

## CUSTOMER SEGMENTATION

A Project report submitted in partial fulfilment of the requirements for the award of the Degree of

**BACHELOR OF TECHNOLOGY**  
**IN**  
**COMPUTER SCIENCE AND SYSTEMS ENGINEERING**

Submitted by

<b>NUTALAPATI SAI LAVANYA</b>	<b>317114110038</b>
<b>PAPPALA SHAMILI KUMARI</b>	<b>317114110039</b>
<b>PARVATHAM HARI CHANDANA</b>	<b>317114110040</b>
<b>POLAMARASETTY LAKSHMI NITHISHA</b>	<b>317114110041</b>

Under the esteemed guidance of

**Ms D. LALITHA KUMARI**

Assistant Professor

Department of Computer Science and Systems Engineering



**DEPARTMENT OF COMPUTER SCIENCE AND SYSTEMS  
ENGINEERING**

**ANDHRA UNIVERSITY COLLEGE OF ENGINEERING FOR WOMEN**

# Visakhapatnam

**July, 2021**

**CERTIFICATE**  
**ANDHRA UNIVERSITY COLLEGE OF ENGINEERING FOR WOMEN**  
**VISAKHAPATNAM**



This is to certify that the main project entitled “**CUSTOMER SEGMENTATION**” is the bonafide work done by **NUTALAPATI SAI LAVANYA (317114110038)**, **PAPPALA SHAMILI KUMARI (317114110039)**, **PARVATHAM HARI CHANDANA (317114110040)**, **POLAMARASETTY LAKSHMI NITHISHA (317114110041)** submitted in the partial fulfilment of the requirement for the award of the Degree of Bachelor of Technology in Computer Science and Systems Engineering during April 2021- July 2021.

**Project Guide**

**Ms. D. LALITHA KUMARI**

Assistant Professor

Department of CS&SE

**Head of the Department**

**Prof. B. PRAJNA**

Head of the Department

Department of CS&SE

## ACKNOWLEDGEMENT

We take this opportunity to express deep gratitude to the people who have been instrumental in the successful completion of the project. We would like to thank our project guide **Ms. D. Lalitha Kumari**, Assistant Professor, Department of Computer Science and Systems Engineering, for the supervision and support, which was greatly helpful in the progression and smoothness of the entire project work.

We would like to thank **Prof. B. Prajna**, Head of the Department, Computer Science and Systems Engineering. We would like to show our gratitude to our beloved Principal, **Prof. S. K. Bhatti** for her encouragement in the aspect of our course, Andhra University College of Engineering for Women, who gave us official support for the progress of our project.

We would also like to extend our gratitude to Lab Technicians and our sincere gratitude towards all the faculty members and Non-Teaching Staff in the Department of Computer Science and Systems Engineering for supporting us. We would also like to thank our parents and friends, who have willingly helped us out with their abilities in completing the project work.

With gratitude,

<b>NUTALAPATI SAI LAVANYA</b>	<b>(317114110038)</b>
<b>PAPPALA SHAMILI KUMARI</b>	<b>(317114110039)</b>
<b>PARVATHAM HARI CHANDANA</b>	<b>(317114110040)</b>
<b>POLAMARASETTY LAKSHMI NITHISHA</b>	<b>(317114110041)</b>

## DECLARATION

We hereby declare that the project work entitled “**Customer Segmentation**”, is a bonafide work done by us, under the esteemed guidance of Ms. D. Lalitha Kumari, Assistant Professor, Department of CS&SE, **Andhra University College of Engineering for Women**. This project report is being submitted in the partial fulfilment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Systems Engineering during the academic year 2020-2021. This project possesses originality as it is not extracted from any source and it has not been submitted to any other institution or university.

NAME	REGD. NO	SIGNATURE
<b>NUTALAPATI SAI LAVANYA</b>	<b>317114110038</b>	
<b>PAPPALA SHAMILI KUMARI</b>	<b>317114110039</b>	
<b>PARVATHAM HARI CHANDANA</b>	<b>317114110040</b>	
<b>POLAMARASETTY LAKSHMI NITHISHA</b>	<b>317114110041</b>	

Visakhapatnam

Date:

# INDEX

<b>Contents</b>	<b>Page No.</b>
ABSTRACT	7
LIST OF FIGURES	8
LIST OF TABLES	9
<b>1. INTRODUCTION</b>	<b>10</b>
1.1 Project Overview	
1.2 Project Deliverables	
1.3 Project Scope	
<b>2. REVIEW OF LITERATURE</b>	<b>13</b>
<b>3. PROBLEM ANALYSIS</b>	<b>14</b>
3.1 Existing System	
3.2 Proposed System	
3.3 Limitations	
<b>4. SYSTEM ANALYSIS</b>	<b>16</b>
4.1 System Requirement Specification	
4.4.1 Functional Requirements	
4.4.2 Non-Functional Requirements	
4.2 Feasibility Study	
4.3 Use Case Scenarios	
4.3.1 Use Case Diagram	
4.4 System Requirements	
4.4.1 Software Requirements	
4.4.2 Hardware Requirements	
<b>5. SYSTEM DESIGN</b>	<b>24</b>
5.1 Introduction	
5.2 UML Diagrams	
5.2.1 Sequence Diagram	
5.2.2 State Chart Diagram	
5.2.3 Activity Diagram	
5.3 System Architecture	
5.3.1 Algorithm Specification	
5.4 User Interface	

<b>6. SYSTEM IMPLEMENTATION</b>	<b>47</b>
6.1 Technology Description	
6.2 System Modules	
6.3 Sample Code	
<b>7. TESTING</b>	<b>63</b>
7.1 Introduction	
7.1.1 Unit Testing	
7.1.2 Integration Testing	
7.1.3 Validation Testing	
7.1.4 System Testing	
7.2 User Training	
7.3 Maintenance	
7.4 Test Cases	
<b>8. OUTPUT SCREENS</b>	<b>67</b>
<b>9. CONCLUSION</b>	<b>75</b>
<b>10. REFERENCES</b>	<b>76</b>

## ABSTRACT

Now a days, maintaining customer loyalty and attention span of the customers are major challenges faced by the retail industry. This leads to the need for reinforcement of marketing strategies from time to time. This project “**CUSTOMER SEGMENTATION**” main objective is to segment the customer by analyzing their behavior, their needs and their interests. It is a systematic approach for targeting customers and providing maximum profit to the organizations. An important initial step is to analyze the data of sales from the purchase history and determine the parameters that have the maximum correlation. Based on respective clusters, proper resources can be channeled towards profitable customers using machine learning algorithms.

The proposed system begins with analyzing the data of sales from purchase history by exploring the customers and products. And then we have an insight on product categories by defining them, creating clusters of products and characterizing the content of clusters. After creating clusters for products, we create the customer categories by creating clusters. Clustering of products and Customers are done using K – Means Clustering .And finally we classify the customers using different algorithms such as Random Forest Algorithm , Logistic Regression, Decision Tree Classifier, Voting Classifier.

## LIST OF FIGURES

<b>S. No</b>	<b>Figure Name</b>	<b>Page No.</b>
1	Use Case Diagram	22
2	Sequence Diagram	28
3	State Chart Diagram	30
4	Activity Diagram	32
5	System Architecture	33
6	Opening Jupyter Notebook	67
7	Dataset used to Train and Test	67
8	Exploring Data	68
9	Data Cleaning	69
10	Categorizing Products	69
11	Visualization of how many times each keyword is used in Description	70
12	Displaying the Silhouette score for clusters in range [3,10]	71
13	Word Cloud Visualization for Product Clusters	71
14	Creating Customer clusters based on Product clusters	72
15	Grouping based on amount spent in each product cluster	72
16	Grouping based on no. of purchases made by the customer	73
17	Creating Customer clusters using K-Means	73
18	PCA Visualization of Customer Clusters	73
19	Analysis of Customer Clusters	74
20	Precision of each classification method	74

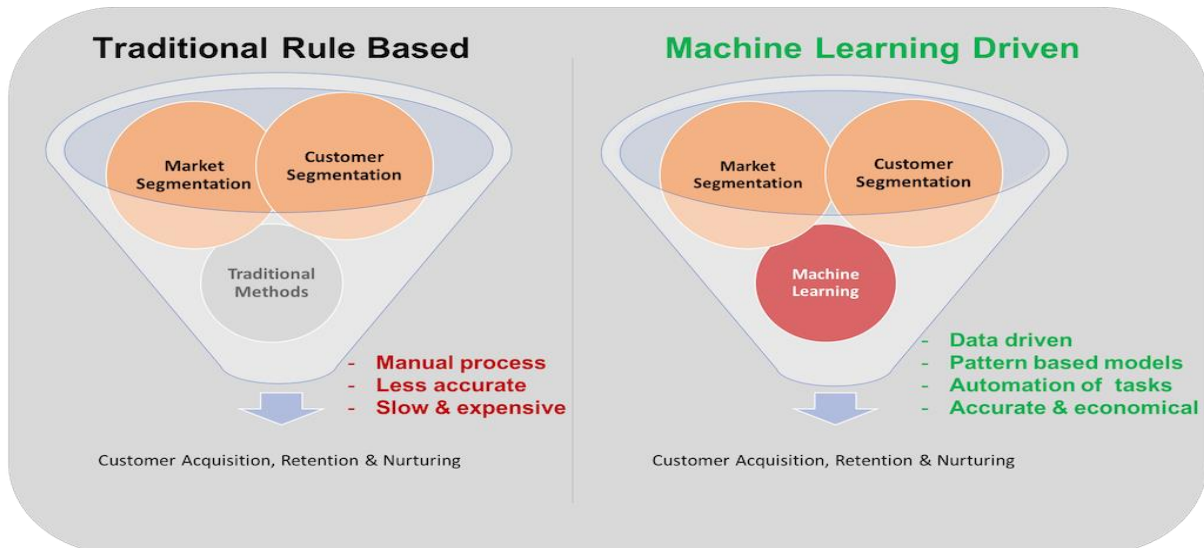


## LIST OF TABLES

S. No	Table Name	Page No.
1	Graphical Representation of Use Case Diagram	21
2	Graphical Notations for Sequence Diagram	27
3	Graphical Notations for State Chart Diagram	29
4	Test Case Representation	67

# **1. INTRODUCTION**

Customer segmentation is the practice of dividing a company's customers into groups that reflect similarity among customers in each group. The goal of segmenting customers is to decide how to relate to customers in each segment in order to maximize the value of each customer to the business.



Mounting consumer expectations and competitive pressures have created a new reality for marketers: Personalization is no longer a luxury but has become a basic standard of service in today's digital economy.

## **Different strategies for customer segmentation:**

- Rule based segmentation
- Machine Learning segmentation

To serve relevant experiences, companies have typically adhered to an approach known as rule-based personalization, which utilizes IF/Then logic to tailor the customer journey according to a set of manually programmed targeting rules.

## **Rule based Segmentation:**

The conditions of rule based segmentation, which can range from simple to complex, are all set by humans, not machines. This is a key factor behind the success of rule-based personalization initiatives, as marketers bring to bear deep industry and brand knowledge that AI may struggle with. Tasked with devising such rules ensures that the segmented and contextualized experiences a brand delivers are based on intuitive insights and real-world experience.

But for brands seeking to scale their personalization efforts, relying on an entirely manual approach to determine the most optimal experience isn't always efficient or manageable. That's why many brands are gravitating towards machine learning algorithms to assist in the decision-making process.

## **Machine Learning based Segmentation**

Through Machine Learning, brands can automate the collection and interpretation of customer insights, with algorithms or decision-making engines determining which variation a customer will be served based on performance. While this approach involves less human input than traditional rule-based personalization, the intention is to augment the marketer, not replace them.

### **Benefits of Customer segmentation:**

- Helps identify least and most profitable customers, thus helping the business to concentrate marketing activities on those most likely to buy your products or services
- Helps build loyal relationships with customers by developing and offering them the products and services they want
- Helps improve customer service
- Helps maximize use of your resources
- Helps improve or tweak products to meet customer requirements
- Helps increase profit by keeping costs down

## **1.1 PROJECT OVER VIEW**

Rule-based personalization will continue to serve as an indispensable tool, providing marketers with the ability to control which audiences are served a particular experience – and in many cases, it will remain the most logical approach for contextualizing portions of the customer journey. But as brands look to scale personalization, machine learning becomes essential.

Crucially, optimization via machine learning saves significant time and resources when it comes to A/B testing, making it a substantial boon to productivity and the bottom line. Take a holiday or back-to-school promotion. Instead of running an A/B test and trying to optimize the customer experience on the fly, machine learning algorithms make it possible to predict positive outcomes for each individual and thus maximize revenue over the duration of the entire campaign. I implore marketers to run short-lived experiments such as this, comparing the optimization mechanisms against their control group and then validating their results.

## How does customer segmentation works?

As depicted in the picture, first we take efficient data set to give input to the Machine Learning algorithm. Then according to our analyst requirements, we add segmentation and clustering algorithms to contrast the desired output visualization forms.

The speciality of ML models is that one generalised model can be used for various data sets and to extract different visualised form that consists of numerous depictions of visualizations from provided input data set information.



Our project focuses on cleaning data taken from word cloud, dividing data based on K- means algorithm and finally giving quality clusters of customers& products perspectives.

## 1.2 PROJECT DELIVERABLES

- Project Information
- Project Documentation
- Proposed System
- Requirements List
- Program

## 1.3 PROJECT SCOPE

The future of **customer segmentation** is going to continue to dig deeper and aim to reach, and truly understand, the significance of customer requirements and what they tell us with the correlativity of frequent products brought together. Customer Segmentation models detect prioritized customers and products within one scan.

Understanding people's requirements is essential for businesses since customers are able to define. By automatically analysing customer feedback, from survey responses to social media conversations, brands are able to listen attentively to their customers, and tailor products and services to meet their needs.

## **2. REVIEW OF LITERATURE**

**1.) Title:** Customer Segmentation using K-means Clustering

**Authors:** Tushar Kansal, Suraj Bahuguna, Vishal Singh, Tanu Priya Choudhury

**Published in:** 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)

### **Overview:**

The zeitgeist of modern era is innovation, where everyone is embroiled into competition to be better than others. Today's business runs on the basis of such innovation having ability to enthrall the customers with the products, but with such a large raft of products leave the customers confounded, what to buy and what to not and also the companies are nonplussed about what section of customers to target to sell their products. This elude concept of which segment to target is made unequivocal by applying segmentation. The process of segmenting the customers with similar behaviours into the same segment and with different patterns into different segments is called customer segmentation. 3 different clustering algorithms (k-Means, Agglomerative, and Mean shift) are been implemented to segment the customers and finally compare the results of clusters obtained from the algorithms. Both the features are the mean of the amount of shopping by customers and average of the customer's visit into the shop annually. However, two new clusters emerged on applying mean shift clustering labelled as High buyers and frequent visitors and High buyers and occasional visitors.

**2.) Title:** A Systematic Approach to customer Segmentation and Buyer Targeting for Profit Maximization.

**Authors:** Anshu Sang and Santosh K. Vishwakarma

**Published in:** 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)

### **Overview:**

Nowadays, maintaining customer loyalty and attention span of the customers are major challenges faced by the retail industry. This leads to the need for reinforcement of marketing strategies from time to time. This paper proposes a systematic approach for targeting customers and providing maximum profit to the organizations. An important initial step is to analyze the data of sales acquired from the purchase history and determine the parameters that have the maximum correlation. Based on respective clusters, proper resources can be channelled towards profitable customers using machine learning algorithms. K-Means clustering is used for customer segmentation and Singular Value Decomposition is used for providing appropriate recommendations to the customers. This paper also deals with the drawbacks of the recommender system like cold start problem, sparsity etc., and how they can be overcome.

### **3. PROBLEM ANALYSIS**

#### **3.1 EXISTING SYSTEM:**

In 21<sup>st</sup> century, online shopping has been developed and introduced to many fields rapidly. Though online shopping system was introduced in 1995, concept of customer segmentation has its introductory roots from the year 1956 by Welldel R. Smith as market segmentation.

In the recent years; urbanization, adaptive mobile technologies and Omni channel retailing convenience has proliferated online business and sprang up the need of focus on various online business strategies.

Customer segmentation creates subsets of a market based on demographics, needs, priorities, common interests and psychological correlativity of product buyers.

When present segmentation scenarios are taken for E-commerce websites, the customers are divided on priority bases so that production of resources can be increased in the areas which are more attractive to those customers by either only concentrating on customers or product perspectives; and changing strategies according to user demo graphs.

Though customer segmentation is a complex system to be implemented in a practical way, concentrating on only customer's point may lead to expensive production which eventually may lead to disturbance of supply and demand chain.

#### **3.2 PROPOSED SYSTEM:**

However, proposal of complete and comprehensive model for customer segmentation has many controversies. On the other side of coin, the importance of it makes us to sneak into different corners of development.

Therefore, we propose an additional aspect as enhancement of existing customer segmentation ideas by starting process of segmentation which starts with considering accurate and only relevant data from data cleaning process and then combine both customer and product perspectives by forming clusters, henceforth leading to quality clusters with better customer segmentation process.

In our scheme, we generate most promising clusters containing non duplicate values. For this, we start the process with data exploration and data cleaning for better quality of cluster. These methods help us to avoid null values, duplicate values and cancelled purchases. Further procedure includes categorizing the products into clusters and customers.

##### **3.2.1 Advantages**

###### **1) Improve Customer Experience –**

Customer segmentation helps to study different expenditures and types of customers therefore leads to better experience of customers to provide products that they tend to buy together and improves business by increasing the market value.

## **2) Build Online Reputation –**

The more a customer finds a relevant product, the more he/she tends to buy more and suggests that website to others. So the segmentation process not only helps the business analysts but also for customers to buy relevant products.

## **3) Identify Opportunities and Enhance Product Features-**

We can't ignore the day to day trends in present situations. Hence this segmentation analysis provides us better opportunities for business by giving information about all these patterns of customers.

### **3.3 LIMITATIONS:**

- Heavy Investment
- Promotion Problems
- Stock and storage problems
- Distribution variances in different places

## **4. SYSTEM ANALYSIS**

Systems analysis is a problem solving technique that decomposes a system into its component pieces for the purpose of studying how well those component parts work and interact to accomplish their purpose. System analysis is the process of studying a procedure in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way.

The development of a computer-based information system includes a systems analysis phase which produces or enhances the data-model which itself is a precursor to creating or enhancing a database. There are a number of different approaches to system analysis. When a computer-based information system is developed, systems analysis would constitute the following steps:

- The development of a feasibility study, involving determining whether a project is economically, socially, technologically and organizationally feasible.
- Conducting fact-finding measures, designed to ascertain the requirements of the system's end-users. These typically span interviews, questionnaires, or visual observations of work on the existing system
- Gauging how the end-users would operate the system (in terms of general experience in using computer hardware or software), what the system would be used for and so on.

### **4.1 SYSTEM REQUIREMENT SPECIFICATION:**

System requirements specification is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide. Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on what the software product is to do as well as what it is not expected to do. Software requirements specifications permit a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirements specifications can help prevent software project failure.

#### **4.1.1 Functional Requirements:**

Functional requirements define what a system is supposed to do. In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). How a system implements functional requirements is detailed in the system design. In some cases a requirements analyst generates



use cases after gathering and validating a set of functional requirements. Each use case illustrates behavioural scenarios through one or more functional requirements. Often, though, an analyst will begin by eliciting a set of use cases, from which the analyst can derive the functional requirements that must be implemented to allow a user to perform each use case.

- **Importing Dataset and Libraries:** The application first imports all the required libraries and extracts the necessary data from the dataset.
- **Data cleaning:** Cleaning of data takes place by eliminating duplicates and unnecessary values.
- **Removal of Cancelled Orders:** Cancelled orders are traced out and then removed from our current processing dataset to ensure good cluster formation.
- **Creation of Product Clusters:** Different Products are categorized into clusters using K-means Algorithm.
- **Creation of Customer Clusters:** Customers are categorized into clusters based on product clusters.
- **Classification of Customers:** Classification of the customers by analyzing their consumption habits is performed.
- **Testing Quality of Classifiers:** The quality of the predictions of the different classifiers was tested using Logistic Regression, Decision Tree and Random Forest Classifier.

#### **4.1.2 Non-Functional Requirements:**

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviour. This should be contrasted with functional requirements that define specific behaviour or functions. In general, functional requirements define what a system is supposed to do whereas non-functional requirements define how a system is supposed to be. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals" and "quality of service requirements," and "non-behavioural requirements." Qualities, that is, non-functional requirements, can be divided into two main categories: Execution qualities (are observable at run time), Evolution qualities (which are embodied in the static structure of the software system).

##### **Usability:**

Ease-of-use requirements address the factors that constitute the capacity of the software to be understood, learned, and used by its intended users. A system that has high usability coefficient makes the work of the user easier.

##### **Availability:**

A system's "availability" or "uptime" is the amount of time that is operational and available for use. As our system will be used by traffic management system, at any time our system must be available. If there are any cases of updations, they must be performed in a short

interval of time without interrupting the normal services made available to the users.

**Efficiency:**

Specifies how well the software utilizes scarce resources: CPU cycles, disk space, memory, bandwidth etc. All of the above mentioned resources can be effectively used by different algorithms such as K-Means; Classifiers such as Logistic Regression, Random forest Classifiers in our Segmentation of Customers.

**Flexibility:**

If the organization intends to increase or extend the functionality of the software after it is deployed, that should be planned from the beginning; it influences choices made during the design, development, testing and deployment of the system. New modules can be easily integrated to our system without disturbing the existing modules or modifying the logical database schema of the existing applications.

**Portability:**

Portability specifies the ease with which the software can be installed on all necessary platforms, and the platforms on which it is expected to run. By using appropriate server versions released for different platforms our project can be easily operated on any operating system, hence can be said highly portable.

**Scalability:**

Software that is scalable has the ability to handle a wide variety of system configuration sizes. The non-functional requirements should specify the ways in which the system may be expected to scale up (by increasing hardware capacity, adding machines etc.). Our system can be easily expandable. Any additional requirements such as hardware or software which increase the performance of the system can be easily added.

**Integrity:**

Integrity requirements define the security attributes of the system, restricting access to features or data to certain users and protecting the privacy of data entered into the software. Certain features access must be disabled to normal users such as adding the details of files, searching etc. which is the sole responsibility of the server. Access can be disabled by providing appropriate logins to the users for only access.

**Performance:**

The performance constraints specify the timing characteristics of the software. The system segments the customers accurately. It won't consider the duplicates or unnecessary values. The various precision levels of different classifiers are compared so that we can use the best classifier who segments the customers accurately. So the overall performance of the system is better than the existing solutions.

## **4.2 FEASIBILITY STUDY:**

The feasibility study is an evaluation and analysis of the potential of a proposed project. It is based on extensive investigation and research to support the process of decision making. Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of an existing or proposed system, opportunities and threats present in the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained.

A well-designed feasibility study should provide a historical background of a project, a description of a service, and details of the operations. Generally, feasibility studies precede technical development and project implementation.

A feasibility study evaluates the project's potential for success. It must therefore be conducted with an objective, unbiased approach to provide information upon which decisions can be based.

Three key considerations involved in the feasibility analysis are

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

### **Economical Feasibility:**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **Technical Feasibility:**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### **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 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.

**Scalability:** Ability to process huge amounts of data.

**Reliability:** It is an efficient way of processing data without loss.

### **Benefits of Conducting a Feasibility Study:**

Conducting a feasibility study is always beneficial to the project as it gives you and other stakeholders a clear picture of your idea. Below are the key benefits of conducting a feasibility study:

Gives project teams more focus and provides an alternative outline:

- Narrows the business alternatives.
- Identifies a valid reason to undertake the project.
- Enhances the success rate by evaluating multiple parameters.
- Aids decision-making on the project.

### **Object-Oriented Analysis:**

Object Oriented Analysis is a popular technical approach for analyzing, designing an application, system or business by applying the object-oriented paradigm and visual modelling throughout the development lifecycle to faster, better, stakeholder communication and product quality. In the case of object-oriented analysis, the process varies. But these two are identical at use case analysis. Actually, the steps involved in the analysis phase are

- Identify the actors.
- Classification-develops a static UML class diagram.
- Develop use cases.
- Identify classes, relationships, attributes, methods.

## **4.3 USE CASE SCENARIOS:**

### **Use Case Model:**

A Use case is a description of the behaviour of the system. That description is written from the point of a user who just told the system to do something particular.




#### **4.3.1 Use case Diagram:**

Use case diagrams are usually referred to as behaviour diagrams used to describe a set of actions that some systems should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

## Graphical Notation:

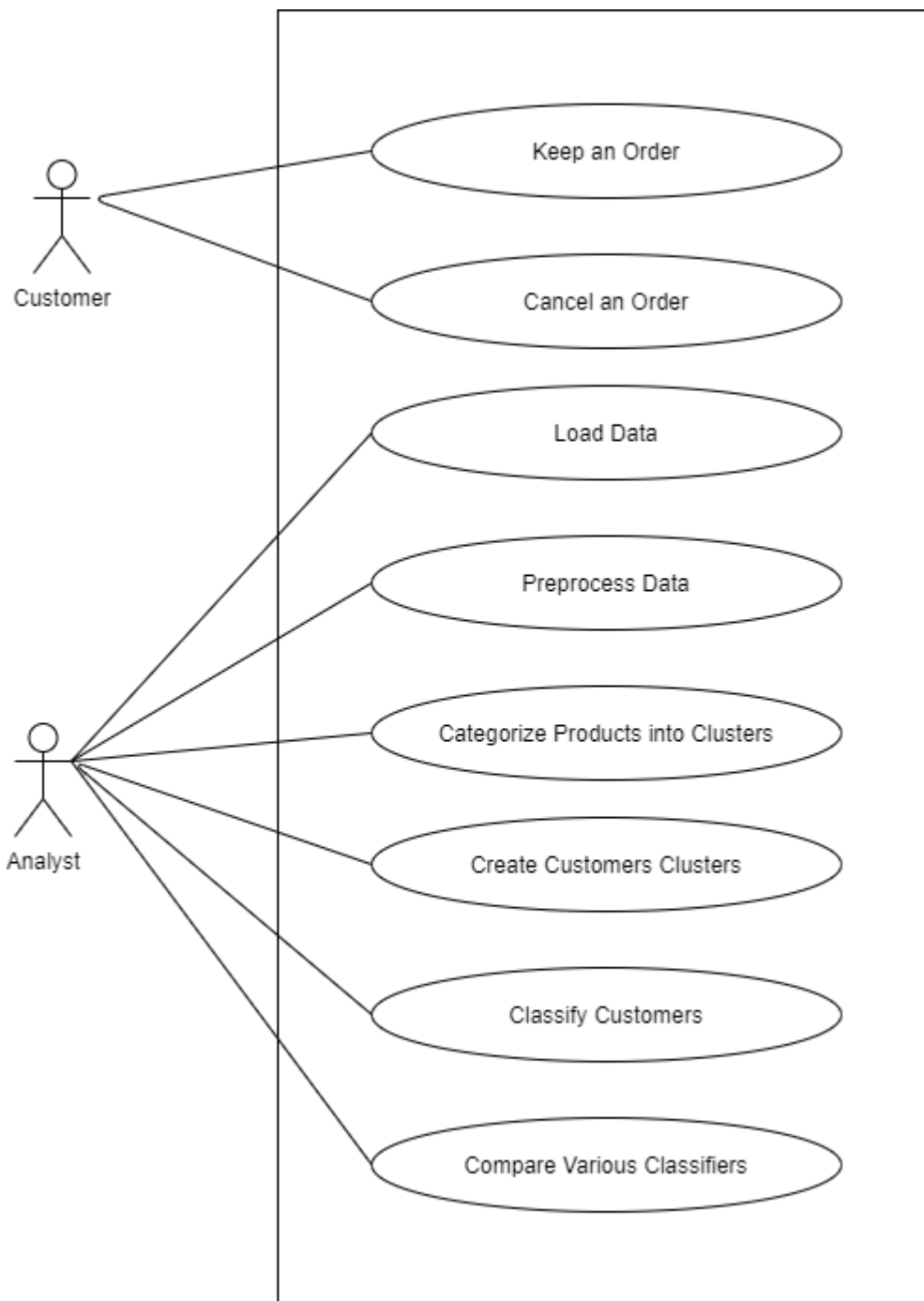
The basic components of Use Case diagrams are the Actor, the Use Case, and the Association.

*Table 4.3.1 Graphical Representation of Use Case Diagram*

<b>Actor</b>	Actor represents users of a system, including human users and other systems.	 <b>Actor Role Name</b>
<b>Use Case</b>	Use case represents functionality or services provided by a system to users.	 <b>Use Case Name</b>
<b>Association</b>	Associations between actors and use cases are indicated by solid lines. An association exists whenever an actor is involved with an interaction described by a use case.	

The main actors of Customer Segmentation Project are: Customer and Analyst who perform different types of use case activities such as:

- Keep an Order
- Cancel an Order
- Load the Data
- Pre-process the Data
- Categorize the Products into Clusters
- Creation of Customer Clusters
- Classify Customers
- Comparison of Various Classifiers



***Fig: 4.3 Use Case Diagram***

#### **4.4 SYSTEM REQUIREMENTS:**

System requirements specification is a detailed statement of the effects that a system is required to achieve. A good specification gives a complete statement of what the system is to do, without making any commitment as to how the system is to do it.

A system requirements specification is normally produced in response to a user requirements specifications or other expression of requirements, and is then used as the basis for system design. The system requirements specification typically differs from expression of requirements in both scope and precision the latter may cover both the envisaged system and the environment in which it will operate, but may leave many broad concepts unrefined.

#### 4.4.1 Software Requirements:

Software Requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

- Operating System : Windows 7 and Above
- Language used : Python
- Database : CSV file
- Libraries : NLTK, Sci-kit, Pandas, Numpy etc.,

#### 4.4.2 Hardware Requirements:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements. Hardware Requirements for present project:

- System : Minimum Pentium IV 2.4 GHz.  
Recommended Intel core i3 3.3 GHz or more.
- Hard disk : Minimum 200 GB. Recommended 500 GB or more.
- RAM : Minimum 4 GB. Recommended 4 GB or more.

## **5. SYSTEM DESIGN**

### **5.1 INTRODUCTION:**

System design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements one could see it as the application of systems theory to product development. There is some overlap and synergy with the disciplines of systems analysis, systems architecture and systems engineering.

Systems design mainly concentrates on defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. Systems design implies a systematic approach to the design of a system. It may take a bottom-up or top-down approach, but either way the process is systematic wherein it takes into account all related variables of the system that needs to be created—from the architecture, to the required hardware and software, right down to the data and how it travels and transforms throughout its travel through the system.

Systems design then overlaps with systems analysis, systems engineering and systems architecture. The systems design approach first appeared right before World War II, when engineers were trying to solve complex control and communications problems. They needed to be able to standardize their work into a formal discipline with proper methods, especially for new fields like information theory, operations research and computer science.

The system design and implementation service provide the following capabilities:

- Design of technical architectures for new IT services.
- Performance and capacity analysis of planned and existing systems.
- System development.

A system approach to design asks:

- For this situation, what is the system?
- What is the environment?
- What goal does the system have in relation to its environment?
- What is the feedback loop by which the system corrects its actions?
- How does the system measure whether it has achieved its goal?
- Who defines the system, environment, goal etc. and monitors it?
- What resources does the system have for maintaining the relationship it desires?
- Are its resources sufficient to meet its purpose?

### **5.2 UNIFIED MODELLING LANGUAGE:**

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

- a) **USER MODEL VIEW:** This view represents the system from the user's perspective. The analysis representation describes a usage scenario from the end-users perspective.



- b) **STRUCTURAL MODEL VIEW:** In this model data and functionality are arrived from inside the system. This model view models the static structures.
- c) **BEHAVIOURAL MODEL VIEW:** It represents the dynamic behaviour as parts of the system, depicting the interactions of collection between various structural elements described various structural elements described in the user model and structural model view.
- d) **IMPLEMENTATION MODEL VIEW:** In this the structural and behavioural as parts of the system are represented as they are to be built.
- e) **ENVIRONMENTAL MODEL VIEW:** In this the structural and behavioural aspect of the environment in which the system is to be implemented are represented.

As the name implies, is a modelling language. It may be used to visualize, specify, construct, and document the artifacts of a software system. It provides a set of notations to create a visual mode of the system. UML has been designed for a broad range of applications. Hence, it provides constructs for a broad range of systems and activities (e.g., distributed systems, analysis, system design, and deployment).

System development focuses on three different models of the system:

- The functional model, represented in UML with use case diagrams, describes the functionality of the system from the user's point of view.
- The object model, represented in 25 UML with class diagrams, describes the structure of the system in terms of objects, attributes, associations, and operations.
- The dynamic model, represented in UML with interaction diagrams, state machine diagrams, and activity diagrams, describes the internal behaviour of the system.

Interaction diagrams describe behaviour as a sequence of messages exchanged among a set of objects, whereas state machine diagrams describe behaviour in terms of states of an individual object and the possible transitions between states. Activity diagrams describe behaviour in terms control and data flows.

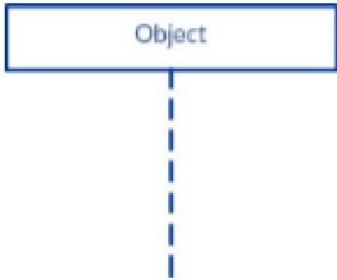



## **5.2.1 SEQUENCE DIAGRAM:**


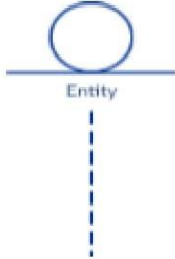
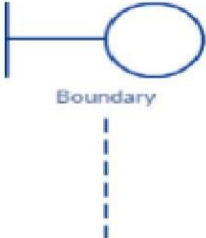

A sequence diagram shows object interactions arranged in time sequence. It depicts - the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

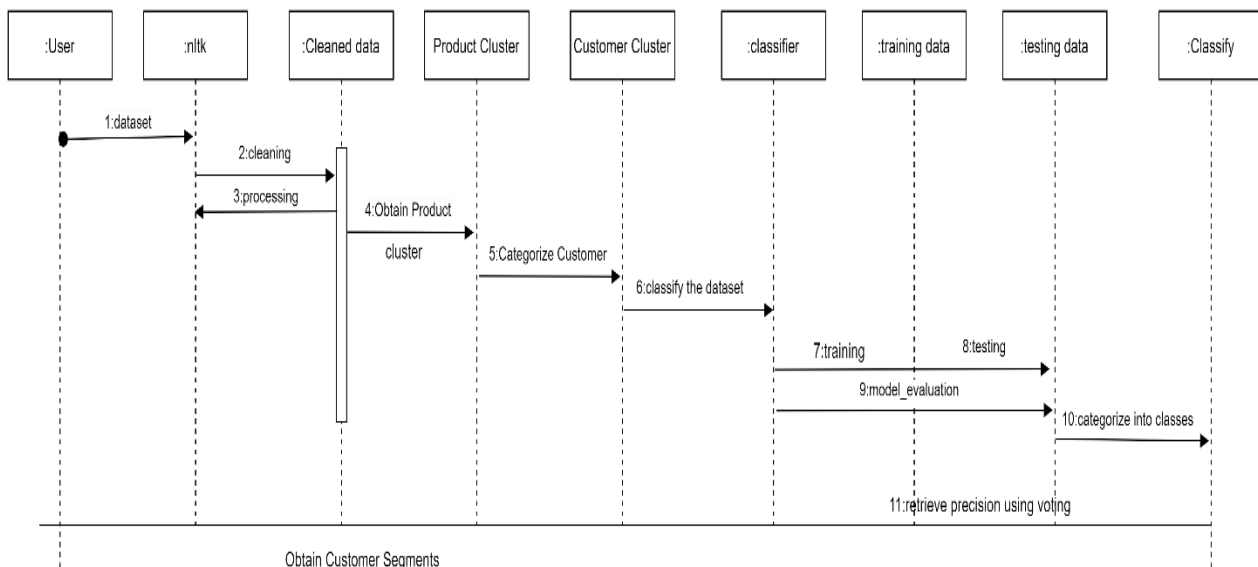
Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram describes an interaction among a set of objects participated in a collaboration (or scenario), arranged in a chronological order; it shows the objects participating in the interaction by their "lifelines" and the messages that they send to each object.

**Table 5.2.1 Graphical Notations for Sequence Diagram**

Object	Objects are instances of classes and are arranged horizontally. The pictorial representation for an Object is class (a rectangle) with the name prefixed by the object name	
Actor	Actor can also communicate with objects so they too can be listed as a column. An Actor is modeled using the stick figure.	
LifeLine	The Lifeline identifies the existence of the object over time. The notation for a life time is a vertical dotted line extending from an object	
Activation	Activation modeled as rectangular boxes on the lifeline indicate when the object is performing an action	

Message	<p>Messages modeled as horizontal arrows between Activations indicate the communication between the objects.</p>	
Entity	<p>A lifeline with an entity element represents system data.</p>	
Boundary	<p>A lifeline with a boundary element indicates a system boundary/ software element in a system</p>	
Control	<p>lifeline with a control element indicates a controlling entity or manager. It organizes and schedules the interactions between the boundaries and entities and serves as the mediator between them.</p>	



**Fig: 5.2.1 Sequence Diagram**

## 5.2.2 STATE CHART DIAGRAM:

State chart diagram is used to describe the states of different objects in its life cycle. Emphasis is placed on the state changes upon some internal or external events. These states of objects are important to analyse and implement them accurately.

State chart diagrams are very important for describing the states. States can be identified as the condition of objects when a particular event occurs.

A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. A state diagram is used to describe the behaviour of systems. This behaviour is analysed and represented in series of events that could occur in one or more possible states. State diagrams require that the system described is composed of a finite number of states. Sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.





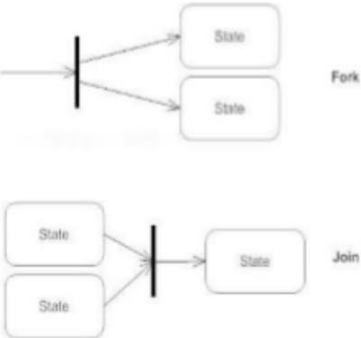
The main purposes of using State Chart diagrams –

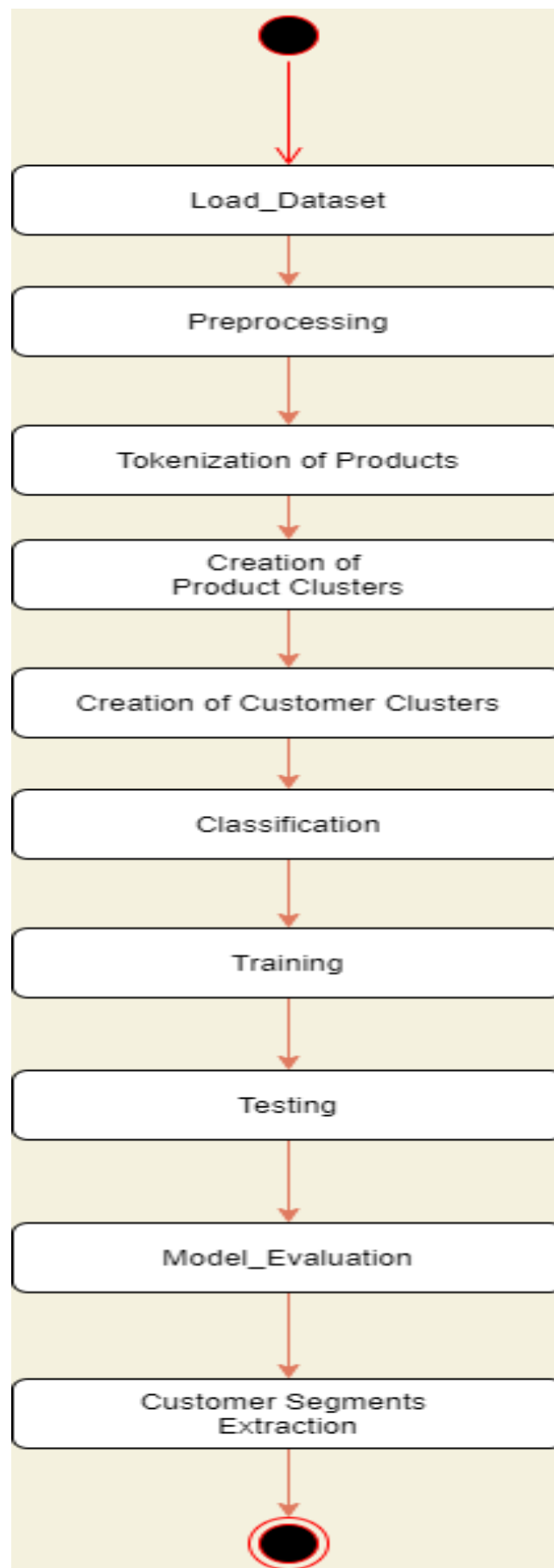
- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

The Main Usage can be described as:

- To Model the object states of a System.
- To Model the reactive System.
- To identify the events responsible for state changes.
- Forward and Reverse Engineering

**Table 5.2.4 Graphical Notations for State Chart Diagram**

Initial State	The initial state represents the source of all objects. A filled circle followed by an arrow represents the object's initial state.	
State	State represents situations during the life of an object. Rectangular boxes with curved edges represent a state	
Transition	A transition represents the change from one state to another. A solid arrow represents the path between different states of an object	
Final State	The final state represents the end of an object's existence. A final state is not a real estate, because objects in this state don't exist anymore. A filled circle represents the final state	
Synchronization and Splitting	A short heavy bar with two transitions entering it represents a synchronization of control. The first bar is called a fork where a single transition splits into concurrent multiple transitions. The second bar is called a join, where the concurrent transitions reduce back to one	



*Fig: 5.2.2 State Chart Diagram*

### 5.2.3 ACTIVITY DIAGRAM:

Activity diagrams are mainly used as a flowchart consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities.

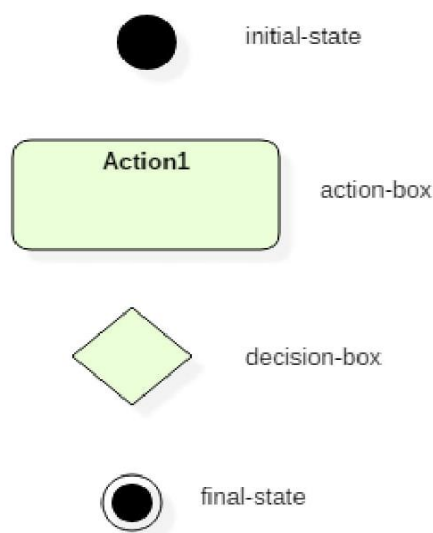
This UML activity diagram shows the Login activity, where the admin will be able to login using username and password.

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modelled can be sequential and concurrent.

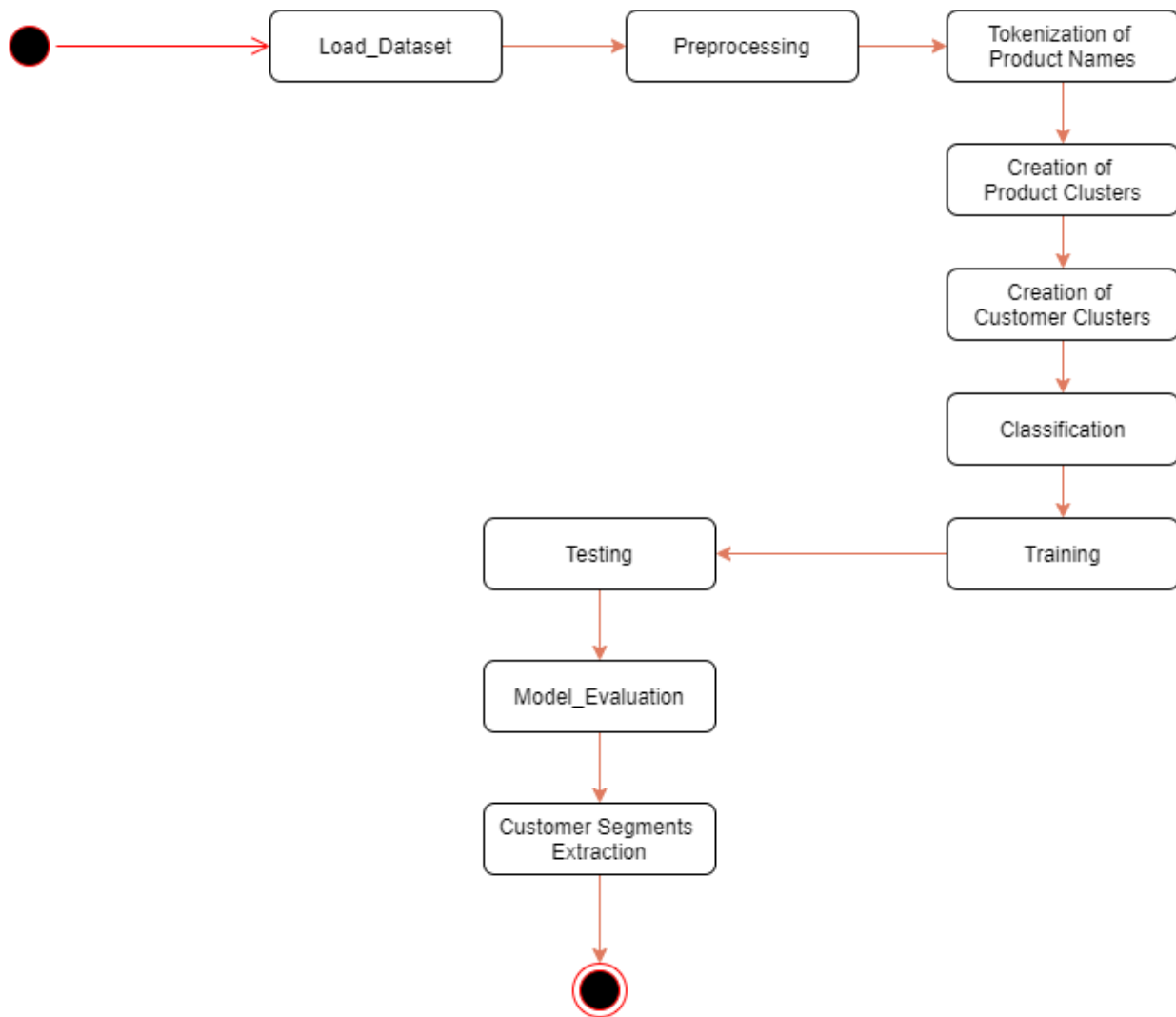
#### Activity Diagram Notations:

Activity diagrams symbol can be generated by using the following notations:

- **Initial states:** The starting stage before an activity takes place is depicted as the initial state
- **Final states:** The state which the system reaches when a specific process ends is known as a Final State
- **State or an activity box**
- **Decision box:** It is a diamond shape box which represents a decision with alternate paths. It represents the flow of control.



#### *Flow of Control of Activity Diagram*

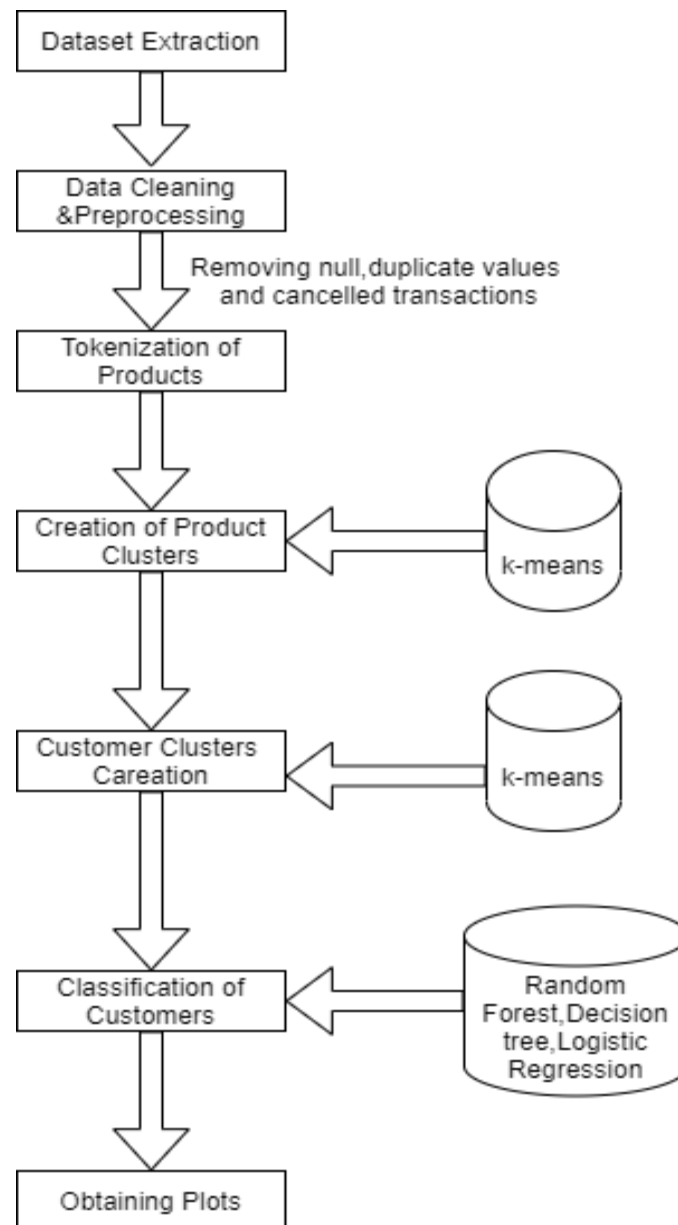


*Fig: 5.2.3 Activity Diagram*

### 5.3 SYSTEM ARCHITECTURE:

Systems architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system.





*Fig: 5.3 System Architecture*

### 5.3.1 Algorithm Specification:

#### Description:

**Step-1:** E-commerce dataset that lists purchases made by 4000 customers approximately undergoes pre-processing and cleaning by tokenizing, removal of cancelled transactions, removal of null and duplicate values etc.

**Step-2:** Find the most repeated words in the Description of Product.

**Step-3:** k-means clustering obtains similar features of the products into 5 clusters which is visualized by WordCloud Data Visualization

**Step-4:** Based on product clusters, type of products they usually buy, the number of purchases made etc., customers are categorized using k-means clustering with 11 clusters.

**Step-5:** These Customer Categories are viewed using Principal Cluster Analysis (PCA).

**Step-6:** Dataset is trained and tested using different algorithms such as Decision Tree, Logistic Regression, and Random Forest Classifier.

**Step-7:** Soft Voting is used for the above algorithms to get better performance than any model individually.

### **K-means Clustering Algorithm:**

K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more. Finally, this algorithm aims at minimizing an objective function known as squared error function given by:

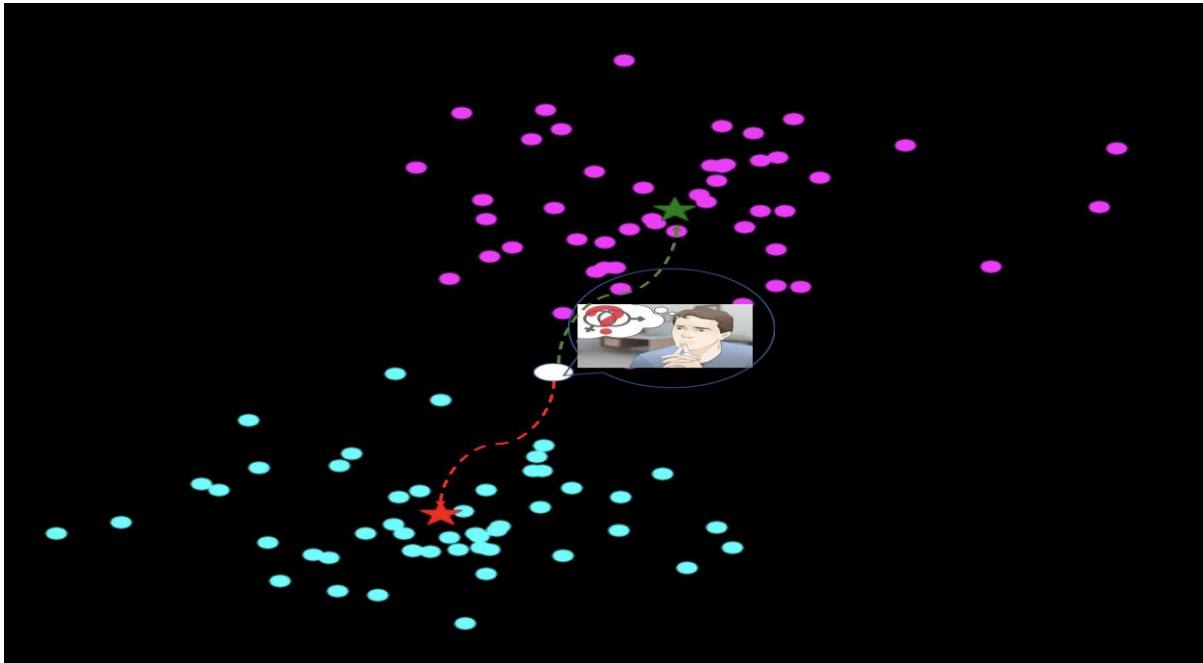
$$J(V) = \sum_{i=1}^c \sum_{j=1}^{c_i} (\|x_i - v_j\|)^2$$

Where

' $\|x_i - v_j\|$ ' is the Euclidean distance between  $x_i$  and  $v_j$ .

' $c_i$ ' is the number of data points in  $i^{th}$  cluster.

' $c$ ' is the number of cluster centers.



### Algorithmic steps for k-means clustering:

Let  $X = \{x_1, x_2, x_3, \dots, x_n\}$  be the set of data points and  $V = \{v_1, v_2, \dots, v_c\}$  be the set of centers.

- 1) Randomly select ' $c$ ' cluster centers.
- 2) Calculate the distance between each data point and cluster centers.
- 3) Assign the data point to the cluster center whose distance from the cluster center is the minimum of all the cluster centers.
- 4) Recalculate the new cluster center.
- 5) Recalculate the distance between each data point and new obtained cluster centers.
- 6) If no data point was reassigned then stop, otherwise repeat from step 3).

### Applications:

K-means algorithm is very popular and used in a variety of applications such as market segmentation, document clustering, image segmentation and image compression, etc. The goal usually when we undergo a cluster analysis is either:

- Get a meaningful intuition of the structure of the data we're dealing with.
- Cluster-then-predict where different models will be built for different subgroups if we believe there is a wide variation in the behaviours of different subgroups. An example of that is clustering patients into different subgroups and builds a model for each subgroup to predict the probability of the risk of having heart attack.

## Word Cloud Visualization:

A word cloud (or *tag cloud*) is a word visualization that displays the most used words in a text from small to large, according to how often each appears.

They give a glance into the most important keywords in news articles, social media posts, and customer reviews, among other text. They can also provide interesting insights when comparing two texts against each other, like political speeches or product reviews.



## **Algorithm:**

It is important to pre-process text before you visualize it with a word cloud. Common pre-processing steps include:

- 1) Remove punctuation: The rule of thumb is to remove everything that is not in the form x, y, z.
- 2) Remove stop words: These are unhelpful words like "the", "is", or "at". These are not helpful because the frequency of such stop words is high in the corpus, but they don't help in differentiating the target classes. The removal of stop words also reduces the data size.
- 3) Conversion to lowercase: Words like "Clinical" and "clinical" need to be considered as one word. Hence, these are converted to lowercase.
- 4) Stemming: The goal of stemming is to reduce the number of inflectional forms of words appearing in the text. This causes words such as "argue," "argued," "arguing," and "argues" to be reduced to their common stem, "argu". This helps in decreasing the size of the vocabulary space.

## Applications:

In the right setting, word cloud visualizations are a powerful tool. Here are a few instances when word clouds excel:

- **Finding customer pain points — and opportunities to connect.** Do you collect feedback from your customers? (You should!) Analyzing your customer feedback can allow you to see what your customers like most about your business and what they like least. Pain points (such as “wait time,” “price,” or “convenience”) are very easy to identify with text clouds.
- **Understanding how your employees feel about your company.** Text cloud visualization can turn employee feedback from a pile of information you’ll read through later to an immediately valuable company feedback that positively drives company culture.
- **Identifying new SEO terms to target.** In addition to normal keyword research techniques, using a word cloud may make you aware of potential keywords to target that your site content already uses.

## PCA(Principle Component Analysis) :

**Principal Component Analysis (PCA)** is a statistical procedure that uses an orthogonal transformation which converts a set of correlated variables to a set of uncorrelated variables. PCA is a most widely used tool in exploratory data analysis and in machine learning for predictive models. Moreover, PCA is an unsupervised statistical technique used to examine the interrelations among a set of variables. It is also known as a general factor analysis where regression determines a line of best fit.

### Objectives of PCA:

- It is basically a non-dependent procedure in which it reduces attribute space from a large number of variables to a smaller number of factors.
- PCA is basically a dimension reduction process but there is no guarantee that the dimension is interpretable.
- Main task in this PCA is to select a subset of variables from a larger set, based on which original variables have the highest correlation with the principal amount.

**Principal Axis Method:** PCA basically search a linear combination of variables so that we can extract maximum variance from the variables. Once this process completes it removes it and search for another linear combination which gives an explanation about the maximum proportion of remaining variance which basically leads to orthogonal factors. In this method, we analyze total variance.

**Eigenvector:** It is a non-zero vector that stays parallel after matrix multiplication. Let’s suppose  $x$  is eigen vector of dimension  $r$  of matrix  $M$  with dimension  $r \times r$  if  $Mx$  and  $x$  are parallel. Then we need to solve  $Mx = Ax$  where both  $x$  and  $A$  are unknown to get eigen vector and eigen values. Under Eigen-Vectors we can say that Principal components show both common and unique variance of the variable. Basically, it is variance focused approach seeking to reproduce total variance and correlation with all components. The principal components are basically the linear combinations of the original variables weighted by their contribution to explain the variance in a particular orthogonal dimension.

**Eigen Values:** It is basically known as characteristic roots. It basically measures the variance in all variables which is accounted for by that factor. The ratio of eigenvalues is the ratio of explanatory importance of the factors with respect to the variables. If the factor is low then it is contributing less in explanation of variables. In simple words, it measures the amount of variance in the total given database accounted by the factor. We can calculate the factor's eigen value as the sum of its squared factor loading for all the variables.

To calculate the following eigenvectors we modify the iterative algorithm. Now we use the calculation formula:

$$w^{k,new} = w^k + \eta y_i (x^{*i} - y^i w^k)$$

$$\text{where } x^{*i} = x^i - \sum_{j=1}^{k-1} u_j^i w^j \quad \text{and } u_j^i = w^j T x^i.$$

This iterative algorithm converges to  $w$  power  $k$  the  $k^{\text{th}}$  eigen vector.

### Applications:

- It is used to find inter-relation between variables in the data.
- It is used to interpret and visualize data.
- As number of variables are decreasing it makes further analysis simpler.
- It's often used to visualize genetic distance and relatedness between populations.

Because of the versatility and interpretability of PCA, it has been shown to be effective in a wide variety of contexts and disciplines. Given any high-dimensional dataset, I tend to start with PCA in order to visualize the relationship between points (as we did with the digits), to understand the main variance in the data (as we did with the eigen faces), and to understand the intrinsic dimensionality (by plotting the explained variance ratio).

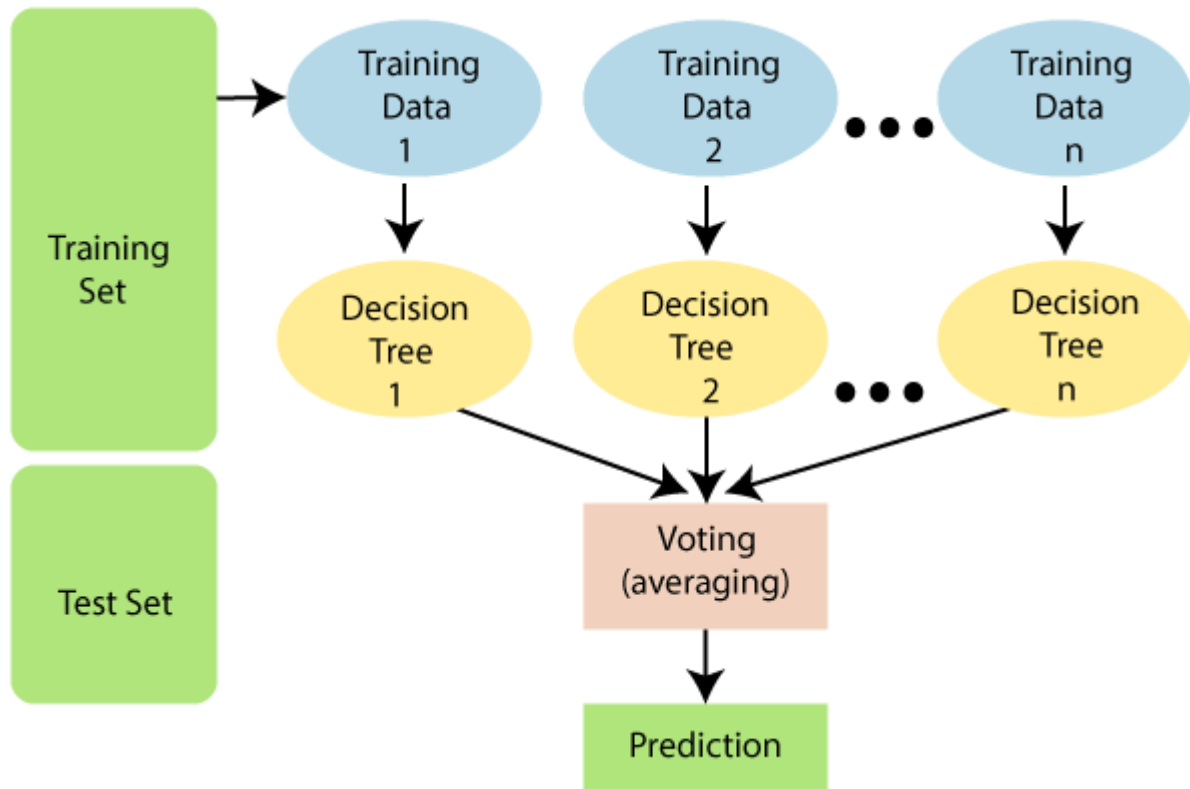
### Random Forest Algorithm:

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 over fitting.**

The below diagram explains the working of the Random Forest algorithm:



### Algorithm:

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

**Step-1:** Select random K data points from the training set.

**Step-2:** Build the decision trees associated with the selected data points (Subsets).

**Step-3:** Choose the number N for decision trees that you want to build.

**Step-4:** Repeat Step 1 & 2.

**Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

### Applications:

There are mainly four sectors where Random forest mostly used:

1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.

2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
3. **Land Use:** We can identify the areas of similar land use by this algorithm.
4. **Marketing:** Marketing trends can be identified using this algorithm.

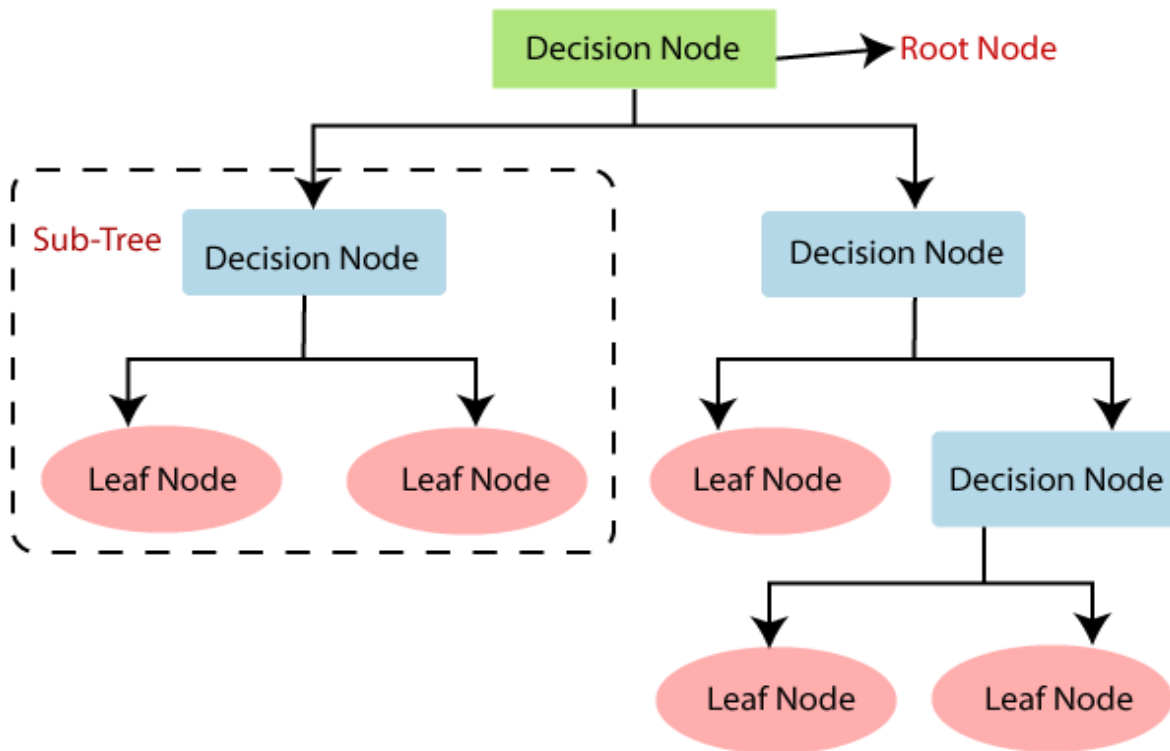
### **Decision Tree Classification Algorithm:**

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.
- *It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.*
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- 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 sub trees.

Below diagram explains the general structure of a decision tree:





### Algorithm:

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

**Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.

**Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM)**.

**Step-3:** Divide the S into subsets that contains possible values for the best attributes.

**Step-4:** Generate the decision tree node, which contains the best attribute.

**Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3.  
Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

### Applications:

1. **Assessing prospective growth opportunities:** One of the applications of decision trees involves evaluating prospective growth opportunities for businesses based on historical data. Historical data on

sales can be used in decision trees that may lead to making radical changes in the strategy of a business to help aid expansion and growth.

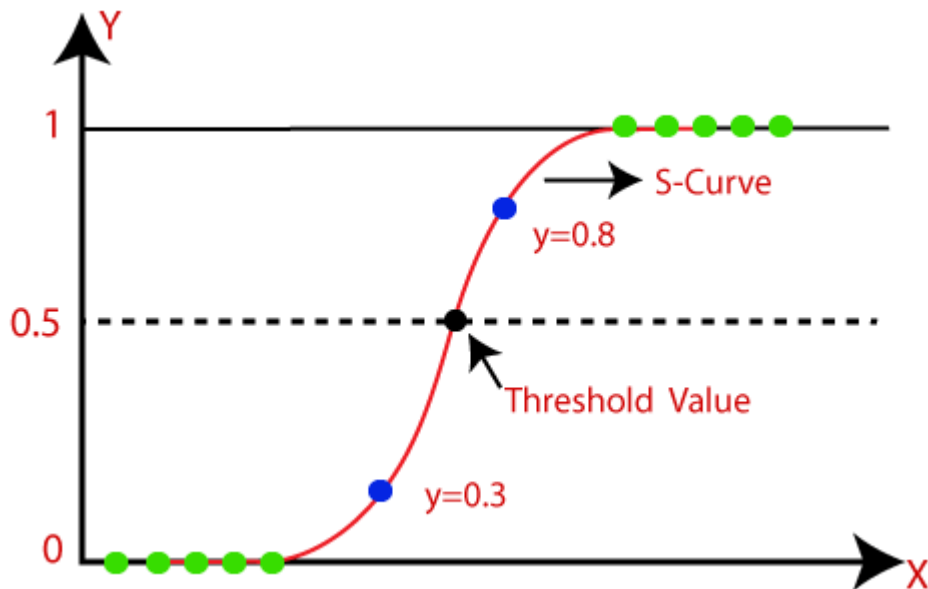
**2. Using demographic data to find prospective clients:** Another application of decision trees is in the use of demographic data to find prospective clients. They can help in streamlining a marketing budget and in making informed decisions on the target market that the business is focused on. In the absence of decision trees, the business may spend its marketing market without a specific demographic in mind, which will affect its overall revenues.

**3. Serving as a support tool in several fields:** Lenders also use decision trees to predict the probability of a customer defaulting on a loan, by applying predictive model generation using the client's past data. The use of a decision tree support tool can help lenders in evaluating the creditworthiness of a customer to prevent losses.

Decision trees can also be used in operations research in planning logistics and strategic management. They can help in determining appropriate strategies that will help a company achieve its intended goals. Other fields where decision trees can be applied include engineering, education, law, business, healthcare, and finance.

## **Logistic Regression in Machine Learning**

- 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 similar to 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 has the ability to 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:



### 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:

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

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

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

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

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

The above equation is the final equation for Logistic Regression.

### Type of Logistic Regression:

On the basis of the categories, Logistic Regression can be classified into three types:

- **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

- **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
- **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

## **Applications:**

Regressions can be used in real world applications such as :

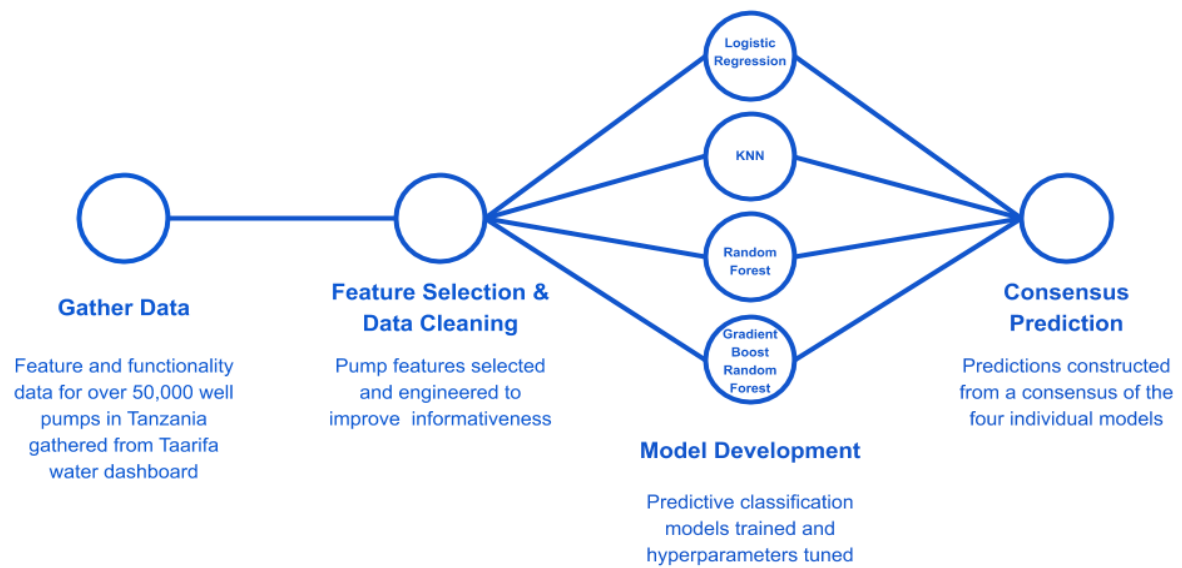
1. Credit Scoring
2. Measuring the success rates of marketing campaigns
3. Predicting the revenues of a certain product
4. Is there going to be an earthquake on a particular day? Etc.,

## **Voting Classifier:**

A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output (class) based on their highest probability of chosen class as the output. It simply aggregates the findings of each classifier passed into Voting Classifier and predicts the output class based on the highest majority of voting. The idea is instead of creating separate dedicated models and finding the accuracy for each of them, we create a single model which trains by these models and predicts output based on their combined majority of voting for each output class.

Voting Classifier supports two types of voting.

- 1) **Hard Voting:** In hard voting, the predicted output class is a class with the highest majority of votes i.e., the class which had the highest probability of being predicted by each of the classifiers. Suppose three classifiers predicted the *output class* ( $A, A, B$ ), so here the majority predicted  $A$  as output. Hence  $A$  will be the final prediction.
- 2) **Soft Voting:** In soft voting, the output class is the prediction based on the average of probability given to that class. Suppose given some input to three models, the prediction probability for class  $A = (0.30, 0.47, 0.53)$  and  $B = (0.20, 0.32, 0.40)$ . So the average for class  $A$  is  $0.4333$  and  $B$  is  $0.3067$ , the winner is clearly class  $A$  because it had the highest probability averaged by each classifier.



### How voting work and applications:

Classification is an important machine learning technique that is often used to predict categorical labels. It is a very practical approach for making binary predictions or predicting discrete values. The classifier, another name for classification model, might have the intention of predicting whether someone is eligible for a job or it could be used to classify the images of multiple objects in a store.

Classification, like other machine learning techniques, uses data sets. A dataset is a combination of multiple values from different variables. After obtaining an optimal dataset, it is split into two: the training and testing set. The training set often has a larger proportion of the dataset. It is likely to take up about 70% to 90% of the dataset.

The training set is inserted into the machine learning algorithm to create a predictive model with an added step called cross-validation. Cross-validation is a great way to ensure that the built model does not over fit the training set and it also optimizes the versatility of the model. Then, the model can be used to predict the labels in the testing set. The predicted labels are further compared to the actual testing set labels via metrics such as confusion matrix, precision score, recall score, F1-score, roc auc score. Once the construction of the classification model is over, a data point's values can be inserted into the algorithm and the algorithm makes a decision by attributing a specific label to this data point based on the variables' inputs.

Now imagine if different classification methods were asked to make decisions based on the data instances' inputs. There are bound to be different answers. This is where **voting classifiers** come into play.

## 5.4 USER INTERFACE:

A user interface, also called a "UI" or simply an "interface," is the means in which a person controls a software application or hardware device. A good user interface provides a "user-friendly" experience, allowing the user to interact with the software or hardware in a natural and intuitive way.

The user interface in the industrial design field of human–computer interaction, is the space where interactions between humans and machines occur. The goal of this interaction is to allow effective operation and control of the machine from the human end, whilst the machine simultaneously feeds back information that aids the operators' decision-making process. Examples of this broad concept of user interfaces include the interactive aspects of computer operating systems, hand tools, heavy machinery operator controls, and process controls. The design considerations applicable when creating user interfaces are related to or involve such disciplines as ergonomics and psychology.

Generally, the goal of user interface design is to produce a user interface which makes it easy (self-explanatory), efficient, and enjoyable (user-friendly) to operate a machine in the way which produces the desired result. This generally means that the operator needs to provide minimal input to achieve the desired output, and also that the machine minimizes undesired outputs to the human. With the increased use of personal computers and the relative decline in societal awareness of heavy machinery, the term user interface is generally assumed to mean the graphical user interface, while industrial control panel and machinery control design discussions more commonly refer to human-machine interfaces.

Nearly all software programs have a graphical user interface, or GUI. This means the program includes graphical controls, which the user can select using a mouse or keyboard. A typical GUI of a software program includes a menu bar, toolbar, windows, buttons, and other controls. The Macintosh and Windows operating systems have different user interfaces, but they share many of the same elements, such as a desktop, windows, icons, etc. These common elements make it possible for people to use either operating system without having to completely relearn the interface. Similarly, programs like word processors and Web browsers all have rather similar interfaces, providing a consistent user experience across multiple programs.

Most hardware devices also include a user interface, though it is typically not as complex as a software interface. A common example of a hardware device with a user interface is a remote control. A typical TV remote has a numeric keypad, volume and channel buttons, mute and power buttons, an input selector, and other buttons that perform various functions. This set of buttons and the way they are laid out on the controller makes up the user interface. Other devices, such as digital cameras, audio mixing consoles, and stereo systems also have a user interface.

While user interfaces can be designed for either hardware or software, most are a combination of both. For example, to control a software program, you typically need to use a keyboard and mouse, which each have their own user interface. Likewise, to control a digital camera, you may need to navigate through the on-screen menus, which is a software interface. Regardless of the application, the goal of a good user interface is to be user-friendly.

## **6. SYSTEM IMPLEMENTATION**

### **6.1 TECHNOLOGY DESCRIPTION:**

#### **6.1.1 Windows:**

Windows is a series of operating systems developed by Microsoft. Each version of Windows includes a graphical user interface, with a desktop that allows users to view files and folders in windows. For the past two decades, Windows has been the most widely used operating system for personal computers PCs.

#### **Features:**

##### **1. Speed:**

Even aside from incompatibilities and other issues that many people had with Vista, one of the most straightforward was speed – it just felt too sluggish compared to XP, even on pumped up hardware. Windows 10 brings a more responsive and sprightlier feel and Microsoft has spent a lot of time and effort getting the Start Menu response just right.

Microsoft has also recognized the need for improved desktop responsiveness, which gives the impression that the computer is responding to the user and that they are in control – something that was often lacking with Vista.

You can also expect faster boot times. And the boot sequence is now not only prettier than it was with Vista, but it's speedier too.

##### **2. Compatibility**

In simple terms, compatibility on Windows 10 will be far better than it was with Vista. Many programs that individuals and companies used on Windows XP did not work immediately and required updates, but with Windows 10 almost all applications that work on Vista should still run.

##### **3. Lower Hardware Requirements**

Vista gained a reputation for making even the beefiest hardware look rather ordinary. Windows 10, however, will run well on lower end hardware, making the transition from Window XP less painful.

##### **4. Search and Organization**

One of the best things about Windows 10 is the improved search tool, which now rivals Mac OS X's Spotlight to be able to find what you need quickly and easily. For example, typing 'mouse' will bring up the mouse option within the control panel or typing a word will display it and split it up neatly into files, folders and applications.

##### **5. Safety and Security**

New security features in Windows include two new authentication methods tailored towards touch screens (PINs and picture passwords), the addition of antivirus capabilities to Windows Defender (bringing it in parity with Microsoft Security Essentials) Smart Screen filtering integrated into Windows, and support for the "Secure Boot" functionality on UEFI systems to protect against malware infecting the

boot process. Family Safety offers Parental controls, which allows parents to monitor and manage their children's activities on a device with activity reports and safety controls.

### **6.1.2 Anaconda (Python Distribution):**

Anaconda is a free and open source distribution of the python and R programming Language for scientific computing such as Data Science, Machine Learning applications, large-scale data processing, predictive analytics etc.

**Anaconda Navigator:** Anaconda Navigator is a desktop Graphical User Interface includes in Anaconda Distribution that allows users to launch applications and manage conda packages, environments and challenges without using command line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them.

The following Applications are available by default in navigator:

- Jupyter Lab
- Jupyter Notebook
- QtConsole
- Spyder
- Glue
- Orange
- RStudio
- Visual Studio Code.

### **Benefits of Using Python Anaconda:**

- It is free and open-source.
- It has more than 1500 Python package.
- It has tools to easily collect data from sources using machine learning and AI.
- Anaconda simplifies package management and deployment.
- Anaconda is the industry standard for developing, testing and training on a single machine.
- It creates an environment that is easily manageable for deploying any project.
- It has good community support- you can ask your questions there.

### **What you get:**

- Download more than 1500 Python/R data science packages.
- Manage libraries, dependencies, and environments with conda.
- Use Dask, NumPy, Pandas and Numba to analyze data scalably and fast.
- Build and train ML and deep learning models with scikit-learn, Tensor Flow and Theano.
- Perform visualization with Matplotlib, Bokeh, Datashader, and Holoviews.

### **Jupyter Notebook:**

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text.



Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

### **Getting Up and Running With Jupyter Notebook:**

The Jupyter Notebook is not included with Python, so if you want to try it out, you will need to install Jupyter.

There are many distributions of the Python language. This will focus on just two of them for the purposes of installing Jupyter Notebook. The most popular is CPython, which is the reference version of Python that you can get from their website. It is also assumed that you are using Python 3.

### **NLP (Natural Language Processing):**

- NLP (Natural Language Processing) is a set of techniques for approaching text problems that enables computer programs and functions to understand human speech as it is spoken.
- We are using the bag of words approach as it takes common concepts from machine learning like feature engineering and model building and tweaks them only slightly.
- With the rise of voice interfaces and chatbots, NLP is one of the most important technologies of the information age, a crucial part of artificial intelligence. Fully understanding and representing the meaning of language is an extremely difficult goal. Why? Because human language is quite special.

### **Why study NLP?**

There's a fast-growing collection of useful applications derived from this field of study. They range from simple to complex. Below are a few of them:

- Spell Checking, Keyword Search, Finding Synonyms.
- Extracting text summarizing using neural networks from websites such as: product price, dates, location, people, or company names.
- Classifying: reading level of school texts, positive/negative sentiment of longer documents.
- Machine Translation.
- Spoken Dialog Systems.
- Complex Question Answering.

Indeed, these applications have been used abundantly in industry: from search (written and spoken) to online advertisement matching; from automated/assisted translation to sentiment analysis model for marketing or finance/trading; and from speech recognition to chat bots/dialog agents (automating customer support, controlling devices, ordering goods).

### **NLTK:**

NLTK stands for Natural Language Processing Toolkit. This toolkit is one of the powerful NLP Libraries which contains packages to make machines understand human language and reply to it with an appropriate response. Tokenization, Stemming, Lemmatization, Punctuation, Character Count, word Count are some of the packages. It ships with graphical demonstrations and sample data.

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, etc. This library provides a practical introduction to programming for language processing. NLTK has been called “a wonderful tool for teaching and working in computational linguistics using Python,” and “an amazing library to play with natural language.”

### 6.1.3 Python:

Python is a general-purpose interpreted, interactive, object-oriented, and high level Programming language. Python is designed to be highly readable. It uses English keywords frequently where as other languages. Python is easy to learn yet powerful and versatile scripting language, which makes it for Application development.

- **Python is interpreted:** Python is processed at runtime by the interpreter. You don't need to compile your programs before executing it. This is similar to PHP.
- **Python is Interactive:** you can actually sit at python prompt and interact with the interpreter directly to write your programs.
- **Python is Object Oriented:** Python supports Object-oriented style programming that encapsulates code within objects.
- **Python is Beginner's Language:** Python is a great language for the beginner-level and support development of wide range applications from simple text processing to WWW browsers to games.

### Applications of Python:

- **Web Applications:** It Provides Libraries to handle internet protocols such as HTML and XML, JSON, Email Processing, request, beautiful Soup, FeedParser etc. It also provides Frameworks such as Django, Pyramid, Flask etc., to design and develop web based applications. Some important developments are: Python Wiki Engines, Pocoo, and Python Blog Software.
- **Desktop GUI Applications:** Python provides Tk GUI Library to develop user interface in python based application. Some other toolkits such as wxWidgets, Kivy, pyqt that are useable on several platforms. The kivy is popular for writing multitouch applications.
- **Software Development:** Python is popular and widely used in scientific and numeric computing. Some useful library and package are SciPy, Pandas, IPython etc., SciPy is group of packages of engineering, science and mathematics.
- **Business Applications:** Python is used to build Business applications like ERP and e-commerce systems. Tryton is a high level application platform.
- **Console Based Applications:** We can use Python to develop console based applications.eg: IPYTHON
- **Audio Video Based Applications:** Python is awesome to perform multiple tasks and can be used to develop multimedia applications. Some of the real applications are: Time Player, cplay etc.,
- **3D CAD Applications:** To create CAD applications Fandango is a real application which provides full features of CAD.
- **Enterprise Applications:** Python can be used to create applications which can be used within an Enterprise or an Organization. Some real time applications: OpenErp, Tryton, Picalo etc.

## **Features of Python Programming:**

### **1. Easy to code:**

Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in a few hours or days. It is also a developer friendly language.

### **2. Free and Open Source:**

Python language is freely available at the official website. Since, it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

### **3. Object-Oriented Language:**

One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, objects encapsulation etc.

### **4. GUI Programming Support:**

Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wxPython or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.

### **5. High-Level Language:**

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

### **6. Extensible feature:**

Python is an Extensible language. We can write our python code into C or C++ language and also we can compile that code in C/C++ language.

### **7. Python is Portable language:**

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, UNIX and Mac then we do not need to change it, we can run this code on any platform.

### **8. Python is integrated language:**

Python is also an integrated language because we can easily integrate Python with other languages like C, C++ etc.

### **9. Interpreted Language:**

Python is an Interpreted Language because python code is executed line by line at a time. Unlike other languages C, C++, Java etc. there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called Byte code.

## 10. Large Standard Library:

Python has a large standard library which provides a rich set of modules and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

## 11. Dynamically Typed Language:

Python is dynamically-typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

### Libraries used:

- **Pandas:** In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.
- **Numpy:** NumPy is a Python package which stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object, provide tools for integrating C, C++ etc., The NumPy arrays takes significantly less amount of memory as compared to python lists. It also provides a mechanism of specifying the data types of the contents, which allows further optimization of the code.
- **Matplotlib:** Matplotlib produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, web application servers, and various graphical user interface toolkits.
- **Matplotlib.pyplot:** Plotting library for 2D graphics plotting.
- **Seaborn:** Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
- **NLTK:** The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning.
- **Datetime:** Encapsulation of date/time values.
- **Warnings:** Warning messages are typically issued in situations where it is useful to alert the user of some condition in a program, where that condition (normally) doesn't warrant raising an exception and terminating the program. For example, one might want to issue a warning when a program uses an obsolete module.
- **Itertool:** Itertool is a module that provides various functions that work on iterators to produce complex iterators. This module works as a fast, memory-efficient tool that is used either by them or in combination to form **iterator algebra**.
- **Pathlib:** Pathlib module in Python provides various classes representing file system paths with semantics appropriate for different operating systems. This module comes under Python's standard utility modules. Path classes in Pathlib module are divided into pure paths and concrete paths.
- **Sklearn:** Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms. It's built upon some of the technology you might already be familiar with, like NumPy, pandas, and Matplotlib.

- **Word cloud:** Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites.
- **IPython:** IPython provides a rich toolkit to help you make the most out of using Python interactively. Its main components are: A powerful interactive Python shell. A Jupyter kernel to work with Python code in Jupyter notebooks and other interactive frontends.
- **Plotly :** The Plotly Python library is an interactive, open-source plotting library that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases. Built on top of the Plotly JavaScript library.

## 6.2 SYSTEM MODULES:

### 1. Reading from CSV file:

We read the dataset of around 5, 00,000 customers that visited a supermarket. We read the file and explored it.

### 2. Pre-processing and Cleaning:

Pre-processing of data is done using NLP functions and cleaning of data undergoes tokenization, removal of stop words and html tags, stemming, deleting the null values and duplicate entries etc.

### 3. Categorizing Products and Visualization:

Products are categorized based on Customer Id and Invoice number followed by finding the keywords used in the description. The number of occurrences of each keyword is visualized.

### 4. Creating Product Clusters and Visualization:

The products are formed into clusters using the k-means clustering algorithm. The keywords considered in each cluster are visualized using word cloud.

### 5. Categorizing Customers:

We introduced new variables i.e., total price and basket price for further analysis usage. Clusters are formed using product clusters formed as basis. Grouping of the customers are done on the basis of amount spent in each product cluster and number of purchases made by the user.

### 6. Creating Customer Clusters and Visualization:

The customers are formed into clusters using the k-means clustering algorithm. The clusters formed are then visualized using Principal Component Analysis (PCA). Further these clusters are useful for analysis of customers.

### 7. Model training and Testing:

After the analysis is done, the data undergoes training and further testing is done using the remaining data using classification method. The classifiers we used are Logistic Regression, Random Forest, Decision tree along with precision of the model is detected.

## 6.3 SAMPLE SOURCE CODE:

### IMPORTING MODULES

```
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import datetime, nltk, warnings
import matplotlib.cm as cm
import itertools
from pathlib import Path
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples, silhouette_score
from sklearn import preprocessing, model_selection, metrics, feature_selection
from sklearn.model_selection import GridSearchCV, learning_curve
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
from sklearn import neighbors, linear_model, svm, tree, ensemble
from wordcloud import WordCloud, STOPWORDS
from sklearn.ensemble import AdaBoostClassifier
from sklearn.decomposition import PCA
from IPython.display import display, HTML
import plotly.graph_objs as go
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
warnings.filterwarnings("ignore")
plt.rcParams["patch.force_edgecolor"] = True
plt.style.use('fivethirtyeight')
mpl.rc('patch', edgecolor = 'dimgray', linewidth=1)
%matplotlib inline
```

### DATA EXPLORING

```
df_initial = pd.read_csv('data.csv', encoding="ISO-8859-1", dtype={'CustomerID': str, 'InvoiceID': str})
tab_info = pd.DataFrame(df_initial.dtypes).T.rename(index={0: 'column type'})
tab_info = tab_info.append(pd.DataFrame(df_initial.isnull().sum()).T.rename(index={0: 'null values (nb)'}))
tab_info = tab_info.append(pd.DataFrame(df_initial.isnull().sum()/df_initial.shape[0]*100).T.rename(index={0: 'null values (%)'}))
display(df_initial[:5])
pd.DataFrame([{'products': len(df_initial['StockCode'].value_counts()), 'transactions': len(df_initial['InvoiceNo'].value_counts()), 'customers': len(df_initial['CustomerID'].value_counts())}, ], columns = ['products', 'transactions', 'customers'], index = ['quantity'])
```

### DATA CLEANING:

```
df_initial['InvoiceDate'] = pd.to_datetime(df_initial['InvoiceDate'])
tab_info = pd.DataFrame(df_initial.dtypes).T.rename(index={0: 'column type'})
```

```

tab_info=tab_info.append(pd.DataFrame(df_initial.isnull().sum()).T.rename(index={0:'null values
(nb)'})))
tab_info=tab_info.append(pd.DataFrame(df_initial.isnull().sum()/df_initial.shape[0]*100).T.rename(in
dex={0:'null values (%)'})))
display(tab_info)
print('Dataframe dimensions of the dataset:', df_initial.shape)

df_initial.dropna(axis = 0, subset = ['CustomerID'], inplace = True)
print("After deleting null values of Customer ID:")
tab_info=pd.DataFrame(df_initial.dtypes).T.rename(index={0:'column type'})
tab_info=tab_info.append(pd.DataFrame(df_initial.isnull().sum()).T.rename(index={0:'null values
(nb)'})))
tab_info=tab_info.append(pd.DataFrame(df_initial.isnull().sum()/df_initial.shape[0]*100).T.rename(in
dex={0:'null values (%)'})))
display(tab_info)
print('Dataframe dimensions after deleting null values:', df_initial.shape)
print('Number of Duplicate entries: {}'.format(df_initial.duplicated().sum()))
df_initial.drop_duplicates(inplace = True)
print('After deleting : Duplicates are {}'.format(df_initial.duplicated().sum()))
df_initial.drop_duplicates(inplace = True)
temp = df_initial.groupby(by=['CustomerID', 'InvoiceNo', as_index=False]['InvoiceDate'].count()
no_products_per_basket = temp.rename(columns = {'InvoiceDate':'Number of products'})
no_products_per_basket[:10].sort_values('CustomerID')
no_products_per_basket['order_canceled'] = no_products_per_basket['InvoiceNo'].apply(lambda
x:int('C' in x))
display(no_products_per_basket[:5])
n1 = no_products_per_basket['order_canceled'].sum()
n2 = no_products_per_basket.shape[0]
print('Number of orders canceled: {}/{} ({:.2f}%)'.format(n1, n2, n1/n2*100))
df_cleaned = df_initial.copy(deep = True)
df_cleaned['QuantityCanceled'] = 0
entry_to_remove = [] ; doubtful_entry = []
for index, col in df_initial.iterrows():
    if (col['Quantity'] > 0) or col['Description'] == 'Discount': continue
    df_test = df_initial[(df_initial['CustomerID'] == col['CustomerID']) &
                        (df_initial['StockCode'] == col['StockCode']) &
                        (df_initial['InvoiceDate'] < col['InvoiceDate']) &
                        (df_initial['Quantity'] > 0)].copy()
    if (df_test.shape[0] == 0):
        doubtful_entry.append(index)
    elif (df_test.shape[0] == 1):
        index_order = df_test.index[0]
        df_cleaned.loc[index_order, 'QuantityCanceled'] = -col['Quantity']
        entry_to_remove.append(index)
    elif (df_test.shape[0] > 1):
        df_test.sort_index(axis=0, ascending=False, inplace = True)
        for ind, val in df_test.iterrows():
            if val['Quantity'] < -col['Quantity']: continue
            df_cleaned.loc[ind, 'QuantityCanceled'] = -col['Quantity']
            entry_to_remove.append(index)
            break
df_cleaned.drop(entry_to_remove, axis = 0, inplace = True)

```

```

df_cleaned.drop(doubtfull_entry, axis = 0, inplace = True)
remaining_entries = df_cleaned[(df_cleaned['Quantity'] < 0) & (df_cleaned['StockCode'] != 'D')]
print("Number of Cancelled orders: {}".format(remaining_entries.shape[0]))
remaining_entries[:5]
pd.DataFrame([{'products':len(df_initial['StockCode'].value_counts()),'transactions':len(df_initial['InvoiceNo'].value_counts()),'customers': len(df_initial['CustomerID'].value_counts())}, columns = ['products', 'transactions', 'customers'], index = ['quantity'])

```

## PRODUCT CATEGORIES

```

print("\033[1m+'"\n\nNumber of products purchased in every transaction:")
temp = df_initial.groupby(by=['CustomerID', 'InvoiceNo'], as_index=False)['InvoiceDate'].count()
no_products_per_basket = temp.rename(columns = {'InvoiceDate':'Number of products'})
no_products_per_basket[:10].sort_values('CustomerID')
df_products = pd.DataFrame(df_initial['Description'].unique()).rename(columns = {0:'Description'})
print(df_products)
is_noun = lambda pos: pos[:2] == 'NN'
def keywords_inventory(dataframe, colonne = 'Description'):
    stemmer = nltk.stem.SnowballStemmer("english")
    keywords_roots = dict() # collect the words / root
    keywords_select = dict() # association: root <-> keyword
    category_keys = []
    count_keywords = dict()
    icount = 0
    for s in dataframe[colonne]:
        if pd.isnull(s): continue
        lines = s.lower()
        tokenized = nltk.word_tokenize(lines)
        nouns = [word for (word, pos) in nltk.pos_tag(tokenized) if is_noun(pos)]
        for t in nouns:
            t = t.lower() ; racine = stemmer.stem(t)
            if racine in keywords_roots:
                keywords_roots[racine].add(t)
                count_keywords[racine] += 1
            else:
                keywords_roots[racine] = {t}
                count_keywords[racine] = 1
    for s in keywords_roots.keys():
        if len(keywords_roots[s]) > 1:
            min_length = 1000
            for k in keywords_roots[s]:
                if len(k) < min_length:
                    clef = k ; min_length = len(k)
            category_keys.append(clef)
            keywords_select[s] = clef
        else:
            category_keys.append(list(keywords_roots[s])[0])
            keywords_select[s] = list(keywords_roots[s])[0]
    print("No of keywords in variable '{}': {}".format(colonne,len(category_keys)))
    return category_keys, keywords_roots, keywords_select, count_keywords
list_products = []
for k,v in count_keywords.items():

```



```

list_products.append([keywords_select[k],v])
list_products.sort(key = lambda x:x[1], reverse = True)
liste = sorted(list_products, key = lambda x:x[1], reverse = True)
plt.rc('font', weight='normal')
fig, ax = plt.subplots(figsize=(7, 25))
y_axis = [i[1] for i in liste[:125]]
x_axis = [k for k,i in enumerate(liste[:125])]
x_label = [i[0] for i in liste[:125]]
plt.xticks(fontsize = 15)
plt.yticks(fontsize = 13)
plt.yticks(x_axis, x_label)
plt.xlabel("No. of occurences", fontsize = 18, labelpad = 10)
ax.barh(x_axis, y_axis, align = 'center')
ax = plt.gca()
ax.invert_yaxis()
plt.title("Words occurence",bbox={ 'facecolor':'k', 'pad':5}, color='w',fontsize = 25)
plt.show()
list_products = []
for k,v in count_keywords.items():
    word = keywords_select[k]
    if word in ['pink', 'blue', 'tag', 'green', 'orange']: continue
    if len(word) < 3 or v < 13: continue
    if ('+' in word) or ('/' in word): continue
    list_products.append([word, v])
list_products.sort(key = lambda x:x[1], reverse = True)
print('Most Repeated Keywords:', len(list_products))
liste_produits = df_initial['Description'].unique()
X = pd.DataFrame()
for key, occurrence in list_products:
    X.loc[:, key] = list(map(lambda x:int(key.upper() in x), liste_produits))
threshold = [0, 1, 2, 3, 5, 10]
label_col = []
for i in range(len(threshold)):
    if i == len(threshold)-1:
        col = '>{ }'.format(threshold[i])
    else:
        col = '{ }<.<{ }'.format(threshold[i],threshold[i+1])
    label_col.append(col)
    X.loc[:, col] = 0
for i, prod in enumerate(liste_produits):
    prix = df_initial[ df_initial['Description'] == prod]['UnitPrice'].mean()
    j = 0
    while prix > threshold[j]:
        j+=1
    if j == len(threshold): break
    X.loc[i, label_col[j-1]] = 1
print("{:<8} {:<20} \n".format('Range', 'no of products') + 20*'-')
for i in range(len(threshold)):
    if i == len(threshold)-1:
        col = '>{ }'.format(threshold[i])
    else:
        col = '{ }<.<{ }'.format(threshold[i],threshold[i+1])

```

```
print("{:<10} {:<20}".format(col, X.loc[:, col].sum()))
```

## CREATING PRODUCT CLUSTERS

```
matrix = X.to_numpy()
for n_clusters in range(3,10):
    kmeans = KMeans(init='k-means++', n_clusters = n_clusters, n_init=30)
    kmeans.fit(matrix)
    clusters = kmeans.predict(matrix)
    silhouette_avg = silhouette_score(matrix, clusters)
    print("For n_clusters =", n_clusters, "The average silhouette_score is :", silhouette_avg)
n_clusters = 5
silhouette_avg = -1
while silhouette_avg < 0.145:
    kmeans = KMeans(init='k-means++', n_clusters = n_clusters, n_init=30)
    kmeans.fit(matrix)
    clusters = kmeans.predict(matrix)
    silhouette_avg = silhouette_score(matrix, clusters)
print("For n_clusters =", n_clusters, "The average silhouette_score is :", silhouette_avg)
pd.Series(clusters).value_counts()
```

## WORD CLOUD VISUALIZATION OF PRODUCT CLUSTERS

```
liste = pd.DataFrame(liste_produits)
liste_words = [word for (word, occurence) in list_products]
occurence = [dict() for _ in range(n_clusters)]
for i in range(n_clusters):
    liste_cluster = liste.loc[clusters == i]
    for word in liste_words:
        if word in ['art', 'set', 'heart', 'pink', 'blue', 'tag']: continue
        occurence[i][word] = sum(liste_cluster.loc[:, 0].str.contains(word.upper()))
def random_color_func(word=None, font_size=None, position=None,
                       orientation=None, font_path=None, random_state=None):
    h = int(360.0 * tone / 255.0)
    s = int(100.0 * 255.0 / 255.0)
    l = int(100.0 * float(random_state.randint(70, 120)) / 255.0)
    return "hsl({}, {}, {}%)".format(h, s, l)
def make_wordcloud(liste, increment):
    ax1 = fig.add_subplot(4,2,increment)
    words = dict()
    trunc_occurrences = liste[0:150]
    for s in trunc_occurrences:
        words[s[0]] = s[1]
    wordcloud = WordCloud(width=1000,height=400, background_color='lightgrey',
                          max_words=1628,relative_scaling=1,
                          color_func = random_color_func,
                          normalize_plurals=False)
    wordcloud.generate_from_frequencies(words)
    ax1.imshow(wordcloud, interpolation="bilinear")
    ax1.axis('off')
    plt.title('cluster n°{}'.format(increment-1))
fig = plt.figure(1, figsize=(14,14))
```

```

color = [0, 160, 130, 95, 280, 40, 330, 110, 25]
for i in range(n_clusters):
    list_cluster_occurences = occurrence[i]
    tone = color[i] # define the color of the words
    liste = []
    for key, value in list_cluster_occurences.items():
        liste.append([key, value])
    liste.sort(key = lambda x:x[1], reverse = True)
    make_wordcloud(liste, i+1)

```

## CUSTOMER CATEGORIES

```

df_cleaned['TotalPrice'] = df_cleaned['UnitPrice'] * (df_cleaned['Quantity']
df_cleaned['QuantityCanceled'])
df_cleaned.sort_values('CustomerID')[5]
temp = df_cleaned.groupby(by=['CustomerID', 'InvoiceNo'], as_index=False)['TotalPrice'].sum()
basket_price = temp.rename(columns = {'TotalPrice':'Basket Price'})
# date de la commande
df_cleaned['InvoiceDate_int'] = df_cleaned['InvoiceDate'].astype('int64')
temp = df_cleaned.groupby(by=['CustomerID', 'InvoiceNo'],
    as_index=False)['InvoiceDate_int'].mean()
df_cleaned.drop('InvoiceDate_int', axis = 1, inplace = True)
basket_price.loc[:, 'InvoiceDate'] = pd.to_datetime(temp['InvoiceDate_int'])
basket_price = basket_price[basket_price['Basket Price'] > 0]
basket_price.sort_values('CustomerID')[6]
corresp = dict()
for key, val in zip (liste_produits, clusters):
    corresp[key] = val
df_cleaned['cluster_product'] = df_cleaned.loc[:, 'Description'].map(corresp)
for i in range(5):
    col = 'cluster_{ }'.format(i)
    df_temp = df_cleaned[df_cleaned['cluster_product'] == i]
    price_temp = df_temp['UnitPrice'] * (df_temp['Quantity'] - df_temp['QuantityCanceled'])
    price_temp = price_temp.apply(lambda x:x if x > 0 else 0)
    df_cleaned.loc[:, col] = price_temp
    df_cleaned[col].fillna(0, inplace = True)
df_cleaned[['InvoiceNo', 'Description', 'cluster_product', 'cluster_0', 'cluster_1', 'cluster_2',
    'cluster_3', 'cluster_4']][5]
temp = df_cleaned.groupby(by=['CustomerID', 'InvoiceNo'], as_index=False)['TotalPrice'].sum()
basket_price = temp.rename(columns = {'TotalPrice':'Basket Price'})
for i in range(5):
    col = 'cluster_{ }'.format(i)
    temp = df_cleaned.groupby(by=['CustomerID', 'InvoiceNo'], as_index=False)[col].sum()
    basket_price.loc[:,col]=temp[col]
df_cleaned['InvoiceDate_int'] = df_cleaned['InvoiceDate'].astype('int64')
temp = df_cleaned.groupby(by=['CustomerID', 'InvoiceNo'],
    as_index=False)['InvoiceDate_int'].mean()
df_cleaned.drop('InvoiceDate_int', axis = 1, inplace = True)
basket_price.loc[:, 'InvoiceDate'] = pd.to_datetime(temp['InvoiceDate_int'])
print("\033[1m'+""\n\n1.)Group formed based on Amount spent in each product cluster:")
basket_price = basket_price[basket_price['Basket Price'] > 0]
basket_price.sort_values('CustomerID', ascending = True)[5]

```

```

print("\033[1m+' '\n\n2.)Group formed based on Number of purchases made by the user:")
transactions_per_user=basket_price.groupby(by=['CustomerID'])['Basket
    Price'].agg(['count','min','max','mean','sum'])
for i in range(5):
    col = 'cluster_{ }'.format(i)
    transactions_per_user.loc[:,col] = basket_price.groupby(by=['CustomerID'])[col].sum() /\
        transactions_per_user['sum']*100
transactions_per_user.reset_index(drop = False, inplace = True)
basket_price.groupby(by=['CustomerID'])['cluster_0'].sum()
transactions_per_user.sort_values('CustomerID', ascending = True)[:5]
n1 = transactions_per_user[transactions_per_user['count'] == 1].shape[0]
n2 = transactions_per_user.shape[0]
print("Number of customers which are unique: {:<2}/{:<5} ({:<2.2f}%)".format(n1,n2,n1/n2*100))
list_cols = ['count','min','max','mean','cluster_0','cluster_1','cluster_2','cluster_3','cluster_4']
selected_customers = transactions_per_user.copy(deep = True)
matrix = selected_customers[list_cols].to_numpy()
scaler = StandardScaler()
scaler.fit(matrix)
print('variables mean values: \n' + 90*'- ' + '\n' , scaler.mean_)
scaled_matrix = scaler.transform(matrix)
n_clusters = 11
kmeans = KMeans(init='k-means++', n_clusters = n_clusters, n_init=100)
kmeans.fit(scaled_matrix)
clusters_clients = kmeans.predict(scaled_matrix)
silhouette_avg = silhouette_score(scaled_matrix, clusters_clients)
print('score of silhouette: {:<.3f}'.format(silhouette_avg))
pd.DataFrame(pd.Series(clusters_clients).value_counts(), columns = ['number of customers']).T

```

## PCA VISUALIZATION OF CUSTOMER CLUSTERS

```

pca = PCA(n_components=6)
matrix_3D = pca.fit_transform(scaled_matrix)
mat = pd.DataFrame(matrix_3D)
mat['cluster'] = pd.Series(clusters_clients)
import matplotlib.patches as mpatches
sns.set_style("white")
sns.set_context("notebook", font_scale=1, rc={"lines.linewidth": 2.5})LABEL_COLOR_MAP =
    {0:'r', 1:'tan', 2:'b', 3:'k', 4:'c', 5:'g', 6:'deeppink', 7:'skyblue', 8:'darkcyan', 9:'orange',10:'yellow',
    11:'tomato', 12:'seagreen'}
label_color = [LABEL_COLOR_MAP[l] for l in mat['cluster']]
fig = plt.figure(figsize = (12,10))
increment = 0
for ix in range(6):
    for iy in range(ix+1, 6):
        increment += 1
        ax = fig.add_subplot(4,3,increment)
        ax.scatter(mat[ix], mat[iy], c= label_color, alpha=0.5)
        plt.ylabel('PCA { }'.format(iy+1), fontsize = 12)
        plt.xlabel('PCA { }'.format(ix+1), fontsize = 12)
        ax.yaxis.grid(color='lightgray', linestyle=':')

```

```

    ax.xaxis.grid(color='lightgray', linestyle=':')
    ax.spines['right'].set_visible(False)
    ax.spines['top'].set_visible(False)
    if increment == 12: break
    if increment == 12: break
comp_handler = []
for i in range(n_clusters):
    comp_handler.append(mpatches.Patch(color = LABEL_COLOR_MAP[i], label = i))
plt.legend(handles=comp_handler, bbox_to_anchor=(1.1, 0.9),
           title='Cluster', facecolor = 'lightgrey',
           shadow = True, frameon = True, framealpha = 1,
           fontsize = 13, bbox_transform = plt.gcf().transFigure)
plt.tight_layout()
selected_customers.loc[:, 'cluster'] = clusters_clients
merged_df = pd.DataFrame()
for i in range(n_clusters):
    test = pd.DataFrame(selected_customers[selected_customers['cluster'] == i].mean())
    test = test.T.set_index('cluster', drop = True)
    test['size'] = selected_customers[selected_customers['cluster'] == i].shape[0]
    merged_df = pd.concat([merged_df, test])
merged_df.drop('CustomerID', axis = 1, inplace = True)
print('number of customers:', merged_df['size'].sum())
merged_df = merged_df.sort_values('sum')
liste_index = []
for i in range(5):
    column = 'cluster_{ }'.format(i)
    liste_index.append(merged_df[merged_df[column] > 45].index.values[0])
liste_index_reordered = liste_index
liste_index_reordered += [ s for s in merged_df.index if s not in liste_index]
merged_df = merged_df.reindex(index = liste_index_reordered)
merged_df = merged_df.reset_index(drop = False)
display(merged_df[['cluster', 'count', 'min', 'max', 'mean', 'sum', 'cluster_0', 'cluster_1', 'cluster_2',
                    'cluster_3', 'cluster_4', 'size']])

```

## CLASSIFICATION OF CUSTOMERS

```

class Class_Fit(object):
    def __init__(self, clf, params=None):
        if params:
            self.clf = clf(**params)
        else:
            self.clf = clf()
    def train(self, x_train, y_train):
        self.clf.fit(x_train, y_train)
    def predict(self, x):
        return self.clf.predict(x)
    def grid_search(self, parameters, Kfold):
        self.grid = GridSearchCV(estimator = self.clf, param_grid = parameters, cv = Kfold)
    def grid_fit(self, X, Y):
        self.grid.fit(X, Y)
    def grid_predict(self, X, Y):
        self.predictions = self.grid.predict(X)

```

```

print("Precision: {:.2f} % ".format(100*metrics.accuracy_score(Y, self.predictions)))
columns = ['mean', 'cluster_0', 'cluster_1', 'cluster_2', 'cluster_3', 'cluster_4']
X = selected_customers[columns]
Y = selected_customers['cluster']
X_train, X_test, Y_train, Y_test = model_selection.train_test_split(X, Y, train_size = 0.8)
lr = Class_Fit(clf = linear_model.LogisticRegression)
lr.grid_search(parameters = [{ 'C':np.logspace(-2,2,20)}], Kfold = 5)
lr.grid_fit(X = X_train, Y = Y_train)
print("Logistic Regression")
lr.grid_predict(X_test, Y_test)
tr = Class_Fit(clf = tree.DecisionTreeClassifier)
tr.grid_search(parameters = [{ 'criterion' : ['entropy', 'gini'], 'max_features' :['sqrt', 'log2']}], Kfold = 5)
tr.grid_fit(X = X_train, Y = Y_train)
print("Decision Tree:")
tr.grid_predict(X_test, Y_test)
rf = Class_Fit(clf = ensemble.RandomForestClassifier)
param_grid = { 'criterion' : ['entropy', 'gini'], 'n_estimators' : [20, 40, 60, 80, 100],
               'max_features' :['sqrt', 'log2']}
rf.grid_search(parameters = param_grid, Kfold = 5)
rf.grid_fit(X = X_train, Y = Y_train)
print("Random Forest Classifier")
rf.grid_predict(X_test, Y_test)
rf_best = ensemble.RandomForestClassifier(**rf.grid.best_params_)
tr_best = tree.DecisionTreeClassifier(**tr.grid.best_params_)
lr_best = linear_model.LogisticRegression(**lr.grid.best_params_)
votingC = ensemble.VotingClassifier(estimators=[('rf', rf_best),('tr', tr_best),('lr', lr_best)],
                                   voting='soft')
votingC = votingC.fit(X_train, Y_train)
predictions = votingC.predict(X)
print("Precision: {:.2f} % ".format(100*metrics.accuracy_score(Y, predictions)))

```

## **7. TESTING & MAINTENANCE**

### **7.1 INTRODUCTION:**

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test have the following

- Meets the requirements that guided its design and development
- Responds correctly to all kinds of inputs
- Performs its functions within an acceptable time
- Can be installed and run in its intended environments

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding software bugs (errors or other defects).

Software testing can provide objective, independent information about the quality of software and risk of its failure to users and/or sponsors.

Software testing can be conducted as soon as executable software (even if partially complete) exists. The overall approach to software development often determines when and how testing is conducted. For example, in a phased process, most testing occurs after system requirements have been defined and then implemented in testable.

#### **Types of Testing**

A software engineering product can be tested in one of two ways:

- Black box testing
- White box testing

#### **Black box testing**

Knowing the specified function that a product has been designed to perform, determine whether each function is fully operational.

#### **White box testing**

Knowing the internal workings of a software product determine whether the internal operation implementing the functions perform according to the specification, and all the internal components have been adequately exercised.

## Testing Strategies

Four Testing Strategies that are often adopted by the software development team include:

- Unit Testing
- Integration Testing
- Validation Testing
- System Testing

### 7.1.1 Unit Testing

We adopt white box testing when using this testing technique. This testing was carried out on individual cells of the jupyter notebook that were designed. Each individual module was tested using this technique during the coding phase. Every component was checked to make sure that they adhere strictly to the specifications spelt out in the data flow diagram and ensure that they perform the purpose intended for them.

All the names of the variables are scrutinized to make sure that they are truly reflected of the element they represent. All the looping mechanisms were verified to ensure that they were as decided. Beside these, we trace through the code manually to capture syntax errors and logical errors.

### 7.1.2 Integration Testing

After finishing the Unit Testing process, next is the integration testing process. Since we are using interpreter, Integration testing is done by top down approach to make sure that the flow is as expected.

The Black box testing technique was employed here. The interactivity and flow among different cells were tested first. This allowed identifying any wrong linkages or parameters passing early in the development process as it just can be passed in a set of data and checked if the result returned is an accepted one.

### 7.1.3 Validation Testing

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input from the dataset, corresponding error messages are displayed.

### 7.1.4 System Testing

System testing is a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all the work should verify that all system elements have been properly integrated and perform allocated functions. System testing also ensures that the project works well in the environment. It traps the errors and allows convenient processing of errors without coming out of the program abruptly.



Recovery testing is done in such a way that failure is forced to a software system and checked whether the recovery is proper and accurate. The performance of the system is highly effective.

Software testing is critical element of software quality assurance and represents ultimate review of specification, design and coding.

## 7.2 USER TRAINING:

Since our system will be mainly having data and business analysts as end users, Naïve user training part can be reduced. End users should be aware of Test dataset and Train dataset compositions and all deliverables. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

## 7.3 MAINTENANCE:

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, this system has been developed to satisfy the needs to the largest possible extent. And as part of maintenance, most of the features and expected outcomes of end users are taken into account while developing this system. With development in cluster forming technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

## 7.4 TEST CASES

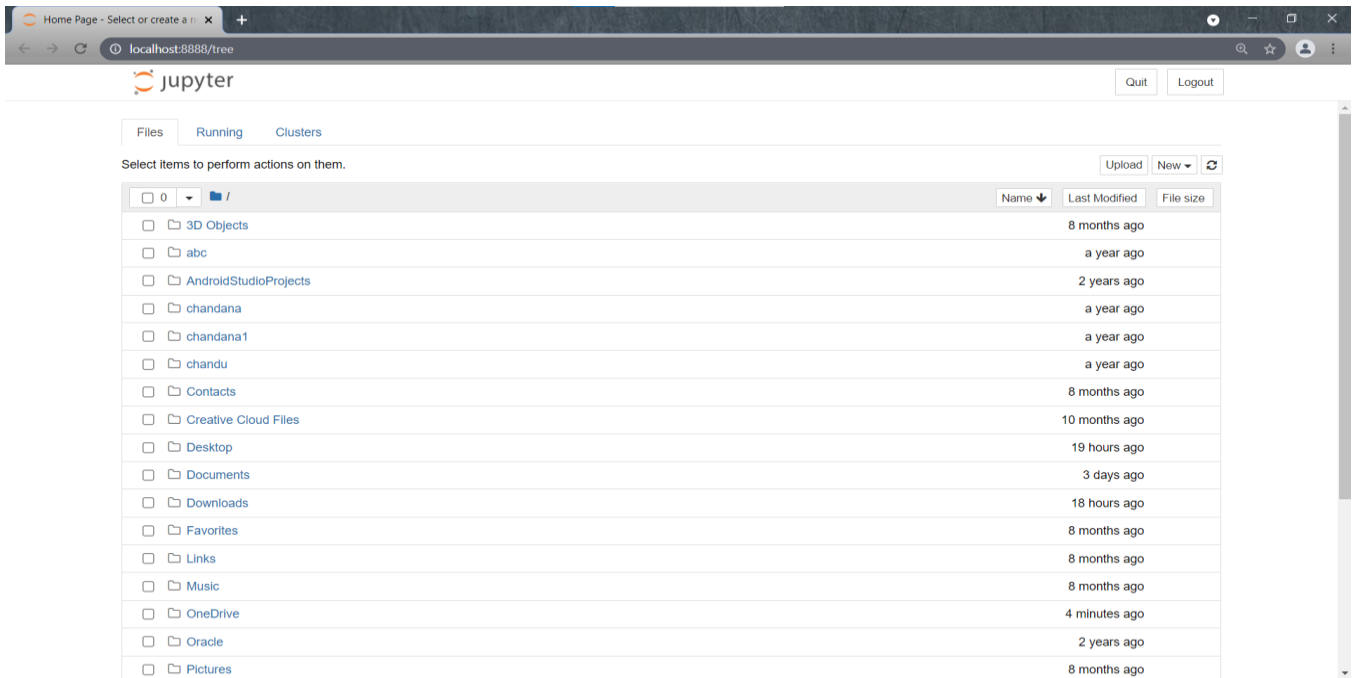
Test case design focuses on a set of technique for the creation of test cases that meet overall testing objectives. Planning and testing of a programming system involve formulating a set of test cases, which are similar to the real data that the system is intended to manipulate.

*Table 7.3.1 Test Cases Representation*

S.No	Test Case	Description & Expected behaviour	Input	Output	Result (Pass/ Fail)
1	Run the code in Jupyter Notebook without having dataset	Data verification & Validation testing, since dataset is not present it should throw error.	Delete the downloaded dataset on testing machine & Run the Jupyter notebook cells	Error is thrown	Pass
2	Keep null values and repeated values in the data set	As per the code design, null values should not be considered & duplicate values are considered	Dataset with null and duplicate values	Duplicate values are delete and null and cancelled order details are displayed	Pass

		only once			
3	Output columns verification testing	Run Jupyter cell to display customer order details accurately	Dataset with various customer order details	Order details are displayed under data exploration	Pass
4	Visualization testing under output verification	Visualization of occurrences of words and word cloud visualization should be interpreted properly	Dataset with customer order details having key words for visualization	Bar graph for count of occurrences and cloud clusters for word cloud visualization	Pass
5	Add new column to the customer data set or to sliced customer dataset.	Adding any new column to the input dataset to check the Integrity & acceptance	Original dataset with customer order details	New column "Total Price" is added to the dataset	Pass
6	Verify cluster formation	Classification testing with the provided algorithms- As per the design, customers should be clustered based on products and grouped based on mentioned criteria like number of purchases and products	Dataset with different values and appropriate algorithms for classification and cluster formation	Clusters are formed after classification according to the code design & silhouette score is displayed for clusters and unique customers, Cluster formed is observed with PCA Visualization	Pass
7	Accuracy and Performance testing for the final outcome of the project	Precisions should be measured for different classifiers and regression methods used in code. Expected precision is at least 96%	Different slices of dataset of customer order details	Precision values are displayed for Logistic regression, Decision Tree, Random Forest Classifier and overall Precision for all slices is recorded as around 97%.	Pass

## 8. OUTPUT SCREENS



**Fig. 8.1 Opening Jupyter Notebook**

The screenshot shows a Microsoft Excel spreadsheet with the following data:

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	Customer	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12-01-2010 08:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	12-01-2010 08:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12-01-2010 08:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12-01-2010 08:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12-01-2010 08:26	3.39	17850	United Kingdom
536365	22752	SET 7 BABUSHKA NESTING BOXES	2	12-01-2010 08:26	7.65	17850	United Kingdom
536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	12-01-2010 08:26	4.25	17850	United Kingdom
536366	22633	HAND WARMER UNION JACK	6	12-01-2010 08:28	1.85	17850	United Kingdom
536366	22632	HAND WARMER RED POLKA DOT	6	12-01-2010 08:28	1.85	17850	United Kingdom
536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	12-01-2010 08:34	1.69	13047	United Kingdom
536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	12-01-2010 08:34	2.1	13047	United Kingdom
536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	12-01-2010 08:34	2.1	13047	United Kingdom
536367	22749	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	12-01-2010 08:34	3.75	13047	United Kingdom
536367	22310	IVORY KNITTED MUG COSY	6	12-01-2010 08:34	1.65	13047	United Kingdom
536367	84969	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	12-01-2010 08:34	4.25	13047	United Kingdom
536367	22623	BOX OF VINTAGE JIGSAW BLOCKS	3	12-01-2010 08:34	4.95	13047	United Kingdom
536367	22622	BOX OF VINTAGE ALPHABET BLOCKS	2	12-01-2010 08:34	9.95	13047	United Kingdom
536367	21754	HOME BUILDING BLOCK WORD	3	12-01-2010 08:34	5.95	13047	United Kingdom
536367	21755	LOVE BUILDING BLOCK WORD	3	12-01-2010 08:34	5.95	13047	United Kingdom
536367	21777	RECIPE BOX WITH METAL HEART	4	12-01-2010 08:34	7.95	13047	United Kingdom
536367	48187	DOORMAT NEW ENGLAND	4	12-01-2010 08:34	7.95	13047	United Kingdom
536368	22960	JAM MAKING SET WITH JARS	6	12-01-2010 08:34	4.25	13047	United Kingdom
536368	22913	RED COAT RACK PARIS FASHION	3	12-01-2010 08:34	4.95	13047	United Kingdom
536368	22912	YELLOW COAT RACK PARIS FASHION	3	12-01-2010 08:34	4.95	13047	United Kingdom
536368	22914	BLUE COAT RACK PARIS FASHION	3	12-01-2010 08:34	4.95	13047	United Kingdom
536369	21756	BATH BUILDING BLOCK WORD	3	12-01-2010 08:35	5.95	13047	United Kingdom
536370	22728	ALARM CLOCK BAKELIKE PINK	24	12-01-2010 08:45	3.75	12583	France
536370	22727	ALARM CLOCK BAKELIKE RED	24	12-01-2010 08:45	3.75	12583	France
536370	22726	ALARM CLOCK BAKELIKE GREEN	12	12-01-2010 08:45	3.75	12583	France
536370	21724	PANDA AND BUNNIES STICKER SHEET	12	12-01-2010 08:45	0.85	12583	France
536370	21893	STARS GIFT TUBE	24	12-01-2010 08:45	0.65	12583	France

**Fig. 8.2 Dataset used to train and test**

## DATA EXPLORING:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850	United Kingdom

*Fig. 8.3 Data Exploring*

## DATA CLEANING:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
column type	object	object	object	int64	datetime64[ns]	float64	object	object
null values (nb)	0	0	1454	0	0	0	135080	0
null values (%)	0.0	0.0	0.268311	0.0	0.0	0.0	24.926694	0.0

Dataframe dimensions of the dataset: (541909, 8)

After deleting null values of Customer ID:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
column type	object	object	object	int64	datetime64[ns]	float64	object	object
null values (nb)	0	0	0	0	0	0	0	0
null values (%)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Dataframe dimensions after deleting null values: (406829, 8)

Out[9]:

	CustomerID	InvoiceNo	Number of products
0	12346	541431	1
1	12346	C541433	1
2	12347	537626	31
3	12347	542237	29
4	12347	549222	24
5	12347	556201	18
6	12347	562032	22
7	12347	573511	47
8	12347	581180	11
9	12348	539318	17

	CustomerID	InvoiceNo	Number of products	order_canceled
0	12346	541431	1	0
1	12346	C541433	1	1
2	12347	537626	31	0
3	12347	542237	29	0
4	12347	549222	24	0

Number of orders canceled: 3654/22190 (16.47%)

*Fig . 8.4 Data Cleaning*

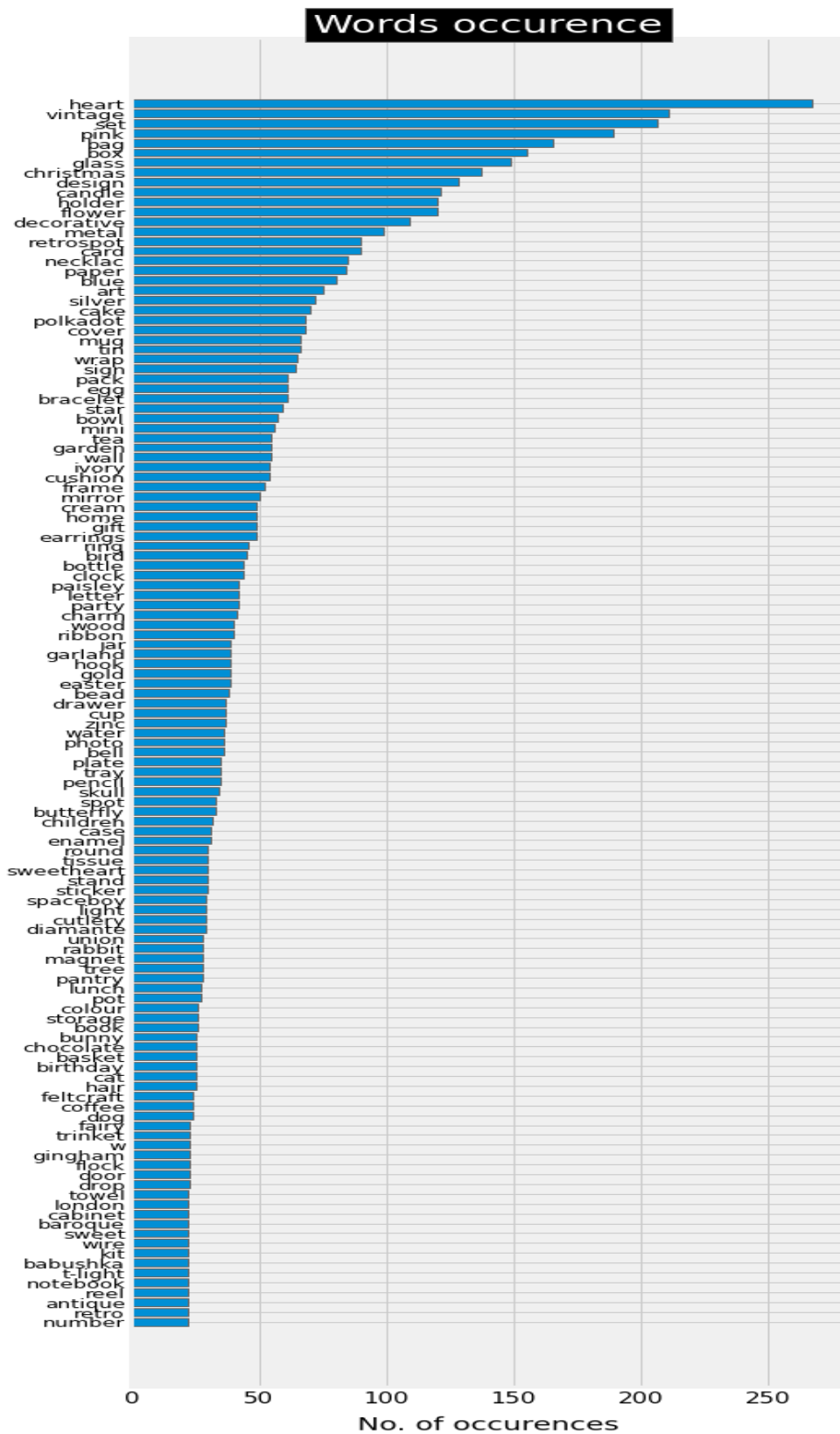
## CATEGORIZING PRODUCTS:

Number of products purchased in every transaction:

Out[14]:

	CustomerID	InvoiceNo	Number of products
0	12346	541431	1
1	12346	C541433	1
2	12347	537626	31
3	12347	542237	29
4	12347	549222	24
5	12347	556201	18
6	12347	562032	22
7	12347	573511	47
8	12347	581180	11
9	12348	539318	17

*Fig .8.5 Displaying the number of products for each transaction till 10 customers*



*Fig .8.6 Visualisation of how many times each keyword is used in description*

```
For n_clusters = 3 The average silhouette_score is : 0.10062159302826501
For n_clusters = 4 The average silhouette_score is : 0.12601189170579502
For n_clusters = 5 The average silhouette_score is : 0.14709895533471118
For n_clusters = 6 The average silhouette_score is : 0.14546205913385046
For n_clusters = 7 The average silhouette_score is : 0.14838651302457997
For n_clusters = 8 The average silhouette_score is : 0.13593683497091483
For n_clusters = 9 The average silhouette_score is : 0.1248652851081635
```

*Fig.8.7 Displaying the silhouette score for clusters in range [3,10]*



**Fig 8.8 Word cloud visualization for product clusters**



## CATEGORIZING CUSTOMERS:

Out[28]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	QuantityCanceled	TotalPrice
61619	541431	23166	MEDIUM CERAMIC TOP STORAGE JAR	74215	2011-01-18 10:01:00	1.04	12346	United Kingdom	74215	0.0
148288	549222	22375	AIRLINE BAG VINTAGE JET SET BROWN	4	2011-04-07 10:43:00	4.25	12347	Iceland	0	17.0
428971	573511	22698	PINK REGENCY TEACUP AND SAUCER	12	2011-10-31 12:25:00	2.95	12347	Iceland	0	35.4
428970	573511	47559B	TEA TIME OVEN GLOVE	10	2011-10-31 12:25:00	1.25	12347	Iceland	0	12.5
428969	573511	47567B	TEA TIME KITCHEN APRON	6	2011-10-31 12:25:00	5.95	12347	Iceland	0	35.7

*Adding new variable total price to the details of 5 customers*

Out[29]:

	CustomerID	InvoiceNo	Basket Price	InvoiceDate
1	12347	537626	711.79	2010-12-07 14:57:00.000001024
2	12347	542237	475.39	2011-01-26 14:29:59.999999744
3	12347	549222	636.25	2011-04-07 10:42:59.999999232
4	12347	556201	382.52	2011-06-09 13:01:00.000000256
5	12347	562032	584.91	2011-08-02 08:48:00.000000000
6	12347	573511	1294.32	2011-10-31 12:25:00.000001280

*Displaying invoice details along with basket price*

Out[31]:

	InvoiceNo	Description	cluster_product	cluster_0	cluster_1	cluster_2	cluster_3	cluster_4
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	4	0.0	0.00	0.0	0.0	15.3
1	536365	WHITE METAL LANTERN	1	0.0	20.34	0.0	0.0	0.0
2	536365	CREAM CUPID HEARTS COAT HANGER	1	0.0	22.00	0.0	0.0	0.0
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	1	0.0	20.34	0.0	0.0	0.0
4	536365	RED WOOLLY HOTTIE WHITE HEART.	1	0.0	20.34	0.0	0.0	0.0

*Fig .8.9 Creating customer clusters based on product clusters*

Out[32]:

	CustomerID	InvoiceNo	Basket Price	cluster_0	cluster_1	cluster_2	cluster_3	cluster_4	InvoiceDate
1	12347	537626	711.79	187.2	293.35	124.44	23.40	83.40	2010-12-07 14:57:00.000001024
2	12347	542237	475.39	130.5	169.20	38.25	84.34	53.10	2011-01-26 14:29:59.999999744
3	12347	549222	636.25	330.9	115.00	38.25	81.00	71.10	2011-04-07 10:42:59.999999232
4	12347	556201	382.52	74.4	168.76	19.90	41.40	78.06	2011-06-09 13:01:00.000000256
5	12347	562032	584.91	109.7	158.16	136.05	61.30	119.70	2011-08-02 08:48:00.000000000

*Fig .8.10 Grouping based on amount spent in each product cluster*



Out[33]:

	CustomerID	count	min	max	mean	sum	cluster_0	cluster_1	cluster_2	cluster_3	cluster_4
0	12347	7	224.82	1294.32	615.714286	4310.00	26.375870	29.540371	12.041531	11.237123	20.805104
1	12348	4	227.44	892.80	449.310000	1797.24	41.953217	0.000000	20.030714	38.016069	0.000000
2	12349	1	1757.55	1757.55	1757.550000	1757.55	26.506216	10.713778	46.021450	4.513101	12.245455
3	12350	1	334.40	334.40	334.400000	334.40	48.444976	0.000000	11.961722	11.692584	27.900718
4	12352	7	144.35	840.30	340.815714	2385.71	15.705178	14.601523	64.322571	1.299404	4.071325

*Fig .8.11 Grouping based on number of purchases made by the user*

## CREATING CUSTOMER CLUSTERS USING K-MEANS:

variables mean values:

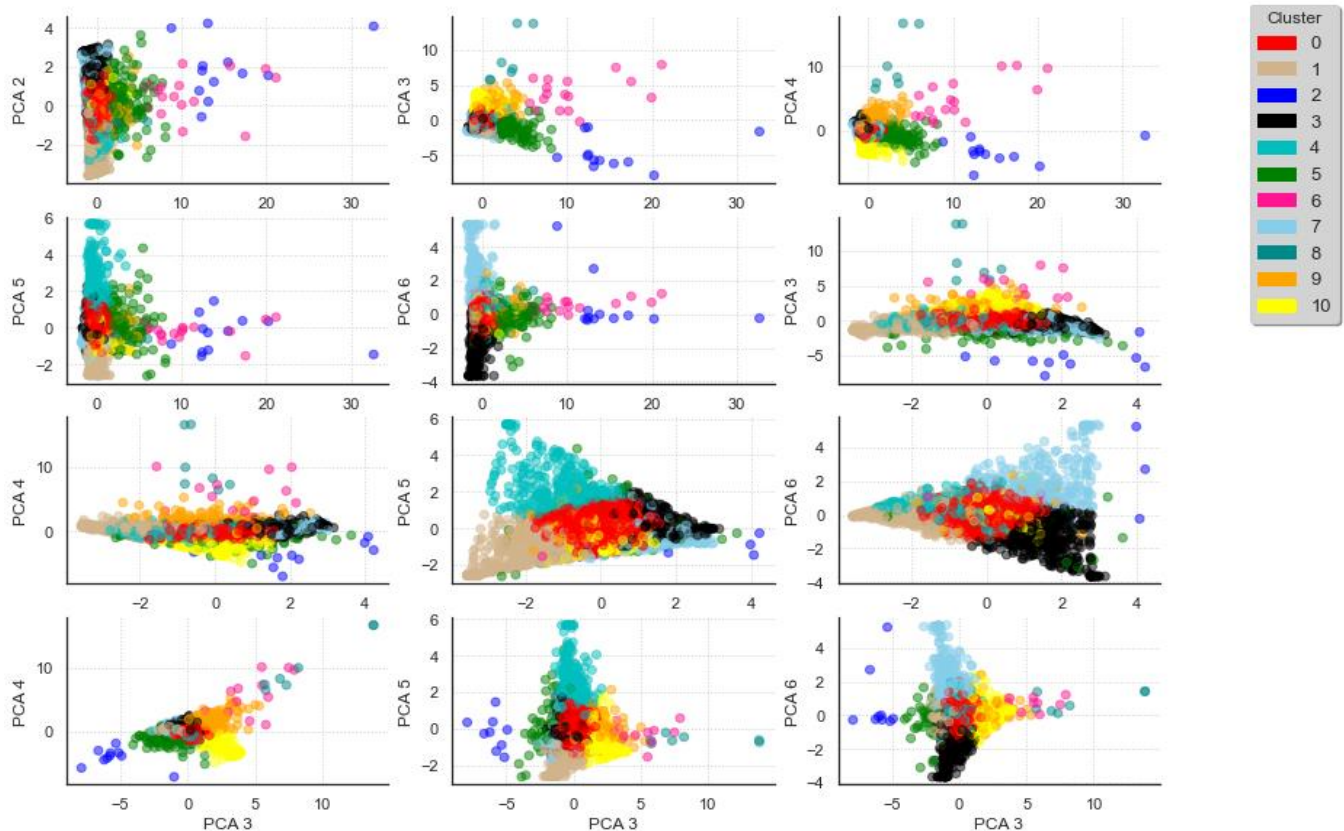
```
[ 4.25190663 241.38253571 578.87676959 372.25705475 25.47624674
 16.96186542 21.9808023 13.98139059 21.60749607]
```

Out[37]:

	0	3	1	4	7	10	9	5	6	2	8
number of customers	1862	569	520	328	319	302	203	188	17	12	7

*Fig .8.12 Creating Customer Clusters using K-Means*

## PCA VISUALIZATION OF CUSTOMER CLUSTERS:



*Fig .8.13 PCA Visualization of Customer Clusters*

number of customers: 4327

	cluster	count	min	max	mean	sum	cluster_0	cluster_1	cluster_2	cluster_3	cluster_4	size
0	3.0	2.711775	203.347364	344.281511	268.643502	750.399844	55.698703	8.355426	8.785099	13.099938	14.073805	569
1	4.0	2.646341	196.592530	355.784909	266.414928	855.547378	13.369769	52.349465	14.364761	6.658548	13.281216	328
2	1.0	2.830769	201.207194	351.803519	266.307844	775.862733	11.227377	10.997307	62.316486	4.785709	10.683972	520
3	7.0	2.360502	184.477649	317.476238	240.404375	609.371536	20.954752	6.908784	7.973653	52.411155	11.752883	319
4	10.0	2.993377	196.552914	400.592914	292.537493	970.275298	13.334250	7.292065	8.659974	6.702878	64.010834	302
5	0.0	3.661654	207.133427	488.959018	333.658561	1249.330640	25.372483	18.112993	20.455797	13.213662	22.850546	1862
6	5.0	2.026596	1015.804149	1493.343729	1244.320109	2700.264739	25.890002	18.903522	22.843303	11.462967	20.900565	188
7	2.0	1.500000	4075.741667	6245.503333	5073.737778	7931.028333	18.187242	15.135140	22.856542	19.206717	24.614359	12
8	9.0	20.798030	70.695271	1336.884778	476.894931	9758.524877	23.445842	17.086203	22.776894	11.776314	24.925233	203
9	8.0	127.285714	10.585714	2248.087143	381.948905	49672.060000	25.349037	14.182605	24.477933	13.849725	22.161895	7
10	6.0	31.058824	85.434118	13750.221765	3025.061266	90558.188235	20.120521	18.515906	22.618822	8.865692	29.879060	17

*Fig . 8.14 Analysis of customer clusters*

## CUSTOMERS CLASSIFICATION:

Logistic Regression  
Precision: 88.68 %

Decision Tree:  
Precision: 83.83 %

Random Forest Classifier  
Precision: 87.99 %

## Overall Precision after using Voting :

Precision: 97.30 %

*Fig .8.15 Precision of each classification method*

## **9. CONCLUSION**

In this project, the database providing details on purchases made on an E-commerce platform is used. Each entry in the dataset describes the purchase of a product, by a particular customer and at a given date. In total, approximately 4000 customers appear in the database.

The first stage of this work consisted in describing the different products sold by the site, which was the subject of a first classification. Then different products are categorized into 5 main categories of products using k-means clustering and the clusters are viewed using WordCloud Visualization which is used to display most used words in a text from small to large and give a glance to most important keywords used in description of products in the dataset.

In a second stage, a classification of the customers by analyzing their consumption habits is performed. Then the customers are classified into 11 major categories based on the type of products they usually buy, the number of purchases made and the amount they spent on each cluster of products. Once these categories established using k-means clustering then the clusters of Customers are viewed using Principle Component Analysis which is used to emphasize variation and bring out strong patterns in a dataset and also makes data easy to explore.

Finally, the quality of the predictions of the different classifiers was tested using Logistic Regression, Decision Tree and Random Forest Classifier. As this project is filled by combining multiple fits of a model trained using Stochastic Learning algorithms, voting ensembles are most effective. In order to get better performance than the any model in ensemble soft voting is used.

This project is used to know the behaviour of the customer using k-means which performs the division of objects into clusters that share similarities and the dissimilarities of the objects are belonging to another cluster and helps for the development of the e-commerce websites for knowing about their customers such as their habits, liked products.

## **10.REFERENCES**

- [1.] R. Siva Subramanian and D.Prabha, "A Survey on Customer Relationship Management", *International Conference on Advanced Computing and Communication Systems*, January 2017.
- [2.] Jayant Tikmani, Sudhanshu Tiwari and Sujata Khedkar, "Telecom customer segmentation based on cluster analysis An Approach to Customer Classification using k-means", *IJIRCCE*, 2015.
- [3.] Raj Bala, Sunil Sikka and Juhi Singh, "A Comparative Analysis of Clustering Algorithms", *International Journal of Computer Applications*, pp. 35-39, August 2014.
- [4.] Yogita Rani and Harish Rohil, "A Study of Hierarchical Clustering Algorithm", *IJICT*, 2013.
- [5.] Ilung Pranata and Geoff Skinner, "Segmenting and targeting customers through clusters selection & analysis", *under review for International Conference on Advanced Computer Science and Information Systems*, October 2015.
- [6.] H.F. Witschel, S. Loo and K. Riesen, "How to Support Customer Segmentation with Useful Cluster Descriptions" in *Advances in Data Mining: Applications and Theoretical Aspects*, Springer, vol. 9165, 2015.
- [7.] Zan Huang, Daniel Zeng and Hsinchun Chen, "A Comparative Study of Recommendation Algorithms in E-Commerce Applications", *Proceedings of the IEEE Region 10 Conference*, pp. 1-23.
- [8.] Ina Maryani and Dwiza Riana, "Clustering and profiling of customers using RFM for customer relationship management recommendations", *5th International Conference on Cyber and IT Service Management*, August 2017.
- [9.] T. Nelson Gnanaraj, K. Ramesh Kumar and N. Monica, "Survey on mining clusters using new k-mean algorithm from structured and unstructured data", *IJACST*, 2014.
- [10.] Chinedu Pascal Ezenkwu and Simeon Ozuomba, "Application of K-Means Algorithm for Efficient Customer Segmentation: A Strategy for Targeted Customer Services", *IJARAI*, 2015.