 33–40, 09 2019.
- [10]. Jadhav, R. J., & Pawar, U. T. (2011). Churn prediction in telecommunication using data mining technology. *International Journal of Advanced Computer Science and Applications*, 2(2).
- [11]. Siu NY, Zhang TJ, Yau CY. The roles of justice and customer satisfaction in customer retention: A lesson from service recovery. *Journal of business ethics*. 2013 Jun 1;114(4):675-86.
- [12]. Hossain MM, Suchy NJ. Influence of customer satisfaction on loyalty: A study on mobile telecommunication industry. *Journal of Social Sciences*. 2013;9(2):73-80.

- [13]. Vafeiadis T, Diamantaras KI, Sarigiannidis G, Chatzisavvas KC. A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*. 2015 Jun 1;55:1-9.
- [14]. Haenlein M. Social interactions in customer churn decisions: The impact of relationship directionality. *International Journal of Research in Marketing*. 2013 Sep 1;30(3):236-48.
- [15]. Rosenberg, L. J., & Czepiel, J. A. (1984). A marketing approach for customer retention. *Journal of consumer marketing*, 1(2), 45-51.
- [16]. Maldonado S, Flores Á, Verbraken T, Baesens B, Weber R. Profit1based feature selection using support vector machines—General framework and an application for customer retention. *Applied Soft Computing*. 2015 Oct 1;35:740- 8.
- [17]. Maga M, Canale P, Bohe A, inventors; Accenture Global Services Ltd, assignee. Churn prediction and management system. United States patent US 8,712,828. 2014 Apr 29.
- [18]. Farquard MA, Ravi V, Raju SB. Churn prediction using comprehensible support vector machine: An analytical CRM application. *Applied Soft Computing*. 2014 Jun 1;19:31-40.
- [19]. Brandusoiu I, Todorean G. Churn prediction in the telecommunications sector using support vector machines. *Margin*. 2013;1:x1.
- [20]. Phadke, C., Uzunalioglu, H., Mendiratta, V. B., Kushnir, D., & Doran, D. (2013). Prediction of subscriber churn using social network analysis. *Bell Labs Technical Journal*, 17(4), 63-76.

APPENDICES

A. SOURCE CODE:

```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
from sklearn.metrics import accuracy_score
from sklearn import metrics
%matplotlib inline

from google.colab import files
uploaded = files.upload()
```

```
df=pd.read_csv("Churn_Modelling.csv")
df.head(5)
```

```
df.isna().sum()
```

```
tenure_churn_no=df[df['Exited']==0]['Tenure']
tenure_churn_yes=df[df['Exited']==1]['Tenure']
plt.xlabel('Tenure')
plt.ylabel('Number of customers')
plt.title('Tenure vs Number of customers')
plt.hist([tenure_churn_no, tenure_churn_yes],
color=['green','red'],label=['Churn=No','Churn=Yes'])
plt.legend()
```

```
CreditScore_churn_no=df[df['Exited']==0]['CreditScore']
CreditScore_churn_yes=df[df['Exited']==1]['CreditScore']
plt.xlabel('CreditScore')
plt.ylabel('Number of customers')
plt.title('CreditScore vs Number of customers')
plt.hist([CreditScore_churn_no, CreditScore_churn_yes],
color=['yellow','red'],label=['Churn=No','Churn=Yes'])
plt.legend()
```

```
NumOfProducts_churn_no=df[df['Exited']==0]['NumOfProducts']
NumOfProducts_churn_yes=df[df['Exited']==1]['NumOfProducts']
plt.xlabel('NumOfProducts')
plt.ylabel('Number of customers')
plt.title('NumOfProducts vs Number of customers')
plt.hist([NumOfProducts_churn_no, NumOfProducts_churn_yes],
color=['black','red'],label=['Churn=No','Churn=Yes'])
plt.legend()
```

```
EstimatedSalary_churn_no=df[df['Exited']==0]['EstimatedSalary']
EstimatedSalary_churn_yes=df[df['Exited']==1]['EstimatedSalary']
plt.xlabel('EstimatedSalary')
plt.ylabel('Number of customers')
plt.title('EstimatedSalary vs Number of customers')
plt.hist([EstimatedSalary_churn_no, EstimatedSalary_churn_yes],
color=['blue','red'],label=['Churn=No','Churn=Yes'])
plt.legend()
```

```
Age_churn_no=df[df['Exited']==0]['Age']
Age_churn_yes=df[df['Exited']==1]['Age']
plt.xlabel('Age')
plt.ylabel('Number of customers')
plt.title('Age vs Number of customers')
plt.hist([Age_churn_no, Age_churn_yes],
```

```
color=['cyan','red'],label=['Churn=No','Churn=Yes'])
plt.legend()
```

```
Balance_churn_no=df[df['Exited']==0]['Balance']
Balance_churn_yes=df[df['Exited']==1]['Balance']
plt.xlabel('Balance')
plt.ylabel('Number of customers')
plt.title('Balance vs Number of customers')
plt.hist([Balance_churn_no, Balance_churn_yes],
color=['orange','red'],label=['Churn=No','Churn=Yes'])
plt.legend()
```

```
Gender_churn_no=df[df['Exited']==0]['Gender']
Gender_churn_yes=df[df['Exited']==1]['Gender']
plt.xlabel('Gender')
plt.ylabel('Number of customers')
plt.title('Gender vs Number of customers')
plt.hist([Gender_churn_no, Gender_churn_yes],
color=['purple','red'],label=['Churn=No','Churn=Yes'])
plt.legend()
```

```
Geography_churn_no=df[df['Exited']==0]['Geography']
Geography_churn_yes=df[df['Exited']==1]['Geography']
plt.xlabel('Geography')
plt.ylabel('Number of customers')
plt.title('Geography vs Number of customers')
plt.hist([Geography_churn_no, Geography_churn_yes],
color=['navy','red'],label=['Churn=No','Churn=Yes'])
plt.legend()
```

```
print(df.Geography.unique())
print(df.Gender.unique())
```

```
df=pd.get_dummies(df,columns=['Geography','Gender'])
df
```

```
X=df.drop(['Exited','RowNumber','Surname','CustomerId'],axis='columns')
Y=df['Exited']
X.head()
```

Before transformation

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
from sklearn.svm import SVC
model=SVC()
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
```

```
model.predict([ [850,43,2,125510.82,1,1,1,79084.10,0,0,1,1,0] ])
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
from sklearn import tree
model=tree.DecisionTreeClassifier()
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
model.predict([ [850,43,2,125510.82,1,1,1,79084.10,0,0,1,1,0] ])
```

```
print(accuracy_score(Y_test,Y_pred))
print(metrics.classification_report(Y_test,Y_pred))
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
from sklearn.ensemble import RandomForestClassifier
model=RandomForestClassifier()
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
model.predict([ [850,43,2,125510.82,1,1,1,79084.10,0,0,1,1,0] ])
```

```
print(accuracy_score(Y_test,Y_pred))
print(metrics.classification_report(Y_test,Y_pred))
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
from sklearn.neighbors import KNeighborsClassifier
model=KNeighborsClassifier(n_neighbors=32)
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
model.predict([ [850,43,2,125510.82,1,1,1,79084.10,0,0,1,1,0] ])
```

```
print(accuracy_score(Y_test,Y_pred))
print(metrics.classification_report(Y_test,Y_pred))
```

```
cols_to_scale=['CreditScore','Age','Tenure','Balance','EstimatedSalary']
from sklearn.preprocessing import MinMaxScaler
mx=MinMaxScaler()
df[cols_to_scale]=mx.fit_transform(df[cols_to_scale])
df.head()
```

```
X=df.drop(['Exited','RowNumber','Surname','CustomerId'],axis='columns')
Y=df['Exited']
X.head()
```

After Transformation

```

from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
from sklearn.svm import SVC
model=SVC()
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
model.predict([ [1.000,0.337838,0.2,0.500246,1,1,1,0.395400,0,0,1,1,0] ])

```

```

print(accuracy_score(Y_test,Y_pred))
print(metrics.classification_report(Y_test,Y_pred))

```

```

from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
from sklearn import tree
model=tree.DecisionTreeClassifier()
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
model.predict([ [1.000,0.337838,0.2,0.500246,1,1,1,0.395400,0,0,1,1,0] ])

```

```

print(accuracy_score(Y_test,Y_pred))
print(metrics.classification_report(Y_test,Y_pred))

```

```

from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
from sklearn.ensemble import RandomForestClassifier
model=RandomForestClassifier()
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
model.predict([ [1.000,0.337838,0.2,0.500246,1,1,1,0.395400,0,0,1,1,0] ])

```

```

print(accuracy_score(Y_test,Y_pred))
print(metrics.classification_report(Y_test,Y_pred))

```

```

from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
from sklearn.neighbors import KNeighborsClassifier
model=KNeighborsClassifier(n_neighbors=32)
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
model.predict([ [1.000,0.337838,0.2,0.500246,1,1,1,0.395400,0,0,1,1,0] ])

```

```

print(accuracy_score(Y_test,Y_pred))
print(metrics.classification_report(Y_test,Y_pred))

```

```

import tensorflow as tf
from tensorflow import keras
model=keras.Sequential([

```

```

        keras.layers.Dense(20,input_dim=13,activation='relu'),
        keras.layers.Dense(10,activation='relu'),
        keras.layers.Dense(1,activation='sigmoid')
])
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
model.fit(X_train,Y_train,epochs=100)

model.evaluate(X_test,Y_test)

Y_pred=model.predict(X_test).reshape(-1)
Y_pred=np.round(Y_pred)
Y_pred[:10]

Y_test[:10]

print(accuracy_score(Y_test,Y_pred))
print(metrics.classification_report(Y_test,Y_pred))

import seaborn as sns
cm=tf.math.confusion_matrix(labels=Y_test,predictions=Y_pred)

plt.figure(figsize=(9,6))
sns.heatmap(cm,annot=True, fmt='d')
plt.xlabel('Predictions')
plt.ylabel('Truth')

Accuracy
round((1521+198)/(1521+198+78+203),2)

Precision for 0 class i.e Precision for customers who did not churn
round(1521/(1521+203),2)

Precision for 1 class i.e Precision for customers who actually churn
round(198/(198+78),2)

Recall for 0 class
round(1521/(1521+78),2)

Recall for 1 class
round(198/(198+203),2)

```

B. SCREENSHOTS

```
[1] import pandas as pd
    from matplotlib import pyplot as plt
    import numpy as np
    from sklearn.metrics import accuracy_score
    from sklearn import metrics
    %matplotlib inline
```

```
[2] from google.colab import files
    uploaded = files.upload()
```

Choose files Churn_Modelling.csv

- Churn_Modelling.csv(text/csv) - 684858 bytes, last modified: 09/12/2021 - 100% done
- Saving Churn_Modelling.csv to Churn_Modelling.csv

```
df=pd.read_csv("Churn_Modelling.csv")
df.head(5)
```

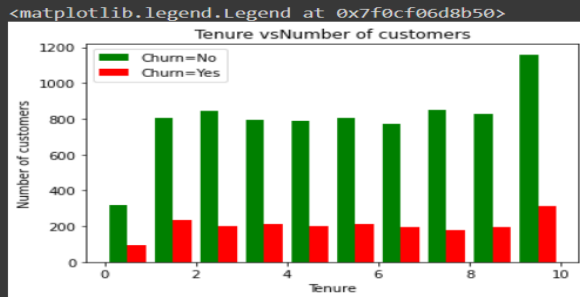
	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

```
[4] df.isna().sum()
```

```
RowNumber      0
CustomerId      0
Surname         0
CreditScore    0
Geography      0
Gender         0
Age            0
Tenure         0
Balance        0
NumOfProducts  0
HasCrCard      0
IsActiveMember 0
EstimatedSalary 0
Exited         0
dtype: int64
```

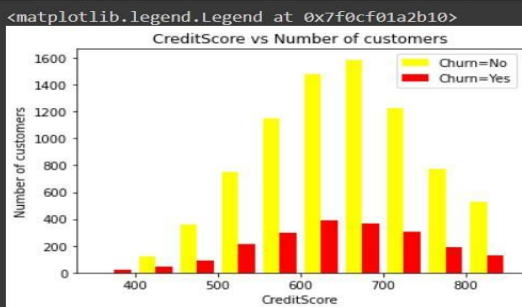
```
[6] tenure_churn_no=df[df['Exited']==0]['Tenure']
tenure_churn_yes=df[df['Exited']==1]['Tenure']

plt.xlabel('Tenure')
plt.ylabel('Number of customers')
plt.title('Tenure vsNumber of customers')
plt.hist([tenure_churn_no, tenure_churn_yes], color=['green','red'],label=['Churn=No','Churn=Yes'])
plt.legend()
```



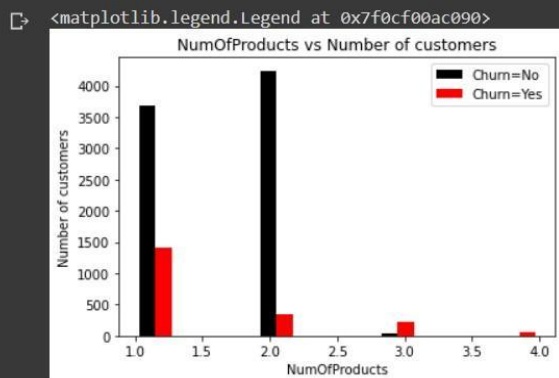
```
[7] CreditScore_churn_no=df[df['Exited']==0]['CreditScore']
CreditScore_churn_yes=df[df['Exited']==1]['CreditScore']

plt.xlabel('CreditScore')
plt.ylabel('Number of customers')
plt.title('CreditScore vs Number of customers')
plt.hist([CreditScore_churn_no, CreditScore_churn_yes], color=['yellow','red'],label=['Churn=No','Churn=Yes'])
plt.legend()
```



```
NumOfProducts_churn_no=df[df['Exited']==0]['NumOfProducts']
NumOfProducts_churn_yes=df[df['Exited']==1]['NumOfProducts']

plt.xlabel('NumOfProducts')
plt.ylabel('Number of customers')
plt.title('NumOfProducts vs Number of customers')
plt.hist([NumOfProducts_churn_no, NumOfProducts_churn_yes], color=['black','red'],label=['Churn=No','Churn=Yes'])
plt.legend()
```



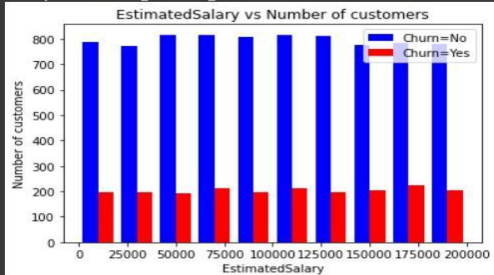
```

EstimatedSalary_churn_no=df[df['Exited']==0]['EstimatedSalary']
EstimatedSalary_churn_yes=df[df['Exited']==1]['EstimatedSalary']

plt.xlabel('EstimatedSalary')
plt.ylabel('Number of customers')
plt.title('EstimatedSalary vs Number of customers')
plt.hist([EstimatedSalary_churn_no, EstimatedSalary_churn_yes], color=['blue','red'],label=['Churn=No','Churn=Yes'])
plt.legend()

```

<matplotlib.legend.Legend at 0x7f0ceffe0d50>



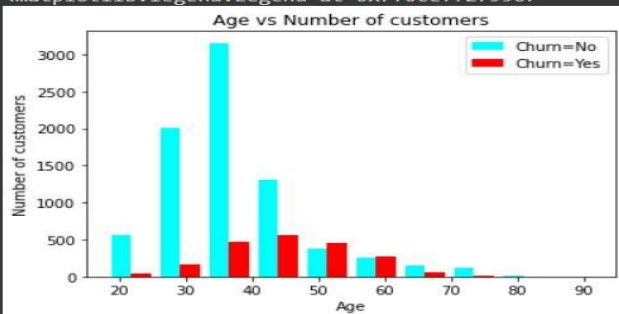
```

[10] Age_churn_no=df[df['Exited']==0]['Age']
Age_churn_yes=df[df['Exited']==1]['Age']

plt.xlabel('Age')
plt.ylabel('Number of customers')
plt.title('Age vs Number of customers')
plt.hist([Age_churn_no, Age_churn_yes], color=['cyan','red'],label=['Churn=No','Churn=Yes'])
plt.legend()

```

<matplotlib.legend.Legend at 0x7f0ceff27990>



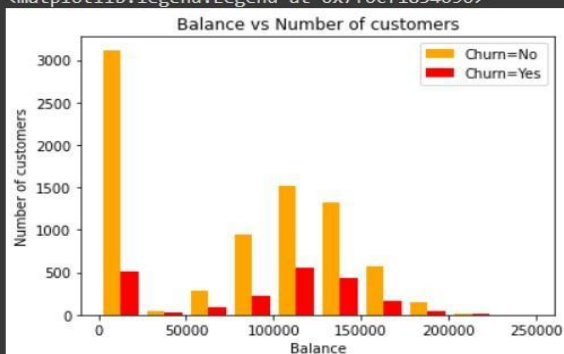
```

[12] Balance_churn_no=df[df['Exited']==0]['Balance']
Balance_churn_yes=df[df['Exited']==1]['Balance']

plt.xlabel('Balance')
plt.ylabel('Number of customers')
plt.title('Balance vs Number of customers')
plt.hist([Balance_churn_no, Balance_churn_yes], color=['orange','red'],label=['Churn=No','Churn=Yes'])
plt.legend()

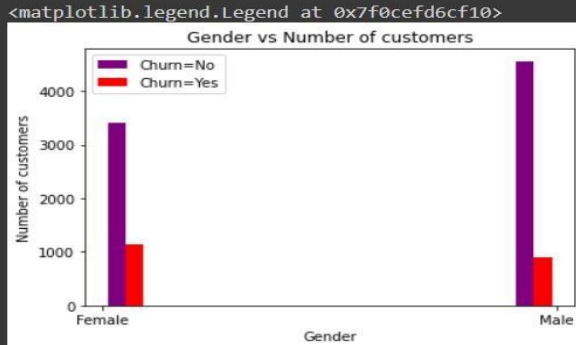
```

<matplotlib.legend.Legend at 0x7f0cf1854690>



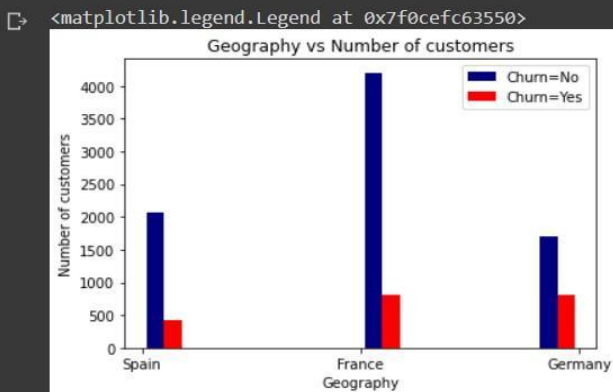
```
[13] Gender_churn_no=df[df['Exited']==0]['Gender']
Gender_churn_yes=df[df['Exited']==1]['Gender']

plt.xlabel('Gender')
plt.ylabel('Number of customers')
plt.title('Gender vs Number of customers')
plt.hist([Gender_churn_no, Gender_churn_yes], color=['purple','red'],label=['Churn=No','Churn=Yes'])
plt.legend()
```



```
Geography_churn_no=df[df['Exited']==0]['Geography']
Geography_churn_yes=df[df['Exited']==1]['Geography']

plt.xlabel('Geography')
plt.ylabel('Number of customers')
plt.title('Geography vs Number of customers')
plt.hist([Geography_churn_no, Geography_churn_yes], color=['navy','red'],label=['Churn=No','Churn=Yes'])
plt.legend()
```



```
[16] df=pd.get_dummies(df,columns=['Geography','Gender'])
df
```

RowNumber	CustomerId	Surname	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_France	Geography_Germany	Geography_Spain
0	1	Hargrave	619	42	2	0.00	1	1	1	101348.88	1	1	0	0
1	2	Hill	608	41	1	83807.86	1	0	1	112542.58	0	0	0	0
2	3	Onio	502	42	8	159660.80	3	1	0	113931.57	1	1	0	0
3	4	Boni	699	39	1	0.00	2	0	0	93826.63	0	1	0	0
4	5	Mitchell	850	43	2	125510.82	1	1	1	79084.10	0	0	0	0

```
x=df.drop(['Exited','RowNumber','Surname','CustomerId'],axis='columns')
y=df['Exited']
x.head()
```

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Geography_France	Geography_Germany	Geography_Spain	Gender_Female	Gender_Male
0	619	42	2	0.00	1	1	1	101348.88	1	0	0	1	0
1	608	41	1	83807.86	1	0	1	112542.58	0	0	1	1	0
2	502	42	8	159660.80	3	1	0	113931.57	1	0	0	1	0
3	699	39	1	0.00	2	0	0	93826.63	1	0	0	1	0
4	850	43	2	125510.82	1	1	1	79084.10	0	0	1	1	0

```
[19] from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
from sklearn.svm import SVC
model=SVC()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
model.predict([ [850,43,2,125510.82,1,1,1,79084.10,0,0,1,1,0] ])
```

/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but array([0])

```
[22] print(accuracy_score(y_test,y_pred))
print(metrics.classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.79	1.00	0.88	1584
1	0.00	0.00	0.00	416
accuracy			0.79	2000
macro avg	0.40	0.50	0.44	2000
weighted avg	0.63	0.79	0.70	2000

```
[24] from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
from sklearn import tree
model=tree.DecisionTreeClassifier()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
model.predict([ [850,43,2,125510.82,1,1,1,79084.10,0,0,1,1,0] ])
```

/usr/lib/python3.7/importlib/_bootstrap.py:219: RuntimeWarning: numpy.ndarray is not a valid Python object
return f(*args, **kwargs)
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but array([0])

```
[25] print(accuracy_score(y_test,y_pred))
print(metrics.classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.88	0.86	0.87	1579
1	0.51	0.54	0.53	421

```
[26] from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
from sklearn.ensemble import RandomForestClassifier
model=RandomForestClassifier()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
model.predict([ [850,43,2,125510.82,1,1,1,79084.10,0,0,1,1,0] ])
```

/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: "X does not have valid feature names, but"
array([0])

```
[27] print(accuracy_score(y_test,y_pred))
print(metrics.classification_report(y_test,y_pred))
```

0.8495		precision	recall	f1-score	support
	0	0.86	0.96	0.91	1573
	1	0.75	0.44	0.56	427
accuracy				0.85	2000
macro avg		0.81	0.70	0.73	2000
weighted avg		0.84	0.85	0.83	2000

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
from sklearn.neighbors import KNeighborsClassifier
model=KNeighborsClassifier(n_neighbors=32)
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
model.predict([ [850,43,2,125510.82,1,1,1,79084.10,0,0,1,1,0] ])
```

/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: "X does not have valid feature names, but"
array([0])

```
[29] print(accuracy_score(y_test,y_pred))
print(metrics.classification_report(y_test,y_pred))
```

0.8		precision	recall	f1-score	support
	0	0.80	1.00	0.89	1600
	1	0.00	0.00	0.00	400
accuracy				0.80	2000
macro avg		0.40	0.50	0.44	2000
weighted avg		0.64	0.80	0.71	2000

```
[30] cols_to_scale=['CreditScore','Age','Tenure','Balance','EstimatedSalary']
from sklearn.preprocessing import MinMaxScaler
mx=MinMaxScaler()
df[cols_to_scale]=mx.fit_transform(df[cols_to_scale])
df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_France	Geography_Germany	Geography_Spain
0	1	15634602	Hargrave	0.538	0.324324	0.2	0.000000	1	1	1	0.506735	1	1	0	
1	2	15647311	Hill	0.516	0.310811	0.1	0.334031	1	0	1	0.562709	0	0	0	
2	3	15619304	Onio	0.304	0.324324	0.8	0.636357	3	1	0	0.566654	1	1	0	
3	4	15701354	Boni	0.698	0.263784	0.1	0.000000	2	0	0	0.469120	0	1	0	
4	5	15737888	Mitchell	1.000	0.337838	0.2	0.500246	1	1	1	0.395400	0	0	0	

```
x=df.drop(['Exited','RowNumber','Surname','CustomerId'],axis='columns')
y=df['Exited']
x.head()
```

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Geography_France	Geography_Germany	Geography_Spain	Gender_Female	Gender_Male
0	0.538	0.324324	0.2	0.000000	1	1	1	0.506735	1	0	0	1	0
1	0.516	0.310811	0.1	0.334031	1	0	1	0.562709	0	0	1	1	0
2	0.304	0.324324	0.8	0.636357	3	1	0	0.569654	1	0	0	1	0
3	0.698	0.283784	0.1	0.000000	2	0	0	0.469120	1	0	0	1	0
4	1.000	0.337838	0.2	0.500246	1	1	1	0.395400	0	0	1	1	0

```
[33] from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
from sklearn.svm import SVC
model=SVC()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
model.predict([ [1.000,0.337838,0.2,0.500246,1,1,1,0.395400,0,0,1,1,0] ])
```

/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X
"X does not have valid feature names, but"
array([0])

```
[36] print(accuracy_score(y_test,y_pred))
print(metrics.classification_report(y_test,y_pred))
```

		precision	recall	f1-score	support
	0	0.87	0.86	0.86	1600
	1	0.46	0.47	0.47	400
	accuracy			0.78	2000
	macro avg	0.66	0.67	0.66	2000
	weighted avg	0.78	0.78	0.78	2000

```
[37] from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
from sklearn import tree
model=tree.DecisionTreeClassifier()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
model.predict([ [1.000,0.337838,0.2,0.500246,1,1,1,0.395400,0,0,1,1,0] ])
```

/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X
"X does not have valid feature names, but"
array([0])

```
[38] print(accuracy_score(y_test,y_pred))
print(metrics.classification_report(y_test,y_pred))
```

		precision	recall	f1-score	support
	0	0.88	0.88	0.88	1573
	1	0.55	0.54	0.54	427
	accuracy			0.81	2000
	macro avg	0.71	0.71	0.71	2000
	weighted avg	0.81	0.81	0.81	2000

```
[40] from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
from sklearn.ensemble import RandomForestClassifier
model=RandomForestClassifier()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
model.predict([ [1.000,0.337838,0.2,0.500246,1,1,1,0.395400,0,0,1,1,0] ])
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X
  "X does not have valid feature names, but"
array([0])
```

```
[41] print(accuracy_score(y_test,y_pred))
print(metrics.classification_report(y_test,y_pred))
```

```
0.8675
          precision    recall  f1-score   support

     0       0.88        0.97        0.92        1591
     1       0.79        0.48        0.60         409

 accuracy
macro avg       0.83        0.73        0.76        2000
weighted avg     0.86        0.87        0.85        2000
```

```
[42] from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
from sklearn.neighbors import KNeighborsClassifier
model=KNeighborsClassifier(n_neighbors=32)
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
model.predict([ [1.000,0.337838,0.2,0.500246,1,1,1,0.395400,0,0,1,1,0] ])
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X
  "X does not have valid feature names, but"
array([0])
```

```
[43] print(accuracy_score(y_test,y_pred))
print(metrics.classification_report(y_test,y_pred))
```

```
0.819
          precision    recall  f1-score   support

     0       0.83        0.97        0.90        1603
     1       0.62        0.23        0.33         397

 accuracy
macro avg       0.73        0.60        0.61        2000
weighted avg     0.79        0.82        0.78        2000
```

```
import tensorflow as tf
from tensorflow import keras
model=keras.Sequential([
    keras.layers.Dense(20,input_dim=13,activation='relu'),
    keras.layers.Dense(10,activation='relu'),
    keras.layers.Dense(1,activation='sigmoid')
])
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
model.fit(x_train,y_train,epochs=100)
```

```
250/250 [=====] - 0s 2ms/step - loss: 0.3279 - accuracy: 0.8627
Epoch 73/100
250/250 [=====] - 0s 2ms/step - loss: 0.3276 - accuracy: 0.8651
Epoch 74/100
250/250 [=====] - 0s 2ms/step - loss: 0.3284 - accuracy: 0.8616
Epoch 75/100
250/250 [=====] - 0s 2ms/step - loss: 0.3264 - accuracy: 0.8637
Epoch 76/100
250/250 [=====] - 0s 2ms/step - loss: 0.3271 - accuracy: 0.8646
Epoch 77/100
250/250 [=====] - 0s 2ms/step - loss: 0.3264 - accuracy: 0.8655
Epoch 78/100
```

```
[46] model.evaluate(x_test,y_test)

63/63 [=====] - 0s 1ms/step - loss: 0.3527 - accuracy: 0.8630
[0.3527148962020874, 0.8629999756813049]

[47] y_pred=model.predict(x_test).reshape(-1)
y_pred=np.round(y_pred)
y_pred[:10]

array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.], dtype=float32)

[48] y_test[:10]

3817    1
9399    0
8070    0
2446    0
5300    0
6361    0
7616    0
8046    0
6199    0
3945    0
Name: Exited, dtype: int64
```

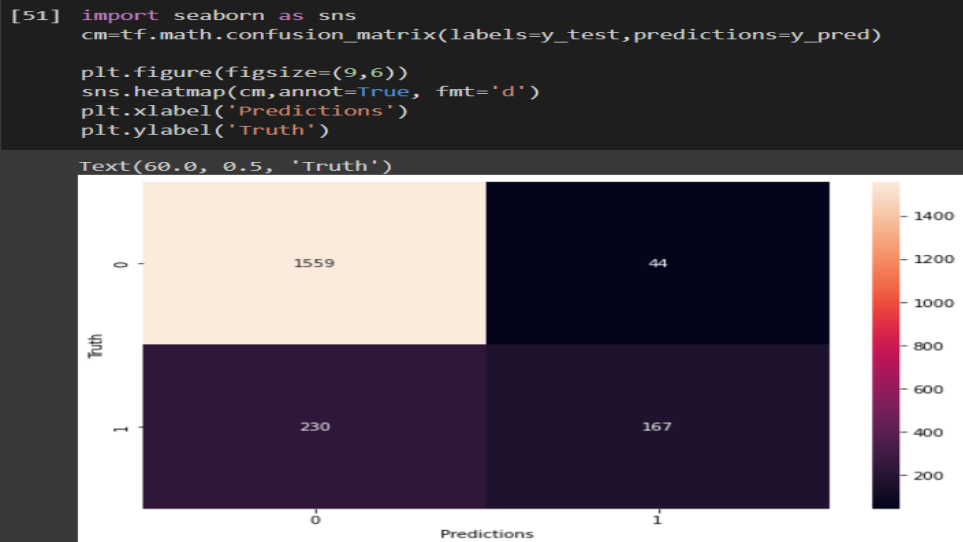
```
[49] print(accuracy_score(y_test,y_pred))
print(metrics.classification_report(y_test,y_pred))

0.863

              precision    recall  f1-score   support

         0       0.87       0.97       0.92       1603
         1       0.79       0.42       0.55        397

 accuracy          0.86
 macro avg          0.83
weighted avg          0.86
```



Accuracy

```
[52] round((1559+167)/(1559+167+44+230),2)
```

0.86

Precision for 0 class i.e Precision for customers who did not churn

```
[53] round(1559/(1559+230),2)
```

0.87

Precision for 1 class i.e Precision for customers who actually churn

```
[54] round(167/(167+44),2)
```

0.79

Recall for 0 class

```
[55] round(1559/(1559+44),2)
```

0.97

Recall for 1 class

```
[56] round(167/(167+230),2)
```

0.42

C. PUBLICATION WITH PLAGIARISM REPORT

