JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

UTILIZING DATA MINING TECHNIQUES FOR PREDICTING STUDENTS' ACADEMIC PERFORMANCE

Dr. P. Ramakrishna Reddy[1], Dr. Hemanth Kumar Molapata[2], Dr. k. Ramakrishna[3], Dr. T. Mahesh Kumar[4]

1.Academic Consultant, Dept. of Statistics, S V University, Tirupati, AP, India.
2.Assistant Professor, Dept. of Statistics, Hindu College, University of Delhi, Delhi, India
3.Academic Consultant, Dept. of Mathematics, S V University, Tirupati, AP, India
4.Academic Consultant, Dept. of Mathematics, S V University, Tirupati, AP, India
1.Sri Venkateswara University, Tirupati, India

Abstract: In the real life of higher education, predicting students' academic performance remains a significant challenge. Employing statistical tools and data mining techniques has become essential to analyse and forecast students' paths through the educational process. This paper explores the utilization of data mining techniques, specifically focusing on the WEKA data mining software, to predict semester-wise student marks based on parameters within a given dataset. The dataset encompasses information from students across five courses within a single college over multiple semesters. By using data mining methodologies, educators can uncover hidden patterns within student data, facilitating a deeper understanding of the factors influencing academic success. This paper discusses the application of data mining techniques as a means to discover valuable insights and improve decision-making processes within the educational domain.

Keywords: Data mining techniques, WEKA tool, Clustering methods: Canopy, EM, Hierarchical, Simple-k means.

INTRODUCTION

Data mining is becoming a critical component in educational institutions and one of the most important areas of research, with the goal of extracting valuable information from large data sets. Educational data mining (EDM) is an essential study area that can help anticipate meaningful information from educational databases in order to improve educational performance, knowledge, and assessment of students' learning processes. Data mining, also known as knowledge discovery, is becoming increasingly important since it aids in the analysis of data from many viewpoints and its summarization into meaningful information.

1. What is Data Mining?

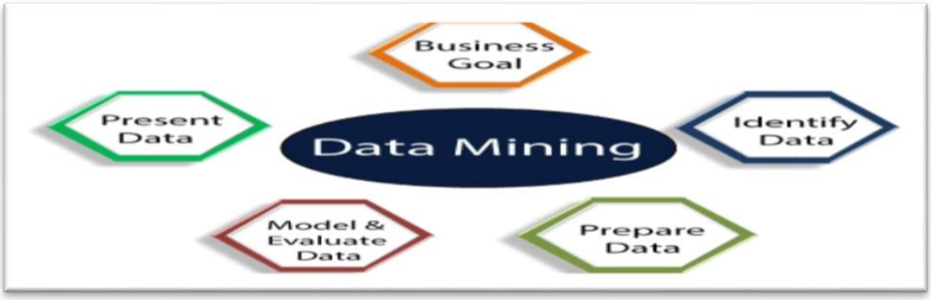
Data Mining is the process of extracting information from large amounts of data to uncover patterns, trends, and usable data that will help a business entity to make data-driven decisions.

Data mining is the process of automatically examining enormous amounts of information to discover trends and patterns that go beyond simple analysis techniques.

Data mining employs complex mathematical algorithms to analyze data segments and determine the likelihood of future events. Data mining is also known as Knowledge Discovery ofData(KDD).

Data mining is a procedure used by corporations to extract specific data from large datasets in order to address business challenges. It primarily converts raw data to useable knowledge.

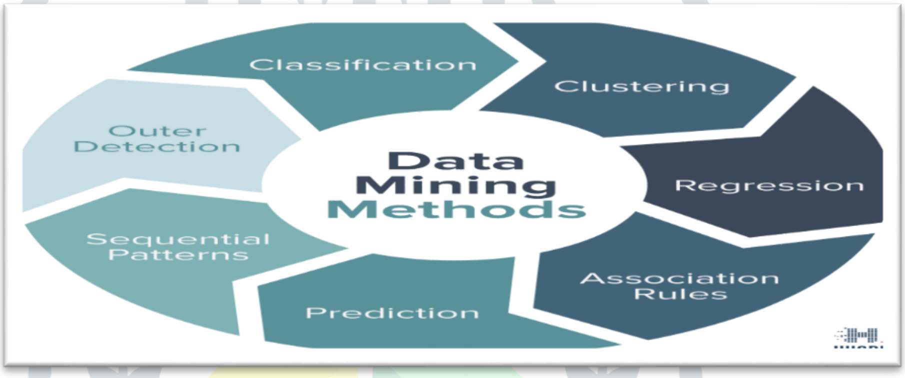
Data Mining is related to Data Science in that it is performed by a person in a specific scenario, on a specific data collection, with a goal in mind. This process involves a variety of services, including text mining, web mining, audio and video mining, graphical data mining, and social media mining. It is accomplished with software that is either simple or highly specific. Outsourcing data mining allows for speedier completion of tasks while keeping operational expenses low. Specialized businesses can also employ new technology to obtain data that is impossible to find manually. There is a wealth of material available across multiple platforms, but relatively little knowledge is accessible. The most difficult challenge is to analyze the data and extract critical information that can be used to address a problem.



2. Data Mining Techniques:

Data mining is the use of sophisticated data analysis technologies to discover previously undiscovered, valid patterns and relationships in large data collections. These technologies can use statistical models, machine learning approaches, and mathematical algorithms like neural networks or decision trees. Thus, data mining includes both analysis and prediction.Professionals in data mining have dedicated their careers to better understanding how to process and draw conclusions from massive amounts of data, but what methods do they use to accomplishthis?

Association, classification, clustering, prediction, sequential patterns, and regression are some of the primary data mining techniques developed and employed in recent projects.



Classification technique:

This strategy is used to gather crucial and relevant data and metadata. This data mining technique aids in classifying data into several categories.
Data mining techniques can be classed using several criteria, as follows:
Data mining frameworks are classified based on the types of data sources mined.
This classification is based on the type of data handled. Examples include multimedia, spatial data, text data, time series data, and the World Wide Web.
Data mining frameworks are classified based on the database involved.
The classification is based on the data model used. For example. There are several types of databases, including object-oriented, transactional, and relational.

Classification of data mining frameworks according to the type of knowledge discovered:This classification is based on the sorts of knowledge obtained or data mining functionalities, such as discrimination, classification, clustering, and characterisation. Some frameworks are comprehensive and provide a variety of data mining features.
Classification of data mining frameworks based on approaches used:This classification is based on the data analysis approach used, such as neural networks, machine learning, genetic algorithms, visualization, statistics, data warehouse- or database-oriented, and so on.
The classification might also consider the extent of user participation engaged in the data mining process, such as query-driven systems, autonomous systems, or interactive exploratory systems.

2.Clustering:

Clustering is the partitioning of information into groups of interconnected things. Describing the data by a few clusters loses some specific details but improves overall performance. It models data using clusters. Data modelling approaches clustering from a historical perspective based on statistics, mathematics, and numerical analysis. Clusters are hidden patterns in machine learning, the search for clusters is unsupervised learning, and the architecture that follows is a data idea. Clustering performs exceptionally well in data mining applications. Examples include scientific data exploration, text mining, information retrieval, spatial database applications, CRM, web analysis, computational biology, medical diagnostics, and much more. In other words, clustering analysis is data.

3.Regression:

Regression analysis is a data mining procedure that identifies and analyzes the relationship between variables based on the existence of another factor. It is used to specify the likelihood of a particular variable. Regression is basically a method of planning and modelling. For example, we may use it to forecast specific costs based on other variables such as availability, consumer demand, and competitiveness. It provides the exact link between two or more variables in a given data collection.

4. Association Rules:

This data mining technique aids in the discovery of a relationship between two or more things. It identifies a hidden pattern in the dataset.
Association rules are if-then statements that help to illustrate the likelihood of interactions between data items inside huge data sets in various types of databases. Association rule mining offers a variety of uses, including improving sales correlations in data or medical data sets.The algorithm works by assuming you have a variety of data. For example, make a list of the groceries goods you've purchased throughout the last six months. It measures the percentage of products purchased together. There are three main measurement techniques:

o Lift:

This measurement technique measures the accuracy of the confidence over how often item B is purchased.

(Confidence) / (item B)/ (Entire dataset) o Support:

This measurement technique measures how often multiple items are purchased and compared it to the overall dataset.

(Item A + Item B) / (Entire dataset) o Confidence:

This measurement technique measures how often item B is purchased when item A is purchased as well.

(Item A + Item B)/ (Item A)

5.Outer detection:

This type of data mining technique relates to the observation of data items in the data set, which do not match an expected pattern or expected behaviour. This technique may be used in various domains like intrusion, detection, fraud detection, etc. It is also known as Outlier analysis or outlier mining. The outlier is a data point that diverges too much from the rest of the dataset. The majority of the real-world datasets have an outlier. Outlier detection plays a significant role in the data mining field. Outlier detection is valuable in numerous fields like network interruption identification, credit or debit card fraud detection, detecting outlying in wireless sensor network data, etc.

6.Sequential Patterns:

The sequential pattern is a data mining technique that focuses on assessing sequential data in order to identify sequential patterns. It entails identifying interesting subsequences within a set of sequences, with the value of a sequence quantified using several parameters such as length, occurrence frequency, and so on. In other words, this data mining technique aids in the discovery or recognition of repeatable patterns in transaction data over time.

7.Prediction:

Prediction used a combination of other data mining techniques such as trends, clustering, classification, etc. It analyses past events or instances in the right sequence to predict a future event.

3. Advantages of Data Mining:

The Data Mining technique enables organizations to obtain knowledge-based data.

Data mining enables organizations to make lucrative modifications in operation and production.

Compared with other statistical data applications, data mining is a cost-efficient. o Data Mining helps the decision-making process of an organization.

It Facilitates the automated discovery of hidden patterns as well as the prediction of trends and behaviour.

It can be induced in the new system as well as the existing platforms.

It is a quick process that makes it easy for new users to analyse enormous amounts of data in a short time.

4 Data Mining Applications:

Data mining is generally utilized by businesses with high consumer demands, such as retail, communication, finance, and marketing firms, to assess pricing, consumer preferences, product placement, and the influence on sales, customer happiness, and corporate profitability. Data mining allows a shop to use point-of-sale records of client purchases to create items and promotions that will assist the company attract customers.

Weka Data Mining

Weka contains a collection of visualization tools and algorithms for data analysis and predictive modelling, together with graphical user interfaces for easy access to these functions. The original non-Java version of Weka was a Tcl/Tk front-end to (mostly third-party) modelling algorithms implemented in other programming languages, plus data preprocessing utilities in C and a make file-based system for running machine learning experiments.

This original version was primarily designed as a tool for analysing data from agricultural domains. Still, the more recent fully Java-based version (Weka 3), developed in 1997, is now used in many different application areas, particularly for educational purposes and research. Weka has the following advantages, such as:Free availability under the GNU General Public License.

Portability, since it is fully implemented in the Java programming language and thus runs on almost any modern computing platform.

A comprehensive collection of data and modelling techniques.

Ease of use due to its graphical user interfaces.

Weka supports several standard data mining tasks, specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection. Input to Weka is expected to be formatted according to the Attribute-Relational File Format and filename with the arff extension.

All Weka's techniques are predicated on the assumption that the data is available as one flat file or relation, where a fixed number of attributes describes each data point (numeric or nominal attributes, but also supports some other attribute types). Weka provides access to SQL databases using Java Database Connectivity and can process the result returned by a database query. Weka provides access to deep learning with Deeplearning4j.

It is not capable of multi-relational data mining. Still, there is separate software for converting a collection of linked database tables into a single table suitable for processing using Weka. Another important area currently not covered by the algorithms included the Weka distribution in sequence modelling.

Farthest First

Farthest first find its variant of k-means, each cluster centre point furthermost from the existing cluster centre is placed by the k mean and this point must be positioned within the data area. So that it greatly speeds up the clustering in most cases but it needs less move and adjustment for their fast performance. It uses heuristic approach for finding its points. It’s arbitrary point is p1, pick an another point p2 far from p1, pick pi to maximize the distance to the nearest of all centroid, the maximize the min dist(pi, p1), dist(pi,p2),...}.After all K representatives are chosen then we define the partition of data area D: cluster is Cj consists of all points closer to pj than to any other representative.

Filtered Cluster

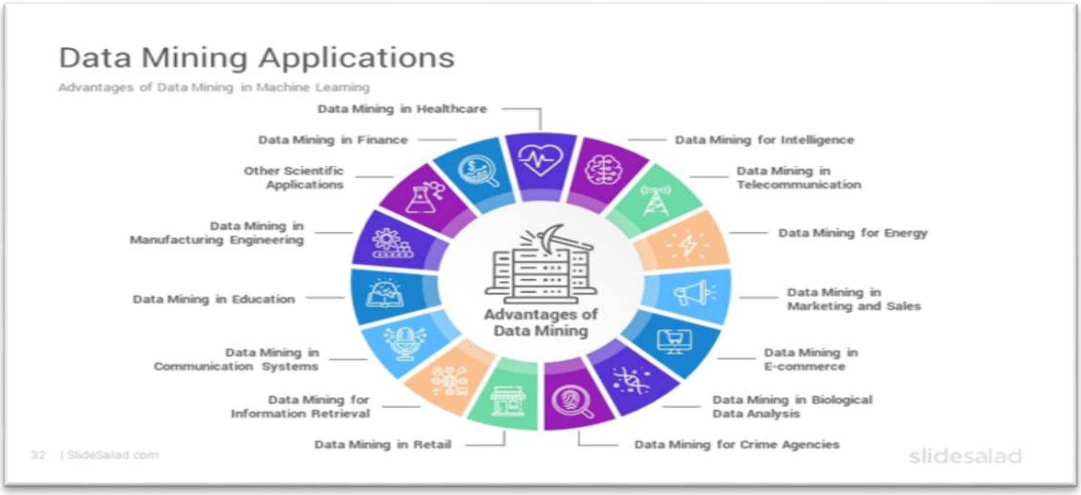
The filtered cluster algorithm is based on storing the multidimensional data points in a kd-tree. The process of the tree is like a binary tree approach, which represents a hierarchical subdivision of its data point set's bounding box using their axis and then splitting is aligned by hyperplanes. Each node of the kd-tree is associated with a closed box, called cell. The root's cell is the bounding box of the point in the dataset. If the cell contains at most one point, then it is declared to be a leaf. Then the finding points in the cell are then partitioned to one side or the other of this hyper plane. The resulting sub cells are the children of the original cell, this leads to a binary tree structure

Density Based Cluster

The DBSCAN is Density-Based clustering algorithms to find clusters based on density of data points in a region and it use only one input parameter for their process so minimal knowledge is required. Density-Based clustering is that for each instance of a cluster the neighbourhood of a given radius (Eps) has to contain at least a minimum number of instances (Min Pts). DBSCAN separates data points into three classes: 1) Core points: points the interior of a cluster, 2) Border points: points neighbourhood of a core point, 3) Noise points: points which is not a core point or a border point. To find a cluster, DBSCAN starts with an arbitrary instance (p) in data set (D) and retrieves all instances of D with respect to Eps and Min Pts.

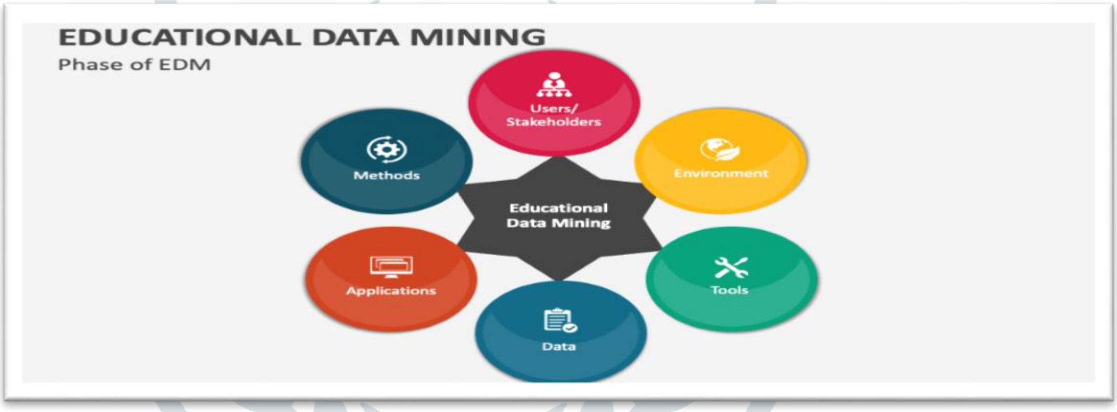
Canopy Algorithm

Canopy Clustering is a very simple, fast and surprisingly accurate method for grouping objects into clusters. All objects are represented as a point in a multidimensional feature space. The algorithm uses a fast-approximate distance metric and two distance thresholds T1 > T2 for processing. The basic algorithm is to begin with a set of points and remove one at random. Create a Canopy containing this point and iterate through the remainder of the point set. At each point, if its distance from the first point is < T1, then add the point to the cluster. If, in addition, the distance is < T2, then remove the point from the set. This way points that are very close to the original will avoid all further processing. The algorithm loops until the initial set is empty, accumulating a set of Canopies, each containing one or more points. A given point may occur in more than one Canopy. Canopy Clustering is often used as an initial step in more rigorous clustering techniques, such as K-Means Clustering . By starting with an initial clustering, the number of more expensive distance measurements can be significantly reduced by ignoring points outside of the initial canopies.


Data Mining Applications
Advantages of Data Mining in Machine Learning
Data Mining in Healthcare
Data Mining for Intelligence
Data Mining in Finance
Other Scientific Applications
Data Mining in Telecommunication
Data Mining in Manufacturing Engineering
Data Mining for Energy
Data Mining in Education
Advantages of Data Mining
Data Mining in Marketing and Sales
Data Mining in Communication Systems
Data Mining in E-commerce
Data Mining for Information Retrieval
Data Mining in Biological Data Analysis
Data Mining in Retail
Data Mining for Crime Agencies
32 | SlideSalad.com
slidesalad

5. Data mining in Education:

Education data mining is a new subject that focuses on developing tools to extract knowledge from data created in educational environments. EDM objectives are identified as validating students' future learning behaviours, researching the impact of educational support, and advancing learning sciences. Data mining may help an organization make precise decisions as well as forecast student outcomes. With the results, the school may focus on what and how to teach.



METHODOLOGY

There are numerous Machine learning algorithms in the literature and some of the important and popular algorithms based on Clustering.

1. Canopy clustering:

Canopy Clustering is a very simple, fast and surprisingly accurate method for grouping objects into clusters. All objects are represented as a point in a multidimensional feature space. The algorithm uses a fast-approximate distance metric and two distance thresholds T1 > T2 for processing.

Step-1: Go to WEKA software

Step-2: Select data by using open file.

Step-3: Go to cluster and select Canopy from choose option.

Step-4: Click start button to run the program.

2. EM Clustering:

For clustering, EM makes use of the finite Gaussian mixtures model and estimates a set of parameters iteratively until a desired convergence value is achieved.

Step-1: Go to WEKA software

Step-2: Select data by using open file.

Step-3: Go to cluster and select EM from choose option.

Step-4: Click start button to run the program.

3. Hierarchical Clustering:

Hierarchical clustering is mainly focussing on building of hierarchy of clusters, i.e., cluster tree and it is represented in a dendrogram. It is either merging smaller clusters into larger clusters or splitting larger clusters into smaller ones. A clustering of the data items is obtained through cutting a dendrogram at a desired level.

Step-1: Go to WEKA software

Step-2: Select data by using open file.

Step-3: Go to cluster and select Hierarchical from choose option.

Step-4: Click start button to run the program.

4. Simple-k means clustering:

K-means clustering is a simple unsupervised learning algorithm. In this, the data objects (‘n’) are grouped into a total of ‘k’ clusters, with each observation belonging to the cluster with the closest mean. It defines ‘k’ sets, one for each cluster k n (the point can be thought of as the centre of a one or two-dimensional figure). The clusters are separated by a large distance.

Step-1: Go to WEKA software

Step-2: Select data by using open file.

Step-3: Go to cluster and select simple k means from choose option.

Step-4: Click start button to run the program.

BACKGROUND

Here is the background tool required for our project.

We have used a data mining software named as WEKA for this project.

Required Software (WEKA) –

What is WEKA?

WEKA (Waikato Environment for Knowledge) Analysis it's a data mining/machine learning tool developed by Department of Computer Science, University of Waikato, New Zealand.

History of WEKA

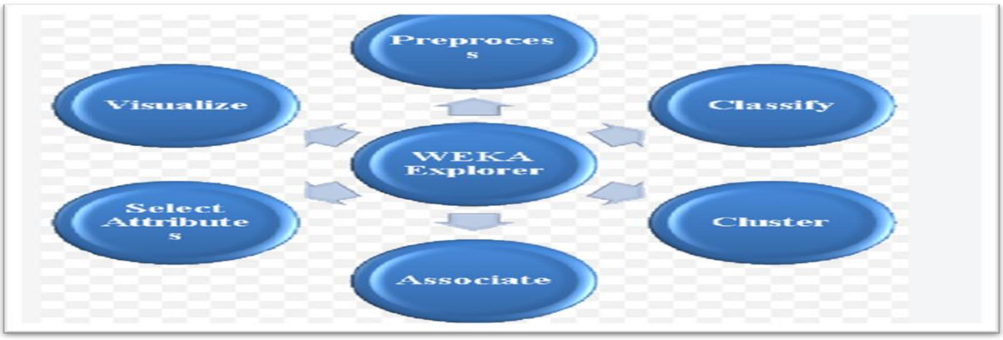
In 1993, the University of Waikato in New Zealand began the development of the original version of Weka, which became a mix of Tcl/Tk, C, and make files.

In 1997, the decision was made to redevelop Weka from scratch in Java, including implementing modelling algorithms.

In 2005, Weka received the SIGKDD Data Mining and Knowledge Discovery Service Award.

In 2006, Pentaho Corporation acquired an exclusive licence to use Weka for business intelligence. It forms the data mining and predictive analytics component of the Pentaho business intelligence suite. Hitachi Vantara has since acquired Pentaho, and Weka now underpins the PMI (Plugin for Machine Intelligence) open-source component.

Features of Weka:



Weka has the following features, such as:

Preprocess:

Preprocessing data is a critical step in data mining. Because the majority of the data is raw, there is a possibility that it contains empty or duplicate entries, junk values, outliers, extra columns, or a different naming convention. All of this degrades the outcomes.

To make data cleaner, better, and more thorough, WEKA provides a broad set of filter choices. The tool supports both supervised and unsupervised activities. Here is a list of some operations for preprocessing.
Replace Missing with User Constant: to resolve the empty or null value issue.
Reservoir Sample: create a random subset of sample data.
Nominal to Binary: Converting data from nominal to binary.
Remove Percentage: To remove

Classify:

One of the most important activities in machine learning is classification, which involves assigning classes or categories to things. Classic examples of classification include labeling a brain tumor as "malignant" or "benign" or categorizing an email as "spam" or "not spam."
Following the selection of the appropriate classifier, we choose test choices for the training set. Some of the alternatives include:
Use the training set: the classifier will be tested using the same training set.
A given test set evaluates the classifier using a separate test set.
Cross-validation Folds: The classifier's performance is evaluated using cross-validation and the number of folds provided.
proportion split: The classifier will be evaluated using a given proportion of the data.
Aside from these, we can also utilize more testing options.

Cluster:

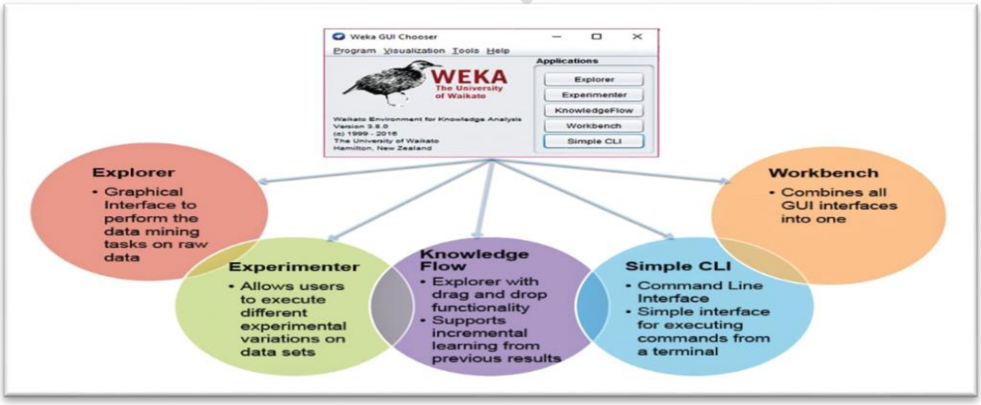
Clustering divides a dataset into groups/clusters based on certain commonalities. In this scenario, the items in the same cluster are identical but distinct from those in other clusters. Clustering can be defined as identifying customers who exhibit similar behaviors and arranging regions based on homogeneous land use.

Associate:

Association rules emphasize all of the relationships and connections between elements in a dataset. In a nutshell, it is an if-then expression that represents the likelihood of associations between data points. A typical example of association is the connection between the selling of milk and bread.
For association rule mining in this category, the tool includes the Apriori, Filtered Associator, and FP Growth algorithms.

Select Attributes:

Every dataset contains many attributes, yet some of them may not be particularly valuable. As a result, reducing extraneous details while retaining vital ones is critical for creating a good model.
Many attribute evaluators and search algorithms exist, including Best First, Greedy Stepwise, and Ranker.
Visualize:
The visualization tab contains a variety of plot matrices and graphs that display the model's discovered trends and mistakes.
Weka Requirements and Installation
We can install WEKA on Windows, Mac OS, and Linux. The current stable versions of Weka require at least Java 8 or above.


Weka GUI Chooser
Program Visualization Tools Help
Applications
Explorer
Experimenter
KnowledgeFlow
Workbench
Simple CLI
WEKA
The University of Waikato
Waikato Environment for Knowledge Analysis
Version 3.8.0
(c) 1999 - 2016
The University of Waikato
Hamilton, New Zealand
Explorer
• Graphical Interface to perform the data mining tasks on raw data
Experimenter
• Allows users to execute different experimental variations on data sets
Knowledge Flow
• Explorer with drag and drop functionality
• Supports incremental learning from previous results
Simple CLI
• Command Line Interface
• Simple interface for executing commands from a terminal
Workbench
• Combines all GUI interfaces into one

Main GUI, Five graphical user interfaces

"The Explorer" (exploratory data analysis)

"The Experimenter" (experimental environment)

"The Knowledge Flow" (new process model inspired interface)

Workbench

Simple CLI (Command prompt)

Offers some functionality not available via the GUI

Explorer:

The WEKA Explorer windows show different tabs starting with preprocessing. Initially, the preprocess tab is active, as first the data set is pre-processed before applying algorithms to it and exploring the dataset.

The tabs are as follows:

Preprocess: Choose and modify the loaded data.

Classify: Apply training and testing algorithms to the data that will classify and regress the data.

Cluster: Form clusters from the data.

Associate: Mine out the association rule for the data.

Select attributes: Attribute selection measures are applied.

Visualize: 2D representation of data is seen.

Status Bar: The bottommost section of the window shows the status bar. This section shows what is happening currently in the form of a message, such as a file being loaded. Right-click on this, Memory information can be seen, and also Run garbage collector to free up space can be run.

Log Button: It stores a log of all actions in Weka with the timestamp. Logs are shown in a separate window when the Log button is clicked.

WEKA Bird Icon: Present on the bottom right corner shows the WEKA bird with represents the number of processes running concurrently (by x.). When the process is running the bird will move around.

Experimenter:

The WEKA experimenter button allows the users to create, run, and modify different schemes in one experiment on a dataset. The experimenter has 2 types of configurations: Simple and Advanced. Both configurations allow users to run experiments locally and on remote computers.

The “Open” and “New” buttons will open a new experiment window that users can do.

Results: Set the result destination file from ARFF, JDFC, and CSV files.

Experiment Type: The user can choose between cross-validation and train/test.

percentage split: The user can choose between Classification and Regression-based upon the dataset and classifier used.

Datasets: The user can browse and select datasets from here. The relative path checkbox is clicked if working on different machines. The format of datasets supported is ARFF, C4.5, CSV, libsvm, bsi, and XRFF.

Iteration: The default iteration number is set to 10. Datasets first and algorithms first help in switching between dataset and algorithms so that algorithms can be run on all datasets.

Algorithms: New algorithms are added by “New Button”. The user can choose a classifier.

Save the experiment using the Save button.Run the experiment using the Run button.

Knowledge flow:

Knowledge flow shows a graphical representation of WEKA algorithms. The user can select the components and create a workflow to analyze the datasets. The data can be handled batch-wise or incrementally. Parallel workflows can be designed and each will run in a separate thread.

Weka Workbench:

The Weka Workbench is an environment that combines all of the GUI interfaces into a single interface.

It is useful if you find yourself jumping a lot between two or more different interfaces, such as between the Explorer and the Experiment Environment. This can happen if you try out a lot of what ifs in the Explorer and quickly take what you learn and put it into controlled experiments.

Weka Simple CLI:

Weka can be used from a simple Command Line Interface (CLI).

This is powerful because you can write shell scripts to use the full API from command line calls with parameters, allowing you to build models, run experiments and make predictions without a graphical user interface.

The Simple CLI provides an environment where you can quickly and easily experiment with the Weka command line interface commands.

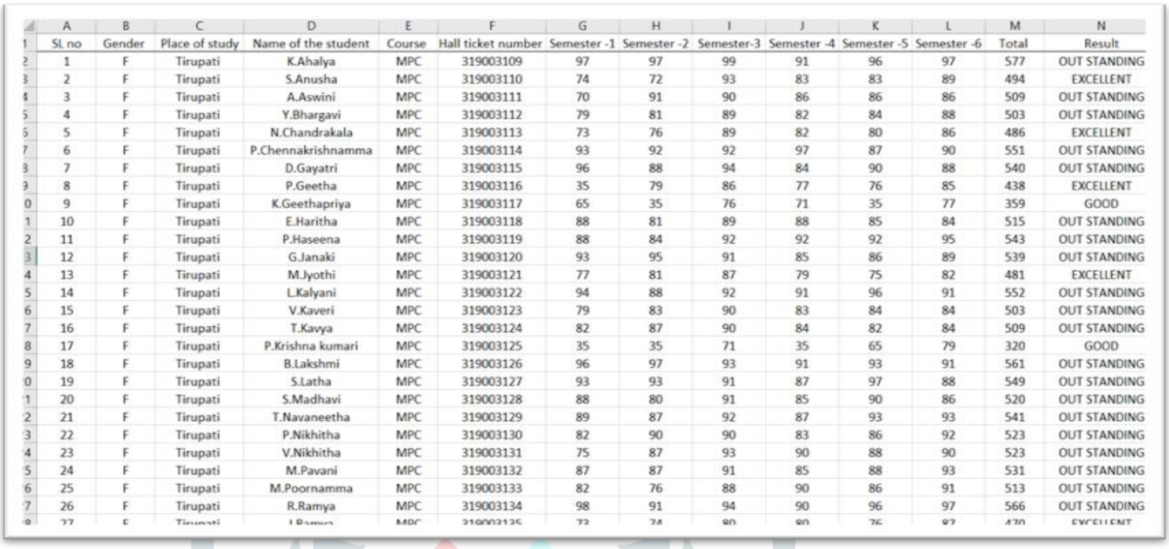
The different components available are Data sources, Data savers, Filters, Classifiers, Clusters, Evaluation, and Visualization.

DESIGN AND IMPLEMENTATION

The following are the step-by-step process of our project evaluation.

Collection of data: We have a collected a data set contains the result of students of overall semesters. The data set contains 255 instances and 14 attributes.

Here is the sample of our data set which is in entered in MS-Excel and saved in 'CSV' format.



SL no	Gender	Place of study	Name of the student	Course	Hall ticket number	Semester -1	Semester -2	Semester-3	Semester -4	Semester -5	Semester -6	Total	Result
1	F	Tirupati	K.Ahalya	MPC	319003109	97	97	99	91	96	97	577	OUT STANDING
2	F	Tirupati	S.Anusha	MPC	319003110	74	72	93	83	83	89	494	EXCELLENT
3	F	Tirupati	A.Aswini	MPC	319003111	70	91	90	86	86	86	509	OUT STANDING
4	F	Tirupati	Y.Bhargavi	MPC	319003112	79	81	89	82	84	88	503	OUT STANDING
5	F	Tirupati	N.Chandrakala	MPC	319003113	73	76	89	82	80	86	486	EXCELLENT
6	F	Tirupati	P.Chennakrishnamma	MPC	319003114	93	92	92	97	87	90	551	OUT STANDING
7	F	Tirupati	D.Gayatri	MPC	319003115	96	88	94	84	90	88	540	OUT STANDING
8	F	Tirupati	P.Geetha	MPC	319003116	35	79	86	77	76	85	438	EXCELLENT
9	F	Tirupati	K.Geethapriya	MPC	319003117	65	35	76	71	35	77	359	GOOD
10	F	Tirupati	E.Haritha	MPC	319003118	88	81	89	88	85	84	515	OUT STANDING
11	F	Tirupati	P.Haseena	MPC	319003119	88	84	92	92	92	95	543	OUT STANDING
12	F	Tirupati	G.Janaki	MPC	319003120	93	95	91	85	86	89	539	OUT STANDING
13	F	Tirupati	M.Jyothi	MPC	319003121	77	81	87	79	75	82	481	EXCELLENT
14	F	Tirupati	L.Kalyani	MPC	319003122	94	88	92	91	96	91	552	OUT STANDING
15	F	Tirupati	V.Kaveri	MPC	319003123	79	83	90	83	84	84	503	OUT STANDING
16	F	Tirupati	T.Kavya	MPC	319003124	82	87	90	84	82	84	509	OUT STANDING
17	F	Tirupati	P.Krishna kumari	MPC	319003125	35	35	71	35	65	79	320	GOOD
18	F	Tirupati	B.Lakshmi	MPC	319003126	96	97	93	91	93	91	561	OUT STANDING
19	F	Tirupati	S.Latha	MPC	319003127	93	93	91	87	97	88	549	OUT STANDING
20	F	Tirupati	S.Madhavi	MPC	319003128	88	80	91	85	90	86	520	OUT STANDING
21	F	Tirupati	T.Navaneetha	MPC	319003129	89	87	92	87	93	93	541	OUT STANDING
22	F	Tirupati	P.Nikhitha	MPC	319003130	82	90	90	83	86	92	523	OUT STANDING
23	F	Tirupati	V.Nikhitha	MPC	319003131	75	87	93	90	88	90	523	OUT STANDING
24	F	Tirupati	M.Pavani	MPC	319003132	87	87	91	85	88	93	531	OUT STANDING
25	F	Tirupati	M.Poornamma	MPC	319003133	82	76	88	90	86	91	513	OUT STANDING
26	F	Tirupati	R.Ramya	MPC	319003134	98	91	94	90	96	97	566	OUT STANDING
27	F	Tirupati	I.Ramya	MPC	319003135	73	74	80	80	76	87	470	EXCELLENT

Preprocessing:

Data Pre-processing is the first step of evaluation of this paper. Now we will choose WEKA Explorer interface. Here the source data file is selected from local machine. After loading the data in explorer, we can refine the data by selecting different options which is known as 'data cleaning' and can also select or remove attributes as per our need

It's the nut shell about WEKA and its application in education system

CLUSTER ANALYSIS IN WEKA TOOL

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters). It is a main task of exploratorydata mining, and a common technique for statistical data analysis, used in many fields, including pattern recognition,image analysis, information retrieval, bioinformatics, data compression, computer graphics and machine learning.

Besides the term clustering, there are a number of terms with similar meanings, including automatic classification, numerical taxonomy, bornology, typological analysis, and community detection. The subtle differences are often in the use of the results: while in data mining, the resulting groups are the matter of interest, in automatic classification the resulting discriminative power is of interest.

Cluster Algorithm using WEKA tool

Clustering is a task for which many algorithms have been proposed. No clustering technique is Universally applicable, and different techniques are in favour for different clustering purposes.

So, an understanding of both the clustering problem and the clustering technique is required to Apply a suitable method to a given problem. In the following, I describe general of a Clustering technique algorithm in weak tool show in figure.

Clustering is the method of dividing a set of abstract objects into groups. Points to Keep in Mind A set of data objects can be viewed as a single entity. When performing cluster analysis, we divide the data set into groups based on data similarity, then assign labels to the groups.

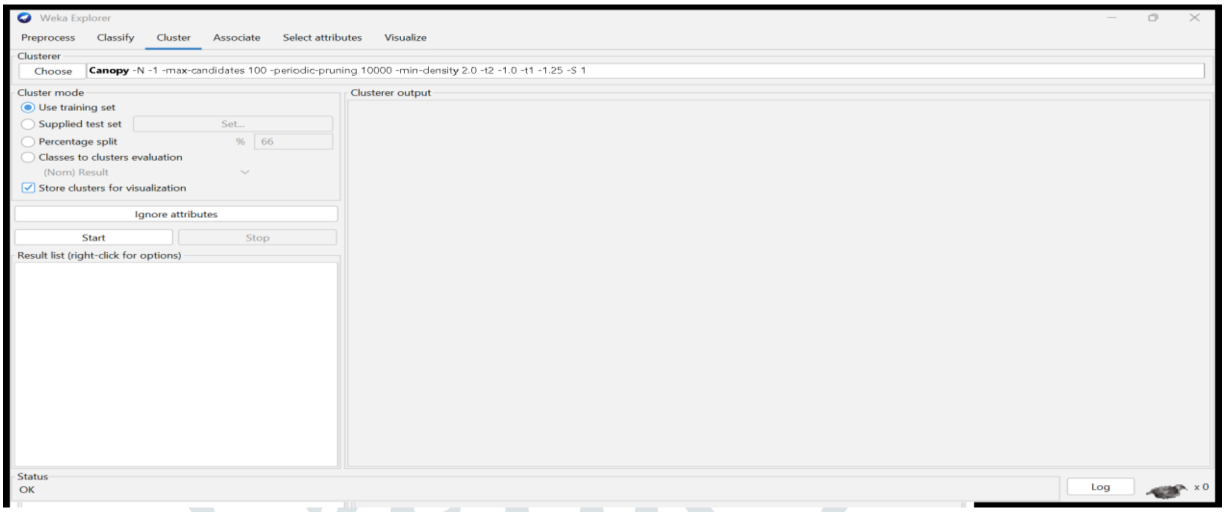
COMWEB Algorithm:

COBWEB is an incremental system for hierarchical conceptual clustering. COBWEB was invented by Professor Douglas H. Fisher, currently at Vanderbilt University. COBWEB incrementally organizes observations into a classification tree. Each node in a classification tree represents a class (concept) and is labelled by a probabilistic concept that summarizes the attribute-value distributions of objects classified under the node. This classification tree can be used to predict missing attributes or the class of a new object.

Steps to be followed:

Step 1: In the preprocessing interface, open the Weka Explorer and load the required dataset, and we are taking the scheme. raff dataset.

Step 2: Find the 'cluster' tab in the explorer and press the choose button to execute clustering. A dropdown list of available clustering algorithms appears as a result of this step and selects the canopy algorithm.



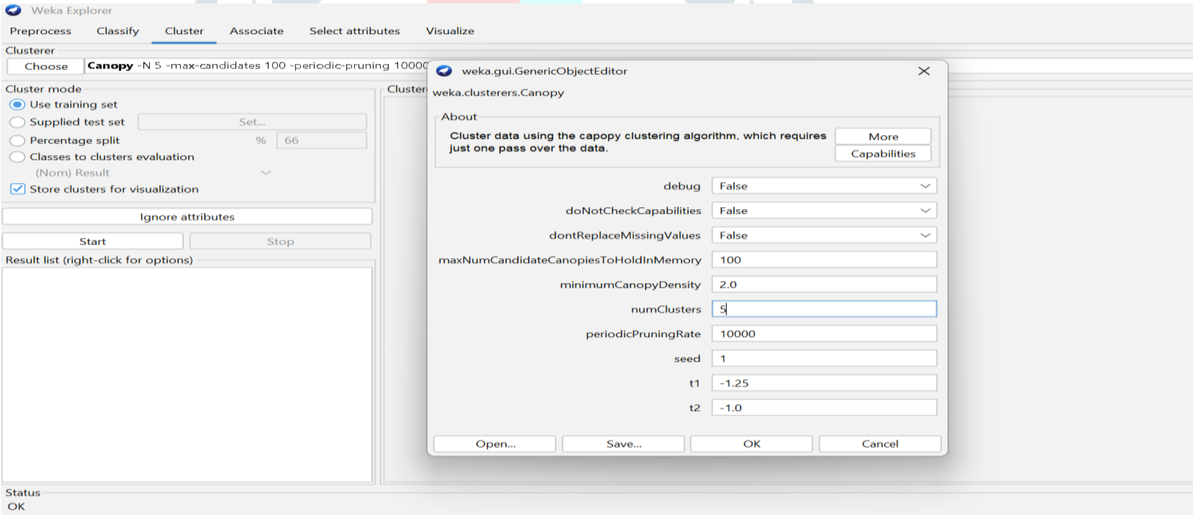
Clusterer: **Canopy** -N -1 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t2 -1.0 -t1 -1.25 -S 1

Cluster mode
- ● Use training set
- ○ Supplied test set
- ○ Percentage split % 66
- ○ Classes to clusters evaluation (Nom) Result
- ☑ Store clusters for visualization

Result list (right-click for options)

Status: OK

Step 3: Then, to the right of the choose icon, press the text button to bring up the popup window. We enter five for the number of clusters in this window and the seed value is one. The seed value is used to generate a random number that is used to make internal assignments of instances of clusters.



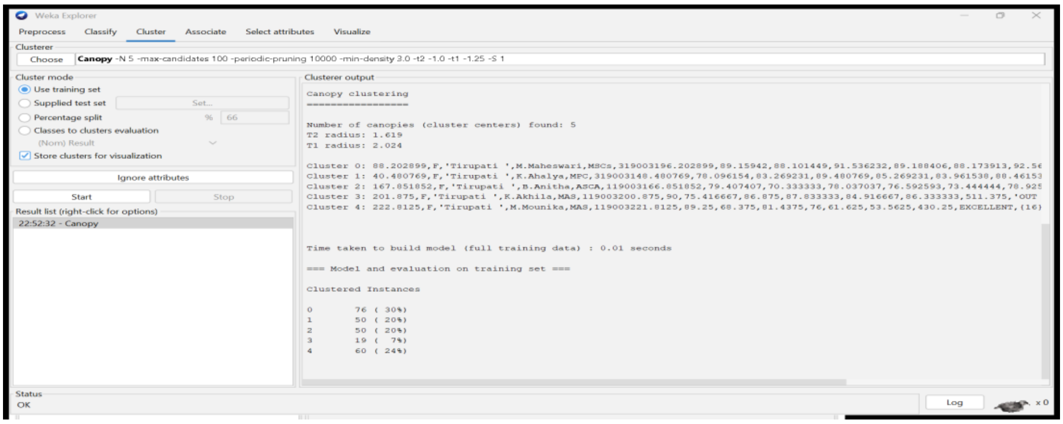
Clusterer: **Canopy** -N 5 -max-candidates 100 -periodic-pruning 10000

weka.clusterers.Canopy

About: Cluster data using the capopy clustering algorithm, which requires just one pass over the data.

debug	False
doNotCheckCapabilities	False
dontReplaceMissingValues	False
maxNumCandidateCanopiesToHoldInMemory	100
minimumCanopyDensity	2.0
numClusters	5
periodicPruningRate	10000
seed	1
t1	-1.25
t2	-1.0

Step 4: One of the choices has been chosen. We must ensure that they are in the 'cluster mode' panel before running the clustering algorithm. The choice to use a training set is selected, and then the 'start' button is pressed.



Clusterer: **Canopy** -N 5 -max-candidates 100 -periodic-pruning 10000 -min-density 3.0 -t2 -1.0 -t1 -1.25 -S 1

Result list (right-click for options): 22:52:32 - Canopy

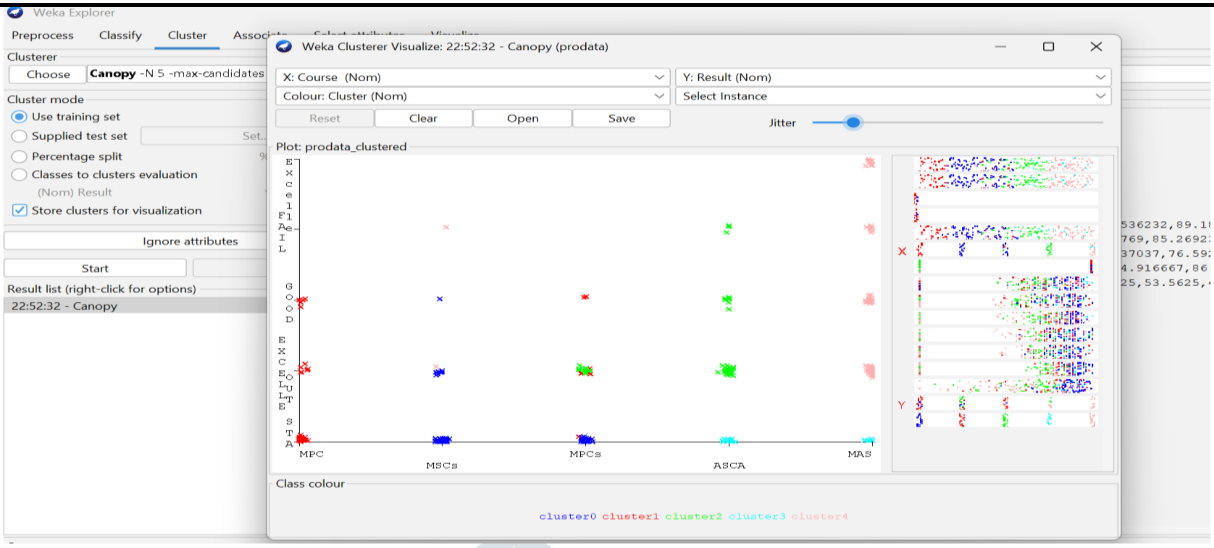
Clusterer output

```
Canopy clustering
=================

Number of canopies (cluster centers) found: 5
T2 radius: 1.619
T1 radius: 2.024

Cluster 0: 88.202899,F,'Tirupati ',M.Maheswari,MSCs,319003196.202899,89.15942,88.101449,91.536232,89.188406,88.173913,92.56
Cluster 1: 40.480769,F,'Tirupati ',K.Ahalya,MPC,319003148.480769,78.096154,83.269231,89.480769,85.269231,83.961538,88.46153
Cluster 2: 167.851852,F,'Tirupati ',B.Anitha,ASCA,119003166.851852,79.407407,70.333333,78.037037,76.592593,73.444444,78.925
Cluster 3: 201.875,F,'Tirupati ',K.Akhila,MAS,119003200.875,90,75.416667,86.875,87.833333,84.916667,86.333333,511.375,'OUT
Cluster 4: 222.8125,F,'Tirupati ',M.Mounika,MAS,119003221.8125,89.25,68.375,81.4375,76,61.625,53.5625,430.25,EXCELLENT,(16)



Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       76 ( 30%)
1       50 ( 20%)
2       50 ( 20%)
3       19 (  7%)
4       60 ( 24%)
```

Status: OK

The centroid of each cluster is shown in the result window, along with statistics on the number and percent of instances allocated to each cluster. Each cluster centroid is represented by a mean vector. This cluster can be used to describe a cluster.

Step 6: Another way to grasp the characteristics of each cluster is to visualize them. To do so, right-click the result set on the result. Selecting to visualize cluster assignments from the list column.



Step 7: In the Cluster visualization we choose Course on X-axis and Result no on Y-axis. Blue colour shows Cluster 0, Red colour shows Cluster 1, Green colour shows Cluster 2, Sky-blue colour shows Cluster 3, Light pink colour shows Cluster 4.

Output:

This canopy algorithm is tested with project dataset in WEKA tool; it produces five different clustered instance clusters 0: 76(30%), cluster 1: 50(20%), cluster 2: 50(20%), cluster 3: 19(7%), cluster 4: 60(24%), and Time taken to build model (full training data): 0.01 seconds.

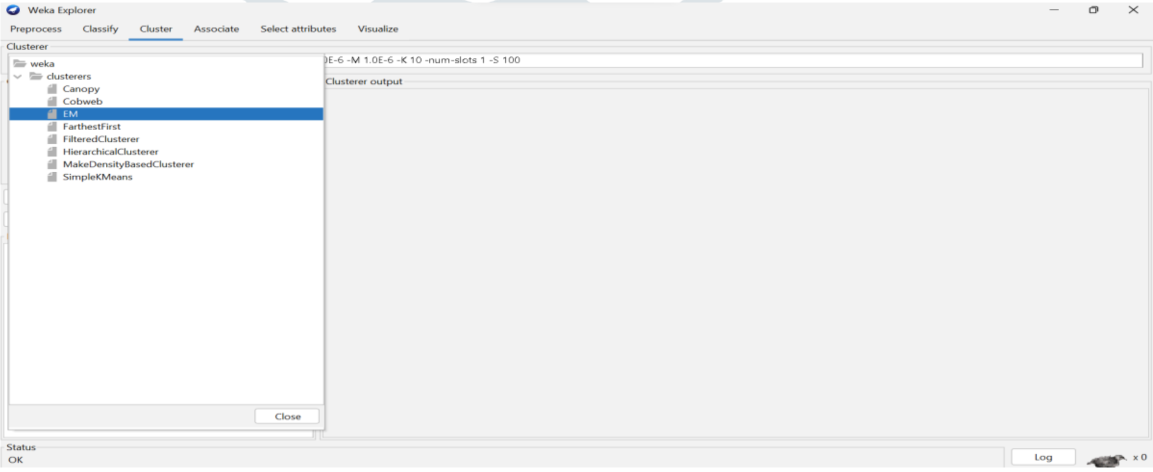
EM:

The EM algorithm (Dempster, Laird, & Rubin 1977) finds maximum likelihood estimates of parameters in probabilistic models. EM is an iterative method which alternates between two steps, expectation (E) and maximization (M). For clustering, EM makes use of the finite Gaussian mixtures model and estimates a set of parameters iteratively until a desired convergence value is achieved. The mixture is defined as a set of K probability distributions and each distribution corresponds to one cluster. An instance is assigned with a membership probability for each cluster.

Steps to be followed:

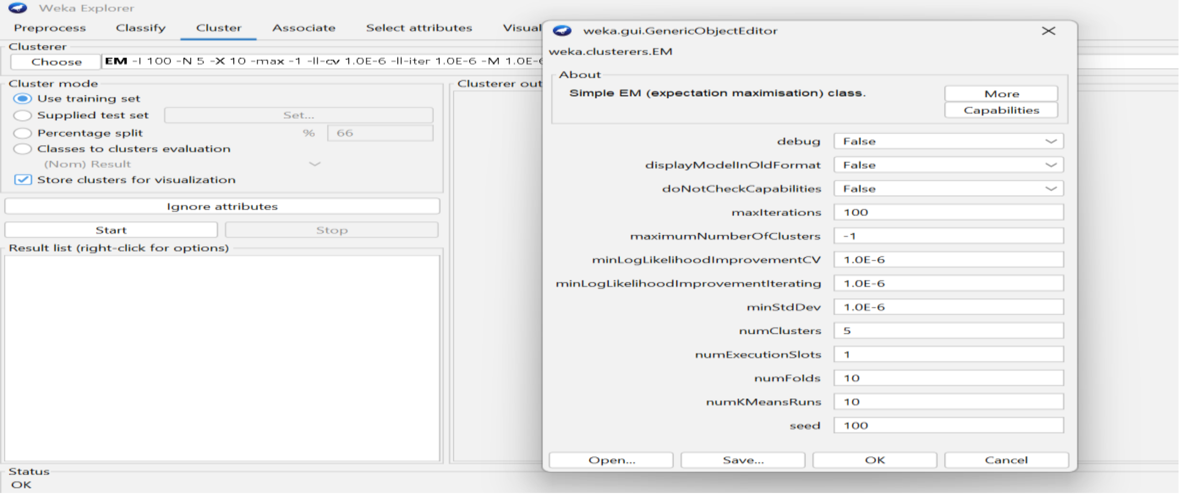
Step 1: In the preprocessing interface, open the Weka Explorer and load the required dataset, and we are taking the project.arff dataset.

Step 2: Find the 'cluster' tab in the explorer and press the choose button to execute clustering. A dropdown list of available clustering algorithms appears as a result of this step and selects the EM algorithm.



0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100

Step 3: Then, to the right of the choose icon, press the text button to bring up the popup window. We enter five for the number of clusters in this window and the seed value is hundred. The seed value is used to generate a random number that is used to make internal assignments of instances of clusters.



Clusterer: EM -I 100 -N 5 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6

Cluster mode: ● Use training set; ○ Supplied test set; ○ Percentage split % 66; ○ Classes to clusters evaluation (Nom) Result; ☑ Store clusters for visualization

weka.gui.GenericObjectEditor

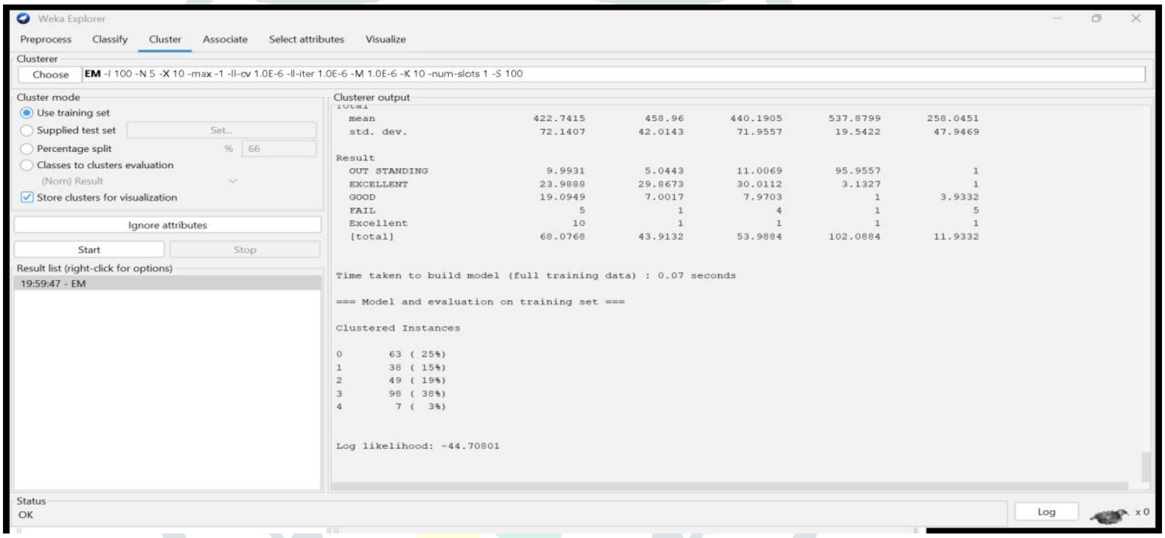
weka.clusterers.EM

About: Simple EM (expectation maximisation) class.

debug	False
displayModelInOldFormat	False
doNotCheckCapabilities	False
maxIterations	100
maximumNumberOfClusters	-1
minLogLikelihoodImprovementCV	1.0E-6
minLogLikelihoodImprovementIterating	1.0E-6
minStdDev	1.0E-6
numClusters	5
numExecutionSlots	1
numFolds	10
numKMeansRuns	10
seed	100

Status: OK

Step 4: One of the choices has been chosen. We must ensure that they are in the 'cluster mode' panel before running the clustering algorithm. The choice to use a training set is selected, and then the 'start' button is pressed.



Clusterer: EM -I 100 -N 5 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100

Cluster mode: ● Use training set; ○ Supplied test set; ○ Percentage split % 66; ○ Classes to clusters evaluation (Nom) Result; ☑ Store clusters for visualization

Result list: 19:59:47 - EM

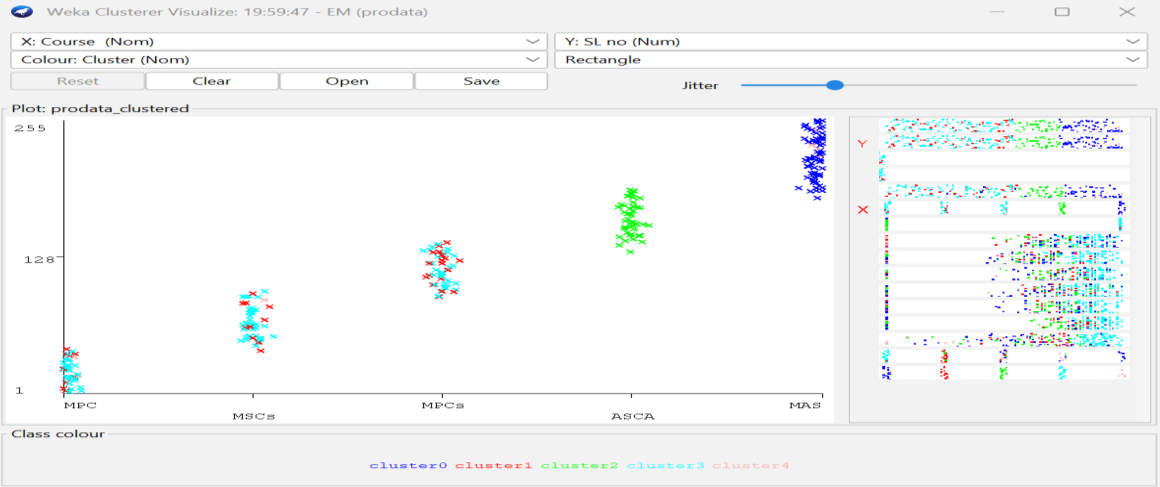
Clusterer output:

```
  mean                        422.7415      458.96    440.1905    537.8799    258.0451
  std. dev.                    72.1407     42.0143     71.9557     19.5422     47.9469

Result
  OUT STANDING                  9.9931      5.0443     11.0069     95.9557           1
  EXCELLENT                    23.9888     29.8673     30.0112      3.1327           1
  GOOD                         19.0949      7.0017      7.9703           1      3.9332
  FAIL                               5           1           4           1           5
  Excellent                         10           1           1           1           1
  [total]                      68.0768     43.9132     53.9884    102.0884     11.9332


Time taken to build model (full training data) : 0.07 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       63 ( 25%)
1       38 ( 15%)
2       49 ( 19%)
3       98 ( 38%)
4        7 (  3%)


Log likelihood: -44.70801
```

Status: OK

Step 5: The centroid of each cluster is shown in the result window, along with statistics on the number and percent of instances allocated to each cluster. Each cluster centroid is represented by a mean vector. This cluster can be used to describe a cluster.

Step 6: Another way to grasp the characteristics of each cluster is to visualize them. To do so, right-click the result set on the result. Selecting to visualize cluster assignments from the list column.



Weka Clusterer Visualize: 19:59:47 - EM (prodata)

X: Course (Nom)

Y: SL no (Num)

Colour: Cluster (Nom)

Rectangle

Plot: prodata_clustered

Step 7: In the Cluster visualization we choose Course on X-axis and SL no on Y-axis.

Blue colour shows Cluster 0, Red colour shows Cluster 1, Green colour shows Cluster 2, Sky-blue colour shows Cluster 3, Light pink colour shows Cluster 4.

Output: This EM algorithm is tested with project dataset in WEKA tool. It produces five different clustered instance clusters 0: 63(25%), cluster 1: 38(15%), cluster 2: 49(19%), cluster 3: 98(38%), cluster 4: 7(3%), and Time taken to build model (full training data): 0.01 seconds.

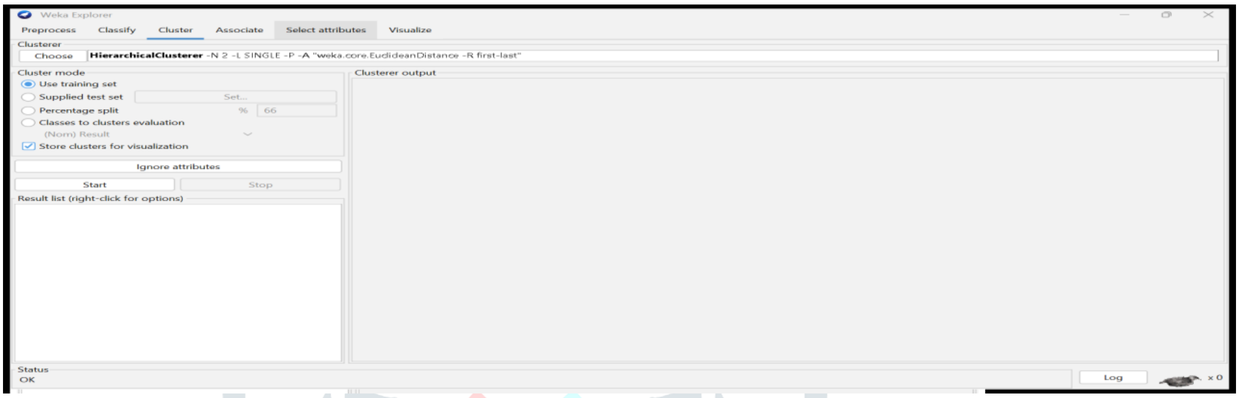
Hierarchical Cluster

Hierarchical clustering is mainly focussing on building of hierarchy of clusters, i.e., cluster tree and it is represented in a dendrogram. It is either merging smaller clusters into larger clusters or splitting larger clusters into smaller ones. A clustering of the data items is obtained through cutting a dendrogram at a desired level. A cluster tree is defined as "a tree showing a sequence of clustering with each clustering being a partition of the data set". Following are the general procedures for performing hierarchical clustering.

Steps to be followed:

Step 1: In the preprocessing interface, open the Weka Explorer and load the required dataset, and we are taking the project. arff dataset.

Step 2: Find the 'cluster' tab in the explorer and press the choose button to execute clustering. A dropdown list of available clustering algorithms appears as a result of this step and selects the Hierarchical Clustered algorithm.



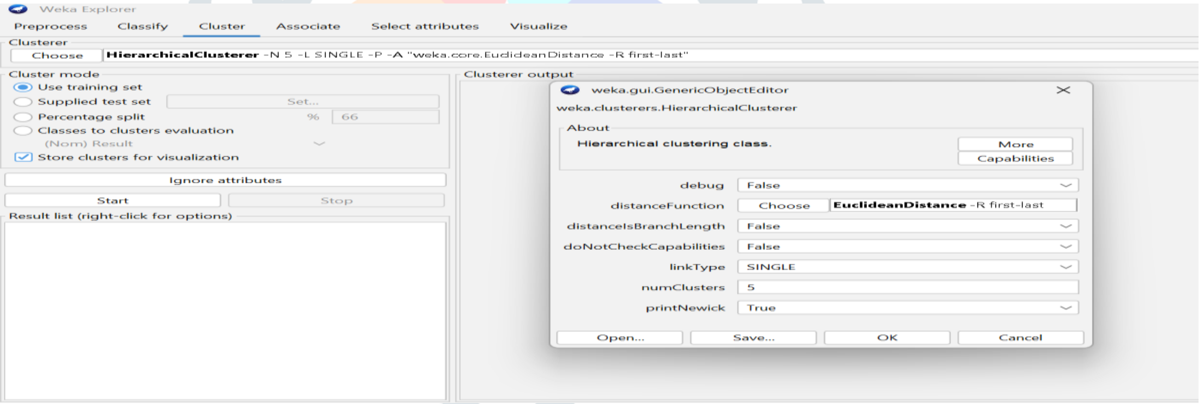
Clusterer: HierarchicalClusterer -N 2 -L SINGLE -P -A "weka.core.EuclideanDistance -R first-last"

Cluster mode: ● Use training set

☑ Store clusters for visualization

Status: OK

Step 3: Then, to the right of the choose icon, press the text button to bring up the popup window. We enter five for the number of clusters in this window.



Clusterer: HierarchicalClusterer -N 5 -L SINGLE -P -A "weka.core.EuclideanDistance -R first-last"

weka.gui.GenericObjectEditor

weka.clusterers.HierarchicalClusterer

About: Hierarchical clustering class.

debug: False

distanceFunction: EuclideanDistance -R first-last

distanceIsBranchLength: False

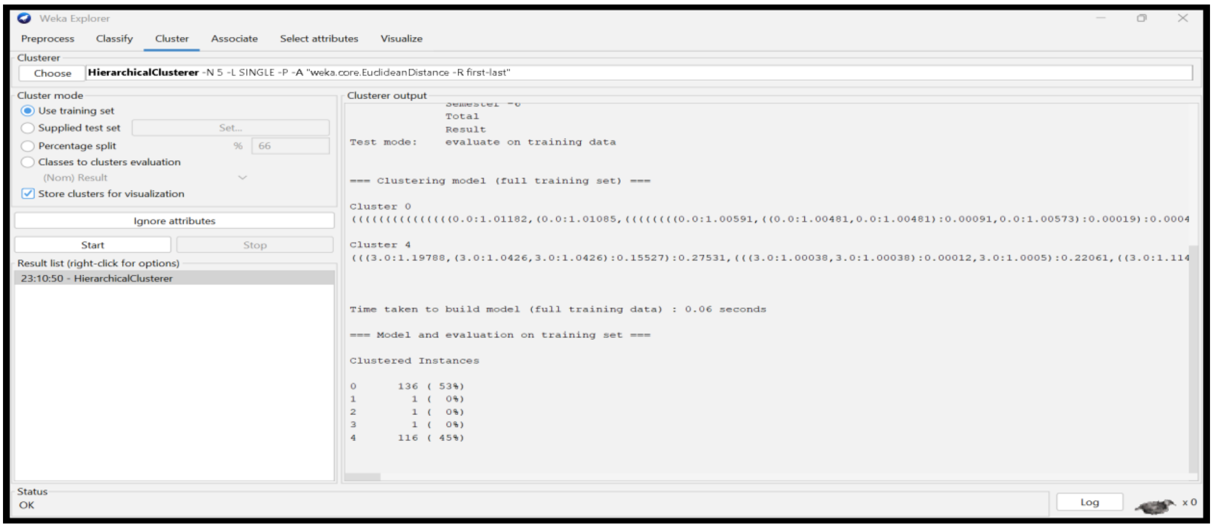
doNotCheckCapabilities: False

linkType: SINGLE

numClusters: 5

printNewick: True

Step 4: One of the choices has been chosen. We must ensure that they are in the 'cluster mode' panel before running the clustering algorithm. The choice to use a training set is selected, and then the 'start' button is pressed.



Clusterer: HierarchicalClusterer -N 5 -L SINGLE -P -A "weka.core.EuclideanDistance -R first-last"

Cluster mode: ● Use training set

☑ Store clusters for visualization

Result list: 23:10:50 - HierarchicalClusterer

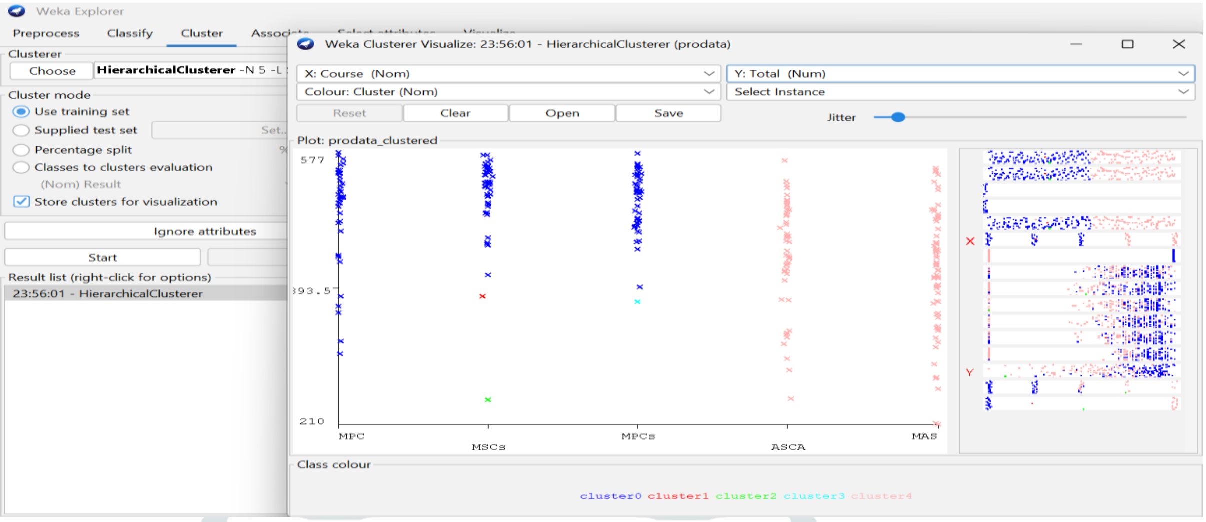
Clusterer output:

```
              Semester -6
              Total
              Result
Test mode:    evaluate on training data


=== Clustering model (full training set) ===

Cluster 0
((((((((((((((0.0:1.01182,(0.0:1.01085,(((((((0.0:1.00591,((0.0:1.00481,0.0:1.00481):0.00091,0.0:1.00573):0.00019):0.0004

Cluster 4
(((3.0:1.19788,(3.0:1.0426,3.0:1.0426):0.15527):0.27531,(((3.0:1.00038,3.0:1.00038):0.00012,3.0:1.0005):0.22061,((3.0:1.114



Time taken to build model (full training data) : 0.06 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       136 ( 53%)
1         1 (  0%)
2         1 (  0%)
3         1 (  0%)
4       116 ( 45%)
```

Status: OK

Step 5: The centroid of each cluster is shown in the result window, along with statistics on the number and percent of instances allocated to each cluster. Each cluster centroid is represented by a mean vector. This cluster can be used to describe a cluster. Merging smaller clusters into larger clusters or splitting larger clusters into smaller ones.

Step 6: Another way to grasp the characteristics of each cluster is to visualize them. To do so, right-click the result set on the result. Selecting to visualize cluster assignments from the list column.



Weka Clusterer Visualize: 23:56:01 - HierarchicalClusterer (prodata)

X: Course (Nom)

Y: Total (Num)

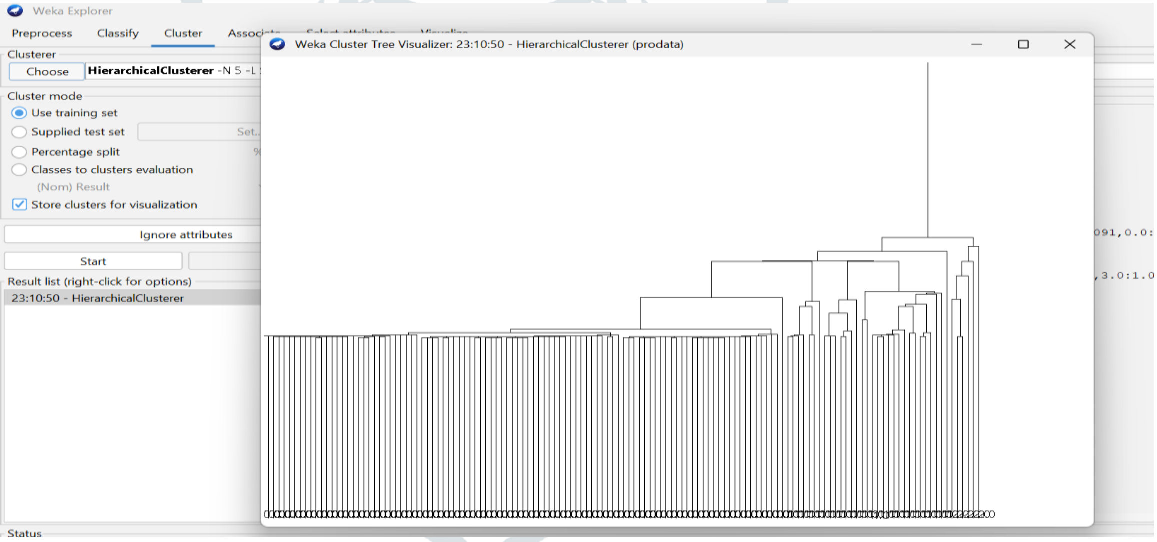
Colour: Cluster (Nom)

Plot: prodata_clustered

Step 7: In the Cluster visualization we choose Course on X-axis and Total no on Y-axis.

Blue colour shows Cluster 0, Red colour shows Cluster 1, Green colour shows Cluster 2, Sky-blue colour shows Cluster 3, Light pink colour shows Cluster 4.

Visualization tree in hierarchical clustered:



Weka Cluster Tree Visualizer: 23:10:50 - HierarchicalClusterer (prodata)

Weka Explorer
Preprocess
Classify
Cluster
Clusterer
Choose
HierarchicalClusterer -N 5 -L
Cluster mode
Use training set
Supplied test set
Percentage split
Classes to clusters evaluation
(Nom) Result
Store clusters for visualization
Ignore attributes
Start
Result list (right-click for options)
23:10:50 - HierarchicalClusterer
Status

Step 8: The above tree diagram shows that the hierarchical Clustering using 5 clusters. A cluster tree is defined as "a tree showing a sequence of clustering with each clustering being a partition of the data set".

Output: This Hierarchical Clustered algorithm is tested with project dataset in WEKA tool; it produces five different clustered instance clusters 0: 136(53%), cluster 1: 1(0%), cluster 2: 1(0%), cluster 3: 1(0%), cluster 4: 116(45%), and Time taken to build model (full training data): 0.06 seconds.

Simple-k means clustering:

K-means clustering is a simple unsupervised learning algorithm. In this, the data objects ('n') are grouped into a total of 'k' clusters, with each observation belonging to the cluster with the closest mean. It defines 'k' sets, one for each cluster k n (the point can be thought of as the centre of a one or two-dimensional figure). The clusters are separated by a large distance.

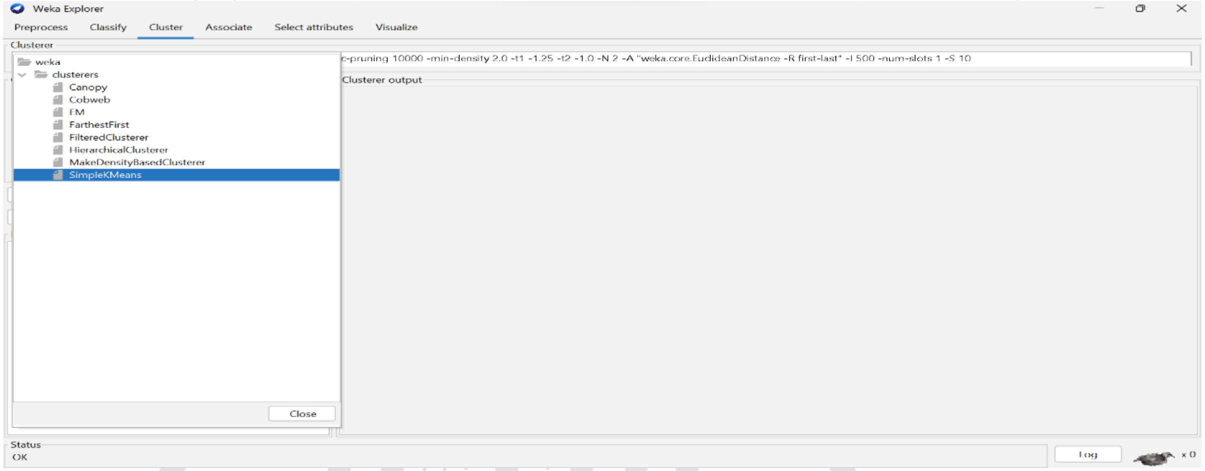
The data is then organized into acceptable data sets and linked to the nearest collection. If no data is pending, the first stage is more difficult to complete; in this case, an early grouping is performed. The 'k' new set must be recalculated as the barycentre's of the clusters from the previous stage.

The same data set points and the nearest new sets are bound together after these 'k' new sets have been created. After that, a loop is created. The 'k' sets change their position step by step until no further changes are made as a result of this loop.

Steps to be followed:

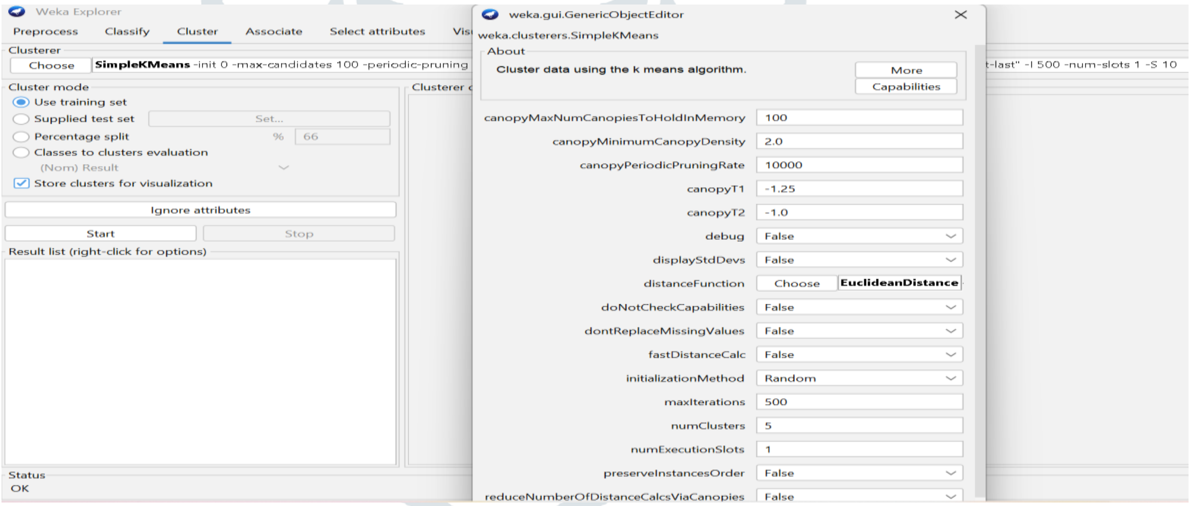
Step 1: In the preprocessing interface, open the Weka Explorer and load the required dataset, and we are taking the project. Arff dataset.

Step 2: Find the ‘cluster’ tab in the explorer and press the choose button to execute clustering. A dropdown list of available clustering algorithms appears as a result of this step and selects the simple-k means algorithm.



```
c-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
```

Step 3: Then, to the right of the choose icon, press the text button to bring up the popup window shown in the screenshots. We enter five for the number of clusters in this window.



```
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning
```

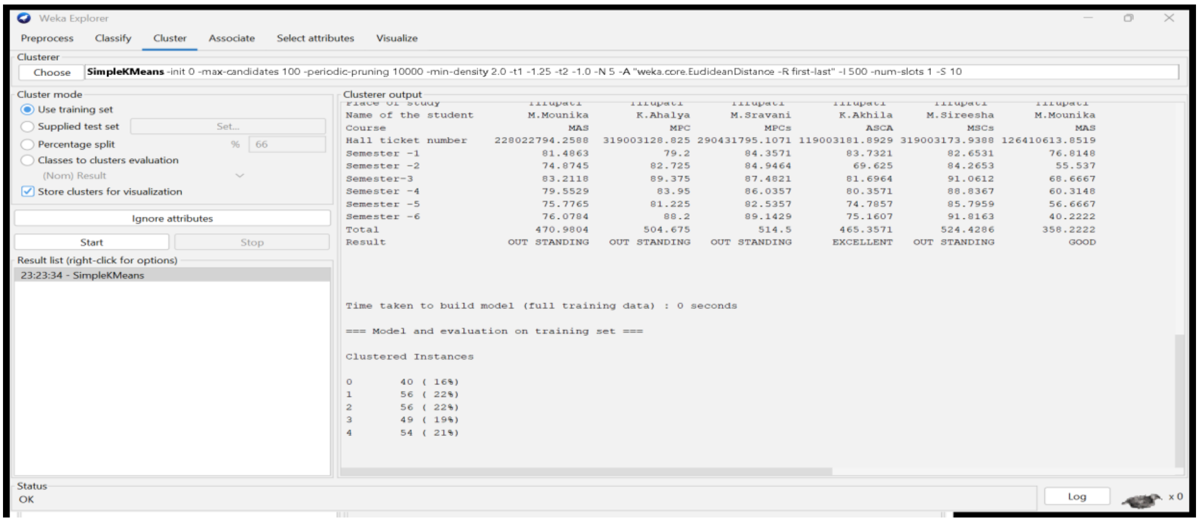
● Use training set
○ Supplied test set
○ Percentage split % 66
○ Classes to clusters evaluation
(Nom) Result
☑ Store clusters for visualization

weka.clusterers.SimpleKMeans

Cluster data using the k means algorithm.

canopyMaxNumCanopiesToHoldInMemory	100
canopyMinimumCanopyDensity	2.0
canopyPeriodicPruningRate	10000
canopyT1	-1.25
canopyT2	-1.0
debug	False
displayStdDevs	False
distanceFunction	Choose EuclideanDistance
doNotCheckCapabilities	False
dontReplaceMissingValues	False
fastDistanceCalc	False
initializationMethod	Random
maxIterations	500
numClusters	5
numExecutionSlots	1
preserveInstancesOrder	False
reduceNumberOfDistanceCalcsViaCanopies	False

Step 4: One of the choices has been chosen. We must ensure that they are in the ‘cluster mode’ panel before running the clustering algorithm. The choice to use a training set is selected, and then the ‘start’ button is pressed.



```
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 5 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
```

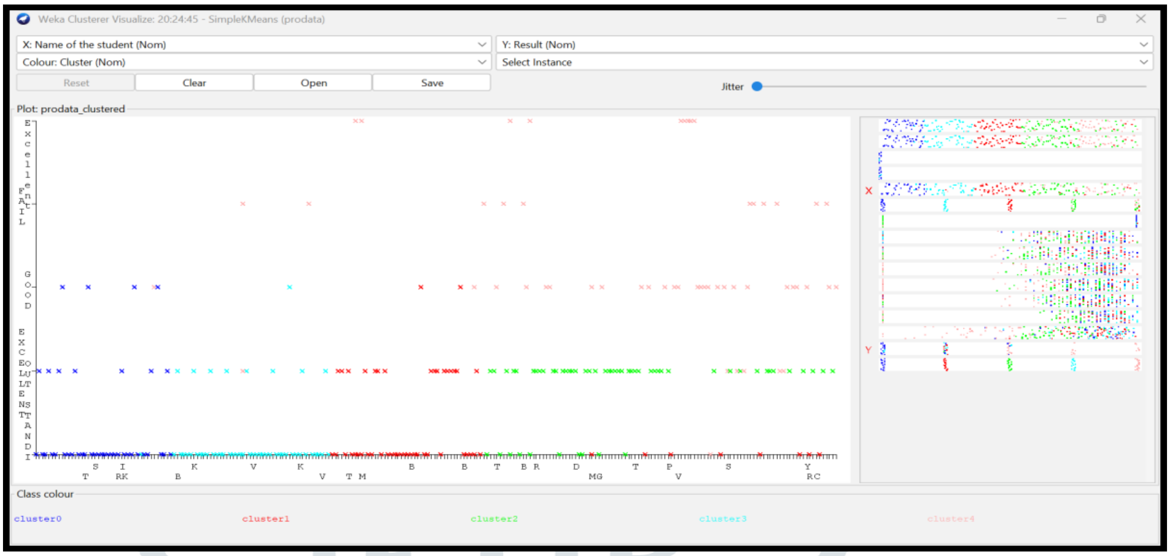
● Use training set
○ Supplied test set
○ Percentage split % 66
○ Classes to clusters evaluation
(Nom) Result
☑ Store clusters for visualization

23:23:34 - SimpleKMeans

```
Place of study              Tirupati     Tirupati     Tirupati     Tirupati     Tirupati     Tirupati
Name of the student        M.Mounika    K.Ahalya    M.Sravani     K.Akhila   M.Sireesha    M.Mounika
Course                           MAS          MPC         MPCs         ASCA         MSCs          MAS
Hall ticket number    228022794.2588 319003128.825 290431795.1071 119003181.8929 319003173.9388 126410613.8519
Semester -1                  81.4863         79.2      84.3571      83.7321      82.6531      76.8148
Semester -2                  74.8745       82.725      84.9464       69.625      84.2653       55.537
Semester-3                   83.2118       89.375      87.4821      81.6964      91.0612      68.6667
Semester -4                  79.5529        83.95      86.0357      80.3571      88.8367      60.3148
Semester -5                  75.7765       81.225      82.5357      74.7857      85.7959      56.6667
Semester -6                  76.0784         88.2      89.1429      75.1607      91.8163      40.2222
Total                       470.9804      504.675        514.5     465.3571     524.4286     358.2222
Result                  OUT STANDING OUT STANDING OUT STANDING    EXCELLENT OUT STANDING         GOOD




Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       40 ( 16%)
1       56 ( 22%)
2       56 ( 22%)
3       49 ( 19%)
4       54 ( 21%)
```

Step 5: The centroid of each cluster is shown in the result window, along with statistics on the number and percent of instances allocated to each cluster. Each cluster centroid is represented by a mean vector. This cluster can be used to describe a cluster.

Step 6: Another way to grasp the characteristics of each cluster is to visualize them. To do so, right-click the result set on the result. Selecting to visualize cluster assignments from the list column.



Weka Clusterer Visualize: 20:24:45 - SimpleKMeans (prodata)

X: Name of the student (Nom)

Y: Result (Nom)

Colour: Cluster (Nom)

Select Instance

Step 7: In the Cluster visualization we choose Name of the student on X-axis and Result no on Y-axis. Blue colour shows Cluster 0, Red colour shows Cluster 1, Green colour shows Cluster 2, Sky-blue colour shows Cluster 3, Light pink colour shows Cluster 4.

Output: K means clustering is a simple cluster analysis method. The number of clusters can be set using the setting tab. The centroid of each cluster is calculated as the mean of all points within the clusters. With the increase in the number of clusters, the sum of square errors is reduced. The objects within the cluster exhibit similar characteristics and properties. The clusters represent the class labels.

This Simple -k means Clusterer algorithm is tested with project dataset in WEKA tool; it produces five different clustered instance clusters 0: 40(16%), cluster 1: 56(22%), cluster 2: 56(22%), cluster 3: 49(19%), cluster 4: 54(21%), and Time taken to build model (full training data): 0 seconds.

RESULT ANALYSIS

Data Mining Techniques (Knowledge Discovery Database):

The KDD (Knowledge Discovery in Databases) paradigm is a step-by-step approach to identifying interesting patterns in enormous volumes of data. Data mining is a step in the process. The algorithms have good potential. Analytic tools for performance evaluation are demonstrated by examining results from a computer performance dataset. It is far easier to store data than to make sense of it. The ability to identify links in huge amounts of stored data can lead to improved analytical tactics in domains such as education, marketing, computer performance analysis, and data analysis in general. KDD addresses the difficulty of finding patterns in huge datasets. Traditionally, data was analyzed manually; however, there are human

The clustering process

Clustering divides data into groupings of related things. Each cluster contains a variety of things that are both similar to one another and distinct to objects from other clusters. Various clustering algorithms are used to produce clusters. The Weka tool is used to evaluate various clustering techniques.

This paper compares the different Weka clustering algorithms to determine which algorithm is best suited to the users. Four clustering algorithms are compared: Canopy clustering algorithm, EM Algorithm, Hierarchical Algorithm, and Simple k-Means clustering algorithm. All of the algorithms discussed above are explained and analyzed using specific assessment parameters. These parameters include the number of clusters formed and clustered instances. The results of all four algorithms are generated.

The following table represents the analysis process of all four algorithms and the accuracy results are shown in table.

Clustering algorithm	No. of Clusters	Clustering instances

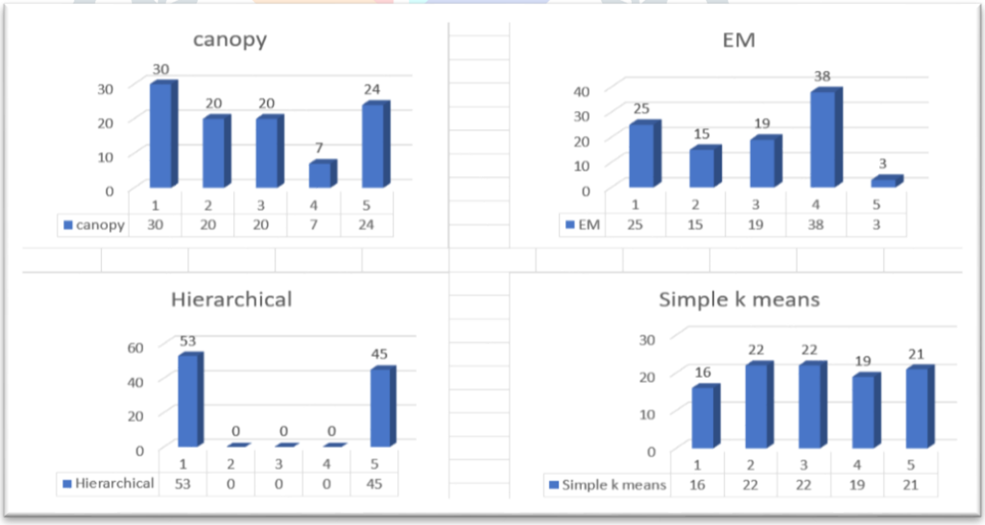
Canopy	5	76(30%)
		50(20%)
		50(20%)
		19(07%)
		60(24%)

EM	5	63(25%)
		38(15%)
		49(19%)
		98(38%)
		7(3%)

Hierarchical	5	136(53%)
		1(0%)
		1(0%)
		1(0%)
		116(45%)

Simple-k Means	5	40(16%)
		56(22%)
		56(22%)
		49(19%)
		54(21%)

In the above table, the Simple-k means clustering algorithm and EM Clustering Algorithm are showing good accuracy than another cluster algorithm.



The above chart shows the performance accuracy of the four-clustering based on different clustering metrics. This metrics shows that Simple-k means, EM algorithms are performing good accuracy result better than other clustering.

CONCLUSION

Data mining translates raw data into information so that predictions can be made. Cluster analysis is a technique for identifying clusters of data with similar features. WEKA has a variety of algorithms for cluster analysis, with simple means being the most commonly employed.
This work describes data mining in the educational context, specifically identifying students' performance trends using the clustering data mining technique. The detected trends are evaluated to provide useful and constructive recommendations to academic planners in higher education institutions to improve their decision-making process.
We conducted analyses using four clustering algorithms: the Canopy clustering algorithm, the EM algorithm, the Hierarchical algorithm, and the Simple k-Means clustering algorithm. All four algorithms give results based on similar items and times.

In future work the authors also interested in working in future on data of student's assessments for each course trying to know what kind of student succeed on what kind of courses. It may define what kinds of courses are adapted for every student's model who shares the same characteristics. It may also provide various multidimensional summary reports and redefine pedagogical learning paths.

REFERENCES

- N. Valarmathy and S. Krishnaveni “Performance Evaluation and Comparison of Clustering Algorithms used in Educational Data Mining.” vol. 7, Apr. 2019.
- Yasir M.A, Fatima D. M. ,2020, predict the grad of student using classification algorithms, International Journal of Science, Environment and Technology, Vol. 9, No 2, 2020, 75-89.
- M. Goyal and R. Vohra, “Applications of Data Mining in Higher Education”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue2, No 1, March 2012.
- T. Devasia, T. P. Vinushree, and V. Hegde,” Prediction of students performance using Educational Data Mining,” in Proceedings of 2016 International Conference on Data Mining and Advanced Computing, SAPIENCE 2016, 2016.
- S. Lailiyah, E. Yulsilviana, and R. Andrea, “Clustering analysis of learning style on anggana high school student,” TELKOMNIKA (Telecommunication Comput. Electron. Control., vol. 17, no. 3, p. 1409, Jun. 2019.
- William Iba and Pat Langley. "Cobweb models of categorization and probabilistic concept formation". In Emmanuel M. Pothos and Andy J. Wills, Formal approaches in categorization. Cambridge: Cambridge University Press. pp. 253—273.
- Zhao Y., Karypis G., "Evaluation of hierarchical clustering algorithms for document datasets", the eleventh international conference on Information and knowledge management,2002, pp. 515-524.
- Z. Huang. "Extensions to the k-means algorithm for clustering large data sets with categorical values". Data Mining and Knowledge Discovery,2:283—304, 1998.