**Customer Churn Analysis**

**ABSTRACT**

Ecommerce transactions are no longer a new thing. Many people shop with ecommerce and many companies use ecommerce to promote and to sell their products. Because of that, overloading information appears on the customers’ side. Overloading information occurs when customers get too much information about a product then feel confused. Personalization will become a solution to overloading problem. In marketing, personalization technique can be used to get potential customers in a case to boost sales. The potential customer is obtained from customer segmentation or market segmentation. This paper will review customer segmentation using data, methods and process from a customer segmentation research. The data for customer segmentation were divided into internal data and external data. Customer profile and purchase history were treated as the internal data while server log, cookies, and survey data were as the external data. In this process, it can take online customer retail dataset to apply principle component analysis to get the best features from the particular feature selection. To apply the K-Means clustering for cluster the data from dataset. At a high level, a computer cluster is a group of two or more computers, or nodes, that run in parallel to achieve a common goal. This allows workloads consisting of a high number of individual, parallelizable tasks to be distributed among the nodes in the cluster. The system is developed the different ML algorithms and estimate some performance metrics such as accuracy and visualizations.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** |  |
|  | **INTRODUCTION** |  |
|  | * 1. General Introduction |  |
|  | * 1. Project Objectives |  |
|  | * 1. Problem Statement |  |
|  | **SYSTEM** **PROPOSAL** |  |
|  | * 1. Existing System |  |
|  | * + 1. Disadvantages |  |
|  | * 1. Proposed System |  |
|  | 2.2.1 Advantages |  |
|  | * 1. Literature Survey |  |
|  | **SYSTEM DIAGRAMS** |  |
|  | * 1. Architecture Diagram |  |
|  | * 1. Flow Diagram |  |
|  | * 1. UML Diagrams |  |
|  | **IMPLEMENTATION** |  |
|  | * 1. Modules |  |
|  | * 1. Modules Description |  |
|  | **SYSTEM** **REQUIREMENTS** |  |
|  | * 1. Hardware Requirements |  |
|  | * 1. Software Requirements |  |
|  | * 1. Software Description |  |
|  | * 1. Testing of Products |  |
|  | **CONCLUSION** **AND** **FUTURE** **ENHANCEMENT** |  |
|  | * 1. Conclusion |  |
|  | * 1. Future Enhancement |  |
|  | **SAMPLE** **CODING** **AND** **SAMPLE** **SCREENSHOT** |  |
|  | **REFERENCES** |  |

**CHAPTER 1**

**INTRODUCTION**

* 1. **General Introduction**

Ecommerce transactions are no longer a new thing. Many people shop with ecommerce and many companies use ecommerce to promote and to sell their products. Because of that, overloading information appears on the customers’ side. Overloading information occurs when customers get too much information about a product then feel confused. Personalization will become a solution to overloading problem. In marketing, personalization technique can be used to get potential customers in a case to boost sales.

The potential customer is obtained from customer segmentation or market segmentation. This paper will review customer segmentation using data, methods and process from a customer segmentation research. The data for customer segmentation were divided into internal data and external data. Customer profile and purchase history were treated as the internal data while server log, cookies, and survey data were as the external data. In this process, it can take online customer retail dataset to apply principle component analysis to get the best features from the particular feature selection.

To apply the K-Means clustering for cluster the data from dataset. At a high level, a computer cluster is a group of two or more computers, or nodes, that run in parallel to achieve a common goal. This allows workloads consisting of a high number of individual, parallelizable tasks to be distributed among the nodes in the cluster.

Customer Segmentation means grouping the customers based on marketing groups which shares the similarity among customers. To be more exact, it means segmenting customers sharing the common characteristics which is the best way of marketing. Customer segmentation is gathering information about each customers and analysing it to identify the different patterns for creating the segments. Some of the best methods for gathering information are face-to-face interviews, Telephonic interviews, through surveys or through research using information which are published related to market categories.

The basic information which includes billing information, shipping information, products purchased, promo codes, payment method etc., Beyond these some companies also collect information like reason for the purchase, advertisement channel which makes them to purchase, age, gender etc.,

In B2B(Business to Business) marketing customers are grouped according to numerous factors like Industries, number of employers, Products purchased from the company in earlier times and location. On other-hand, in B2C(Business to Consumer) marketing companies segment the customers based on Age, Gender of the customers, their marital status, life stage of the customers like single, married, divorced, retired etc.,

On of the main factor of B2C is Location of customers (Rural,suburban, urban).Customer segmentation can be practiced for all the businesses nevertheless of size or industry. Common segmentation types include Demographic, RFM (Recency, Frequency, Monetary) analysis, HVCs (High-value customer), customer status, Behavioural, psychographic etc., Some of the major benefits of customer segmentation include Marketing strategy, promotion strategy, Budget efficiency ,product development etc., In this article we applied the basic analytics functionality to provide the decision makers(in our case the business investors) with the required information to make the right decision. In this article we define a solution for reducing risk factors and also contribute to decision making for new business investments. We proposed to use K-means technique for customer segmentation. Our solution is segmenting the customers based on information analytics.

Consumers can be divided into groups in relation to common behaviours they share. Such behaviours link to their knowledge of, attitude toward, use of, or spending score or response to a product. We used machine learning Clustering algorithm K-Means for this customer segmentation.

Customer segmentation is the division of likely clients in a given market into discrete groups. That division depends on clients having comparative necessities, purchasing qualities and so on; This guide will focus on the worth based approach, which permits extension stage organizations to obviously characterize and focus on their best possibilities (in light of its momentum information available) and fulfill the greater part of their requirements for division in the development stage without consuming the time and assets of a conventional, unmistakable division research process.

The customer segmentation has the significance as it incorporates, the capacity to change the projects of market so it is appropriate to every one of the client fragment, support in business choice; ID of items related with every client portion and to deal with the interest and supply of that item; recognizing and focusing on the potential client base, and The customer segmentation has the significance as it incorporates, the capacity to change the projects of market so it is appropriate to every one of the client fragment, support in business choice; ID of items related with every client portion and to deal with the interest and supply of that item foreseeing client deserting, giving bearings in tracking down the arrangements.

Clients into bunches in light of orientation, age, geographic area and spending examples to give some examples. Notwithstanding, in this report a more surprising customer clustering approach in view of client conduct in the Pick E-commercial center will be assessed. This cycle did not depend on any previous relations or rules. Rather, the actual information uncovers potential similitudes between clients. This issue is regularly alluded to as data over-burden and can be addressed by giving particular article streams to every client contingent upon his/her inclinations.

This report will cover an investigation of the Plick client base, preferably, the examination ought to reveal a few examples in the information that can be utilized to join clients into more modest gatherings. Customer segmentation is right now performed by handling client data set, for example segment information or buy history.

* 1. **Project Objectives**
* To apply the principle component analysis for finding best features.
* To effectively create the Clusters.
* The principle objective of this task is the field of cluster analysis it is essential to pick a calculation appropriate for the accessible dataset.
* To implement the different ML algorithms.

.

* 1. **Problem Statement**

According to business, a company may create three segments like Medium and Low. This is where Machine Learning provides a crucial solution, several algorithms are applied for revealing the hidden patterns in data for better decision making. In this paper we proposed a Customer segmentation concept in which the customer bases of an establishment is divided into segments based on the customers’ characteristics and attributes. This idea can be used by the B2C companies to outperform the competition by developing uniquely appealing products and services and make it reach to potential customers. This approach is implemented using “k-means”

**CHAPTER 2**

**SYSTEM PROPOSAL**

* 1. **Existing System:**

The segmentation process done by manually in before, Since the previous models are predicted by constant data, the system needs the updated values and methods. In old methods the whole process is done by using mathematical methods with probability and permutations.

* + 1. **Disadvantages**
* Limited availability and the need for time-consuming offline data reduction.
* The process is implemented without removing unwanted data.
  1. **Proposed System**

Machine learning approaches are an incredible instrument for dissecting customer information and tracking down bits of knowledge and examples. Misleadingly wise models are useful assets for chiefs. They can exactly recognize client fragments, which is a lot harder to do physically or with ordinary logical techniques. There are many machine learning algorithms, each reasonable for a particular sort of issue. One extremely normal AI calculation that is appropriate for client division issues is the k-means clustering algorithm. The system is developed the different ML algorithm such as logistic regression and decision tree. The experimental results shows that accuracy, precision, recall and f1-score.

* + 1. **Advantages**
* The proposed methods provide Better Feature selection for k-Means Algorithm
* The process is implemented with removing unwanted data.
  1. **LITERATURE SURVEY**

**Title**: Study on various clustering techniques”.

**Year**: 2015

**Author**: Jainendra Shukla, Miguel Barreda-Angeles, Joan Oliver, G. C. Nandi, Domenec Puig

**Methodology**

Customer Segmentation is the subdivision of a market into discrete customer groups that share similar characteristics. Customer Segmentation can be a powerful means to identify unsatisfied customer needs. Using the above data companies can then outperform the competition by developing uniquely appealing products and services. The most common ways in which businesses segment their customer base are, such as gender, age, familial and marital status, income, education, and occupation, which differs depending on the scope of the company.

**Advantage**

* Feature extraction is very useful improve our accuracy in Emotional Recognition.
* It is easy to speed up our trained data to improve high accuracy.

**Disadvantage**

* One drawback of feature extraction is that the new features generated are not interpretable by humans. The data in the new variables would appear like random numbers to human eyes. PCA is a popular dimensionality reduction and unsupervised learning technique.08-Mar-2020

**Title**: Robust clustering algorithms

**Year**: 2011

**Author:** P.Gupta

**Methodology**

Clustering is a method of unsupervised learning and a common technique for statistical data analysis where a data set is segmented into subsets such that the elements within each subset are somehow more similar to each other than to elements assigned to other subsets in some sense. Cluster analysis divides data into groups (clusters) that are either meaningful or useful or both. If meaningful clusters are the goal, then the clusters should capture the natural structure of the data. In some cases, however, cluster analysis is only a useful starting point for other purposes, such as data summarization.

**Advantage**

* We show that our algorithms can be used to cluster accurately in cases where the data satisfies a number of natural properties and where the traditional agglomerative algorithms fail.
* In particular, if the data satisfies the (α, ν)-good neighborhood property, the RHL will be successful in generating a hierarchy such that the target clustering is a pruning of that hierarchy, whereas WNL will be successful if the data satisfies the (α, 0)-good neighbourhood property

**Disadvantage**

* The wave form does not researchers to distinguish between activities originating in different but closely adjacent locations.

**Title**: “Cluster analysis and mathematical programming

**Year**: 2016

**Author**: Hansen

**Methodology**

Consider a set of entities together with observations or measurements describing them. Cluster Analysis deals with the problem of finding subsets of interest called clusters, within such a set. Usually, clusters are required to be homogeneous and/or well separated. Homogeneity means that entities within the same cluster should resemble one another and separation that entities in different clusters should differ one from the other . This problem is old. It can be traced back to Aristotle and was already much studied by xviiith century naturalists such as Buffon, Cuvier and Linn´e. It is also ubiquitous, with applications in the natural sciences, psychology, medicine, engineering, economics, marketing and other fields. As a consequence, the cluster analysis literature is vast and heterogeneous (the yearly Classification Literature Automated Search Service lists many books and hundreds of papers on that topic in each issue). Cluster analysis algorithms draw upon statistics, mathematics and computer science. Closely related fields are pattern recognition, computer vision, computational geometry and subfields of operations research such as location theory and scheduling.

**Advantage**

* The key benefits of these systems are their small size, lightness, low-power consumption and, of course, their wearable.

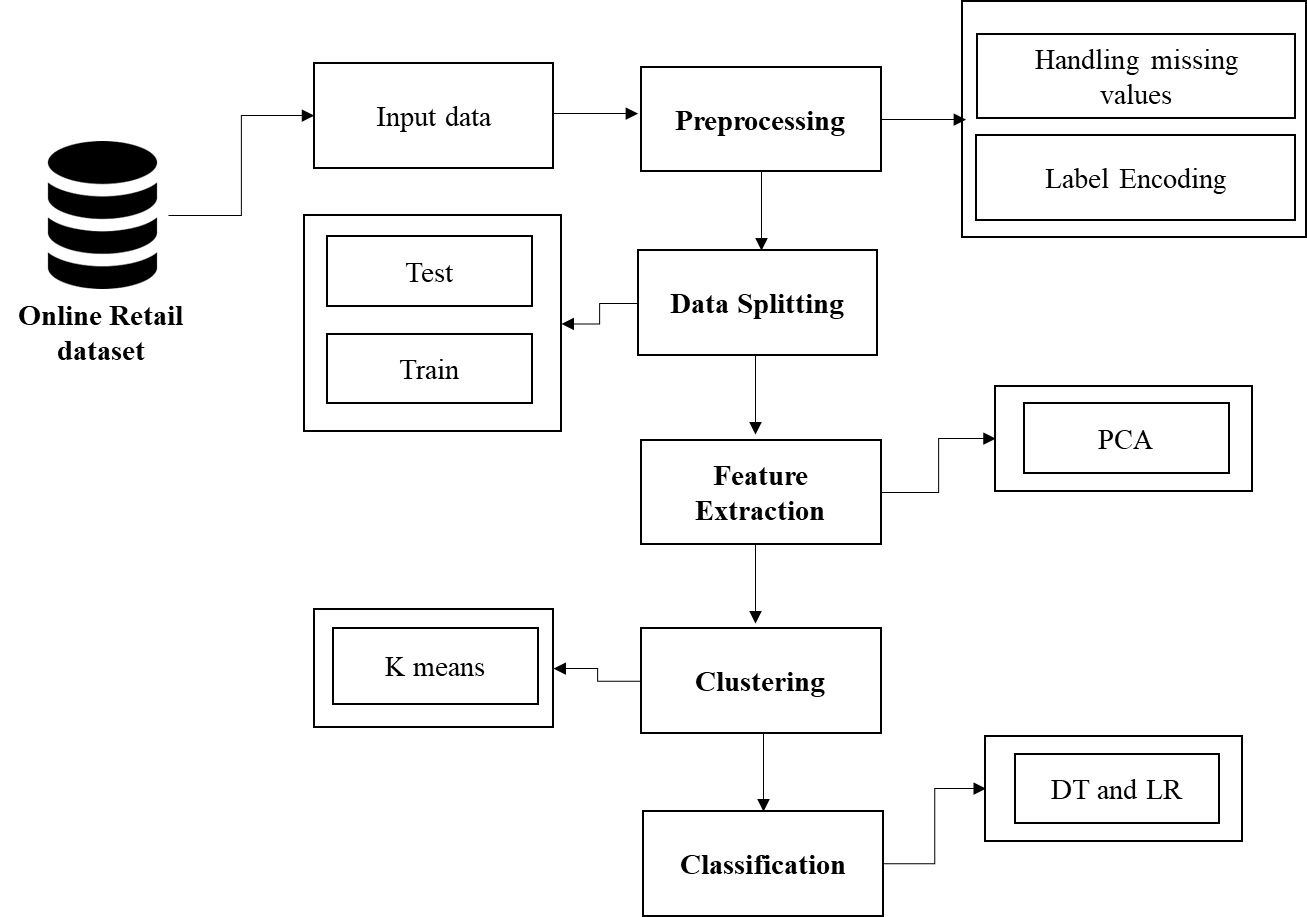
**Disadvantage**

* There is no affective computing research that tries to automatically recognize the user’s mood in an IVE through physiological signals and machine learning algorithms.

**CHAPTER 3**

**SYSTEM DIAGRAMS**

* 1. **ARCHITECTURE DIAGRAM**

****

* 1. **FLOW DIAGRAM**

**View Data**

**Select Dataset**

**Start**

**Data Preprocessing**

**Label Encoding**

**Feature Selection**

**K-Means Algorithm**

**Clustering**

* 1. **UML Diagram**

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS**:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

2. Provide extendibility and specialization mechanisms to extend the core concepts.

3. Be independent of particular programming languages and development process.

4. Provide a formal basis for understanding the modeling language.

5. Encourage the growth of OO tools market.

6. Support higher level development concepts such as collaborations, frameworks, patterns and components.

7. Integrate best practices.

**3.3.1 USECASE DIAGRAM**

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

A use case is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modelling Language (UML) as an actor) and a system to achieve a goal. The actor can be a human or other external system.

UML use case diagrams are ideal for:

* Representing the goals of system-user interactions
* Defining and organizing functional requirements in a system
* Specifying the context and requirements of a system
* Modelling the basic flow of events in a use case

**Notations:**

* **Use cases**: Horizontally shaped ovals that represent the different uses that a user might have.
* **Actors**: Stick figures that represent the people actually employing the use cases.
* **Associations**: A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.
* **System boundary boxes**: A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system. For example, Psycho Killer is outside the scope of occupations in the chainsaw example found below.
* **Packages**: A UML shape that allows you to put different elements into groups. Just as with component diagrams, these groupings are represented as file folders.

USER

SYSTEM

**3.3.2 ER DIAGRAM**

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research.

Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.

They mirror grammatical structure, with entities as nouns and relationships as verbs.

**Notation:**

### **Entity**

A definable thing—such as a person, object, concept or event—that can have data stored about it. Think of entities as nouns. Examples: a customer, student, car or product. Typically shown as a rectangle.

**Entity type:**A group of definable things, such as students or athletes, whereas the entity would be the specific student or athlete. Other examples: customers, cars or products.

**Entity set:** Same as an entity type, but defined at a particular point in time, such as students enrolled in a class on the first day.

Other examples: Customers who purchased last month, cars currently registered in Florida. A related term is instance, in which the specific person or car would be an instance of the entity set.

**Entity categories:** Entities are categorized as strong, weak or associative. A **strong entity** can be defined solely by its own attributes, while a **weak entity** cannot. An associative entity associates entities (or elements) within an entity set.

**Entity keys:** Refers to an attribute that uniquely defines an entity in an entity set. Entity keys can be super, candidate or primary. **Super key:**A set of attributes (one or more) that together define an entity in an entity set.

**Candidate key:**A minimal super key, meaning it has the least possible number of attributes to still be a super key. An entity set may have more than one candidate key. **Primary key:**A candidate key chosen by the database designer to uniquely identify the entity set. **Foreign key:**Identifies the relationship between entities.

### **Relationship**

How entities act upon each other or are associated with each other. Think of relationships as verbs.

For example, the named student might register for a course.

The two entities would be the student and the course, and the relationship depicted is the act of enrolling, connecting the two entities in that way.

Relationships are typically shown as diamonds or labels directly on the connecting lines.

**DATA SELECTION & LOAD**

**Feature Selection**

**Clustering**

**3.3.3 CLASS DIAGRAM**

|  |
| --- |
| **DATASET** |
| Select dataset () |
| Import dataset () |
| View dataset () |

|  |
| --- |
| **DATA PREPROCESSING** |
| Label Encoding() |

|  |
| --- |
| **Feature Selection** |
| Splitting the training and testing dataset () |

|  |
| --- |
| **Clustering** |
| K-Means Clustering() |
| PCA() |

**3.3.4 SEQUENCE DIAGRAM**

Sequence diagrams document the interactions between classes to achieve a result, such as a use case. Because UML is designed for object-oriented programming, these communications between classes are known as messages. The Sequence diagram lists objects horizontally, and time vertically, and models these messages over time.

**Graphical Notation**: In a Sequence diagram, classes and actors are listed as columns, with vertical lifelines indicating the lifetime of the object over time.

**Object**: Objects are instances of classes, and are arranged horizontally. The pictorial representation for an Object is a class (a rectangle) with the name prefixed by the object.

**Lifeline** The Lifeline identifies the existence of the object over time. The notation 2for a Lifeline is a vertical dotted line extending from an object.

**Activation**: Activations, modelled as rectangular boxes on the lifeline, indicate when the object is performing an action.

Feature selection

Clustering

Splitting

Prediction

Select

Select dataset

 Load dataset

Start

Result Generation

**3.3.5 ACTIVITY DIAGRAM**

This shows the flow of events within the system. The activities that occur within a use case or within an objects behaviour typically occur in a sequence. An activity diagram is designed to be simplified look at what happens during an operations or a process. Each activity is represented by a rounded rectangle the processing within an activity goes to compilation and then an automatic transmission to the next activity occurs. An arrow represents the transition from one activity to the next. An activity diagram describes a system in terms of activities. Activities are the state that represents the execution of a set of operations.

These are similar to flow chart diagram and dataflow.

**Initial state**: which state is starting the process?

**Action State**: An action state represents the execution of an atomic action, typically the invocation of an operation. An action state is a simple state with an entry action whose only exit transition is triggered by the implicit event of completing the execution of the entry action.

**Transition**: A transition is a directed relationship between a source state vertex and a target state vertex. It may be part of a compound transition, which takes the static machine from one static configuration to another, representing the complete response of the static machine to a particular event instance.

**Final state:** A final state represents the last or "final" state of the enclosing composite state. There may be more than one final state at any level signifying that the composite state can end in different ways or conditions.

When a final state is reached and there are no other enclosing states it means that the entire state machine has completed its transitions and no more transitions can occur.

**Decision**: A state diagram (and by derivation an activity diagram) expresses decision when guard conditions are used to indicate different possible transitions that depend on Boolean conditions of the owning object.

INPUT DATA

PRE-PROCESSING

FEATURE SELECTION

TESTING DATASET

TRAINING DATASET

PCA & K-Means Clustering

**CHAPTER 4**

**IMPLEMENTATION**

* 1. **Modules**
* Data Selection and Loading
* Data Preprocessing
* Feature Selection
* Clustering
* Classification
* Performance Analysis
  1. **Modules Description**

**4.2.1 DATA SELECTION AND LOADING**

* The data selection is the process of selecting the data for online retail dataset.
* The dataset was collected from dataset repository.
* The dataset which contains the information about the Mean, Covmat, Entropy, Label.
* With the help of panda’s, we have to read our input dataset.
* The dataset is in the format is ‘.csv’

**4.2.2 DATA PREPROCESSING**

* Data pre-processing is the process of removing the unwanted data from the dataset.
* Pre-processing data transformation operations are used to transform the dataset into a structure suitable for machine learning.
* This step also includes cleaning the dataset by removing irrelevant or corrupted data that can affect the accuracy of the dataset, which makes it more efficient.
* Missing data removal
* Encoding Categorical data
* Missing data removal: In this process, the null values such as missing values and Nan values are replaced by 0.
* Missing and duplicate values were removed and data was cleaned of any abnormalities.
* Encoding Categorical data: That categorical data is defined as variables with a finite set of label values.
* That most machine learning algorithms require numerical input and output variables.

**4.3.3 DATA SPLITTING :**

* Data splitting is the act of partitioning available data into two portions, usually for cross-validate purposes.
* One Portion of the data is used to develop a predictive model and the other to evaluate the model's performance.
* Separating data into training and testing sets is an important part of evaluating image processing models.
* Typically, when you separate a data set into a training set and testing set, most of the image data is used for training, and a smaller portion of the data is used for testing.

**4.4.4 CLUSTERING**

* **K-Means** Clustering is an Unsupervised Learning algorithm, which groups the unlabelled dataset into different clusters.
* Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.
* **Principal Component Analysis** is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning.
* It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation.

**4.4.5 CLASSIFICATION:**

* In our process, we have to implement the different classification algorithm such as Logistic Regression and decision tree.
* A **decision tree** is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.
* **Logistic regression** is used to obtain odds ratio in the presence of more than one explanatory variable. The procedure is quite similar to multiple linear regression, with the exception that the response variable is binomial. The result is the impact of each variable on the odds ratio of the observed event of interest.

**4.4.6 PERFORMANCE ANALYSIS:**

The Final Result will get generated based on the overall classification and prediction. The performance of this proposed approach is evaluated using some measures like,

* **Accuracy:**

Accuracy of classifier refers to the ability of classifier. It predicts the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.

AC= (TP+TN)/ (TP+TN+FP+FN)

* **Precision**

Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives.

Precision=TP/ (TP+FP)

* **Recall**

Recall is the number of correct results divided by the number of results that should have been returned. In binary classification, recall is called sensitivity. It can be viewed as the probability that a relevant document is retrieved by the query.

Recall=TP/ (TP+FN)

**CHAPTER 5**

**SYSTEM REQUIREMENTS**

* 1. **Hardware Requirement**
* System : Pentium IV 2.4 GHz
* Hard Disk : 200 GB
* Mouse : Logitech.
* Keyboard : 110 keys enhanced
* Ram : 4GB
  1. **Software Requirement**
* O/S : Windows 7.
* Language : Python
* Front End: Anaconda Navigator – Spyder
  1. **Software Description**

**Python**

Python is one of those rare languages which can claim to be both simple and powerful. You will find yourself pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in. The official introduction to Python is Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. I will discuss most of these features in more detail in the next section.

**Features of Python**

**Simple**

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

**Easy to Learn**

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

**Free and Open Source**

Python is an example of a FLOSS (Free/Libré and Open Source Software). In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and is constantly improved by a community who just want to see a better Python.

**High-level Language**

When you write programs in Python, you never need to bother about the low-level details such as managing the memory used by your program, etc.

**Portable**

Due to its open-source nature, Python has been ported to (i.e. changed to make it work on) many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features.

You can use Python on GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, and # -\*- coding: utf-8 -\*-

z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and PocketPC!

You can even use a platform like Kivyto create games for your computer and for iPhone, iPad, and Android.

**Interpreted**

This requires a bit of explanation.

A program written in a compiled language like C or C++ is converted from the source language i.e. C or C++ into a language that is spoken by your computer (binary code i.e. 0s and 1s) using a compiler with various flags and options. When you run the program, the linker/loader software copies the program from hard disk to memory and starts running it.

Python, on the other hand, does not need compilation to binary. You just run the program directly from the source code. Internally, Python converts the source code into an intermediate form called byte codes and then translates this into the native language of your computer and then runs it. All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc. This also makes your Python programs much more portable, since you can just copy your Python program onto another computer and it just works!

**Object Oriented**

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

Extensible

If you need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, you can code that part of your program in C or C++ and then use it from your Python program.

**Embeddable**

You can embed Python within your C/C++ programs to give scripting capabilities for your program's users.

**Extensive Libraries**

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, FTP, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), and other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the Batteries Included philosophy of Python.

Besides the standard library, there are various other high-quality libraries which you can find at the Python Package Index.

* 1. **Testing of Products**

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. . A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that “al gears mesh”, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercise.

**UNIT TESTING:**

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as ‘module testing’.

The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

**INTEGRATION TESTING:**

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

i) Top-down integration testing. ii) Bottom-up integration testing.

**TESTING TECHNIQUES/STRATEGIES:**

**WHITE BOX TESTING:**

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we

Derived test cases that guarantee that all independent paths within a module have been exercised at least once.

**BLACK BOX TESTING:**

* Black box testing is done to find incorrect or missing function
* Interface error
* Errors in external database access
* Performance errors.
* Initialization and termination errors

In ‘functional testing’, is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called ‘black box testing’. It tests the external behaviour of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

**SOFTWARE TESTING STRATEGIES**

**VALIDATION TESTING:**

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many,

But a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

**USER ACCEPTANCE TESTING:**

User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

**OUTPUT TESTING**:

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

### **Accessibility Testing**

Accessibility testing is the practice of ensuring your mobile and web apps are working and usable for users without and with disabilities such as vision impairment, hearing disabilities, and other physical or cognitive conditions.

### **Acceptance Testing**

Acceptance testing ensures that the end-user (customers) can achieve the goals set in the business requirements, which determines whether the software is acceptable for delivery or not. It is also known as user acceptance testing (UAT).

### **End to End Testing**

End to end testing is a technique that tests the application’s workflow from beginning to end to make sure everything functions as expected.

### **Functional Testing**

Functional testing checks an application, website, or system to ensure it’s doing exactly what it’s supposed to be doing.

### **Interactive Testing**

Also known as manual testing, interactive testing enables testers to create and facilitate manual tests for those who do not use automation and collect results from external tests.

### **Load Testing**

This type of non-functional software testing process determines how the software application behaves while being accessed by multiple users simultaneously.

### **Non-Functional Testing**

Non-functional testing verifies the readiness of a system according to nonfunctional parameters (performance, accessibility, UX, etc.)  Which are never addressed by functional testing.

### **Performance Testing**

Performance testing examines the speed, stability, reliability, scalability, and resource usage of a software application under a specified workload.

### **Regression Testing**

Regression testing is performed to determine if code modifications break an application or consume resources.

### **Sanity Testing**

Performed after bug fixes, sanity testing determines that the bugs are fixed and that no further issues are introduced to these changes.

### **Security Testing**

Security testing unveils the vulnerabilities of the system to ensure that the software system and application are free from any threats or risks. These tests aim to find any potential flaws and weaknesses in the software system that could lead to a loss of data, revenue, or reputation per employees or outsides of a company.

### **Single User Performance Testing**

Single user performance testing checks that the application under test performs fine according to specified threshold without any system load. This benchmark can be then used to define a realistic threshold when the system is under load.

### **Smoke Testing**

This type of software testing validates the stability of a software application, it is performed on the initial software build to ensure that the critical functions of the program are working.

### **Stress Testing**

Stress testing is a software testing activity that tests beyond normal operational capacity to test the results.

**FEASIBILITY STUDY**

The feasibility study is carried out to test whether the proposed system is worth being implemented. The proposed system will be selected if it is best enough in meeting the performance requirements. The feasibility carried out mainly in three sections namely.

• Economic Feasibility

• Technical Feasibility

• Behavioural Feasibility

**Economic Feasibility**

Economic analysis is the most frequently used method for evaluating effectiveness of the proposed system. More commonly known as cost benefit analysis. This procedure determines the benefits and saving that are expected from the system of the proposed system. The hardware in system department if sufficient for system development.

**Technical Feasibility**

This study centre around the system’s department hardware, software and to what extend it can support the proposed system department is having the required hardware and software there is no question of increasing the cost of implementing the proposed system. The criteria, the proposed system is technically feasible and the proposed system can be developed with the existing facility.

**Behavioural Feasibility**

People are inherently resistant to change and need sufficient amount of training, which would result in lot of expenditure for the organization. The proposed system can generate reports with day-to-day information immediately at the user’s request, instead of getting a report, which doesn’t contain much detail.

**CHAPTER 6**

**CONCLUSION AND FUTURE ENHANCEMENT**

* 1. **Conclusion**

Customer segmentation is effectively implemented by using K-means partitioning method. Segmentation implemented like product wise, season wise, region wise, combination of season wise and gender wise. Partition method is work well large databases. Segmentation provides loyalty between marketer and customers and provide customer relationship management, improves business and get more profit.

* 1. **Future Enhancement**

All things considered, the spotlight during this task was on the bunching investigation, the preprocessing phase of this undertaking could be improved by joining preferences and discussion to the appraisals computations utilizing some weighting of these three elements.

**CHAPTER 7**

**SAMPLE CODINGS AND SAMPLE SCREEN SHOTS**

**7.1 Sample coding:**

#======================= IMPORT PACKAGES =============================

import pandas as pd

from sklearn.model\_selection import train\_test\_split

import warnings

warnings.filterwarnings('ignore')

import matplotlib.pyplot as plt

#===================== DATA SELECTION ==============================

#=== READ A DATASET ====

data\_frame=pd.read\_excel("online\_retail\_II.xlsx")

print("-------------------------------------------------------")

print("================== 1.Data Selection ===================")

print("-------------------------------------------------------")

print()

print(data\_frame.head(20))

#===================== 2.DATA PREPROCESSING ==========================

#=== CHECK MISSING VALUES ===

print("=====================================================")

print(" 2.Preprocessing ")

print("=====================================================")

print()

print("------------------------------------------------------")

print("================ 2.1 Checking missing values =========")

print("------------------------------------------------------")

print()

print(data\_frame.isnull().sum())

print()

print("----------------------------------------------------------")

print("=========== After Checking Missing Values ================")

print("-----------------------------------------------------------")

print()

data\_frame=data\_frame.fillna(0)

print(data\_frame.isnull().sum())

#=== DROP UNWANTED COLUMNS ===

data\_frame=data\_frame.drop(["StockCode","Description","InvoiceDate","Invoice","Country"],axis=1)

#====================== 3. DATA SPLITTING =============================

x=data\_frame.drop('Quantity',axis=1)

y=data\_frame['Quantity']

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(x,y,test\_size=0.2,random\_state=50)

print("=========================================================")

print("-------------------- 3.Data Splitting -------------------")

print("==========================================================")

print()

print("Total number of rows in dataset:", data\_frame.shape[0])

print()

print("Total number of rows in training data:", X\_train.shape[0])

print()

print("Total number of rows in testing data:", X\_test.shape[0])

#========================== 4. PCA ===================================

from sklearn.decomposition import PCA

pca = PCA(random\_state=1).fit(x)

print("=========================================================")

print("-------------------- 4.PCA ----------------------------")

print("==========================================================")

print()

pca\_samp = pca.transform(x)

plt.scatter(pca\_samp, x)

plt.show()

#====================== 5. K MEANS =============================

print("=========================================================")

print("-------------------- 5.K MEANS ----------------------------")

print("==========================================================")

print()

from sklearn.cluster import KMeans

wcss = []

for i in range(1, 11):

kmeans = KMeans(n\_clusters = i, init = 'k-means++',

max\_iter = 300, n\_init = 10, random\_state = 0)

kmeans.fit(x)

wcss.append(kmeans.inertia\_)

plt.subplots(figsize=(5,5))

plt.plot(range(1, 11), wcss)

plt.title('The Elbow Method', color="RED",size=10)

plt.xlabel('Number of clusters', color="#FDD017",size=10)

plt.ylabel('WCSS', color="#FDD017",size=15)

kmeans = KMeans(n\_clusters = 2, init = 'k-means++',

max\_iter = 300, n\_init = 10, random\_state = 0)

y\_kmeans = kmeans.fit\_predict(pca\_samp)

y\_kmeans

import matplotlib.pyplot as plt

plt.subplots(figsize=(8,5))

plt.scatter(x[y\_kmeans == 0], x[y\_kmeans == 0], s = 100, c = '#FF2400', label = 'Quantity')

plt.scatter(x[y\_kmeans == 1], x[y\_kmeans == 1], s = 100, c = '#306EFF', label = 'Price')

plt.scatter(x[y\_kmeans == 2], x[y\_kmeans == 2], s = 100, c = '#FBB917', label = 'Customer ID')

# Plotting the centroids of the clusters

plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:,1], s = 200, c = '#FF00FF',marker = '\*', label = 'Segmentation')

\_ = plt.legend()

plt.show()

plt.scatter(y, y)

#==== cluster customers =====

kmeans = KMeans(n\_clusters=10)

k\_fit = kmeans.fit(pca\_samp)

k\_fit

k\_fit.n\_clusters

k\_fit.cluster\_centers\_

k\_fit.labels\_

k\_means = KMeans(n\_clusters=10).fit(pca\_samp)

kumeler = k\_means.labels\_

cus\_kmeans = pd.DataFrame(pca\_samp)

plt.scatter(cus\_kmeans.iloc[:, 0],

cus\_kmeans.iloc[:, 1],

c=kumeler,

s=50,

cmap="viridis")

plt.show()

merkezler = k\_means.cluster\_centers\_

plt.scatter(cus\_kmeans.iloc[:, 0],

cus\_kmeans.iloc[:, 1],

c=kumeler,

s=50,

cmap="viridis")

plt.scatter(merkezler[:, 0],

merkezler[:, 1],

c="black",

s=200,

alpha=0.5)

plt.show()

#====================== CLASSIFICATION =============================

# === DT ====

from sklearn import tree

clf = tree.DecisionTreeClassifier()

clf.fit(X\_train[0:1500],Y\_train[0:1500])

y\_pred\_dt=clf.predict(X\_test[0:1500])

from sklearn import metrics

mae\_dt=metrics.mean\_absolute\_error(y\_pred\_dt,Y\_test[0:1500])

acc\_dt= 100- mae\_dt

print("--------------------------------------")

print(" LOGISTIC REGRESSION")

print("--------------------------------------")

print()

print("1. Accuracy = ", acc\_dt)

print()

print("2. MAE =", mae\_dt)

print()

# ==== LR =====

from sklearn import linear\_model

lr=linear\_model.LogisticRegression()

lr.fit(X\_train[0:1500],Y\_train[0:1500])

y\_pred\_lr=lr.predict(X\_test[0:1500])

from sklearn import metrics

mae\_lr=metrics.mean\_absolute\_error(y\_pred\_lr,Y\_test[0:1500])

acc\_lr= 100- mae\_lr

print("--------------------------------------")

print(" LOGISTIC REGRESSION")

print("--------------------------------------")

print()

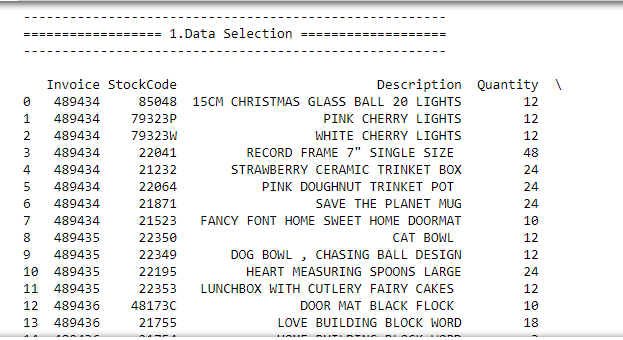
print("1. Accuracy = ", acc\_lr)

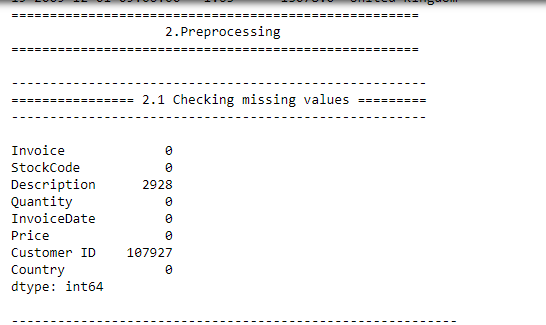
print()

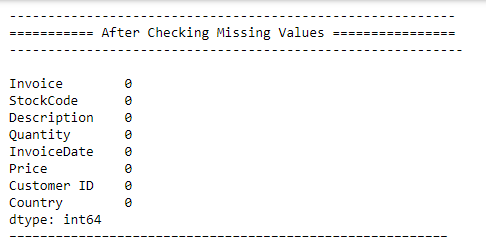
print("2. MAE =", mae\_lr)

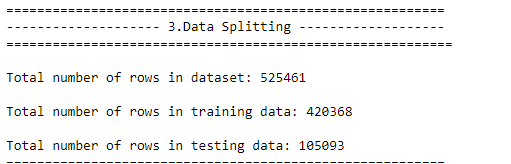
print()

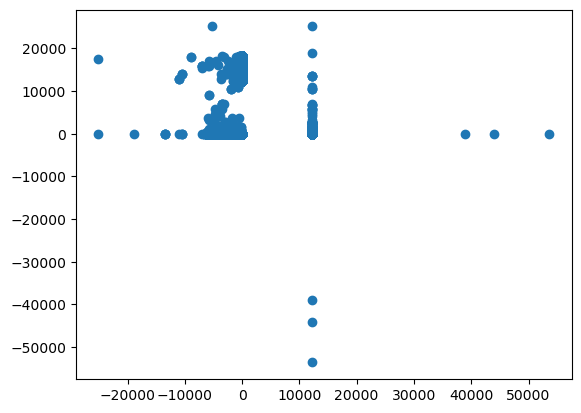
**7.2 sample Screenshots:**

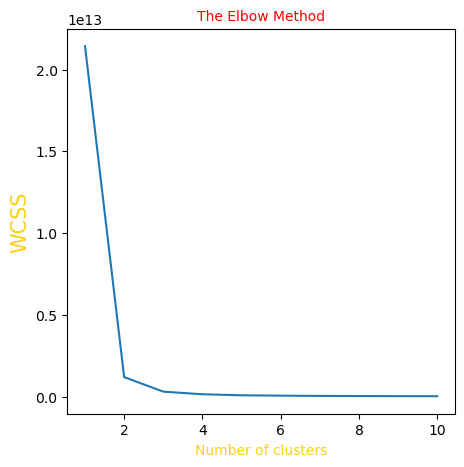


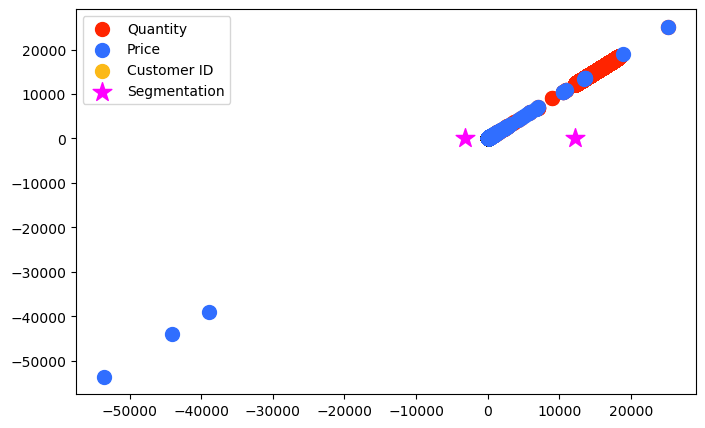


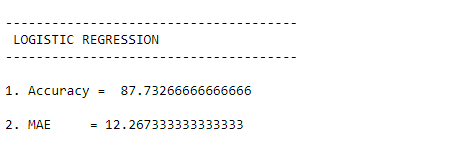


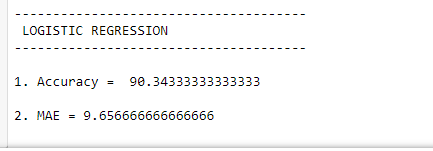


****

****

****





**CHAPTER 7**

**REFERENCES**

1. D. L Davies and D. W Bouldin.“A cluster separation measure”. In: IEEE transactions on pattern analysis and AI intelligence 2 (1979), pp. 224–227.
2. Pranata and Geo Skinner. “Segmenting and selecting customers through clustering selection & analysis”. In: Advanced Computer Science and Information Systems (ICACSIS), 2015 International Conference on. IEEE. 2015, pp. 303–308.
3. Minghua Han. “Customer segmentation model based on shopping consumer behavior analysis”. International Symposium on. IEEE. 2018, pp. 914–917.
4. Ardabili, S.F.; Mosavi, A.; Várkonyi-Kóczy, A.R. Systematic Review of Deep Learning and AI Models in Biofuels Research. In Proceedings of the 18th International Conference on Global Research and Education, Budapest, Hungary, 4–7 Sept 2019.
5. Saroj, T. C., and Chaudhary, T., 2015. “Study on various clustering techniques”. International Journal of Computer Science and Information Technologies, 6(3),pp. 3031–3033.
6. Tan, P.-N., Steinbach, M., and Kumar, V., 2013. “Data mining cluster analysis: basic concepts and algorithms”. Introduction to data mining.
7. Gupta, P., 2011. “Robust clustering algorithms”. PhD thesis, Georgia Institute of Technology.
8. Gupta, P., 2011. “Robust clustering algorithms”. PhD thesis, Georgia Institute of Technology.
9. Xiong, T., Wang, S., Mayers, A., and Monga, E., 2012. “Dhcc: Divisive hierarchical clustering of categorical data”. Data Mining and Knowledge Discovery, 24(1), pp. 103–135.
10. Berkhin, P., 2006. “A survey of clustering data mining techniques”. In Grouping multidimensional data. Springer, pp. 25–71
11. Everitt, B., Landau, S., Leese, M., and Stahl, D., 2001.“Cluster analysis. 4th”. Arnold, London.
12. Hansen, P., and Jaumard, B., 1997. “Cluster analysis and mathematical programming”. Mathematical programming, 79(1-3), pp. 191–215.
13. Jain, A. K., and Dubes, R. C., 1988. “Algorithms for clustering data”.
14. Jain, A. K., Murty, M. N., and Flynn, P. J., 1999. “Data clustering: a review”. ACM computing surveys (CSUR), 31(3), pp. 264–323.
15. Kolatch, E., et al., 2001. “Clustering algorithms for spatial databases: A survey”. PDF is available on the Web, pp. 1–22.
16. Wilks, D. S., 2011. “Cluster analysis”. In International geophysics, Vol. 100. Elsevier, pp. 603–616. [13] Grabmeier, J., and Rudolph, A., 2002. “Techniques of cluster algorithms in data mining”. Data Mining and knowledge discovery, 6(4), pp. 303–360.
17. Migu´eis, V. L., Camanho, A. S., and e Cunha, 2012. “Customer data mining for lifestyle segmentation”. Expert Systems with Applications, 39(10), pp. 9359– 9366.
18. Ng, R. T., and Han, J., 2002. “Clarans: A method for clustering objects for spatial data mining”. IEEE transactions on knowledge and data engineering, 14(5), pp. 1003–1016.