EXPERIMENT 1

FAMILIARIZATION OF THE WEKA DATA MINING/MACHINE LEARNING TOOLKIT

AIM: To familiarize the Weka data mining/machine learning toolkit, understand the features and explore the datasets.

THEORY:

WEKA (Waikato Environment for Knowledge Analysis) is an open-source software that provides tools for data pre-processing, implementation of several Machine Learning algorithms, and visualization tools. It can be used to develop machine learning techniques and apply them to real-world data mining problems. The software is fully developed using the Java programming language. Weka provides access to SQL databases using Java Database Connectivity (JDBC) and allows using the response for an SQL query as the source of data.

I. WEKA-FUNCTIONALITIES

The various functionalities offered by WEKA can be summarized as given in the Figure 1.

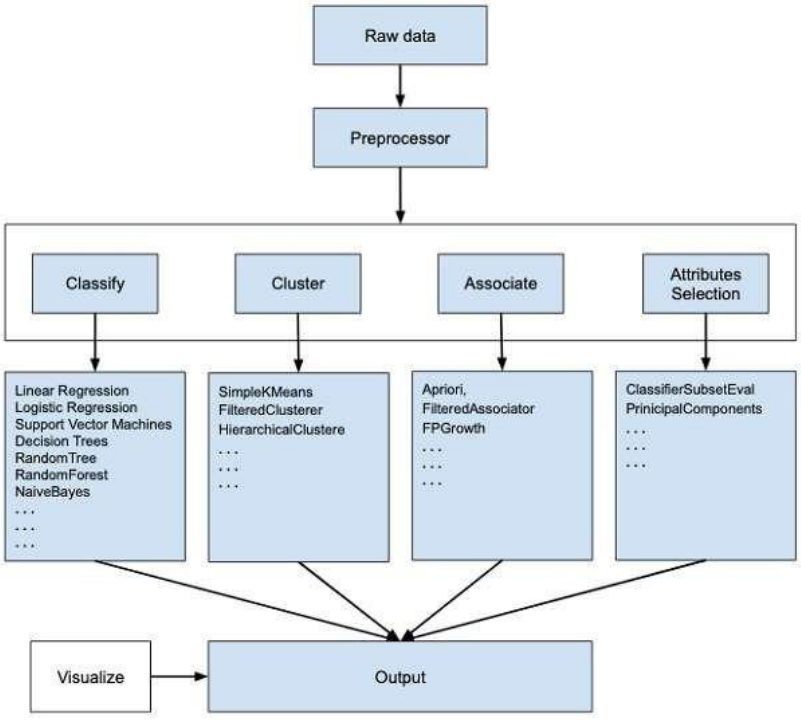

Figure 1: Functionalities of WEKA

There are many stages in dealing with big data to make it suitable for machine learning.

1. **Raw data collection** from the field.

2. **Data pre-processing**: The collected raw data may contain several null values and irrelevant fields. The pre-processing tools provided by WEKA can be sued to cleanse the raw data.

3. **Application of Machine Learning (ML) algorithms**: Depending on the kind of ML model being developed, one of the options such as Classify, Cluster, or Associate would be selected. The Attributes Selection allows the automatic selection of features to create a reduced dataset. Under each category, WEKA provides the implementation of several algorithms. An algorithm specific to the application can be selected, the desired parameters be set and the algorithm can be run on the dataset.

4. **Output visualization:** The statistical output of the ML model processing is provided by WEKA along with visualization tools to inspect the data.

5. **Selection of best ML model:** The various ML models can be applied on the same dataset; their outputs be compared and the best model that suits the specific application can be selected.

II. INSTALLATION OF WEKA

To install WEKA on a machine, visit WEKA's official website and download the installation file. Run the installation file and complete the installation.

III. LAUNCHING WEKA APPLICATIONS

Open the WEKA toolkit by double clicking the WEKA shortcut icon. The WEKA GUI Chooser application will display the screen as shown in Figure 2.

Figure 2: WEKA GUI Chooser application

The GUI Chooser application allows to run five different types of applications as listed here −

- Explorer
- Experimenter
- Knowledge Flow
- Workbench
- Simple CLI

IV. WEKA EXPLORER

On clicking the Explorer button in the Applications selector, it opens the screen shown in Figure 3.

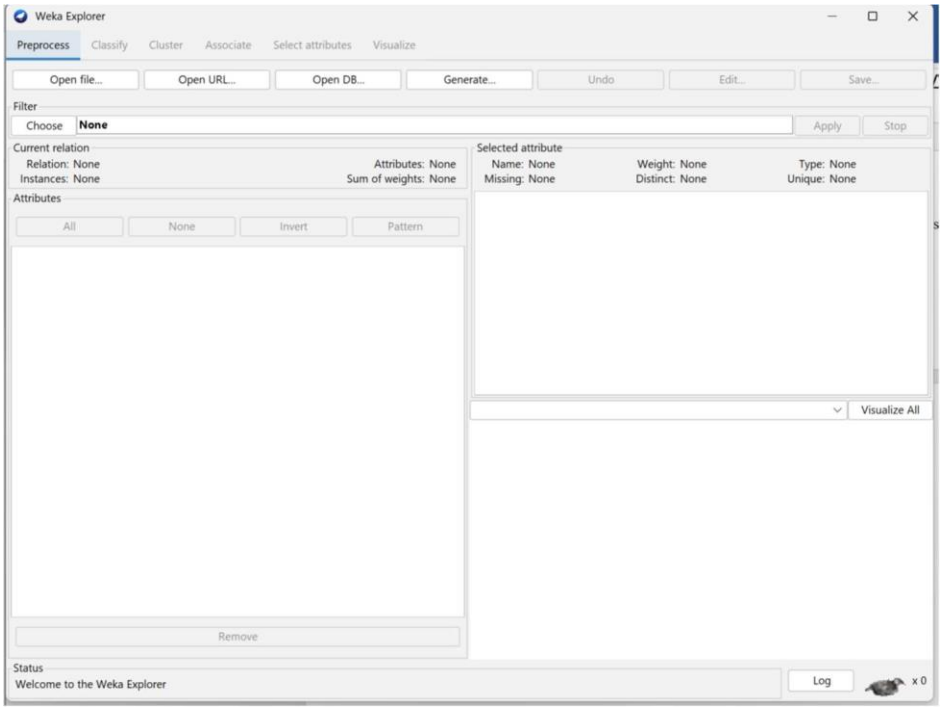

Figure 3: Weka Explorer

On the top, several Machine Learning tabs can be seen, as:

- Pre-process
- Classify
- Cluster
- Associate
- Select Attributes
- Visualize

Under these tabs, there are several pre-implemented machine learning algorithms.

- **Preprocess Tab:** Initially on opening the explorer, only the Preprocess tab is enabled. The first step in machine learning is to preprocess the data. Thus, in the Preprocess

option, select the data file, process it, and make it fit for applying the various machine learning algorithms.

- **Classify Tab:** The Classify tab provides several supervised and unsupervised machine learning algorithms for data classification. Algorithms such as Linear Regression, Logistic Regression, Support Vector Machines, Decision Trees, Random Tree, Random Forest, Naive Bayes etc. may be applied.

- **Cluster Tab:** Under the Cluster tab, there are several clustering algorithms provided - such as Simple K-Means, Filtered Clusterer, Hierarchical Clusterer, and so on.

- **Associate Tab:** Under the Associate tab, the available algorithms are Apriori, Filtered Associator and FPGrowth.

- **Select Attributes Tab:** Select Attributes allows feature selections based on several algorithms such as ClassifierSubsetEval, Principal Components, etc.

- **Visualize Tab:** Lastly, the Visualize option allows to visualize the processed data for analysis.

V. LOADING DATA

The data can be loaded from the following sources −

- Local file system
- Web
- Database

1. **Loading Data from Local File System**
   Just under the Machine Learning tabs, the following three buttons can be seen:
   i. Open file ...
   ii. Open URL ...
   iii. Open DB ...

   Click on the **Open file** ... button. A directory navigator window opens as shown in Figure 4.

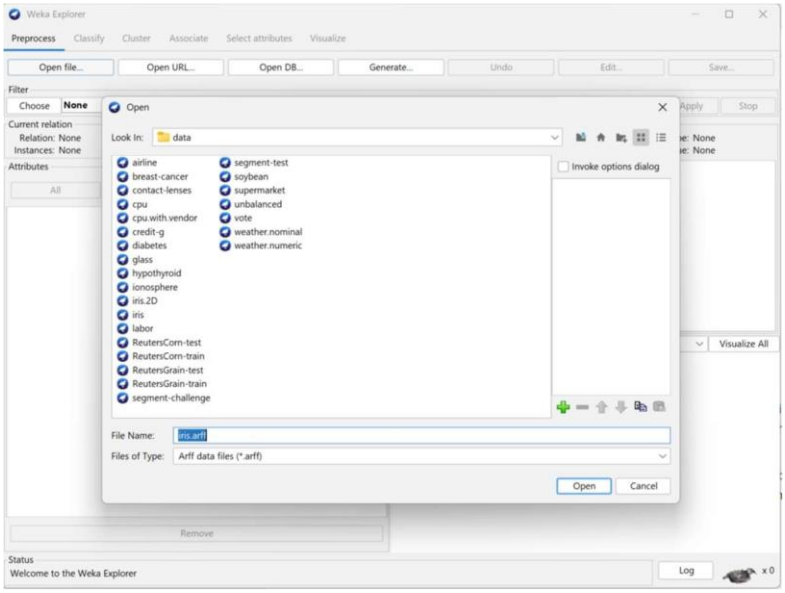

Figure 4: Directory navigator window

Now, navigate to the folder where the data files are stored. WEKA installation comes up with many sample databases for users to experiment. These are available in the data folder of the WEKA installation. (For eg., " C:\Program Files\Weka-3-8-6\data"). Select any data file from this folder. The contents of the file would be loaded in the WEKA environment.

2. Loading Data from Web

On clicking the Open URL ... button, a window as in Figure 5 will be displayed to open the file from a public URL.

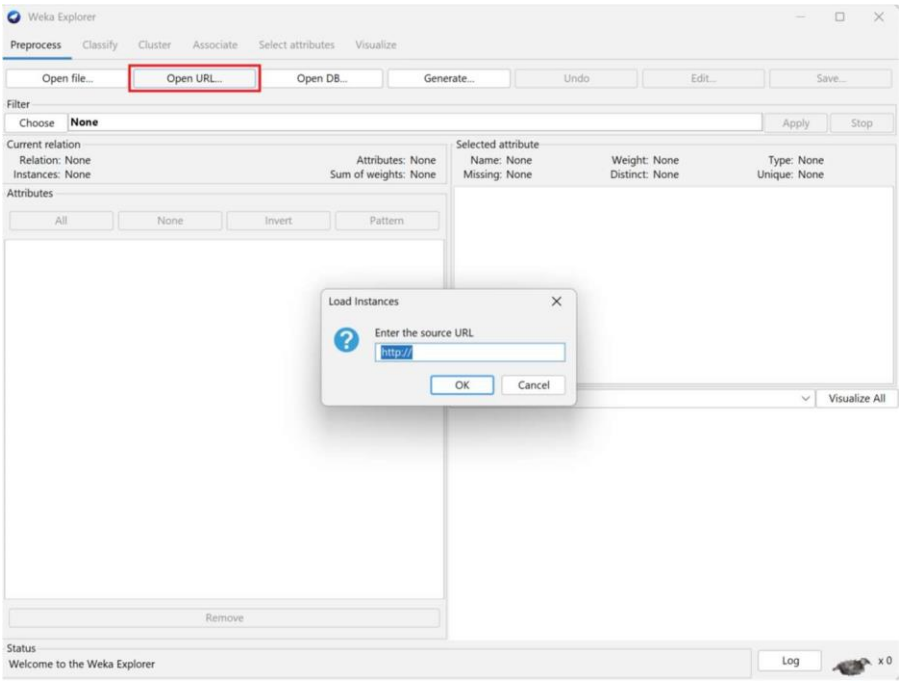

Figure 5: Loading data from web

Type the following URL in the popup box −

https://storm.cis.fordham.edu/~gweiss/data-mining/weka-data/weather.nominal.arff

Any other URL where data is stored can be specified. The Explorer will load the data from the remote site into its environment.

3. Loading Data from DB

On clicking the Open DB ... button, a window as in Figure 6 opens up.

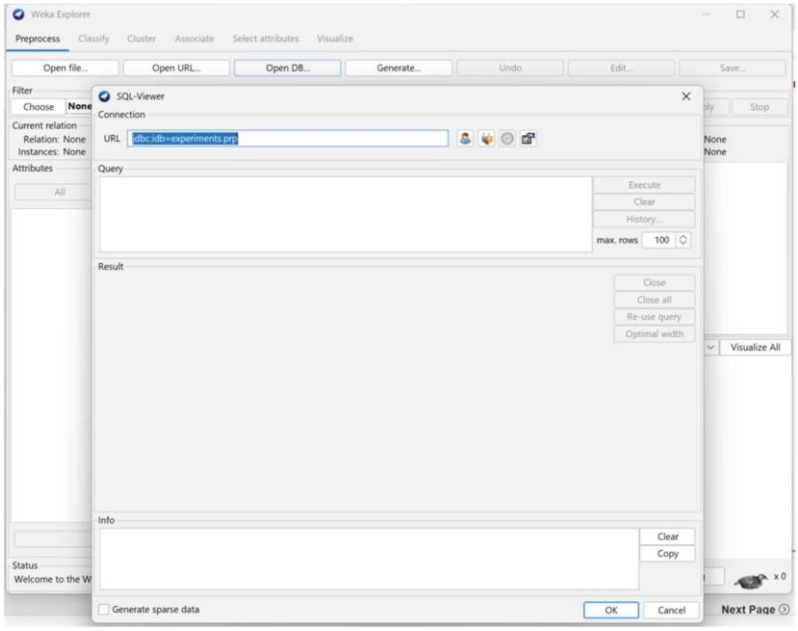

Figure 6: Loading data from DB

Set the connection string to the input database, set up the query for data selection, process the query and load the selected records in WEKA.

VI. FILE FORMATS

WEKA supports many file formats for the data. The complete list includes:

- arff
- arff.gz
- bsi
- csv
- dat
- data
- json
- json.gz
- libsvm
- m
- names
- xrff
- xrff.gz

The default file type is Arff.

Arff Format (Attribute-Relation File Format)

An Arff file contains two sections - header and data.

- The header describes the attribute types.
- The data section contains a comma separated list of data.

As an example for Arff format, the Weather data file loaded from the WEKA sample databases is shown in Figure 7.

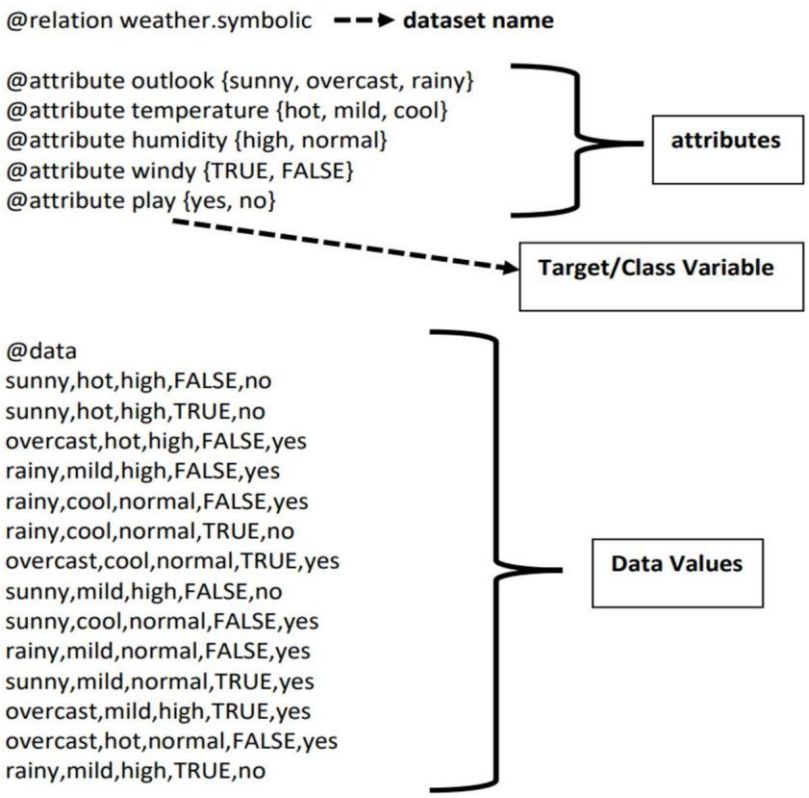

Figure 7: Weather data file

The following points can be inferred from the sample database shown in Figure 7.

- The @relation tag defines the name of the database.
- The @attribute tag defines the attributes.
- The @data tag starts the list of data rows each containing the comma separated fields.
- The attributes can take nominal values as in the case of ***outlook*** shown as

@attribute outlook (sunny, overcast, rainy)

- The attributes can take real values as in this case −

  @attribute temperature real

- A Target or a Class variable can also be set, called ***play*** as shown here

  @attribute play (yes, no)

- The Target assumes two nominal values yes or no.

The Explorer can load the data in any of the earlier mentioned formats. As arff is the preferred format in WEKA, the data can be loaded from any format and saved to arff format for later use. After pre-processing the data, just save it to arff format for further analysis.

VII. EXPLORING DATASETS IN WEKA

Use the **Weather** database that is provided in the installation. Using the **Open file** ... option under the **Preprocess** tag select the **weather-nominal.arff** file.

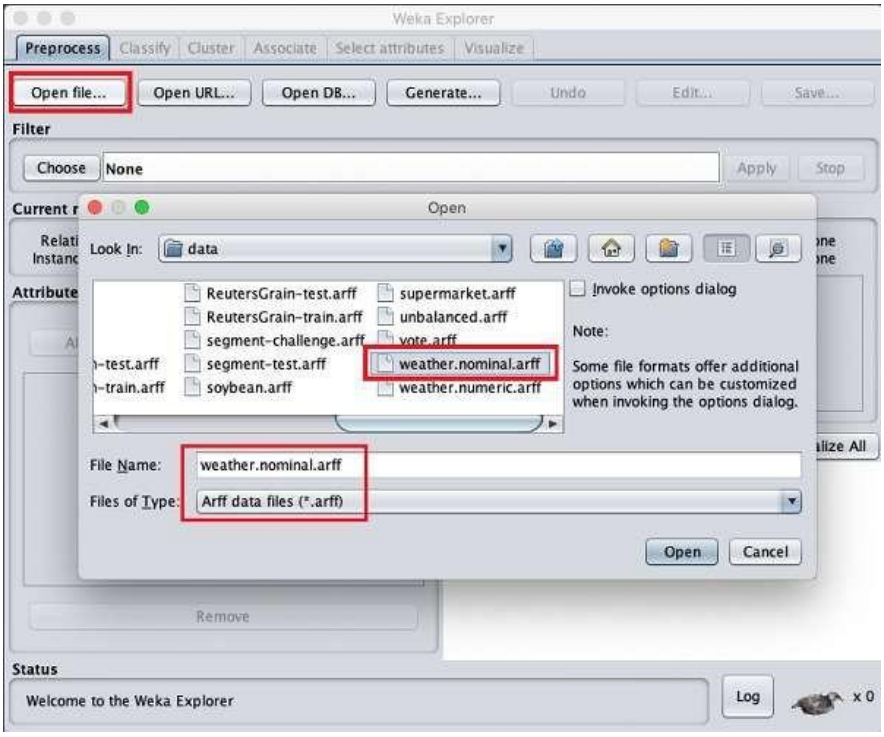

Figure 8: Opening the weather database

When you open the file, your screen looks like as shown in Figure 9.

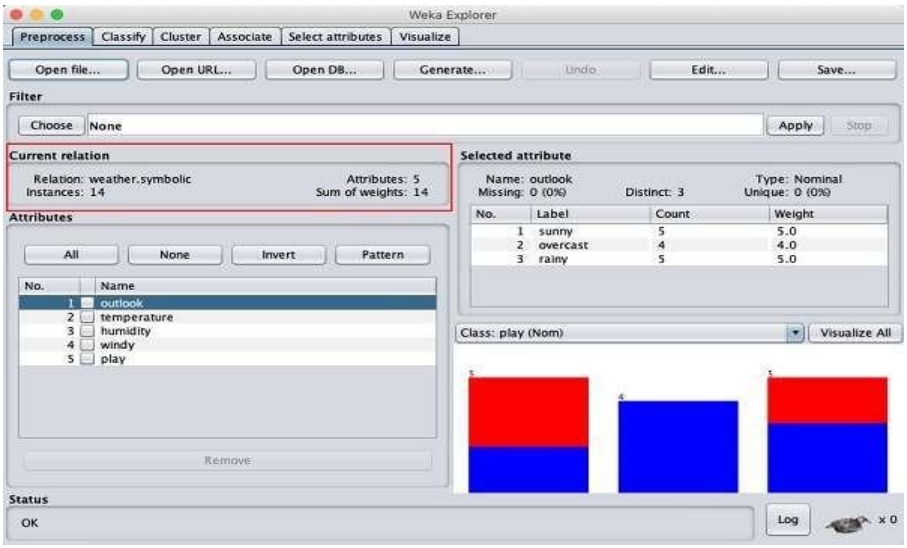

Figure 9: The weather.arff database

Current Relation sub window

It shows the name of the database that is currently loaded. The following two points can be inferred from this sub window:

- There are 14 instances - the number of rows in the table.
- The table contains 5 attributes - the fields

Attributes sub window

The Attributes sub window appears on the left side, that displays the various fields in the database, as shown in Figure 10.

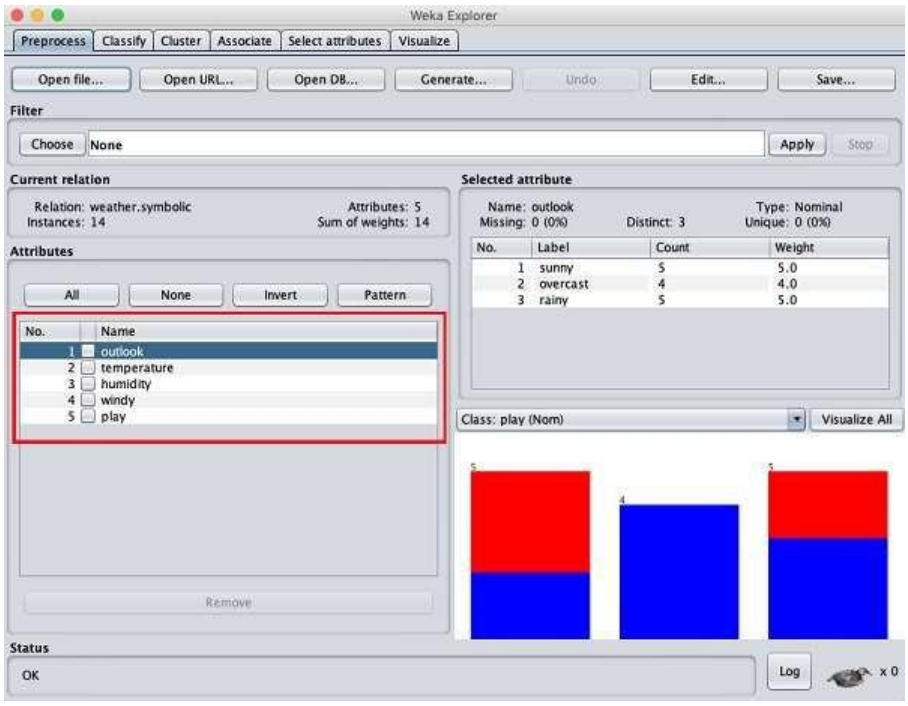

Figure 10: the attributes subwindow

The weather database contains five fields - outlook, temperature, humidity, windy and play.

Selected Attribute subwindow

On selecting an attribute from this list by clicking on it, further details on the attribute itself are displayed on the right-hand side. For example, select the temperature attribute first. The screen shown in Figure 11 appears on selecting the temperature attribute.

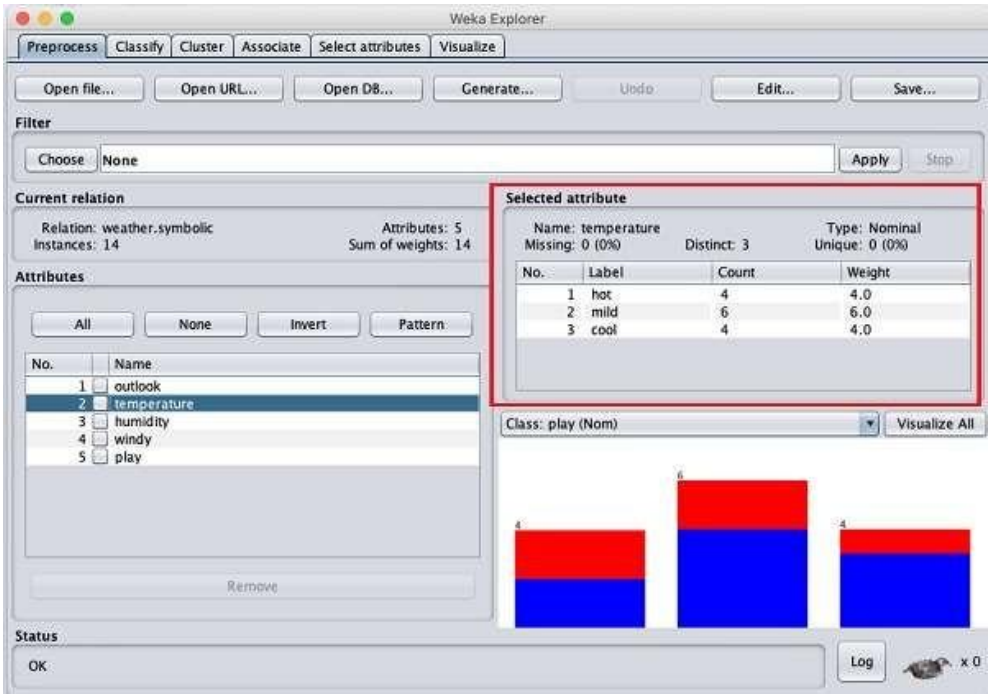

Figure 11: Selected attribute subwindow

In the Selected Attribute subwindow, you can observe the following –

- The name and the type of the attribute are displayed.
- The type for the temperature attribute is Nominal.
- The number of Missing values is zero.
- There are three distinct values with no unique value.
- The table underneath this information shows the nominal values for this field as hot, mild, and cold.
- It also shows the count and weight in terms of a percentage for each nominal value.

Visualization of attributes

At the bottom of the window, the visual representation of the class values can be seen. On clicking the Visualize All button, all features can be seen in one single window as shown in Figure 12.

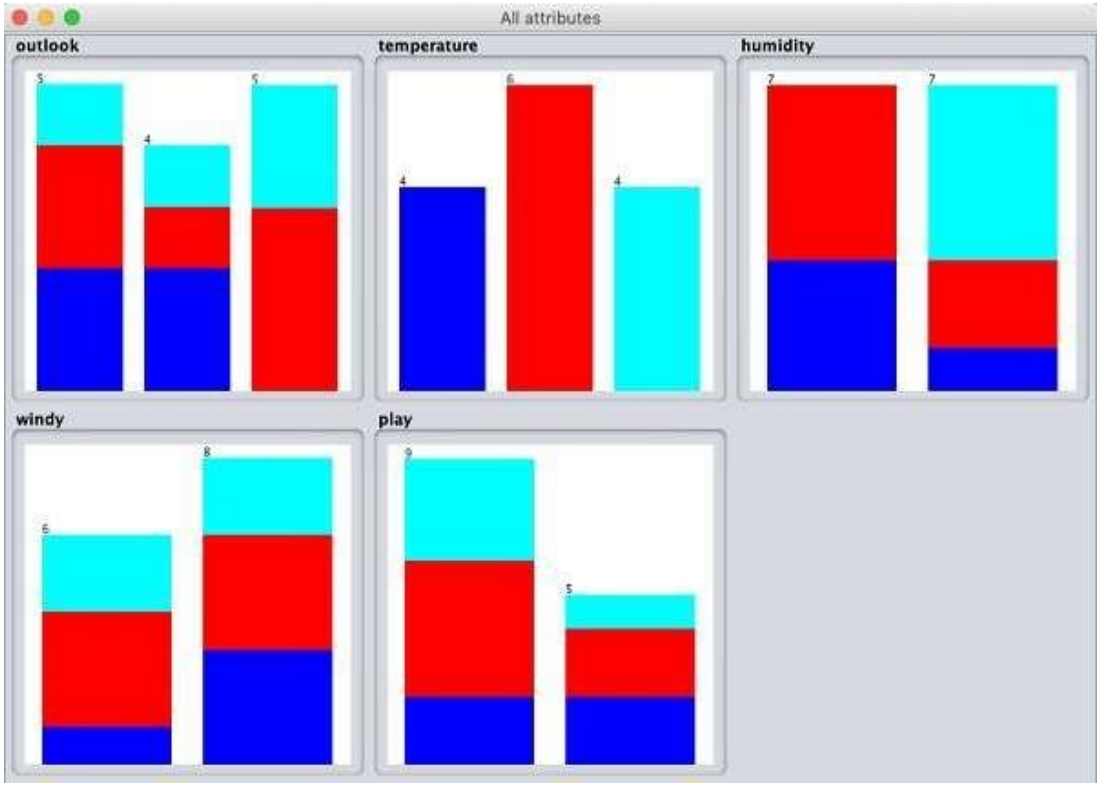

Figure 12: Attribute visualization

EXPERIMENT 2

DATA PRE-PROCESSING

AIM: To understand and perform the basic data pre-processing operations on a given dataset.

THEORY:

Data pre-processing is a data mining technique that performs a series of operations to transform the raw data in a useful and efficient format. These transformations are applied to the data before feeding it to algorithm. Whenever data is gathered from different sources, it is collected in raw format, which is not feasible for analysis. Pre-processing converts raw format into readable format (graphs, documents, etc.), so that it can be interpreted by computers and utilized by employees throughout an organization.

STEPS INVOLVED IN DATA PRE-PROCESSING:

1. DATA CLEANING:

The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

(a) Missing Data:

This situation arises when some data is missing in the data. It can be handled in various ways.

Some of them are:

i. **Ignore the tuples:** This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.

ii. **Fill the Missing values:** There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

(b) Noisy Data:

Noisy data is a meaningless data that cannot be interpreted by machines. It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways:

i. **Binning Method:** This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.

ii. **Regression:** Here data can be made smooth by fitting it to a regression function. The regression used may be linear (having one independent variable) or multiple (having multiple independent variables).

iii. **Clustering:** This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

2. DATA TRANSFORMATION:

This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

(a) Normalization:

It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)

(b) Attribute Selection:

In this strategy, new attributes are constructed from the given set of attributes to help the mining process.

(c) Discretization:

This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.

(d) Concept Hierarchy Generation:

Here attributes are converted from lower level to higher level in hierarchy. For Example-The attribute “city” can be converted to “country”.

3. DATA REDUCTION:

Since data mining is a technique that is used to handle huge amount of data. While working with huge volume of data, analysis became harder in such cases. In order to get rid of this, we use data reduction technique. It aims to increase the storage efficiency and reduce data storage and analysis costs.

The various steps to data reduction are:

(a) Data Cube Aggregation:

Aggregation operation is applied to data for the construction of the data cube.

(b) Attribute Subset Selection:

The highly relevant attributes should be used, rest all can be discarded. For performing attribute selection, one can use level of significance and p- value of the attribute. The attribute having p-value greater than significance level can be discarded.

(c) Numerosity Reduction:

This enables to store the model of data instead of whole data, for example: Regression Models.

(d) Dimensionality Reduction:

This reduces the size of data by encoding mechanisms. It can be lossy or lossless. If after reconstruction from compressed data, original data can be retrieved, such reduction are called lossless reduction else it is called lossy reduction. The two effective methods of dimensionality reduction are: Wavelet transforms and PCA (Principal Component Analysis).

DATA CLEANING USING WEKA

Some of the common filters available in Weka for data pre-processing are:

1. ***ReplaceMissingValues:*** Replaces all missing values for nominal and numeric attributes in a dataset with the modes and means from the training data.
2. ***RemoveWithValues:*** Filters instances according to the value of an attribute.
3. ***InterQuartileRange:*** A filter for detecting outliers and extreme values based on interquartile ranges.
4. ***Discretize:*** An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.
5. ***Normalize:*** Normalizes all numeric values in the given dataset (apart from the class attribute, if set).
6. ***NominalToBinary:*** Converts all nominal attributes into binary numeric attributes.
7. ***NumericToNominal:*** A filter for turning numeric attributes into nominal ones.
8. ***Remove:*** A filter that removes a range of attributes from the dataset.
9. ***RemoveByName:*** Removes attributes based on a regular expression matched against their names but will not remove the class attribute.
10. ***RenameAttribute:*** This filter is used for renaming attributes.