CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING

PROJECT REPORT

(BTCS-603-18)

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELORS OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Sharpening Minds, Brightening Tomorrow.
Gulzar Group of Institutes

UNDER THE GUIDANCE OF

Dy. Dean & Prof. Dr. Amandeep Singh

SUBMITTED BY:

Sakshi Jha (1906138)

ਆਈ.ਕੇ. ਗੁਜਰਾਲ ਪੰਜਾਬ ਟੈਕਨੀਕਲ ਯੂਨੀਵਰਸਿਟੀ, ਜਲੰਧਰ

I.K. GUJRAL PUNJAB TECHNICAL UNIVERSITY, JALANDHAR

CANDIDATE'S DECLARATION AND CERTIFICATE

We hereby certify that the work, which is being presented in this report entitled, Credit Card Fraud Detection Using Machine Learning, in partial fulfillment of the requirements for the degree of **B. TECH** submitted in the **Computer Science and Engineering**, Gulzar Group of Institutions, Khanna, Punjab; by **Sakshi(1906138)** is the authentic record of our own work carried out under the supervision of **Dy. Dean & Prof. Dr.Amandeep Singh, Computer science and Engineering**, Gulzar Group of Institutions, Khanna, Punjab.

We further declare that the matter embodied in this report has not been submitted by us for the award of any other degree.

Candidate(s) Signature

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Signature of HOD **Signature of Supervisor**

Er. Jai Parkash **Dy. Dean & Prof. Dr. Amandeep Singh**

Date:

ACKNOWLEDGEMET

It is our pleasure to acknowledge the contributions of all who have helped us and supported us during this Project report.

First, we thank God for helping us in one way or another and providing strength and endurance to us. We wish to express my sincere gratitude and indebtedness to our supervisor, Dy. Dean & Prof. Dr. Amandeep Singh, Computer Science and Engineering, Gulzar Group of Institutions, Khanna, Punjab; for his intuitive and meticulous guidance and perpetual inspiration in completion of this report. In spite of his busy schedule, he rendered help whenever needed, giving useful suggestions and holding informal discussions. His invaluable guidance and support throughout this work cannot be written down in few words. I also thank him for providing facilities for my work in the department name.

I am also humbly obliged by the support of our group members and friends for their love and caring attitude. The sentimental support they rendered to us is invaluable and everlasting. They have helped us through thick and thin and enabled us to complete the work with joy and vigor. We thank the group members for entrusting in each other and following directions, without them this report would never have been possible.

We are also thankful to our parents, elders and all family members for their blessing, motivation and inspiration throughout our work and bearing with us even during stress and bad temper. They have always provided us a high moral support and contributed in all possible ways in completion of this Capstone report.

ABSTRACT

Fraud detection is a process of monitoring the transaction behavior of a cardholder in order to detect whether an incoming transaction is done by the cardholder or others. Credit cards are widely used due to the popularization of ecommerce and the development of mobile intelligent devices. It is estimated that losses are increased yearly at double digit rates by 2020. Since the physical card is not needed in the online transaction environment and the card's information is enough to complete a payment, it is easier to conduct a fraud than before. Hence, fraud detection is essential and necessary.

The performance of fraud detection in credit card transactions is greatly affected by the sampling method of the dataset and the choice of variables and the detection techniques used. This paper investigates the performance of logistic regression (LR), k-nearest-neighbor(KNN), Support vector machine(SVM), Decision Tree(DT) and Random Forest Classifier and Naïve Bayes on credit card fraud data. The dataset of credit card transactions obtained from European cardholders containing 284,807 transactions. A mixture of under-sampling and oversampling techniques applied to the unbalanced data. The five strategies used to the raw and preprocessed data, respectively. This work implemented in Python. The results are shown in comparison.

The main objectives of the projects on credit card fraud detection system are to manage the details of credit card, transactions, datasets, files, prediction. It manages all the information about credit card, customers, prediction, credit card. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the credit card, transactions, customers, datasets. It tracks all the details about the datasets, files, prediction.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER – 1 INTRODUCTION

Credit cards are widely used due to the popularization of ecommerce and the development of mobile intelligent devices. The Credit Card Is a Small Plastic Card Issued to Users as a System of Payment. It Allows Its Cardholder to Buy Goods and Services Based on The Cardholder's Promise to Pay for These Goods and Services. Credit Card Security Relies on The Physical Security of The Plastic Card as Well as The Privacy of The Credit Card Number. Card-not-present transactions (i.e., online transaction without a physical card) is more popular, especially all credit card operations are performed by web payment gateways, e.g., PayPal and Alipay. Credit card has made an online transaction easier and more convenient.

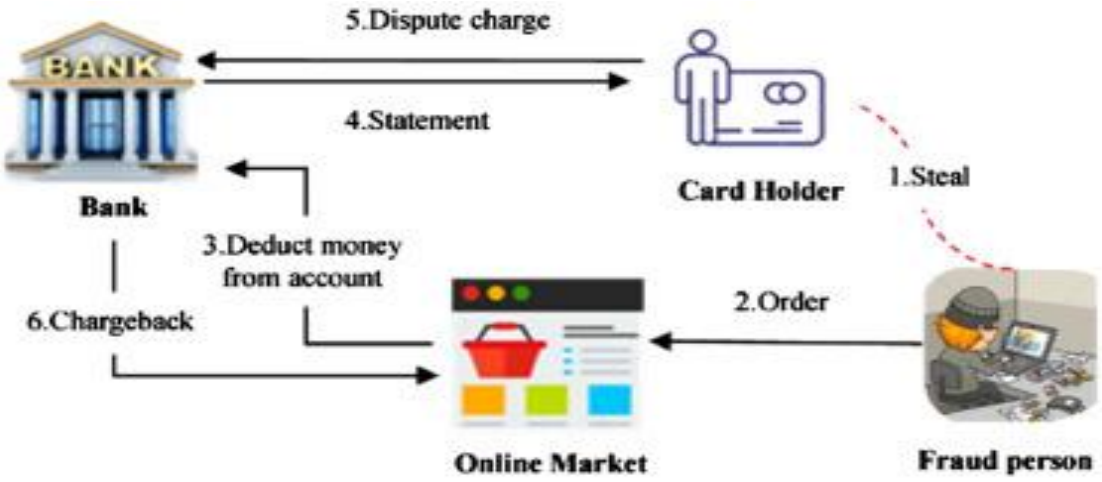


Fig. 1.1 Overview of Credit Card Fraud [4]

However, there is a growing trend of transaction frauds resulting in a great loss of money every year. It is estimated that losses are increased yearly at double digit rates by 2020. Since the physical card is not needed in the online transaction environment and the card's information is enough to complete a payment, it is easier to conduct a fraud than before. Transaction fraud has become a top barrier to the development of e-commerce and has a dramatic influence on the economy. Hence, fraud detection is essential and necessary. Fraud detection is a process of monitoring the transaction behavior of a cardholder in order to detect whether an incoming transaction is done by the cardholder or others. In 2018 Credit card fraud losses in London estimated US dollar 844.8 million. Use of credit cards for online purchases has dramatically increased and it caused an explosion in the credit card fraud.

As credit card becomes the most popular mode of payment for both online as well as regular purchase, cases of fraud associated with it are also rising. In real life, fraudulent transactions are scattered with genuine transactions and simple pattern matching techniques are not often sufficient to detect those frauds accurately. Implementation of efficient fraud detection systems has thus become imperative for all credit card issuing banks to minimize their losses. These frauds are classified as:

- Credit Card Frauds: Online and Offline
- Card Theft
- Account Bankruptcy
- Device Intrusion
- Application Fraud
- Counterfeit Card

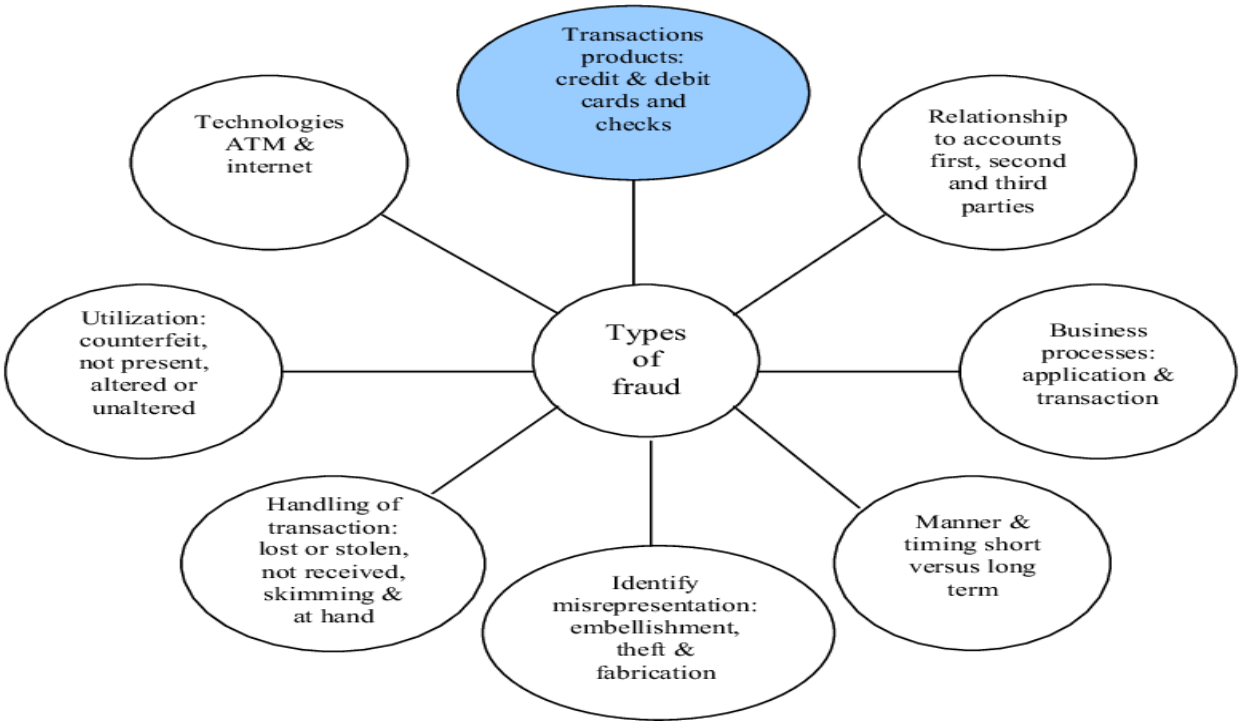


Fig. 1.2 Types of Fraud [1]

Credit card fraud events take place frequently and then result in huge financial losses. Criminals can use some technologies such as Trojan or Phishing to steal the information of other people's credit cards. Therefore, an effective fraud detection method is important since it can identify a fraud in time when a criminal uses a stolen card to consume.

CHAPTER - 2 LITERATURE REVIEW

In previous studies, many methods have been implemented to detect fraud using supervised, unsupervised algorithms and hybrid ones. Fraud types and patterns are evolving day by day. It is important to have a clear understanding of technologies behind fraud detection. Here discuss machine learning models, algorithms and fraud detection models used in earlier studies.

These classifiers were evaluated using a credit card fraud detection dataset generated from European cardholders in 2013. This dataset is highly imbalanced. The researcher used the classification accuracy to assess the performance of each ML approach.

- Prajwal Save et al. [2018] have proposed a model based on a decision tree and a combination of Luhan's and Hunt's algorithms. Luhan's algorithm is used to determine whether an incoming transaction is fraudulent or not. It validates credit card numbers via the input, which is the credit card number. Address Mismatch and Degree of Outlier are used to assess the deviation of each incoming transaction from the cardholder's normal profile. In the final step, the general belief is strengthened or weakened using Bayes Theorem, followed by recombination of the calculated probability with the initial belief of fraud using advanced combination heuristic.

- Vimala Devi. J et al. [2019] To detect counterfeit transactions, three machine-learning algorithms were presented and implemented. There are many measures used to evaluate the performance of classifiers or predictors, such as the Vector Machine, Random Forest, and Decision Tree. These metrics are either prevalence-dependent or prevalence-independent. Furthermore, these techniques are used in credit card fraud detection mechanisms, and the results of these algorithms have been compared.

- Pawan and Chaudhary et al. [2020] supervised algorithms were presented Deep learning, Logistic Regression, Nave Bayesian, Support Vector Machine (SVM), Neural Network, Artificial Immune System, K Nearest Neighbour, Data Mining, Decision Tree, Fuzzy logic based System, and Genetic Algorithm are some of the techniques used. Credit card fraud detection algorithms identify transactions that have a high probability of being fraudulent. We compared machine-learning algorithms to prediction, clustering, and outlier detection.

- Shenyang Xuan et al. [2021] For training the behavioural characteristics of credit card transactions, the random Forest classifier was used. The following types are used to train the normal and fraudulent behaviour features Random forest-based on random trees and random forest based on CART. To assess the model's effectiveness, performance measures are computed.

- Krishna Modi et al. [2016] investigated several techniques that were used for detecting the fraudulent transactions and provided a comparative study amongst them. The fraudulent transactions can be detected by utilizing either one of these or integrating any of these methods. The model can possibly be trained in a more accurate manner by adding new features. Several data mining techniques are being used by bank and credit card companies for detecting fraud behaviors.

- Zahra Kazmi et al. [2013] proposed Deep autoencoder which is used to extract the best characteristics of the information from the credit card transaction. This will further add SoftMax software to resolve the class labels issues. An overcomplete autoencoder is used to map the data into a high dimensional space and a sparse model was used in a descriptive manner which provides benefits for the classification of a type of fraud. Deep learning is one of the most motivated and powerful techniques being employed for the detection of fraud in the credit card. These types of networks have a complex distribution of data which is very difficult to recognize. Deep autoencoder has been used in some stages to extract the best features of the data and for the classification purposes. Also, higher accuracy and low variance are achieved within these networks.

- John O. Awoyemi et al. [2014] proposed an investigation through which the performances of several algorithms were evaluated when they were applied on credit card fraud data that is highly skewed. The European cardholders‟ 284,807 transactions were used as a source to generate the dataset of credit card transactions. On the skewed data, a hybrid approach of under-sampling and oversampling is performed. On raw and preprocessed data, there are three different techniques applied in Python. Based on certain parameters like precision, sensitivity, accuracy, balanced classification rate and so on the performances of these techniques are evaluated. It is seen through the achieved results that in comparison to naïve Bayes and logistic regression approaches, the performance of k-NN is better.

CHAPTER – 3 PROBLEM STATEMENT

3.1 Problem Statement: The Credit Card Fraud Detection Problem includes modeling past credit card transactions with the knowledge of the ones that turned out to be fraud. This model is then used to identify whether a new transaction is fraudulent or not.

3.2 Objective of the Project: -

The main objectives of the projects on credit card fraud detection system are to manage the details of credit card, transactions, datasets, files, prediction. It manages all the information about credit card, customers, prediction, credit card. The purpose of the project is to build an application program to reduce the manual work for managing the credit card, transactions, customers, datasets. It tracks all the details about the datasets, files, prediction.

Functionalities provided by Credit Card Fraud Detection System are as follows: -

- It tracks all the information of transaction, customers, files etc.
- Manage the information of transactions.
- It deals with monitoring the information and transactions of files.
- Manages the information of credit card.
- Manages the information of files.

3.3 Scope of the Project: -

Can be highly developed and reduce more fraud activities. Highly complexity can increase the detection of the irregular activities. We used supervised machine learning algorithms to detect credit card fraud transactions using real datasets. We use these algorithms to build classification using machine learning methods. We found key variables that lead to greater accuracy in detecting credit card mall fraudulent transactions.

3.4 Feasibility Study of the Project

After doing the project Credit Card Fraud Detection System, study and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible - given unlimited resources and infinite time.

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirement.

3.4.1 Economic Feasibility: -

This is a very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor.

- All hardware and software cost must be borne by the organization.
- Overall, we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later running cost for system.

3.4.2 Technical Feasibility: -

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of frontend and backend platforms.

3.4.3 Social Feasibility: -

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of Register Module: confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system

3.5 Significance of the Project

The primary significance in this project is to help users to well-organized every transaction and minimize the fraud detection. Credit card fraud detection is the process of identifying purchase attempts that are fraudulent and rejecting them rather than processing the order. There are a variety of tools and techniques available for detecting fraud, with most merchants employing a combination of several of them.

Due to Behavior and location analysis approach**,** there is a drastic reduction in the number of False Positives transactions identified as malicious by an FDS although they are genuine**.** The system stores previous transaction patterns for each u ser.

CHAPTER- 4 METHODOLOGY

4.1 System Design of Credit Card Fraud Detection: -

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the client's requirements into a logically working system.

Normally, design is performed in the following in the following two steps:

- **Primary Design Phase:** In this phase, the system is designed at block level. The blocks are created based on analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimizing the information flow between blocks. Thus, all activities which require more interaction are kept in one block.

- **Secondary Design Phase:** In the secondary phase the detailed design of every block is performed.

The general tasks involved in the design process are the following:

- Design various blocks for overall system processes.

- Design smaller, compact, and workable modules in each block.

- Design various database structures.

- Specify details of programs to achieve desired functionality.

- Design the form of inputs, and outputs of the system.

- Perform documentation of the design.

- **User Interface Design**: - User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue.

The following steps are various guidelines for User Interface Design:

- The system user should always be aware of what to do next.
- The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.
- Message, instructions, or information should be displayed long enough to allow the system user to read them.
- Use display attributes sparingly.
- Default values for fields and answers to be entered by the user should be specified.
- A user should not be allowed to proceed without correcting an error.

4.2 TOOLS AND TECHNOLOGY USED:

4.2.1 PYTHON

Python is developed by **Guido van Rossum**. Guido van Rossum started implementing Python in 1989. Python is a very simple programming language so even if you are new to programming, you can learn python without facing any issues. Some of the best Python features are:

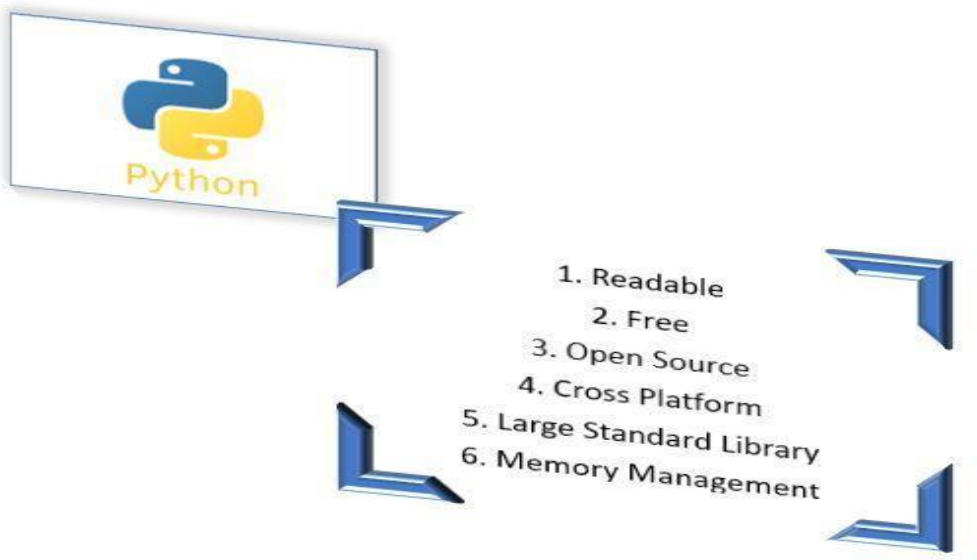


Fig. 4.1 Python Programming language [5]

Python is a dynamic, interpreted (bytecode-compiled) language. There are no type declarations of variables, parameters, functions, or methods in source code. This makes the code short and flexible, and you lose the compile-time type checking of the source code. Python tracks the types of all values at runtime and flags code that does not make sense as it runs. Python source files use the ". Pie" extension and are called "modules". With a Python module "hello.py," the easiest way to run it is with the shell command "python hello.py Alice" which calls the Python interpreter to execute the code in "hello.py", passing it the command line argument "Alice". Python can be used as:

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can be used to handle big data and perform complex mathematics.
- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.).

It has a simple syntax like the English language. It has syntax that allows developers to write programs with fewer lines than some other programming languages. It runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. It can be treated in a procedural way, an object orientated way or a functional way.

4.2.2 Data Types in Python

Variables can hold values, and every value has a data-type. Python is a dynamically typed language; hence we do not need to define the type of the variable while declaring it. The interpreter implicitly binds the value with its type. (Example: x=5)

The variable **x** holds integer value five and we did not define its type. Python interpreter will automatically interpret variables **x** as an integer type.

Python enables us to check the type of the variable used in the program. Python provides us the **type ()** function, which returns the type of the variable passed.

Python provides various standard data types that define the storage method on each of them. The data types defined in Python are given below.

- Numeric
- Sequence Type
- Boolean
- Set
- Dictionary

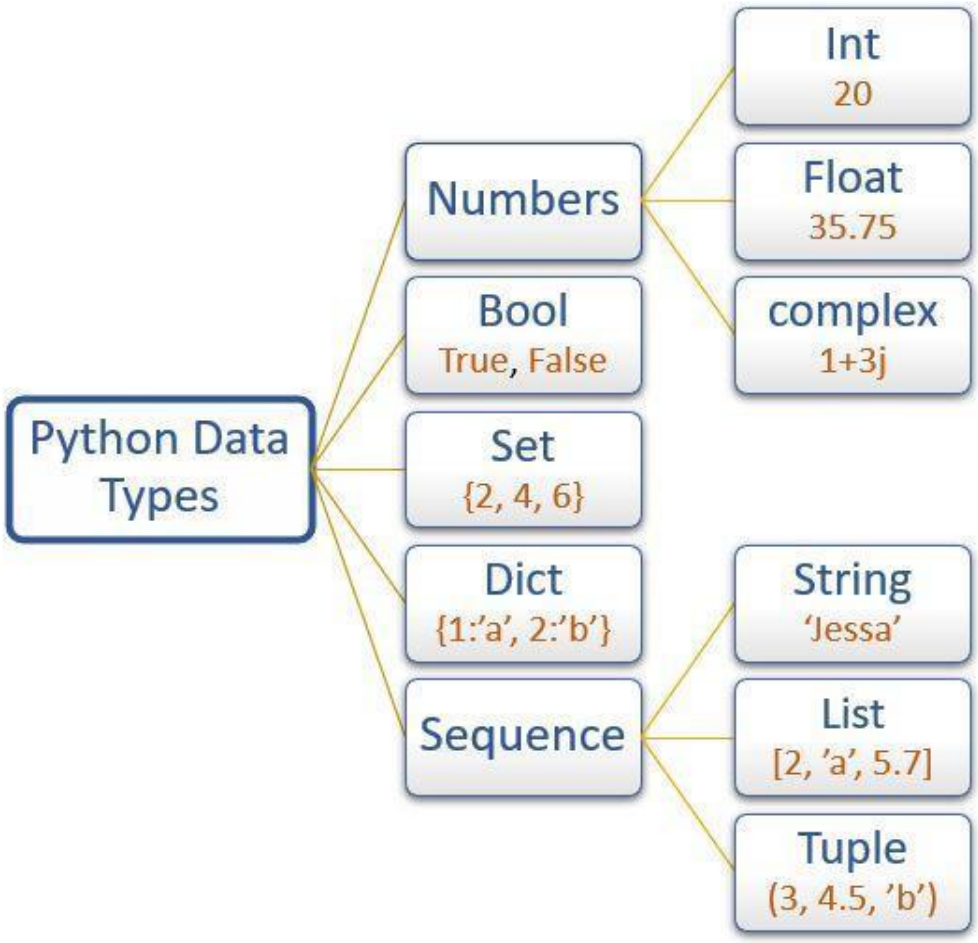


Fig. 4.2 Data Types in Python[3]

Number stores numeric values. The integer, float, and complex values belong to a Python Numbers data-type. Python provides the **type ()** function to know the data-type of the variable. Similarly, the **is instance ()** function is used to check an object belongs to a particular class.

Python supports three types of numeric data.

- **Int -** Integer value can be any length such as integers 10, 2, 29, -20, -150 etc. Python has no restriction on the length of an integer. Its value belongs to int.
- **Float -** Float is used to store floating-point numbers like 1.9, 9.902, 15.2, etc. It is accurate up to 15 decimal points.
- **Complex -** A complex number contains an ordered pair, i.e., x + is where x and y denote the real and imaginary parts, respectively. The complex numbers like 2.14j, 2.0 + 2.3j, etc.

4.2.2.1 Strings:

The string can be defined as the sequence of characters represented in the quotation marks. In Python, we can use single, double, or triple quotes to define a string.

String handling in Python is a straightforward task since Python provides built-in functions and operators to perform operations in the string.

In the case of string handling, the operator + is used to concatenate two strings as the operation "hello"+" python" returns "hello python".

4.2.2.2 Lists:

Python Lists are like arrays in C. However, the list can contain data of different types. The items stored in the list are separated with a comma (,) and enclosed within square brackets [].

We can use slice [:] operators to access the data of the list. The concatenation operator (+) and repetition operator (*) works with the list in the same way as they were working with the strings.

4.2.2.3 Sets:

Python Set is the unordered collection of the data type. It is inerrable, mutable (can modify after creation), and has unique elements. In set, the order of the elements is undefined; it may return the changed sequence of the element. The set is created by using a built-in function **set (),** or a sequence of elements is passed in the curly braces and separated by the comma. It can contain various types of values.

4.2.2.4 Dictionary:

Dictionary is an unordered set of a key-value pair of items. It is like an associative array or a hash table where each key stores a specific value. Key can hold any primitive data type, whereas value is an arbitrary Python object.

The items in the dictionary are separated with the comma (,) and enclosed in the curly braces {}.

4.3 Libraries:

First step is to import all the important libraries. Libraries are-

- NumPy
- Pandas
- Seaborn
- Matplotlib
- Scikit-learn

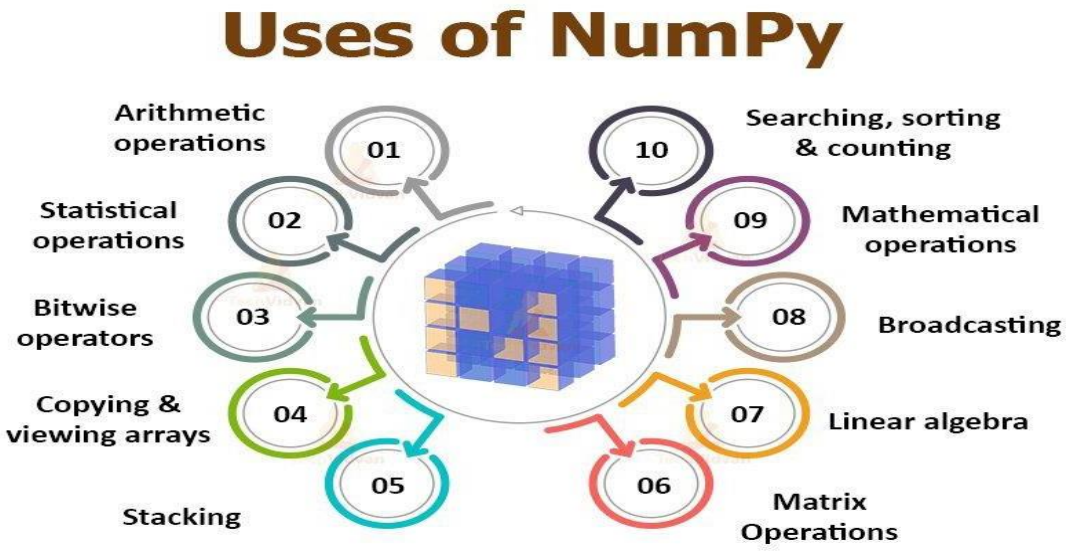
4.3.1 NumPy:

⇨ NumPy is a Python library used for working with arrays.

⇨ It also has functions for working in domain of linear algebra, Fourier transform, and matrices.

⇨ NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely.

⇨ NumPy stands for Numerical Python.



⇨

Fig. 4.3 Uses of NumPy in Python[8]

4.3.2 Pandas:

=>Pandas is a Python library used for working with data sets.

=> It has functions for analyzing, cleaning, exploring, and manipulating data.

=> The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

Fig. 4.4 Pandas Operations in Python[9]

4.3.3 Matplotlib:

Matplotlib is a multi-platform data visualization library built on NumPy arrays, and designed to work with the broader SciPy stack. It was conceived by John Hunter in 2002, originally as a patch to Python for enabling interactive MATLAB-style plotting from the Python command line.

4.3.4 Seaborn:

Seaborn helps to visualize the statistical relationships, to understand how variables in a dataset are related to one another and how that relationship is dependent on other variables, we perform statistical analysis. This Statistical analysis helps to visualize the trends and identify various patterns in the dataset.

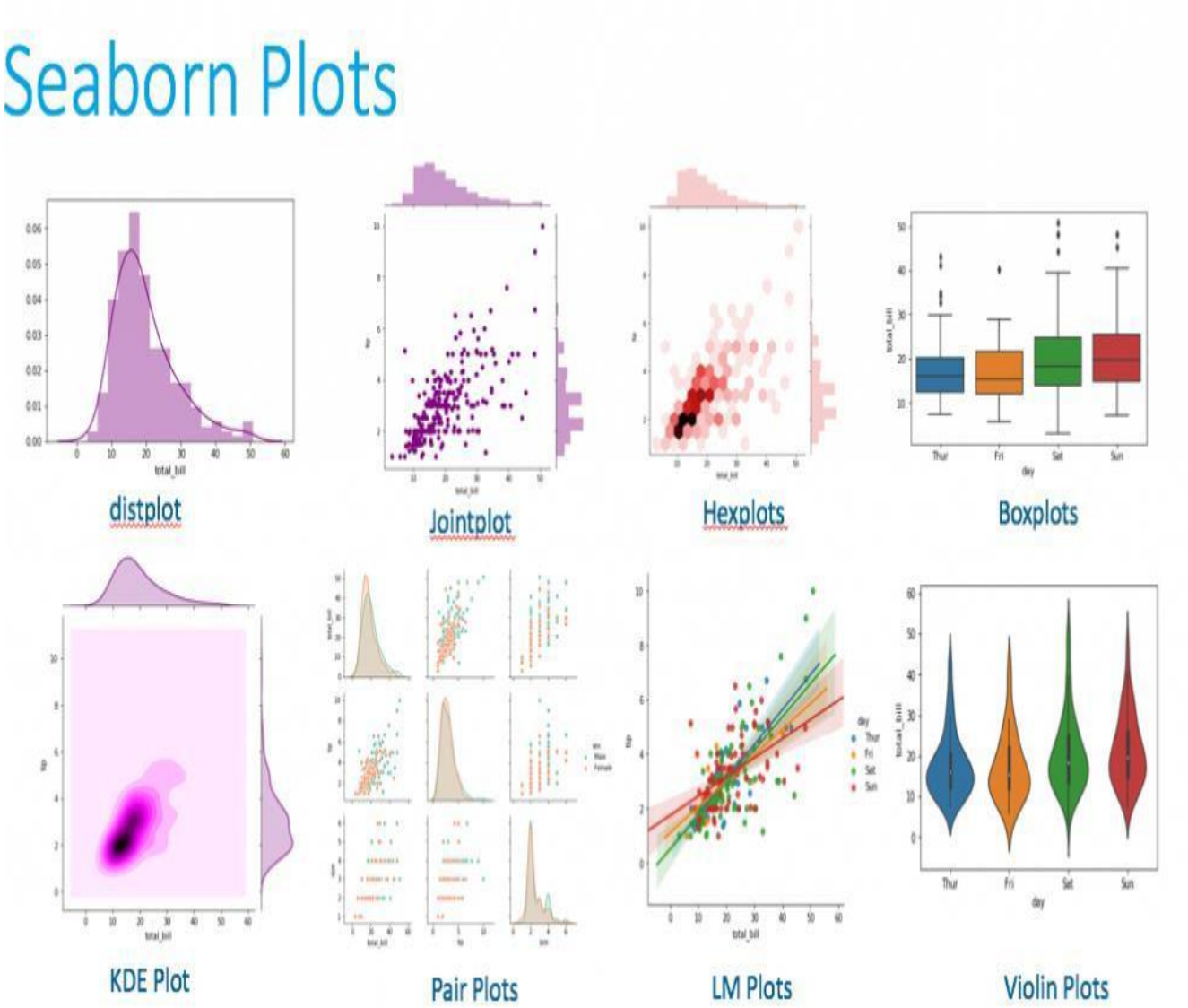


Fig. 4.5 Types of plots in Seaborn [9]

4.3.5 Scikit-learn:

Scikit-learn (Sklearn) 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 consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

For example-

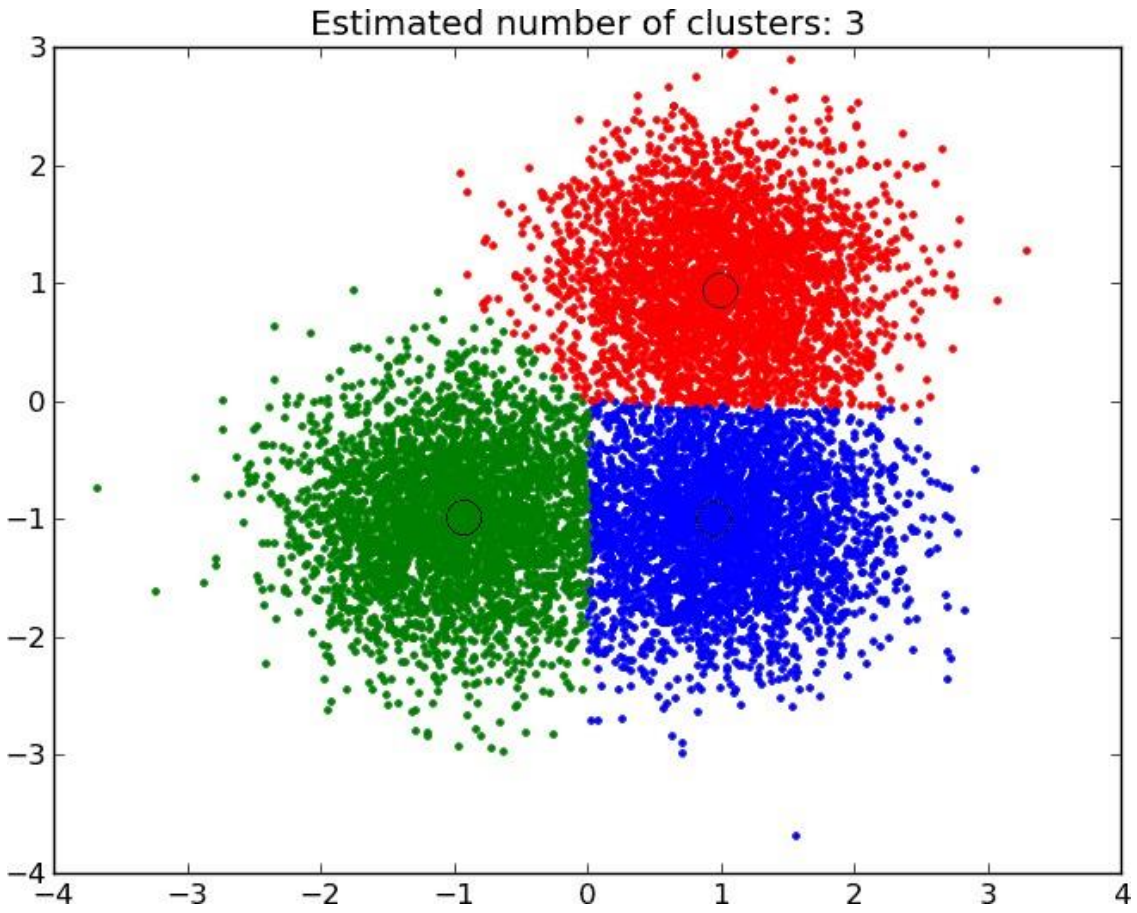


Fig. 4.6 Clusters in sklearn[10]

4.4 Time Frame Required for Various Stages:

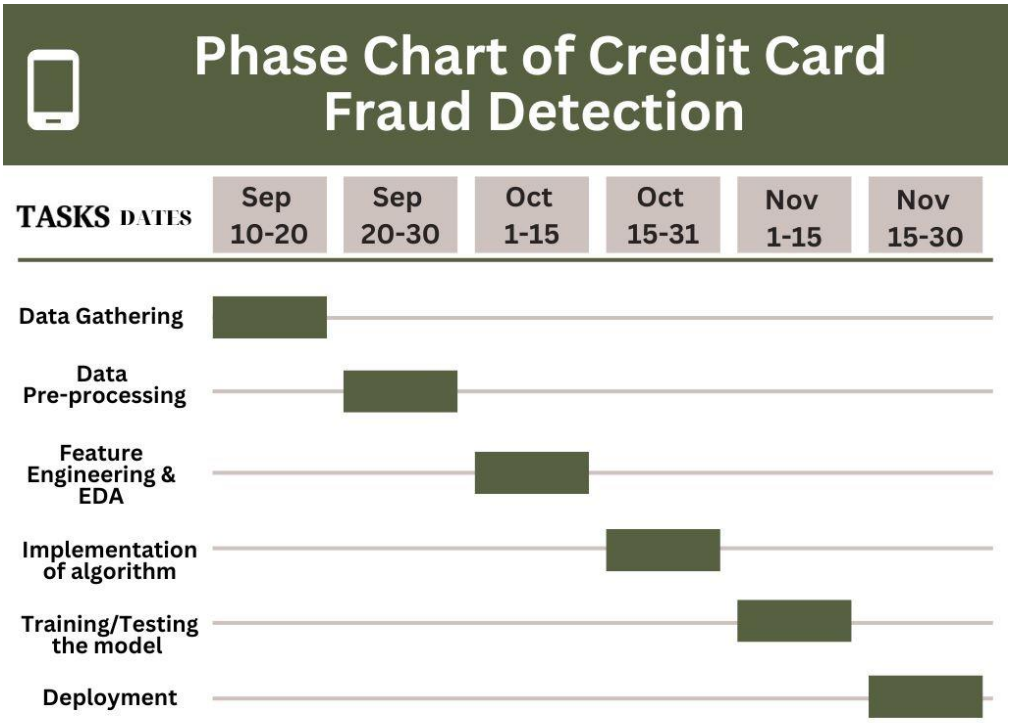


Fig. 4.8 Time Frame Required Various Stages

- Problem Definition
- Data Collection
- Data Preparation
- Data Visualization
- ML Modeling
- Feature Engineering

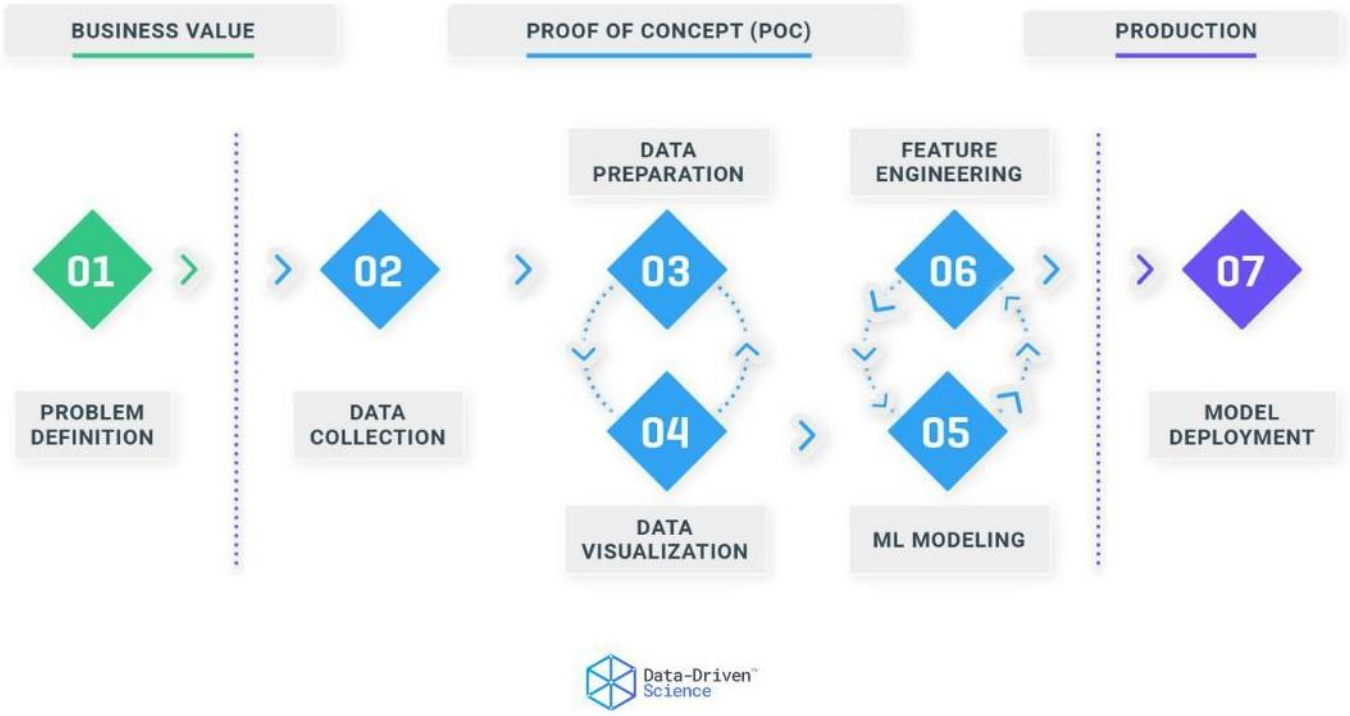


Fig. 4.9 Stages of ML Model [7]

These 7 stages are the key steps in our framework. We have categorized them additionally into groups to get a better understanding of the larger picture.

The stages are grouped into 3 phases:

1. Business Value
2. Proof of Concept (POC)
3. Production

4.5 Problem Definition:

The Credit Card Fraud Detection Problem includes modeling past credit card transactions with the knowledge of the ones that turned out to be fraud. This model is then used to identify whether a new transaction is fraudulent or not.

Few possible questions:

- What is the business?
- Why does the problem need to be solved?
- Is a traditional solution available to solve the problem?
- If probabilistic in nature, then does available data allow to model it?
- What is a measurable business goal?

4.6 Data Collection:

Once the goal is clearly defined, one must start getting the data that is needed from various available data sources.

At this stage, some of the questions worth considering are:

What data do I need for my project?

- Where is that data available?
- How can I obtain it?
- What is the most efficient way to store and access all of it?

There are many ways to collect data that is used for Machine Learning. For example, focus groups, interviews, surveys, and internal usage & user data. Also, public data can be another source and is usually free. These include research and trade associations such as banks, publicly-traded corporations, and others. If data is not publicly available, one could also use web scraping to get it (however, there are some legal restrictions).

- **Dataset is look like:**

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CRE
0	100002	1	Cash loans	M	N	Y	0	202500.000	4065
1	100003	0	Cash loans	F	N	N	0	270000.000	12935
2	100004	0	Revolving loans	M	Y	Y	0	67500.000	1350
3	100006	0	Cash loans	F	N	Y	0	135000.000	3126
4	100007	0	Cash loans	M	N	Y	0	121500.000	5130
5	100008	0	Cash loans	M	N	Y	0	99000.000	4904
6	100009	0	Cash loans	F	Y	Y	1	171000.000	15607
7	100010	0	Cash loans	M	Y	Y	0	360000.000	15300
8	100011	0	Cash loans	F	N	Y	0	112500.000	10196
9	100012	0	Revolving loans	M	N	Y	0	135000.000	4050
10	100014	0	Cash loans	F	N	Y	1	112500.000	6525
11	100015	0	Cash loans	F	N	Y	0	38419.155	1483
12	100016	0	Cash loans	F	N	Y	0	67500.000	808
13	100017	0	Cash loans	M	Y	N	1	225000.000	9184
14	100018	0	Cash loans	F	N	Y	0	189000.000	7736

Fig. 4.10 Attributes of Dataset

4.7 Data Preparation:

The third stage is the most time-consuming and labor-intensive. Data Preparation can take up to 70% and sometimes even 90% of the overall project time. But what is the purpose of this stage?

Well, the type and quality of data that is used in a Machine Learning model affects the output considerably. In Data Preparation one explores, pre-processes, conditions, and transforms data prior to modeling and analysis. It is essential to understand the data, learn about it, and become familiar before moving on to the next stage.

Some of the steps involved in this stage are:

- Data Filtering
- Data Validation & Cleansing
- Data Formatting
- Data Aggregation & Reconciliation

4.7.1 Memory Usage:

When dealing with a large amount of data, we must be careful with how we use memory. Shortage of memory is a common issue when we have a large amount of data at hand. In case the entire RAM space is consumed, the program can crash and throw a Memory Error, which can be tricky to handle at times. Limiting the memory usage becomes important in this case. Reducing memory usage also speeds up computation and helps save time.

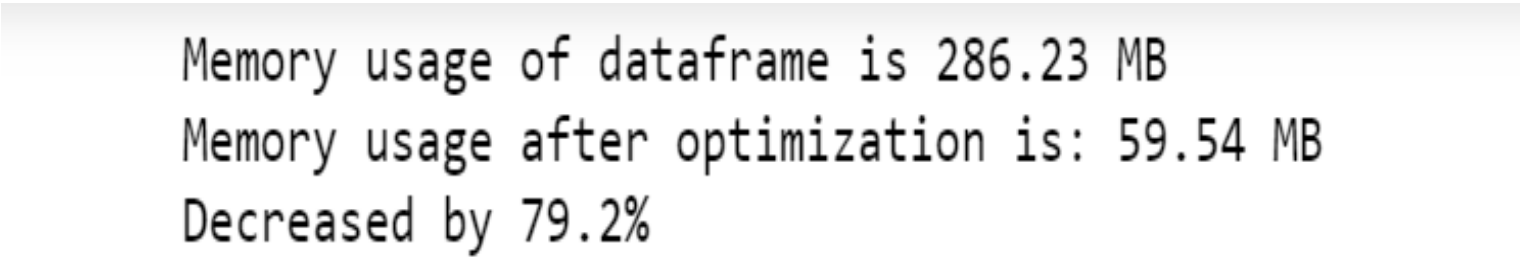


Fig. 4.11 Memory usage Function

4.11.2 Finding Missing Values: The real-world data often has a lot of missing values. The cause of missing values can be data corruption or failure to record data. The handling of missing data is very important during the preprocessing of the dataset as many machine learning algorithms do not support missing values.

4.7.2.1 Categorical Variable's:

a categorical variable (also called qualitative variable) is a variable that can take on one of a limited, and usually fixed, number of possible values, assigning each individual or other unit of observation to a particular group or nominal category based on some qualitative property.

```
Missing Values of Categorical variables are as :
```

:

	"Total	Percentage
FONDKAPREMONT_MODE	210295	68.386172
WALLSMATERIAL_MODE	156341	50.840783
HOUSETYPE_MODE	154297	50.176091
EMERGENCYSTATE_MODE	145755	47.398304
OCCUPATION_TYPE	96391	31.345545
NAME_TYPE_SUITE	1292	0.420148

Table 4.1 Categorical Missing Values

4.7.2.2 Integer Variable's:

Integer variable are **variables that must take an integer value (0, 1, 2, ...)**. A special kind of integer variables is binary variables. Binary variables can only take the value 0 or 1. They are integer variables with a maximum of 1 on them (and do not forget there is always an implicit minimum of 0 on each variable).

```
Missing values of integer variables are as :
```

	"Total	Percentage

Here, in Integer Type Variables we see there is no missing values are present.

Table 4.2 Integer Missing Values

4.7.2.3 Floating Point Variable's:

A floating-point type variable is **a variable that can hold a real number**, such as 4320.0, -3.33, or 0.01226. The floating part of the name floating point refers to the fact that the decimal point can "float;" that is, it can support a variable number of digits before and after the decimal point.

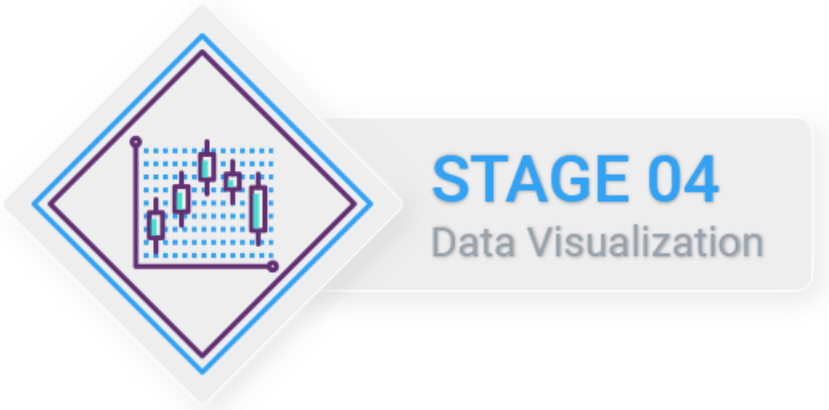
Nkbm

Missing values of Floating Type Variables are as :

	"Total	Percentage
COMMONAREA_MODE	214865	69.872297
COMMONAREA_MEDI	214865	69.872297
COMMONAREA_AVG	214865	69.872297
NONLIVINGAPARTMENTS_MODE	213514	69.432963
NONLIVINGAPARTMENTS_MEDI	213514	69.432963
NONLIVINGAPARTMENTS_AVG	213514	69.432963
LIVINGAPARTMENTS_MODE	210199	68.354953
LIVINGAPARTMENTS_MEDI	210199	68.354953
LIVINGAPARTMENTS_AVG	210199	68.354953
FLOORSMIN_MEDI	208642	67.848630
FLOORSMIN_AVG	208642	67.848630
FLOORSMIN_MODE	208642	67.848630
YEARS_BUILD_AVG	204488	66.497784
YEARS_BUILD_MEDI	204488	66.497784
YEARS_BUILD_MODE	204488	66.497784
OWN_CAR_AGE	202929	65.990810
LANDAREA_AVG	182590	59.376738
LANDAREA_MODE	182590	59.376738
LANDAREA_MEDI	182590	59.376738
BASEMENTAREA_MEDI	179943	58.515956
BASEMENTAREA_AVG	179943	58.515956
BASEMENTAREA_MODE	179943	58.515956
EXT_SOURCE_1	173378	56.381073
NONLIVINGAREA_MEDI	169682	55.179164
NONLIVINGAREA_MODE	169682	55.179164
NONLIVINGAREA_AVG	169682	55.179164
ELEVATORS_AVG	163891	53.295980
ELEVATORS_MODE	163891	53.295980
APARTMENTS_MEDI	156061	50.749729
ENTRANCES_MODE	154828	50.348768
ENTRANCES_AVG	154828	50.348768
ENTRANCES_MEDI	154828	50.348768
LIVINGAREA_MODE	154350	50.193326
LIVINGAREA_MEDI	154350	50.193326
LIVINGAREA_AVG	154350	50.193326
FLOORSMAX_AVG	153020	49.760822
FLOORSMAX_MEDI	153020	49.760822
FLOORSMAX_MODE	153020	49.760822
YEARS_BEGINEXPLUATATION_MODE	150007	48.781019
YEARS_BEGINEXPLUATATION_AVG	150007	48.781019
YEARS_BEGINEXPLUATATION_MEDI	150007	48.781019
TOTALAREA_MODE	148431	48.268517
EXT_SOURCE_3	60965	19.825307
AMT_REQ_CREDIT_BUREAU_HOUR	41519	13.501631
AMT_REQ_CREDIT_BUREAU_DAY	41519	13.501631
AMT_REQ_CREDIT_BUREAU_WEEK	41519	13.501631
AMT_REQ_CREDIT_BUREAU_MON	41519	13.501631
AMT_REQ_CREDIT_BUREAU_QRT	41519	13.501631
AMT_REQ_CREDIT_BUREAU_YEAR	41519	13.501631
OBS_30_CNT_SOCIAL_CIRCLE	1021	0.332021
DEF_30_CNT_SOCIAL_CIRCLE	1021	0.332021
OBS_60_CNT_SOCIAL_CIRCLE	1021	0.332021
DEF_60_CNT_SOCIAL_CIRCLE	1021	0.332021
EXT_SOURCE_2	660	0.214626
AMT_GOODS_PRICE	278	0.090403
AMT_ANNUITY	12	0.003902
CNT_FAM_MEMBERS	2	0.000650
DAYS_LAST_PHONE_CHANGE	1	0.000325

Table 4.3 Floating Missing Values

4.8 Data Visualization



Data Visualization is used to perform Exploratory Data Analysis (EDA). When one is dealing with large volumes of data, building graphs is the best way to explore and communicate findings. Visualization is an incredibly helpful tool to identify patterns and trends in data, which leads to clearer understanding and reveals important insights. Data Visualization also helps for faster decision making through the graphical illustration.

4.8.1 Exploratory Data Analysis:

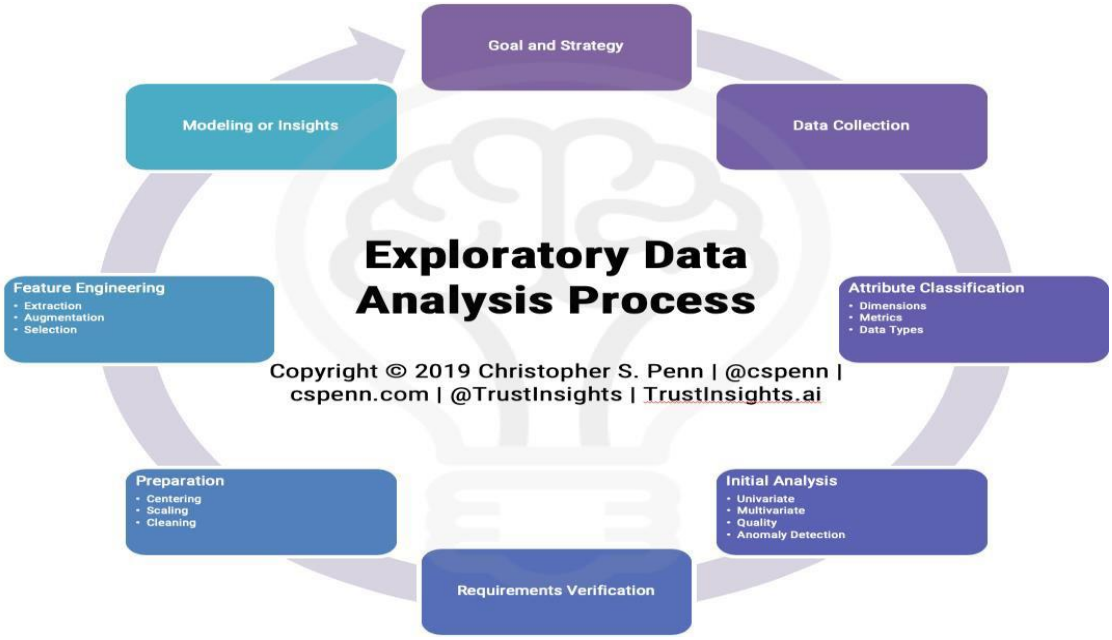


Fig. 4.12 Exploratory Data Analysis Process [14]

The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

Data scientists can use exploratory analysis to ensure the results they produce are valid and applicable to any desired business outcomes and goals. EDA also helps stakeholders by confirming they are asking the right questions. EDA can help answer questions about standard deviations, categorical variables, and confidence intervals. Once EDA is complete and insights are drawn, its features can then be used for more sophisticated data analysis or modeling, including machine learning.

Exploratory data analysis tools:

Specific statistical functions and techniques you can perform with EDA tools include:

- Clustering and dimension reduction techniques, which help create graphical displays of high-dimensional data containing many variables.
- Univariate visualization of each field in the raw dataset, with summary statistics.
- Bivariate visualizations and summary statistics that allow you to assess the relationship between each variable in the dataset and the target variable you are looking at.
- Multivariate visualizations, for mapping and understanding interactions between different fields in the data.
- K-means Clustering is a clustering method in unsupervised learning where data points are assigned into K groups, i.e. the number of clusters, based on the distance from each group's centroid. The data points closest to a particular centroid will be clustered under the same category. K-means Clustering is commonly used in market segmentation, pattern recognition, and image compression.
- Predictive models, such as linear regression, use statistics and data to predict outcomes.

Types of exploratory data analysis:

There are four primary types of EDA:

- **Univariate non-graphical.** This is simplest form of data analysis, where the data being analyzed consists of just one variable. Since it is a single variable, it doesn't deal with causes or relationships. The main purpose of univariate analysis is to describe the data and find patterns that exist within it.
- **Univariate graphical.** Non-graphical methods do not provide a full picture of the data. Graphical methods are therefore required. Common types of univariate graphics include:
    - Stem-and-leaf plots, which show all data values and the shape of the distribution.
    - Histograms, a bar plot in which each bar represents the frequency (count) or proportion (count/total count) of cases for a range of values.
    - Box plots, which graphically depict the five-number summary of minimum, first quartile, median, third quartile, and maximum.
- **Multivariate nongraphical:** Multivariate data arises from more than one variable. Multivariate non-graphical EDA techniques generally show the relationship between two or more variables of the data through cross-tabulation or statistics.
- **Multivariate graphical:** Multivariate data uses graphics to display relationships between two or more sets of data. The most used graphic is a grouped bar plot or bar chart with each group representing one level of one of the variables and each bar within a group representing the levels of the other variable.

4.8.2 Relationship Analysis: There are two kinds of relationship of analysis of correlation: Positive correlation A positive correlation is a relationship between 2 variables which the increase of one variable causes an increase for another variable

Other common types of multivariate graphics include:

- Scatter plot, which is used to plot data points on a horizontal and a vertical axis to show how much one variable is affected by another.
- Multivariate chart, which is a graphical representation of the relationships between factors and a response.

Here are some common ways of visualization:

- Area Chart
- Bar Chart
- Box-and-whisker Plots
- Bubble Cloud
- Dot Distribution Map
- Heat Map
- Histogram

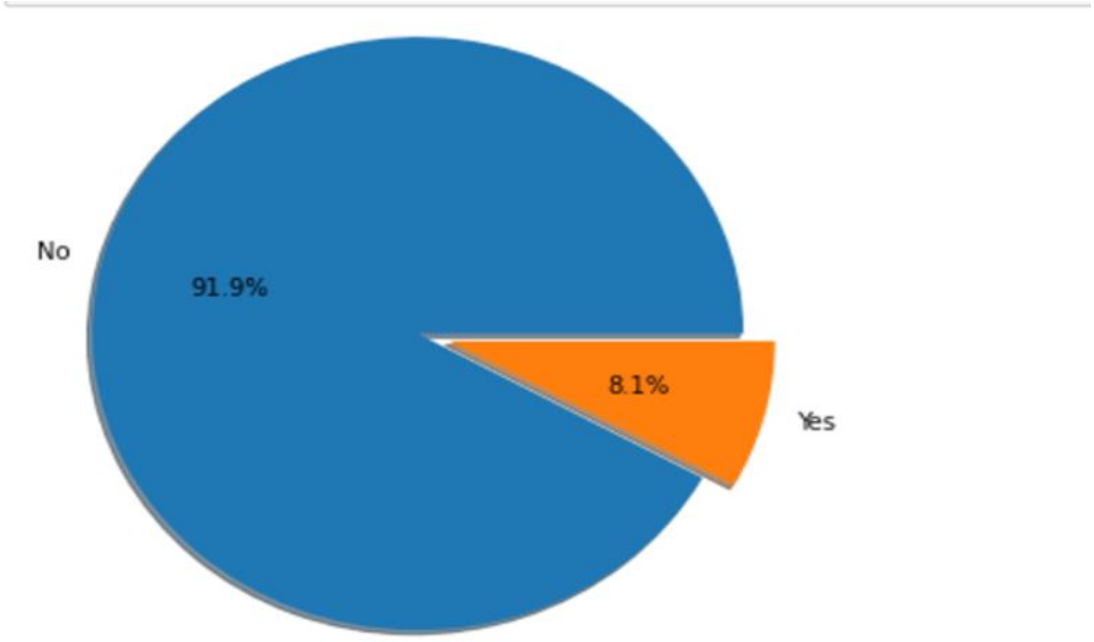
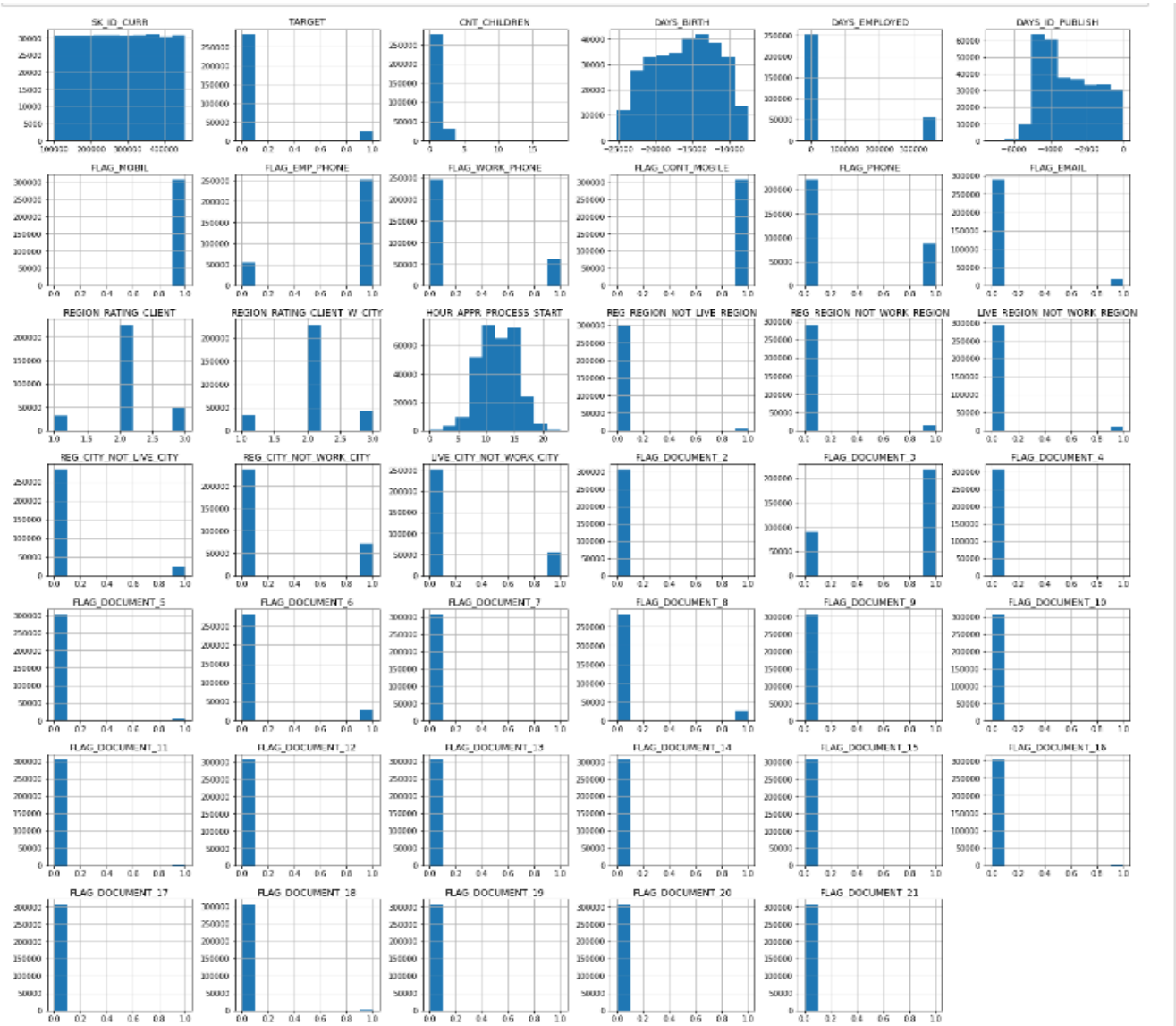


Fig. 4.13 Pie Chart of Target variable

4.8.3 Histogram's :

A histogram is **a display of statistical information that uses rectangles to show the frequency of data items in successive numerical intervals of equal size**. In the most common form of histogram, the independent variable is plotted along the horizontal axis and the dependent variable is plotted along the vertical axis.

Histogram visualization for Integer Type Variable's :



Histogram visualization for Floating Type Variable's :

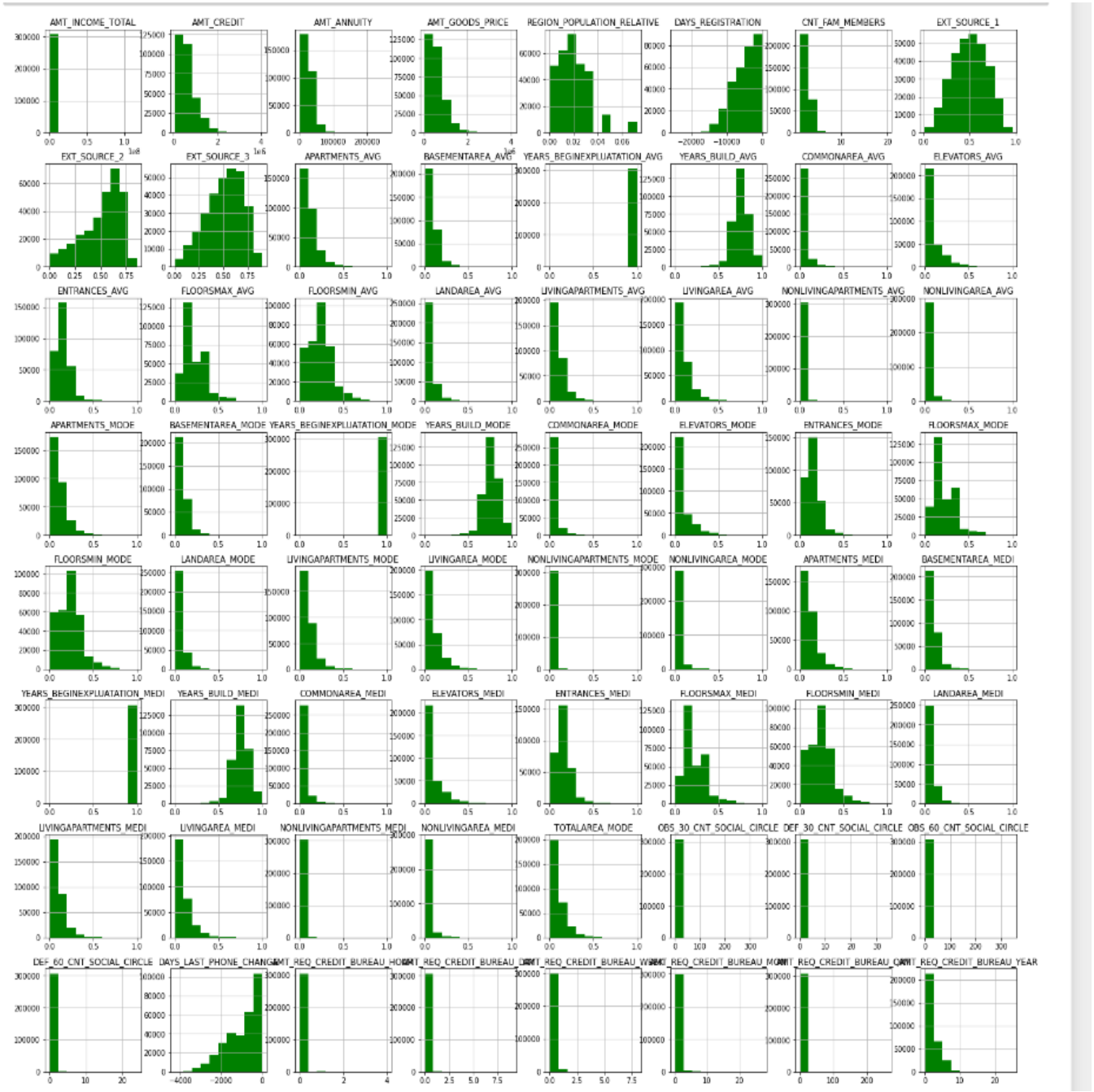
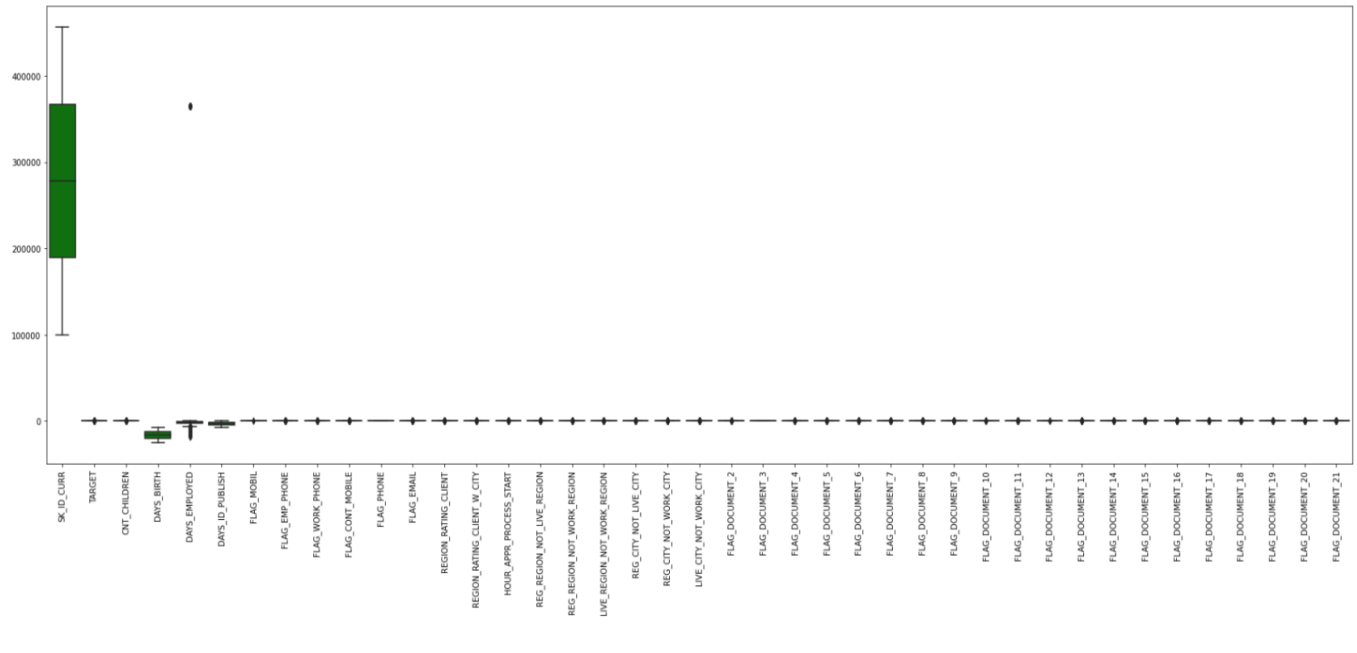


Fig. 4.14 Histogram's Visualization

4.8.4 Outliers: Outliers are an important part of a dataset. They can hold useful information about your data. Outliers can give helpful insights into the data you are studying, and they can have an effect

on statistical results. This can potentially help you discover inconsistencies and detect any errors in your statistical processes.

So, knowing how to find outliers in a dataset will help you better understand your data.



❖ Outlier Detection for Floating Point Variables:

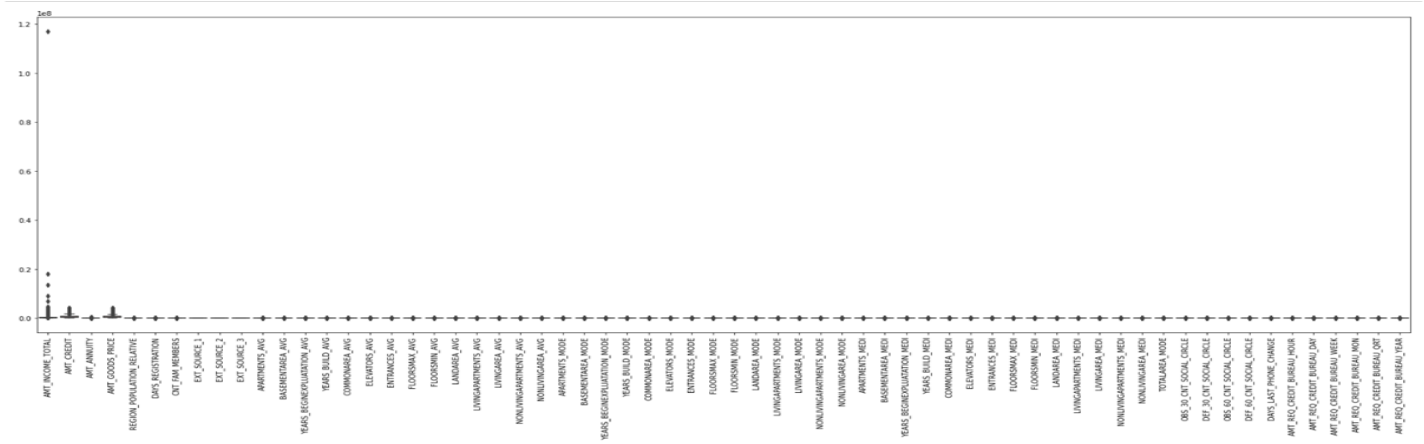


Fig. 4.15 Outlier ‘s Visualization

4.8.5 Correlation Matrix: Correlation Matrix is a statistical method of showing the relationship between two or more variables and the interrelation in their movements etc. In short, it helps in defining the relationship and dependence among the variables. It is a very commonly used mechanism and finds its application in the field of Investment Management, Risk Management, Statistics as well as Economics to name a few.

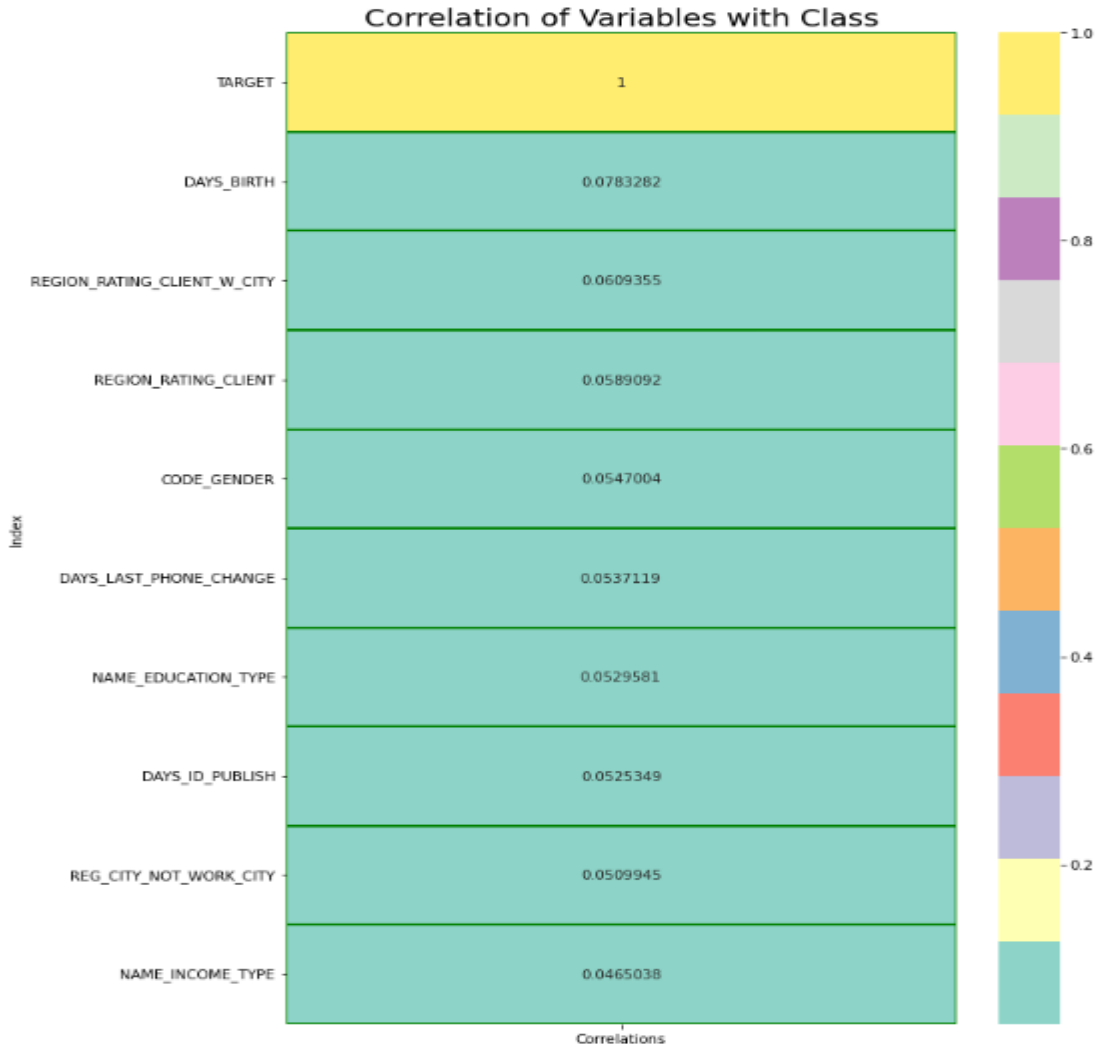


Fig. 4.16 Correlation Matrix

4.9 ML Modeling

Finally, this is where 'the magic happens.' Machine Learning is finding patterns in data, and one can perform either supervised or unsupervised learning. ML tasks include regression, classification, forecasting, and clustering.

In this stage of the process, one must apply mathematical, computer science, and business knowledge to train a Machine Learning algorithm that will make predictions based on the provided data. It is a crucial step that will determine the quality and accuracy of future predictions in new situations. Additionally, ML algorithms help to identify key features with high predictive value.

4.10 Feature Engineering

Feature Engineering is a process to achieve a set of features by performing mathematical, statistical, and heuristic procedures. It is a collection of methods for identifying an optimal set of inputs to the Machine Learning algorithm. Feature Engineering is extremely important because well-engineered features make learning possible with simple models.

Following are the characteristics of good features:

- Represents data in an unambiguous way
- Ability to captures linear and non-linear relationships among data points
- Capable of capturing the precise meaning of input data
- Capturing contextual details

4.10.1 Feature Selection Techniques:

Why is Feature Selection important?

In the machine learning process, feature selection is used to make the process more accurate. It also increases the prediction power of the algorithms by selecting the most critical variables and eliminating the redundant and irrelevant ones.

Information gain:

Information gain can also be used for feature selection, by **evaluating the gain of each variable in the context of the target variable**. In this slightly different usage, the calculation is referred to as mutual information between the two random variables.

Top Best Features after Feature Selection using Information Gain

	Feature	Score
1	FLAG_MOBIL	0.064405
0	FLAG_CONT_MOBILE	0.064195
2	FLAG_EMP_PHONE	0.044550
3	NAME_TYPE_SUITE	0.043994
4	NAME_EDUCATION_TYPE	0.042289
5	NAME_HOUSING_TYPE	0.040196
6	REGION_RATING_CLIENT_W_CITY	0.039676
7	REGION_RATING_CLIENT	0.038408
8	FLAG_DOCUMENT_3	0.033042
9	FLAG_OWN_REALTY	0.030635
10	EMERGENCYSTATE_MODE	0.029579
11	NAME_INCOME_TYPE	0.023722

Table 4.4 Top Features of Dataset

4.11 Implementation of Algorithm's:

4.11.1 SVM (SUPPORT VECTOR MACHINE)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

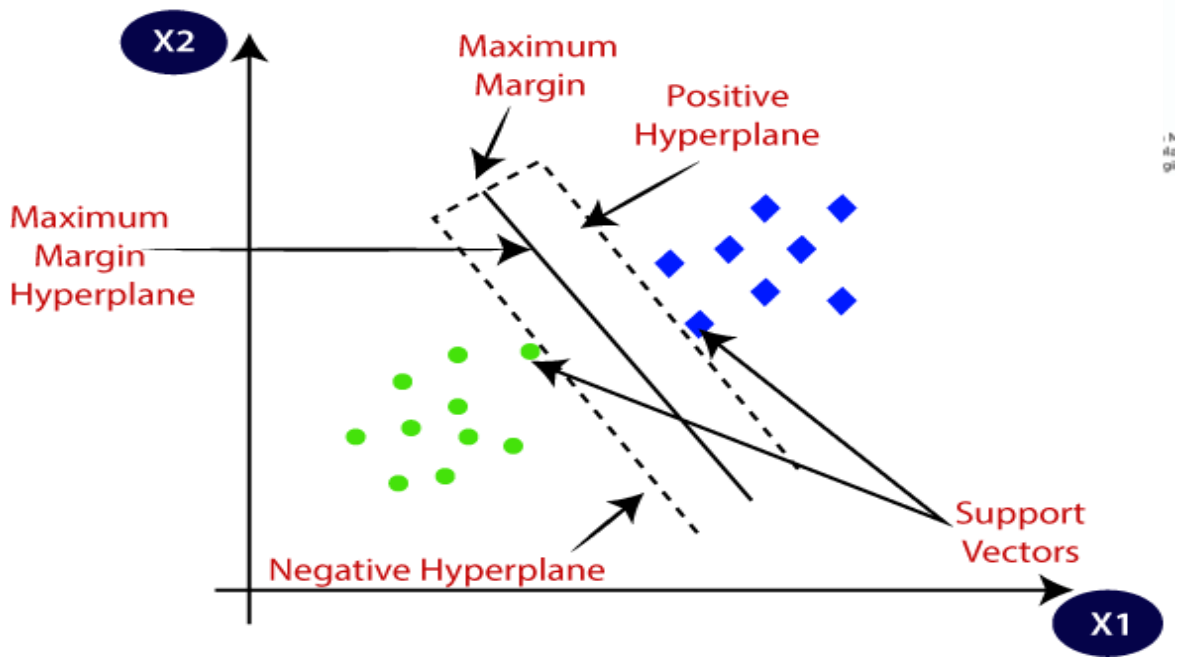


Fig. 4.17 SVM Algorithm [2]

Types of SVM:

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data.

4.11.2 LOGISTIC REGRESSION ALGORITHM:

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.
- Logistic Regression is much like the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it can provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

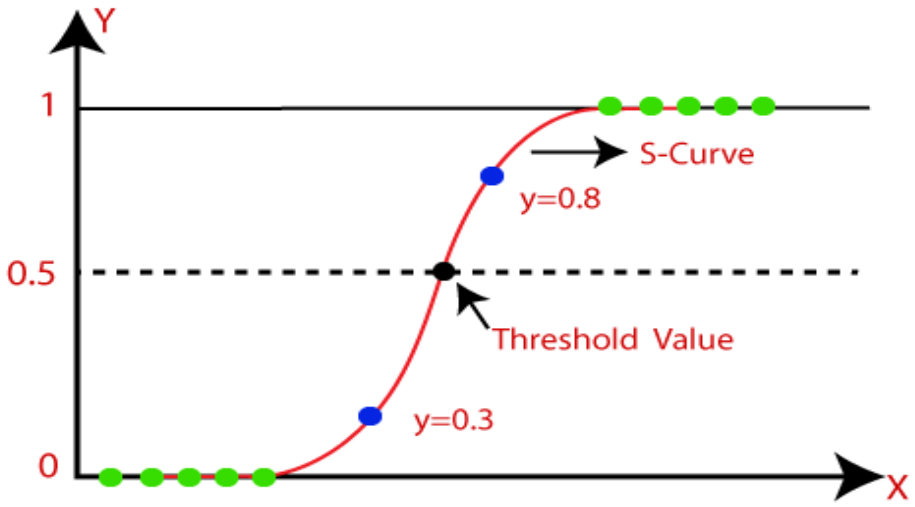


Fig. 4.18 Logistic Regression Graph[11]

Logistic Function (Sigmoid Function):

- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Assumptions for Logistic Regression:

- The dependent variable must be categorical in nature.
- The independent variable should not have multi-collinearity.

Logistic Regression Equation:

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

- We know the equation of the straight line can be written as:

$$y= b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

- In Logistic Regression y can be between 0 and 1 only, so for this let us divide the above equation by (1-y):

$$\frac{y}{1-y}$$; 0 for y= 0, and infinity for y=1

- But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

4.11.3 K-Nearest Neighbor(KNN) Algorithm:

- K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most like the available categories.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- **Example:** Suppose, we have an image of a creature that looks like cat and dog, but we want to know either it is a cat or dog. So, for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs' images and based on the most similar features it will put it in either cat or dog category.

Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

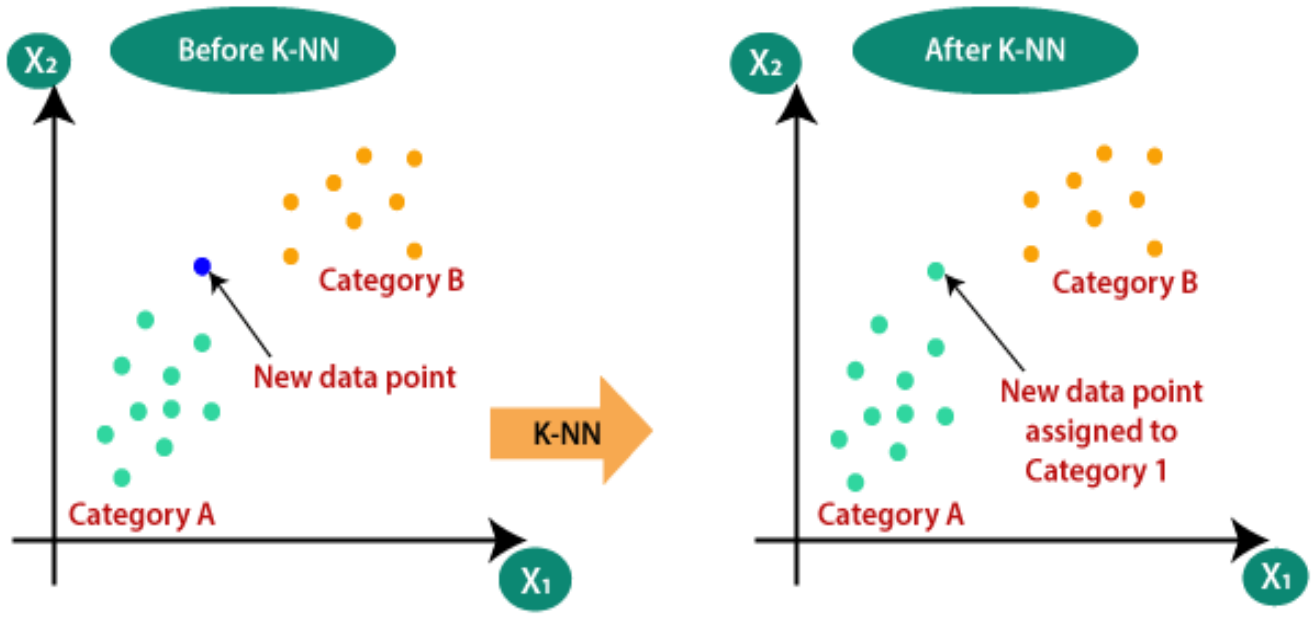


Fig. 4.19 K-NN Algorithm[12]

How does K-NN work?

The K-NN working can be explained based on the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

4.11.4 Decision Tree Algorithm:

- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules **and** each leaf node represents the outcome.
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
- Below diagram explains the general structure of a decision tree:

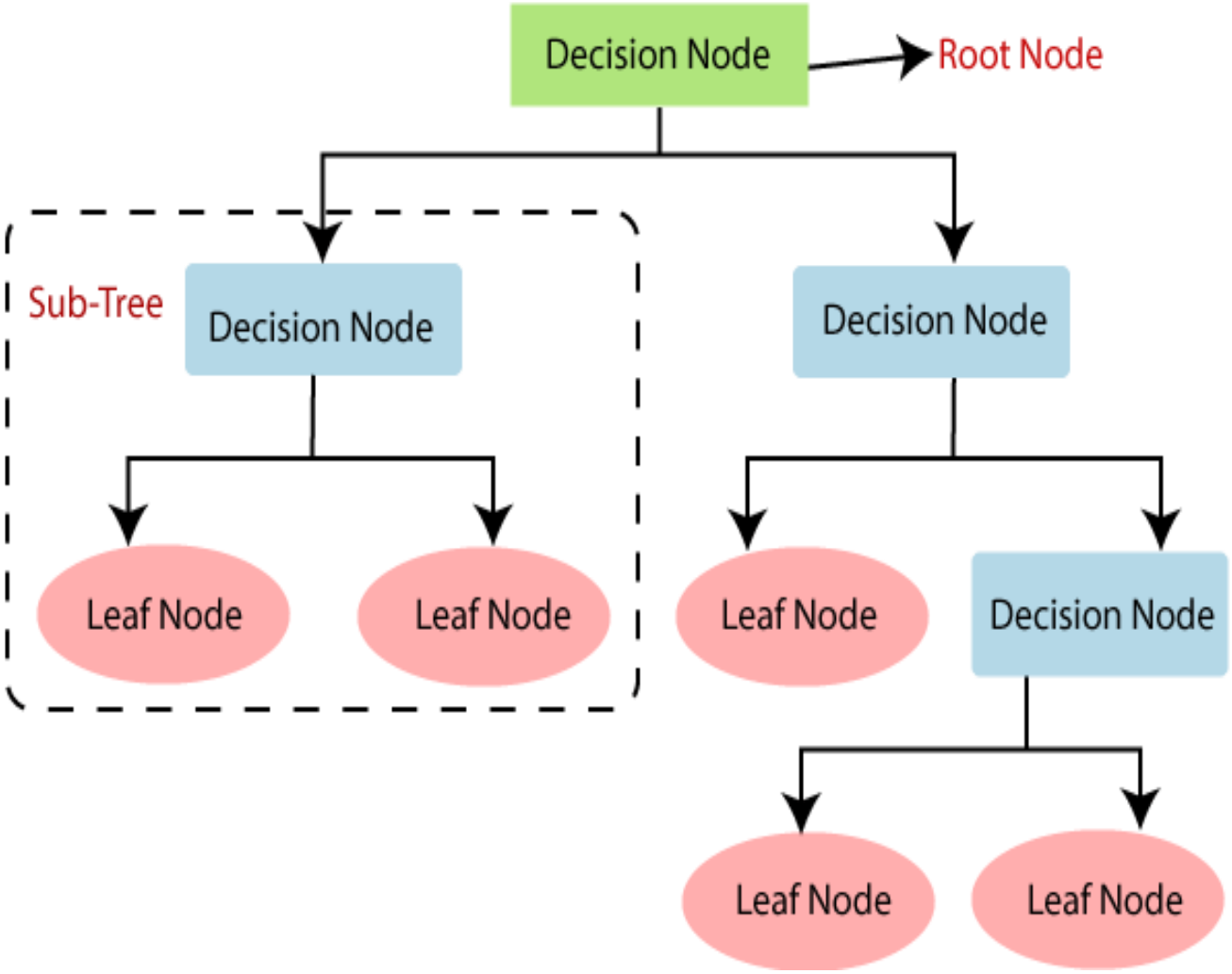


Fig. 4.20 Decision Tree Classifier [14]

4.11.5 Random Forest Algorithm:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, **"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random Forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on most results, the Random Forest classifier predicts the final decision. Consider the below image:

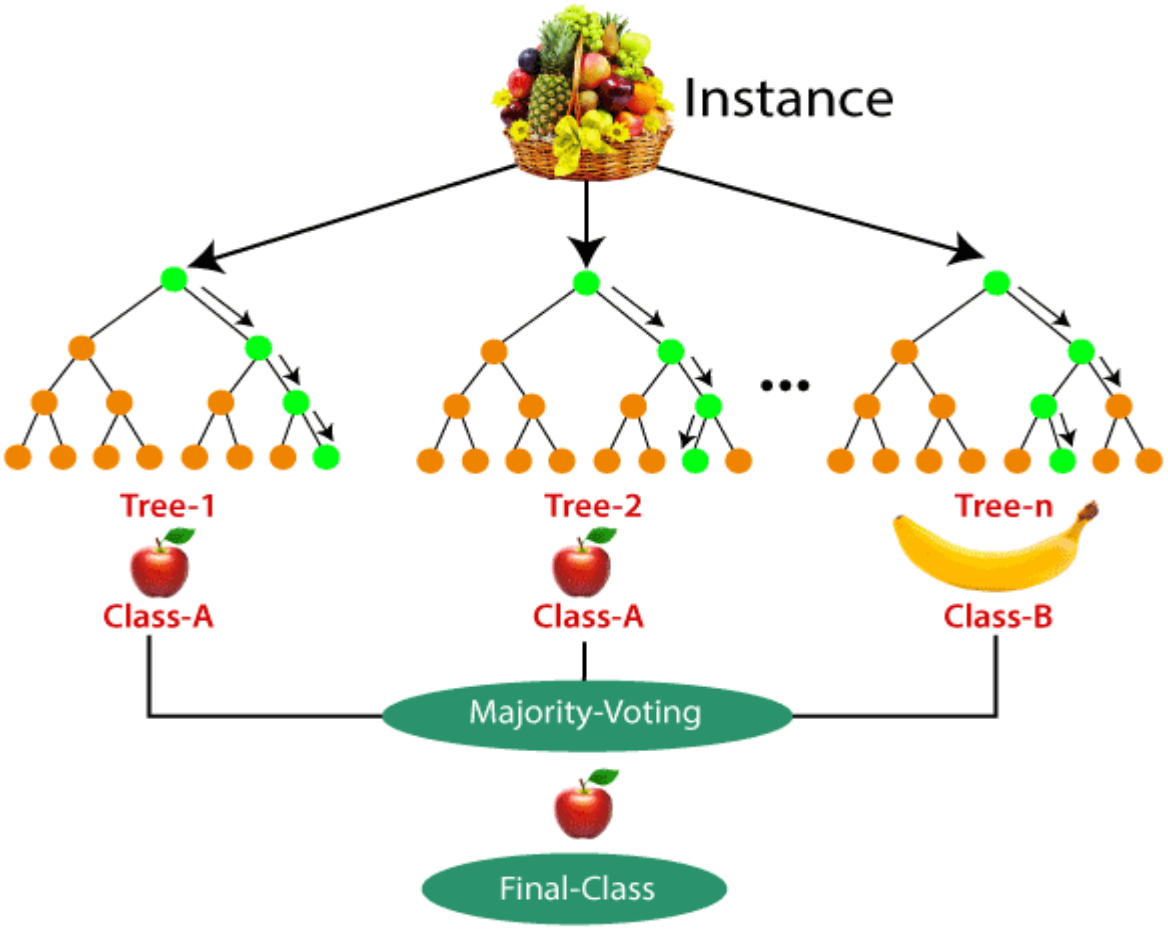


Fig. 4.21 Random Forest Classifier[15]

4.11.6 Naïve Bayes Algorithm:

Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e., every pair of features being classified is independent of each other.

- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts based on the probability of an object.
- Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

Bayes' Theorem:

- Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- The formula for Bayes' theorem is given as:

$$P(A|B)= \frac{P(B|A)P(A)}{P(B)}$$

Were,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

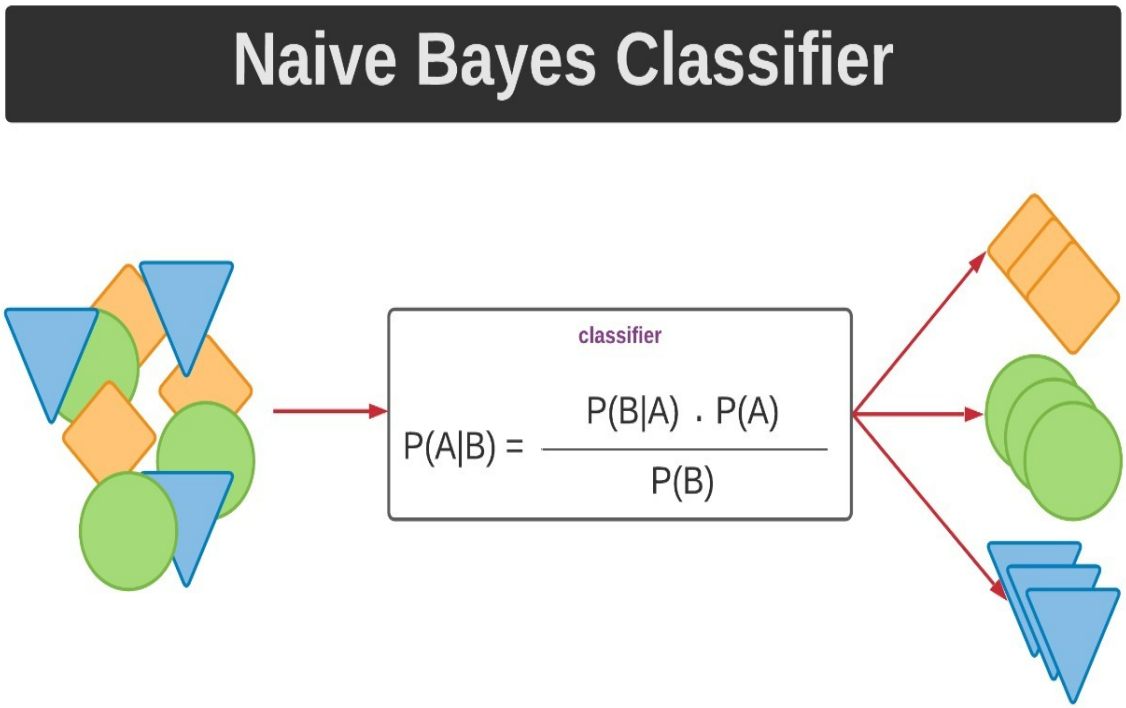
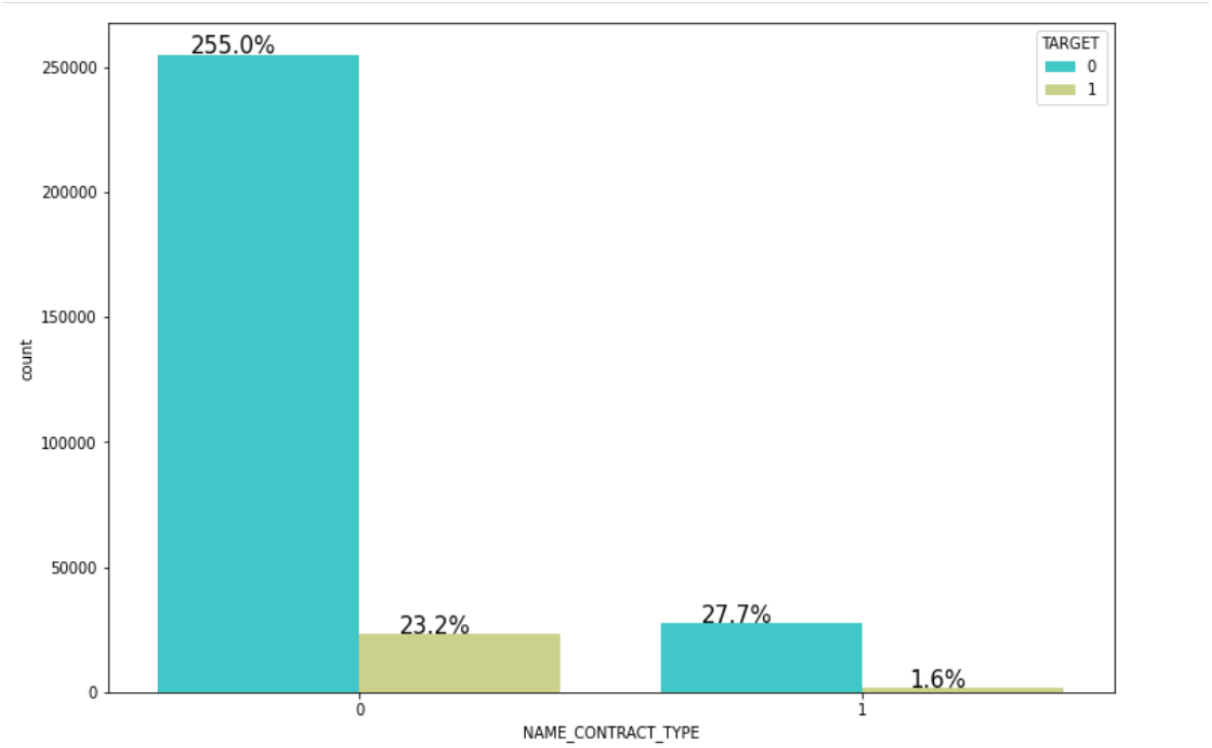


Fig.4.22 Naïve Bayes Classifier[10]

CHAPTER – 5 RESULT

5.1 Bar chart Screenshots :



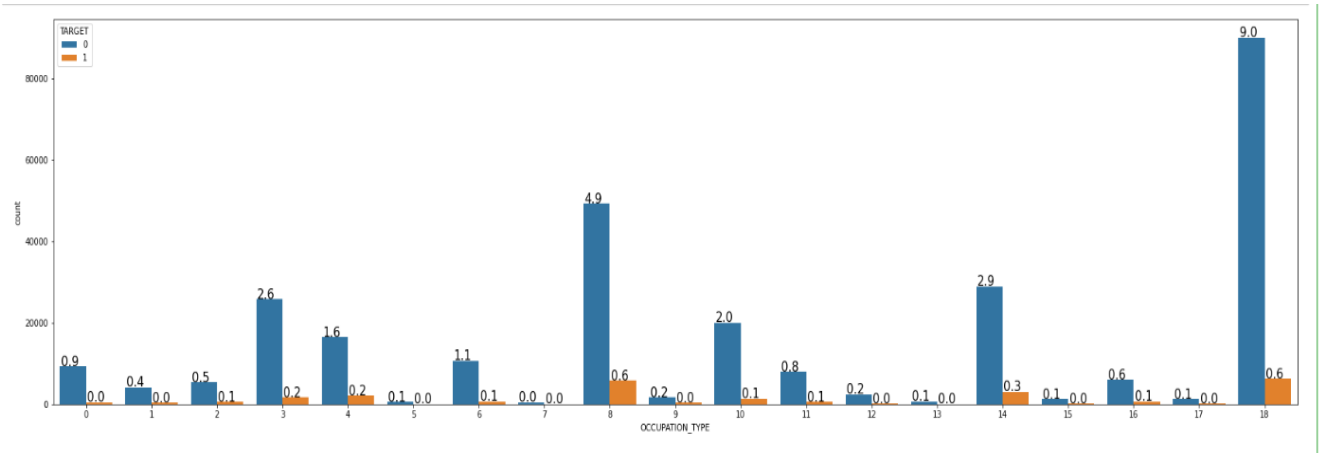


Fig. 5.1 Bar Chart Visualization

5.2 Accuracy of all the algorithm's:

Logistic Regression Classifier:

In [104]:
```
#PREDICTING THE TEST RESULTS AND CALCULATING THE ACCURACY
y_pred = logreg.predict(X_test)
LR_score=logreg.score(X_test, Y_test)*100
print('Accuracy of logistic regression classifier on test set: {:.2f}%'.format(LR_score))
```

```
Accuracy of logistic regression classifier on test set: 91.93%
```

K- Nearest Neighbor Algorithm:

In [60]:
```
ac_knn = accuracy_score(Y_test,y_pred)*100
print('Accuracy of KNN algorithm on test set: {:.2f}%'.format(ac_knn))
```

```
Accuracy of KNN algorithm on test set: 91.92%
```

Decision Tree Classifier:

In [65]:
```
ac_dt=accuracy_score(Y_test,y_pred)*100
```

In [66]:
```
print('Accuracy of Decision Tree algorithm on test set: {:.2f}%'.format(ac_dt))
```

```
Accuracy of Decision Tree algorithm on test set: 91.93%
```

Random Forest Classifier:

In [71]:
```
ac_rf=accuracy_score(Y_test,y_pred)*100
```

In [72]:
```
print('Accuracy of Random Forest algorithm on test set: {:.2f}%'.format(ac_rf))
```

```
Accuracy of Random Forest algorithm on test set: 91.93%
```

Naïve Bayes Classifier:

In [76]:
```
ac_nb = accuracy_score(Y_test,y_pred)*100
```

In [95]:
```
print('Accuracy of NAIVE BAYES algorithm on test set: {:.2f}%'.format(ac_nb))
```

```
Accuracy of NAIVE BAYES algorithm on test set: 89.83%
```

Support Vector Machine SVM) Algorithm:

```
In [100]: classifier = accuracy_score(Y_test,y_pred)*100

In [101]: print('Accuracy of SVM algorithm on test set: {:.2f}%'.format(classifier))
          Accuracy of SVM algorithm on test set: 91.93%
```

5.3 Comparison of the Model:

Hence, on Comparing all the algorithm Except Naïve Bayes all give 91.93% accuracy.

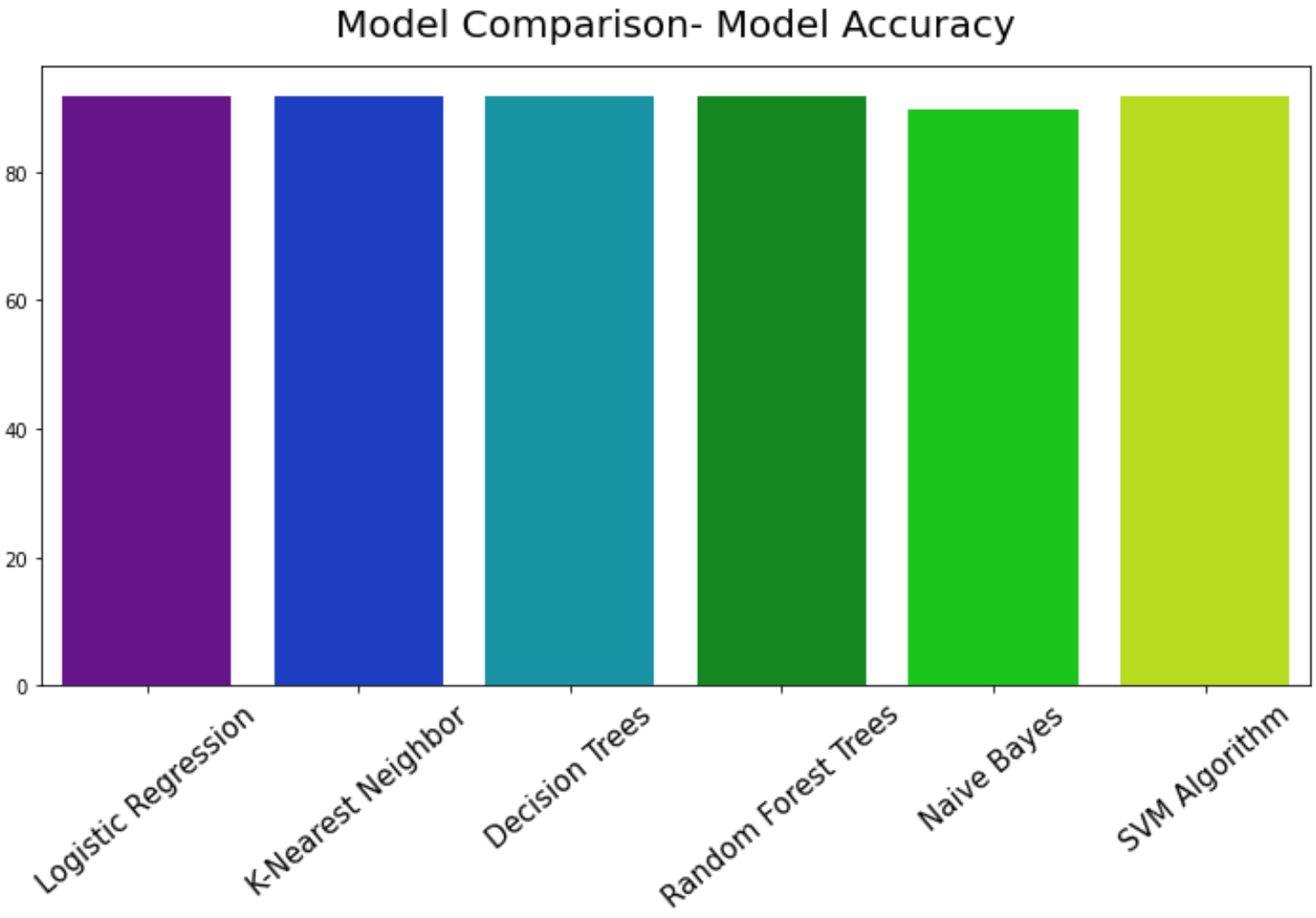


Fig. 5.2 Comparison of Algorithm's

CHAPTER - 6 CONCLUSION & FUTURE SCOPE

- **Conclusion**

At the end, it is concluded that after applying different – different Machine learning Algorithm and Feature Selection technique like Information Gain Top Features are then taken and all algorithm's give accuracy as, Logistic Regression give 91.93% accuracy, K-Nearest neighbor algorithm give 91.92% accuracy, Decision Tree classifier give 91.93% accuracy , Random Forest classifier give 91.93% accuracy , Naïve Bayes classifier give 89.83% accuracy and Support vector machine give 91.93% accuracy. So, it is good accuracy as compared to previous done work.

- **Future scope**

- Can be highly developed and reduce more fraud activities.
- Highly complexity can increase the detection of the irregular activities.
- Making model which is give up to 99 to 100% accuracy.

References:

[1] Meena, I.S.L. Sarani, S.V.S.S. Lakshmi," Web Service mining and its techniques in Web Mining" Volume 2, Issue 1, Page No.385-389.

[2] https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

[3] https://pynative.com/python-data-types/

[4] F. N. Roguelike, "Data Mining Application in Credit Card Fraud Detection System", Journal of Engineering Science and Technology, vol. 6, no. 3, pp. 311-322, 2020

[5] Aihua, S. et al. 2007. Application of Classification Models on Credit Card Fraud Detection. IEEE.

Al Daoud, E. J. I. J. o. C. and Engineering, I. 2019. Comparison between XGBoost, LightGBM and CatBoost Using a Home Credit Dataset. 13(1), pp. 6-10.

[6] Alghamdi, M. et al. 2017. Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: The Henry Ford Exercise Testing (FIT) project. 12(7), p. e0179805.

[7] Awoyemi, J. O. et al. eds. 2017. Credit card fraud detection using machine learning techniques: A comparative analysis. 2017 International Conference on Computing Networking and Informatics (ICCNI). IEEE.

[8] Bahnsen, A. C. et al. eds. 2014. Improving credit card fraud detection with calibrated probabilities. Proceedings of the 2014 SIAM international conference on data mining. SIAM.

[9] Barandela, R. et al. eds. 2004. The imbalanced training sample problem: Under or over sampling? Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR). Springer.

[10] Bhatla, T. et al. 2003. Understanding credit card frauds. Cards Business Review# 2003–1. Bhattacharyya, S. et al. 2011. Data mining for credit card fraud: A comparative study. 50(3), pp. 602- 613. Contributors, W. W. 2020. Bibliographic details for "Credit card fraud". Available :https://en.wikipedia.org/w/index.php?title=Credit_card_fraud&oldid=970300096 [Accessed: 10 September 2020].

[11] Dal Pozzolo, A. et al. eds. 2015. Calibrating probability with undersampling for unbalanced classification. 2015 IEEE Symposium Series on Computational Intelligence. IEEE.

Dornadula, V. N. and Geetha, S. J. P. C. S. 2019. Credit Card Fraud Detection using Machine Learning Algorithms. 165, pp. 631-641.

[12] Dorogush, A. V. et al. 2018. CatBoost: gradient boosting with categorical features support.

Duman, E. et al. eds. 2013. A novel and successful credit card fraud detection system implemented in a turkish bank. 2013 IEEE 13th International Conference on Data Mining Workshops. IEEE.

[13] Foulsham, M. 2019. Living with the new general data protection regulation (GDPR).Financial Compliance. Springer, pp. 113-136. FTC.gov, C. 2012. Protecting Against Credit Card Fraud (2012). Available at: https://www.consumer.ftc.gov/articles/0216-protecting-against-credit-card-fraud [Accessed: 4 September 2020].

[14] Rushin, G. et al. 2017. Horse race analysis in credit card fraud-deep learning, logistic regression, and Gradient Boosted Tree. IEEE.

[15] Sahin, Y. and Duman, E. 2011. Detecting credit card fraud by ANN and logistic regression. IEEE. Seeja, K. and Zareapoor, M. J. T. S. W. J. 2014.

[16] FraudMiner: A novel credit card fraud detection model based on frequent itemset mining. 2014, Shen, A. et al. eds. 2007. Application of classification models on credit card fraud detection. 2007 International conference on service systems and service management. IEEE. Singh, G. et al. 2012. A machine learning approach for detection of fraud based on svm. 1(3), pp. 192- 196.

[17] Stoloff, S. et al. eds. 1997. Credit card fraud detection using meta-learning: Issues and initial results.