SRM Institute of Science and Technology
Delhi – Meerut Road, Sikri Kalan, Ghaziabad, Uttar Pradesh – 201204
Department of Computer Applications
Circular – 2023-24
BCA DS 5th Sem

Data Warehousing and Data Mining (UDS23D01J)

Lab Manual

Lab 1: Installation of WEKA Tool - Investigating the Application Interfaces of the Weka Tool

Title: WEKA Installation and Interface Exploration

Aim: To install the WEKA data mining tool and explore its various application interfaces.

Procedure:

1. Download WEKA from the official website (cs.waikato.ac.nz/ml/weka/).
2. Install WEKA following the installation instructions for your operating system.
3. Launch WEKA.
4. Explore the following interfaces:

   Explorer: For data exploration, preprocessing, and applying mining algorithms.

   Experimenter: For conducting experiments and comparing algorithms.

   Knowledge Flow: For designing data mining workflows.

   Workbench:The main GUI

Identify the key components of each interface.

Source Code: (N/A - This lab is about software installation and exploration, not coding)

Input: WEKA installation file.

Expected Output: WEKA successfully installed and the ability to navigate and describe the Explorer, Experimenter, and Knowledge Flow interfaces.

Lab 2: Overview - Working in the Console

Title: Working with WEKA Console

Aim: To become familiar with using the WEKA console for executing data mining tasks.

Procedure:

1. Launch WEKA and open the Console interface.
2. Learn basic WEKA command-line syntax.
3. Load a dataset (e.g., using the `java weka.core.converters.CSVLoader` command).
4. Apply a simple classifier (e.g., J48) from the console.
5. Display the results.
6. Practice using help command.

Source Code: (Conceptual - Adapt to your WEKA installation)

```
java weka.core.converters.CSVLoader <your_data.csv> > <your_data.arff>  #
Convert CSV to ARFF
java weka.classifiers.trees.J48 -t <your_data.arff> -d model.model # Build
J48 Decision Tree
java weka.classifiers.trees.J48 -l model.model -T test_data.arff -p 0 # Test
the model
```

Input: A dataset (e.g., in CSV format).

Expected Output: Successful execution of WEKA commands in the console, including loading data, running a classifier, and viewing the output.

Lab 3: Getting Help in WEKA Tool and Quitting WEKA

Title: Using WEKA's Help System and Exiting the Application

Aim: To learn how to access WEKA's built-in help resources and to properly exit the application.

Procedure:

1. Launch WEKA.
2. Explore the help menu within each interface (Explorer, Experimenter, Knowledge Flow).
3. Use the command-line help within the Console interface (e.g., `java weka.jar -h`).
4. Identify how to find information on specific algorithms or options.
5. Practice different ways to quit WEKA (e.g., using the menu, closing the window, using a command in the console).

Source Code: (N/A - This lab is about using the WEKA interface, not writing code)

Input: N/A

Expected Output: Ability to navigate WEKA's help system to find information and to quit the application correctly.

Lab 4: Pre-process a given dataset based on Attribute selection

Title: Attribute Selection using WEKA

Aim: To learn how to use WEKA to select relevant attributes from a dataset for improved data mining.

Procedure:

1. Load a dataset into WEKA Explorer.
2. Go to the "Select attributes" tab.
3. Choose an attribute selection method (e.g., InfoGainAttributeEval, GainRatioAttributeEval).
4. Select a search method (e.g., Ranker, BestFirst).
5. Start the attribute selection process.
6. Analyze the results to see which attributes were selected.
7. Apply the selected attributes to the dataset.

Source Code: (Conceptual - WEKA Explorer) (This lab primarily uses the WEKA Explorer interface. The equivalent console commands are complex and depend on the specific evaluator and search method.)

Input: A dataset with multiple attributes.

Expected Output: A reduced dataset containing only the most relevant attributes, as determined by the chosen attribute selection method.

Lab 5: Pre-process a given dataset based on Handling Missing Values

Title: Handling Missing Values in WEKA

Aim: To learn how to use WEKA to handle missing values in a dataset.

Procedure:

1. Load a dataset with missing values into WEKA Explorer.
2. Go to the "Preprocess" tab.
3. Select a filter to handle missing values (e.g., ReplaceMissingValues).
4. Apply the filter to the dataset.
5. Observe how the missing values are replaced (e.g., with the mean, median, or mode).

Source Code: (Conceptual - WEKA Explorer) (This lab primarily uses the WEKA Explorer interface. The equivalent console commands are complex.)

Input: A dataset with missing values.

Expected Output: A modified dataset where missing values have been replaced using a suitable method.

Lab 6: Create a Weather Table with the help of Data Mining Tool WEKA.

Title: Creating a Weather Table in WEKA

Aim: To create a dataset representing weather conditions using WEKA.

Procedure:

1. Open WEKA Explorer.
2. Use the "Preprocess" tab.
3. Create a new dataset using "Edit..." button, defining attributes (e.g., outlook, temperature, humidity, windy, play) and their types (nominal, numeric).
4. Enter the data rows for the weather conditions.
5. Save the dataset in ARFF format.

Source Code: (Conceptual - WEKA Explorer Data Input) (This lab primarily involves manual data entry within the WEKA Explorer. There is no direct "source code" in the traditional sense. The ARFF file that WEKA creates represents the data.)

Input: Weather data (e.g., from a textbook or online source).

Expected Output: A WEKA-compatible ARFF file representing the weather data, ready for further analysis.

Lab 7: Generate Association Rules using the Apriori Algorithm

Title: Association Rule Mining with Apriori

Aim: To use the Apriori algorithm in WEKA to discover association rules in a dataset.

Procedure:

1. Load a dataset into WEKA Explorer.
2. Go to the "Associate" tab.
3. Select the Apriori algorithm.
4. Configure the Apriori parameters (e.g., minimum support, minimum confidence).
5. Start the Apriori algorithm.
6. Analyze the generated association rules.

Source Code: (Conceptual - WEKA Explorer)

Input: A transactional dataset (e.g., market basket data).

Expected Output: A set of association rules in the form "IF X THEN Y", indicating relationships between data items.

Lab 8: Generate Association Rules using the Apriori Algorithm

Title: Association Rule Mining with Apriori (Continued)

Aim: To further explore and apply the Apriori algorithm in WEKA with different parameter settings.

Procedure:

1. Repeat the steps from Lab 7 with the same or a different dataset.
2. Experiment with different values for minimum support and minimum confidence.
3. Observe how the change in parameters affects the number and quality of the generated rules.
4. Analyze the new set of association rules.

Source Code: (Conceptual - WEKA Explorer)

Input: A transactional dataset.

Expected Output: Multiple sets of association rules generated with varying parameter settings, along with an analysis of the impact of those settings.

Lab 9: Build a Decision Tree by using J48 algorithm

Title: Decision Tree Induction with J48

Aim: To build a decision tree classifier using the J48 algorithm in WEKA.

Procedure:

1. Load a dataset into WEKA Explorer.
2. Go to the "Classify" tab.
3. Select the J48 algorithm (a WEKA implementation of C4.5).
4. Set the class attribute.
5. Start the J48 algorithm.
6. Analyze the resulting decision tree (e.g., size, structure, accuracy).
7. Interpret the rules represented by the tree.

Source Code: (Conceptual - WEKA Explorer)

Input: A labeled dataset (i.e., a dataset with a class attribute).

Expected Output: A decision tree model that can be used to classify instances.

Lab 10: Naïve Bayes classification on a given data set

Title: Naïve Bayes Classification

Aim: To apply the Naïve Bayes classifier in WEKA to a given dataset.

Procedure:

1. Load a dataset into WEKA Explorer.
2. Go to the "Classify" tab.
3. Select the Naïve Bayes classifier.
4. Set the class attribute.
5. Start the Naïve Bayes algorithm.
6. Evaluate the performance of the classifier (e.g., accuracy, precision, recall).

Source Code: (Conceptual - WEKA Explorer)

Input: A labeled dataset.

Expected Output: A Naïve Bayes classification model and its performance metrics.

Lab 11: Finding Association Rules for Employee data.

Title: Association Rule Mining for Employee Data

Aim: To apply association rule mining to a dataset containing employee information.

Procedure:

1. Obtain or create a dataset of employee information (e.g., department, job title, salary range, years of service). You may need to discretize numerical attributes like salary range.
2. Load the employee dataset into WEKA Explorer.
3. Go to the "Associate" tab.
4. Select the Apriori algorithm (or another association rule mining algorithm).
5. Configure the algorithm parameters.
6. Run the algorithm.
7. Interpret the resulting association rules (e.g., "IF department=Sales AND job title=Junior THEN salary range=Low").

Source Code: (Conceptual - WEKA Explorer)

Input: A dataset containing employee information.

Expected Output: Association rules that describe relationships between different attributes of employees.

Lab 12: To Construct Decision Tree for Weather data and classify it.

Title: Decision Tree Classification of Weather Data

Aim: To construct a decision tree model for weather data and use it for classification.

Procedure:

1. Use the weather data created in Lab 6 or load a suitable weather dataset into WEKA Explorer.
2. Go to the "Classify" tab.
3. Select a decision tree algorithm (e.g., J48).
4. Set the class attribute (e.g., "play").
5. Build the decision tree.
6. Evaluate the tree's performance.
7. Use the tree to classify new, unseen weather conditions.

Source Code: (Conceptual - WEKA Explorer)

Input: A weather dataset.

Expected Output: A decision tree model that predicts the class attribute (e.g., "play") based on weather conditions, along with an evaluation of the model's accuracy.

Lab 13: Applying k-means clustering on a given data set

Title: k-means Clustering

Aim: To apply the k-means clustering algorithm to a dataset using WEKA.

Procedure:

1. Load a dataset into WEKA Explorer.
2. Go to the "Cluster" tab.
3. Select the k-means algorithm (SimpleKMeans).
4. Set the number of clusters (k).
5. Run the algorithm.
6. Analyze the resulting clusters (e.g., cluster centroids, distribution of instances within clusters).

Source Code: (Conceptual - WEKA Explorer)

Input: A dataset.

Expected Output: A set of k clusters, where each data point is assigned to one of the clusters.

Lab 14: Distance Measures in Algorithmic Methods

Title: Exploring Distance Measures

Aim: To explore different distance measures within WEKA.

Procedure:

1. Load a dataset into WEKA.
2. This lab will involve using the Explorer, and potentially the console, to investigate how distance measures are used within different algorithms.
3. For example, within the k-means clustering algorithm, the Euclidean distance is used by default.
4. Investigate the documentation to find how to change the distance function, if possible.
5. Experiment with different distance measures (if available) and observe the effect on the results of clustering or classification.

Source Code: (Conceptual - WEKA Explorer and Console)

Input: A dataset.

Expected Output: An understanding of how distance measures affect the behavior of data mining algorithms.

Lab 15: Write a procedure for Employee data using MakeDensityBased Cluster Algorithm

Title: Density-Based Clustering with MakeDensityBasedCluster

Aim: To apply a density-based clustering algorithm to employee data using WEKA.

Procedure:

1. Obtain or create a dataset of employee information.
2. Load the employee data into WEKA Explorer.
3. Go to the "Cluster" tab.
4. Select the "MakeDensityBasedCluster" algorithm (note: you might need to choose an underlying density-based algorithm, like DBSCAN, which MakeDensityBasedCluster wraps).
5. Configure the algorithm's parameters (e.g., epsilon, minPoints for DBSCAN).
6. Run the clustering algorithm.
7. Analyze the resulting clusters.

Source Code: (Conceptual - WEKA Explorer)

Input: A dataset containing employee information.

Expected Output: Clusters of employees based on the density of data points in the attribute space.