**INTRODUCTION:**

First of all why do we require data mining? We require data mining because of big data that is the term used for collection of large and complex datasets. A large data has several problems like capturing it, storing, searching, sharing and most importantly. All the industries have huge amount of data but they don’t have appropriate knowledge extracting tools to get benefit out of it. There are various algorithms in data mining to help these industries for their better decision making. WEKA is a tool with capabilities of performing many data mining tasks such as data preprocessing, attribute selection, classification, clustering and improving the knowledge discovery using various meta classifiers. In this paper we’ll discuss what is facilitated by the WEKA tool and what the steps to perform any activity on WEKA are. There are some constraints in Weka as it doesn’t accept data in every format.

**Using Weka Tool:**

Weka is a very good tool used for solving various purposes of data mining. There are four weka application interfaces: explorer, experimenter, knowledge flow and simple command line. The task can be processed using any of these interfaces.

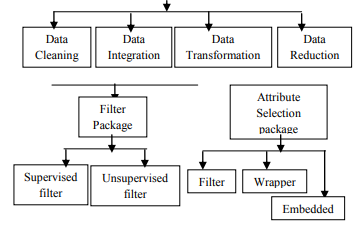
Not only can the interfaces, the open source code of weka also be used.

**Datasets in WEKA:**

WEKA accepts the data in ARFF format that is attribute relation file format, CSV format that is comma separated values, . Though it can accept data in CSV format also and can be converted into ARFF format. ARFF file consists of: @RELATION gives the relation declaration. @ATTRIBUTE depicts attribute declaration showing the name of the attribute and its datatype. @DATA illustrates the data declaration that is the start of the data segment in file. @DATA 5.1,3.5,1.4,0.2 Shows the data values means the values of each attribute in every sample. Data types can be numeric, nominal, string and date. Numeric, string and date are case insensitive. Data can be accepted from a database using JDBC connectivity. In Weka, there is option of importing the data as well as generating it automatically.

**Data Preprocessing in WEKA:**

Preprocessing is one of the important and prerequisite step in data mining. Feature selection (FS) is a process to select features which are more informative but some features may be redundant, and others may be irrelevant and noisy [3]. When the data set consists of meaningless data that is incomplete (missing), noisy (outliers) and inconsistent data, preprocessing of the dataset is required. Preprocessing step includes: (i) Data Cleaning: Handling the missing values by ignoring that particular tuple, filling that value with some specific value and handling noisy data using binning methods, clustering, combined human & machine inspection and regression. Inconsistency may be handled manually. (ii) Data Integration: Sometimes we have data from various sources in data warehouse and we may require to combine them for further analysis. Schema integration and redundancy are major problems in data integration. (iii) Data Transformation: Data Transformation is to transform the data in given format to required format for data mining. Normalization, smoothing, aggregation and generalization are few methods to perform transformation. (iv) Data Reduction: Data analysis on filters: Supervised and unsupervised. In both the categories we have filters for attributes and instances separately. After data cleaning, integration and transformation the data reduction is performed to get the task relevant data. For data reduction we have “Attribute Selection” option. It consists of various types of feature selection programs for wrapper approach, filter approach and embedded approach huge amount of data takes a very long time. It can be performed using data cube aggregation, dimension reduction, data compression, numerosity reduction, discretization and concept hierarchy generation. For the first 3 ways of preprocessing we have option of “filter” in WEKA. In filter option itself there are two types of



# **How to use the package manager?**

Usually, the term "package" is used to refer to Java's concept of organizing classes. From version 3.7.2, Weka has the concept of a package as a bundle of additional functionality, separate from that supplied in the main weka.jar file. A package consists of various jar files, documentation, meta data, and possibly source code. Many learning algorithms and tools that were present in earlier versions of Weka have become separate packages from version 3.7.2. This simplifies the core Weka system and allows users to install just what they need or are interested in. It also provides a simple mechanism for people to use when contributing to Weka. There are a number of packages available for Weka that add learning schemes or extend the functionality of the core system in some fashion. Many are provided by the Weka team and others are from third parties.

Weka includes a facility for the management of packages and a mechanism to load them dynamically at runtime. There is both a command-line and GUI package manager. If the package manager does not start when you try to run it, take a look at [this](https://weka.wikispaces.com/Weka%20package%20manager%20does%20not%20start) page.

# **Command line package management**

Assuming that the weka.jar file is in the classpath, the package manager can be accessed by typing:

*java weka.core.WekaPackageManager*

