Roll No.	Experiment No. 07	Marks :
BATCH -		Sign :

Aim: Perform data Pre-processing task and Demonstrate Association algorithm on data sets using data mining tools (WEKA, R tool, XL Miner, Orange etc.)

Apparatus: Google Colab, WEKA 3.8.6

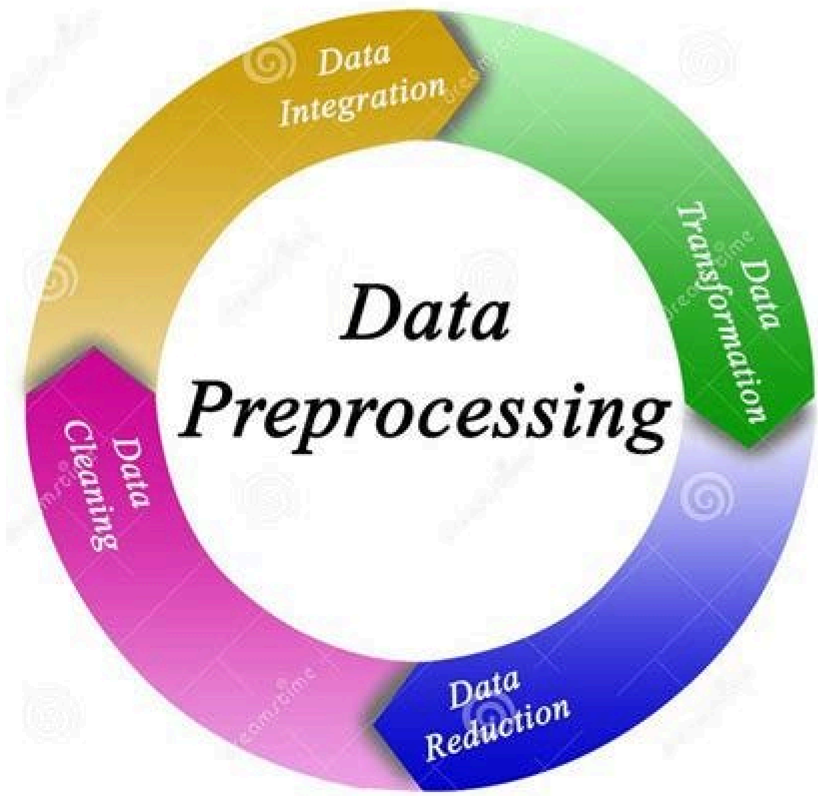
Theory:

Why is Data Preprocessing important?

Data preprocessing steps or data preprocessing techniques in machine learning is important for varied reasons. They are: –

- **Enhancing Data Quality**
  Data preprocessing in machine learning is crucial for enhancing data quality, forming the bedrock of reliable insights. Cleaning and refining raw data eliminates inaccuracies, missing values, and inconsistencies, ensuring that subsequent analyses and models are built on a solid foundation. This meticulous data preprocessing in machine learning directly impacts the accuracy and credibility of the conclusions drawn from the data.
- **Handling Missing Data**
  Addressing missing data preprocessing in machine learning is a pivotal aspect of data preprocessing. By employing techniques such as imputation or removal, the gap in information is effectively mitigated. This ensures that analytical models are not skewed by the absence of crucial data points, contributing to more robust and accurate outcomes.
- **Standardizing and Normalizing**
  Standardizing and normalizing data during data preprocessing steps ensure consistency in measurements, a critical factor in data analysis. This step transforms diverse scales and units into a standardized format, facilitating fair comparisons and preventing certain features from dominating others. The result is a leveled playing field where each variable contributes proportionately to the analysis.
- **Eliminating Duplicate Records**
  Steps in data preprocessing involves identifying and eliminating duplicate records, a key element in maintaining data integrity. Duplicate entries can distort analyses and mislead decision-making processes. By removing redundancies, the dataset retains its accuracy, and subsequent analyses yield trustworthy and actionable insights.

There are the major steps involved in data preprocessing, namely, data cleaning, data integration, data reduction, and data transformation as follows −



Data Cleaning − Data cleaning routines operate to "clean" the information by filling in missing values, smoothing noisy information, identifying or eliminating outliers, and resolving deviation. If users understand the data are dirty, they are unlikely to trust the results of some data mining that has been used.

Moreover, dirty data can make confusion for the mining phase, resulting in unstable output. Some mining routines have some phase for dealing with incomplete or noisy information, they are not always potent. Instead, they can concentrate on preventing overfitting the information to the function being modeled.

Data Integration − Data integration is the procedure of merging data from several disparate sources. While performing data integration, it must work on data redundancy, inconsistency, duplicity, etc. In data mining, data integration is a record preprocessing method that includes merging data from a couple of the heterogeneous data sources into coherent data to retain and provide a unified perspective of the data.

Data integration is especially important in the healthcare industry. Integrated data from multiple patient data and clinics assist clinicians in recognizing medical disorders and diseases by integrating data from multiple systems into an individual perspective of beneficial data from which beneficial insights can be derived.

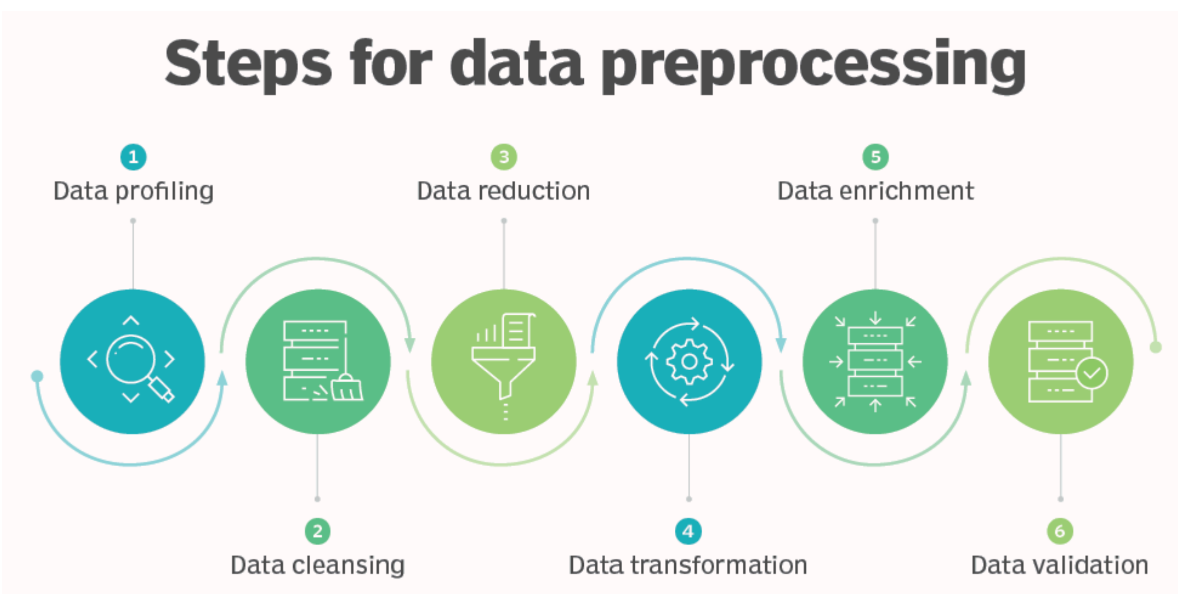
Data Reduction − The objective of Data reduction is to define it more compactly. When the data size is smaller, it is simpler to use sophisticated and computationally high-cost algorithms. The reduction of the data can be in terms of the multiple rows (records) or terms of the multiple columns (dimensions).
In dimensionality reduction, data encoding schemes are used so as to acquire a reduced or "compressed" description of the initial data. Examples involve data compression methods (e.g., wavelet transforms and principal components analysis), attribute subset selection (e.g., removing irrelevant attributes), and attribute construction (e.g., where a small set of more beneficial attributes is changed from the initial set).

Data transformation − In data transformation, where data are transformed or linked into forms applicable for mining by executing summary or aggregation operations. In Data transformation, it includes −

Smoothing : It can work to remove noise from the data. Such techniques include binning, regression, and clustering.
Aggregation : In aggregation, where summary or aggregation services are used to the data. For instance, the daily sales data can be aggregated to calculate monthly and annual total amounts. This procedure is generally used in developing a data cube for the analysis of the records at several granularities.

What are the key steps in data preprocessing?

The steps used in data preprocessing include the following:



1. **Data profiling**
   Data profiling is the process of examining, analyzing and reviewing data to collect statistics about its quality. It starts with a survey of existing data and its characteristics. Data scientists identify data sets that are pertinent to the problem at hand, inventory its significant attributes, and form a hypothesis of features that might be relevant for the proposed analytics or machine learning task. They also relate data sources to the relevant business concepts and consider which preprocessing libraries could be used.

2. **Data cleansing**
   The aim here is to find the easiest way to rectify quality issues, such as eliminating bad data, filling in missing data or otherwise ensuring the raw data is suitable for feature engineering.

3. **Data reduction**
   Raw data sets often include redundant data that arise from characterizing phenomena in different ways or data that is not relevant to a particular ML, AI or analytics task. Data reduction uses techniques like principal component analysis to transform the raw data into a simpler form suitable for particular use cases.

4. **Data transformation**
   Here, data scientists think about how different aspects of the data need to be organized to make the most sense for the goal. This could include things like structuring unstructured data, combining salient variables when it makes sense or identifying important ranges to focus on.

5. **Data enrichment**
   In this step, data scientists apply the various feature engineering libraries to the data to effect the desired transformations. The result should be a data set organized to achieve the optimal balance between the training time for a new model and the required compute.

6. **Data validation**
   At this stage, the data is split into two sets. The first set is used to train a machine learning or deep learning model. The second set is the testing data that is used to gauge the accuracy and robustness of the resulting model. This second step helps identify any problems in the hypothesis used in the cleaning and feature engineering of the data. If the data scientists are satisfied with the results, they can push the preprocessing task to a data engineer who figures out how to scale it for production. If not, the data scientists can go back and make changes to the way they implemented the data cleansing and feature engineering steps.

DATA CLEANING

Employee_Data:

Employee_ID	Name	Age	Department	Salary	Join_Date
101	John Doe	25	Marketing	50000	2018-05-15
102	Jane Smith	32	HR	60000	2019-02-20
103	Michael Johnson		Finance	55000	2020-07-10
104	Sarah Williams	28		52000	2017-11-30
105	Robert Brown	35	IT		2016-09-25
106	Emily Davis	40	Sales	58000	2021-04-12

```python
import pandas as pd

# Load the dataset
df = pd.read_csv('Employee_data.csv')
print(df)
# Handling missing values
df['Age'].fillna(df['Age'].median(), inplace=True)  # Impute missing values in 'Age' with median
df['Department'].fillna('Unknown', inplace=True)  # Replace missing values in 'Department' with 'Unknown'
df.dropna(inplace=True)                          # Drop rows with missing values in other columns

# Correcting data types
df['Join_Date'] = pd.to_datetime(df['Join_Date'])   # Convert 'Join_Date' to datetime

# Standardizing text data
df['Name'] = df['Name'].str.title()              # Convert 'Name' to title case
# Removing duplicates
df.drop_duplicates(inplace=True)

# Display the cleaned dataset
print(df)
df.to_csv("Cleaned_Employee_data.csv")
```

OUTPUT (Cleaned_Employee_data):

Employee_ID	Name	Age	Department	Salary	Join_Date
101	John Doe	25	Marketing	50000	2018-05-15
102	Jane Smith	32	HR	60000	2019-02-20
103	Michael Johnson	32	Finance	55000	2020-07-10
104	Sarah Williams	28	Unknown	52000	2017-11-30
106	Emily Davis	40	Sales	58000	2021-04-12

DATA INTEGRATION

Employee_Job_Role:

Employee_ID	Education	Job_Title
101	Bachelor's Degree	Marketing Manager
102	Master's Degree	HR Specialist
103	Bachelor's Degree	Financial Analyst
104		Marketing Coordinator
105	Bachelor's Degree	Systems Administrator
107	Master's Degree	Sales Director

```
import pandas as pd

# Load the dataset
df = pd.read_csv('Cleaned_Employee_data.csv', index_col=False)
# Load the additional dataset
additional_df = pd.read_csv('Employee_Job_Role.csv', index_col=False)
# Data integration
merged_df = pd.merge(df, additional_df, on='Employee_ID', how='left')
# Display the integrated dataset
print(merged_df)

merged_df.to_csv("Integrated_Employee_data.csv")
```

OUTPUT (Integrated_Employee_data):

Employee_ID	Name	Age	Department	Salary
101	John Doe	25	Marketing	50000
102	Jane Smith	32	HR	60000
103	Michael Johnson	32	Finance	55000
104	Sarah Williams	28	Unknown	52000
106	Emily Davis	40	Sales	58000

Join_Date	Education	Job_Title
2018-05-15	Bachelor's Degree	Marketing Manager
2019-02-20	Master's Degree	HR Specialist
2020-07-10	Bachelor's Degree	Financial Analyst
2017-11-30		Marketing Coordinator
2021-04-12		

DATA TRANSFORMATION

```
import pandas as pd

# Load the dataset
merged_df = pd.read_csv('integrated_Employee_data.csv')

# 1. Creating Derived Columns: Calculate years of experience
merged_df['Join_Date'] = pd.to_datetime(merged_df['Join_Date'])
merged_df['Years_Experience'] = pd.Timestamp.now().year - merged_df['Join_Date'].dt.year

# 2. Aggregation: Calculate average salary by department
avg_salary_by_department = merged_df.groupby('Department')['Salary'].mean().reset_index()
avg_salary_by_department.columns = ['Department', 'Avg_Salary']

# 3. Normalization/Standardization: Scale the salary column to a common range
merged_df['Normalized_Salary'] = (merged_df['Salary'] - merged_df['Salary'].min()) / (merged_df['Salary'].max() - merged_df['Salary'].min())

# 4. One-Hot Encoding: Create binary vectors for department
department_dummies = pd.get_dummies(merged_df['Department'], prefix='Dept')
merged_df = pd.concat([merged_df, department_dummies], axis=1)
```

```
# Save the transformed dataset to CSV
merged_df.to_csv('transformed_dataset.csv', index=False)

# Save the average salary by department to CSV
avg_salary_by_department.to_csv('avg_salary_by_department.csv', index=False)
```

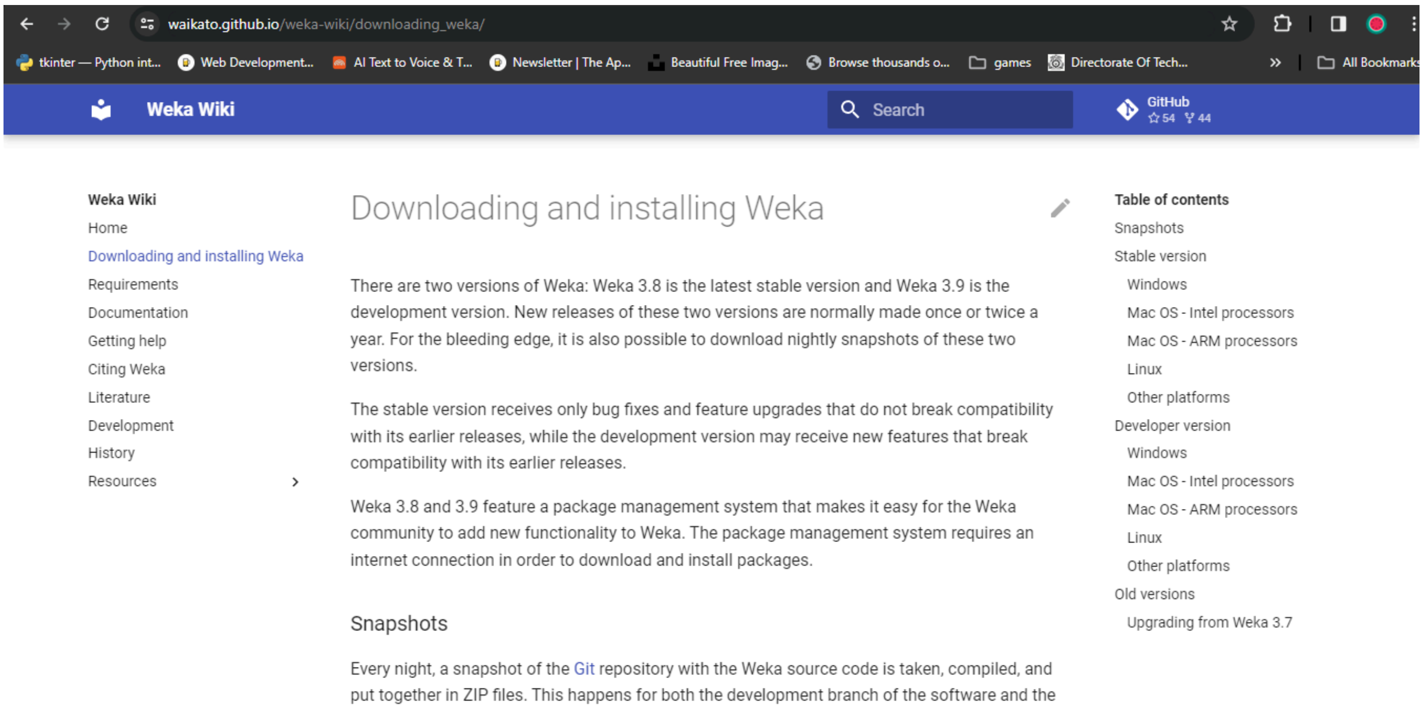
OUTPUT(Transformed_dataset):

Employee_ID	Name	Age	Department	Salary	Join_Date	Education
101	John Doe	25	Marketing	50000	2018-05-15	Bachelor's Degree
102	Jane Smith	32	HR	60000	2019-02-20	Master's Degree
103	Michael Johnson	32	Finance	55000	2020-07-10	Bachelor's Degree

Job_Title	Years_Experience	Normalized_Salary	Dept_Finance	Dept_HR	Dept_Marketing
Marketing Manager	6	0	FALSE	FALSE	TRUE
HR Specialist	5	1	FALSE	TRUE	FALSE
Financial Analyst	4	0.5	TRUE	FALSE	FALSE

WEKA Installation:

1. **Download Weka:**



Visit the official Weka website at https://www.cs.waikato.ac.nz/ml/weka/downloading.html to download the latest version of Weka. You'll find versions for different operating systems, including Windows, macOS, and Linux.

2. **Choose Installation Package:**

Stable version

Weka 3.8 is the latest stable version of Weka. This branch of Weka only receives bug fixes and upgrades that do not break compatibility with earlier 3.8 releases, although major new features may become available in packages. There are different options for downloading and installing it on your system:

WINDOWS

- Click here to download a self-extracting executable for 64-bit Windows that includes Azul's 64-bit OpenJDK Java VM 17 (weka-3-8-6-azul-zulu-windows.exe; 133.2 MB)

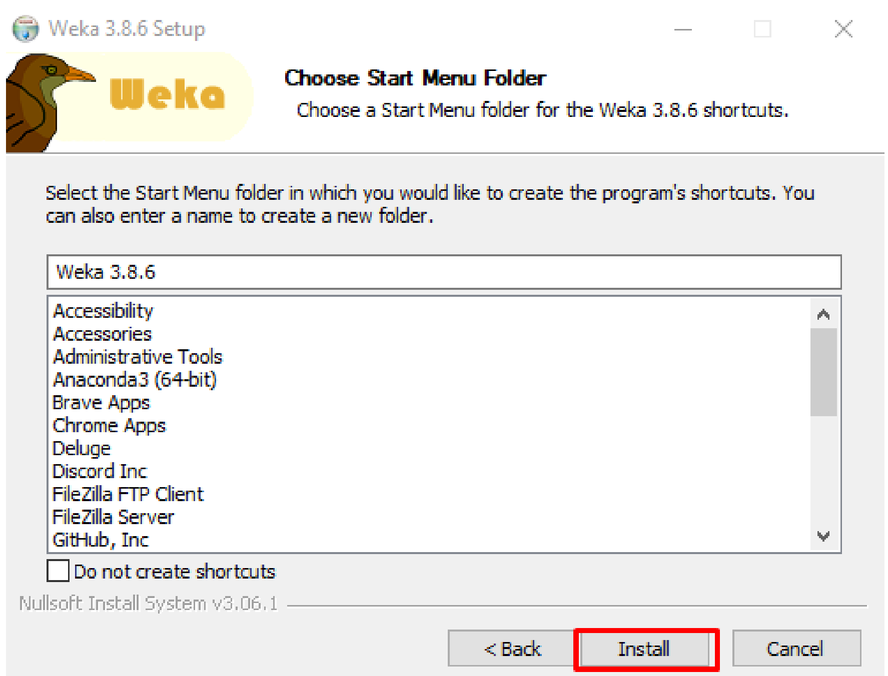
This executable will install Weka in your Program Menu. Launching via the Program Menu or shortcuts will automatically use the included JVM to run Weka.

MAC OS - INTEL PROCESSORS

- Click here to download a disk image for Mac OS that contains a Mac application including

Weka is available in various installation packages, including executable installers, zip files, and macOS disk images. Choose the appropriate package for your operating system.

3. **Install Weka:**



Run the downloaded installer and follow the on-screen instructions to install Weka.

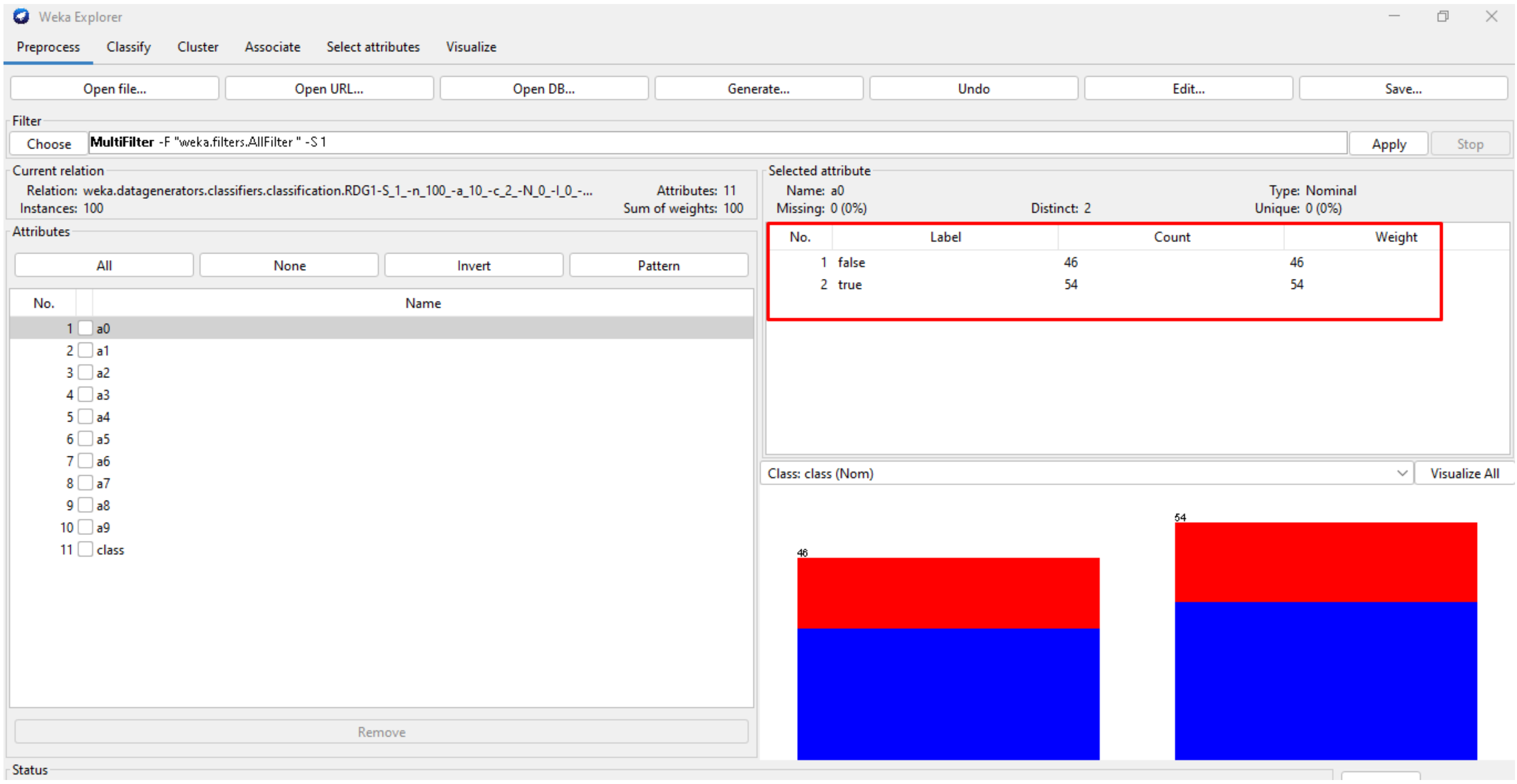
DEMONSTRATION (Apriori)

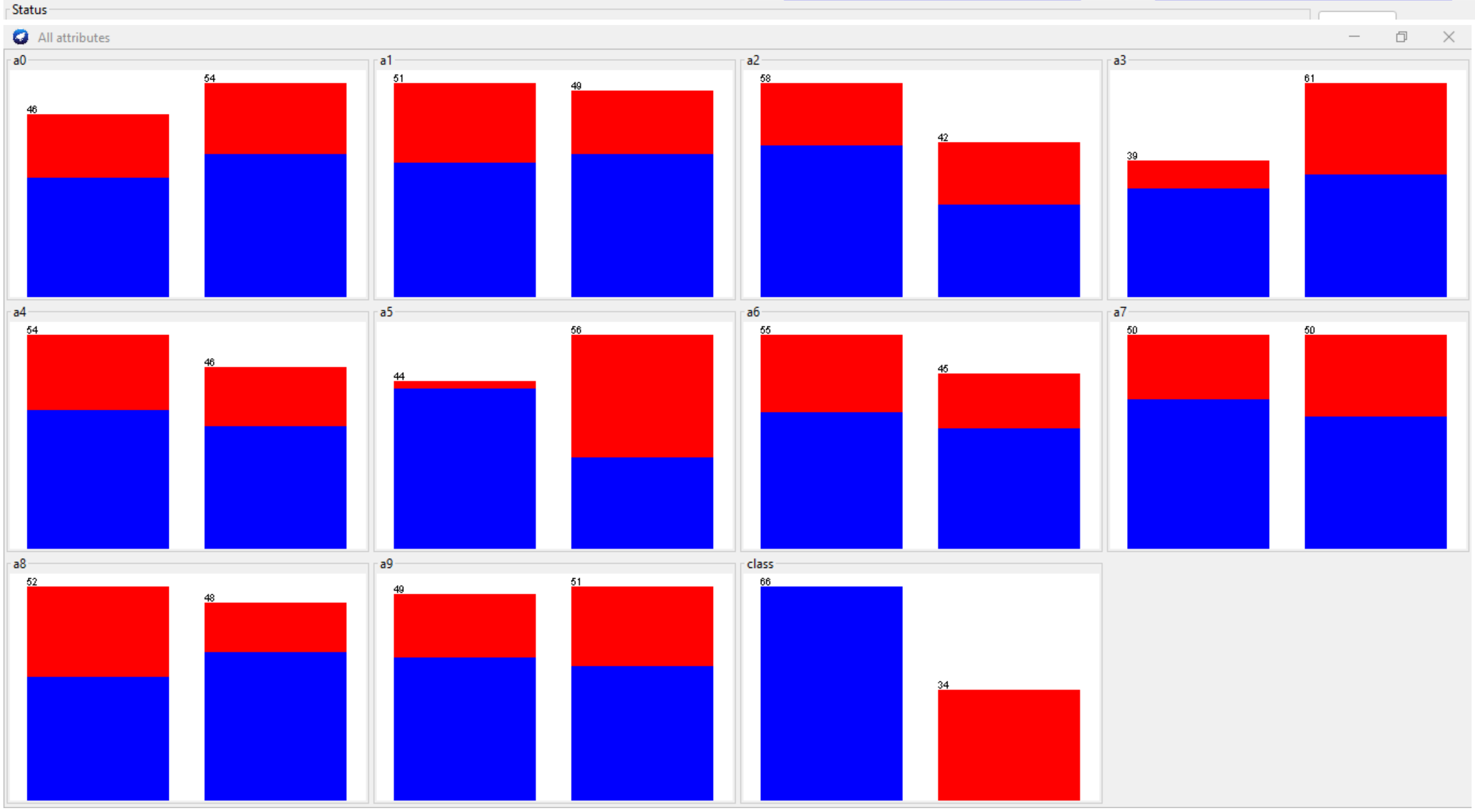
Generated Dataset:

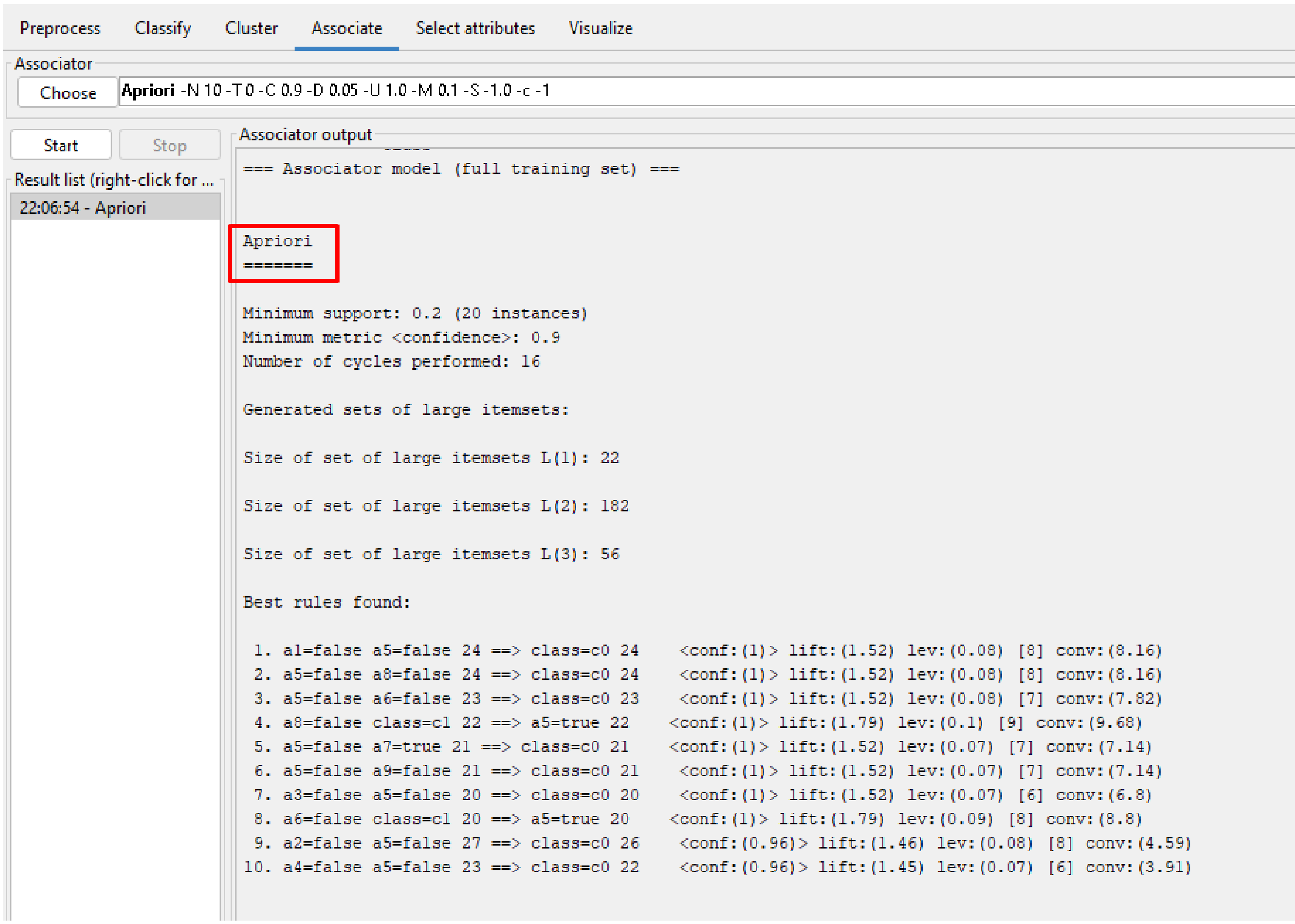
Viewer

Relation: weka.datagenerators.classifiers.classification.RDG1-S_1_-n_100_-a_10_-c_2_-N_0_-I_0_-M_1_-R_10-wek

No.	1: a0 Nominal	2: a1 Nominal	3: a2 Nominal	4: a3 Nominal	5: a4 Nominal	6: a5 Nominal	7: a6 Nominal	8: a7 Nominal	9: a8 Nominal	10: a9 Nominal	11: class Nominal
1	true	false	false	false	true	false	true	true	true	true	c0
2	true	true	false	false	false	true	true	false	true	true	c0
3	false	false	true	true	false	true	false	false	false	false	c1
4	false	true	false	true	true	false	true	true	false	false	c0
5	true	true	false	false	false	true	true	false	false	true	c0
6	true	true	true	true	false	false	true	false	true	true	c1
7	true	true	false	true	true	false	true	false	false	true	c0
8	false	false	true	false	true	true	false	true	false	true	c1
9	true	true	false	true	false	true	false	true	true	true	c0
10	false	false	true	true	true	true	true	true	false	true	c1
11	false	true	false	false	false	true	true	false	true	false	c0
12	true	false	true	false	true	false	false	false	false	true	c0
13	true	true	false	true	true	true	false	false	false	false	c1
14	false	false	false	true	true	true	false	false	true	false	c1
15	false	false	false	true	true	false	false	false	true	false	c0
16	true	false	false	true	false	false	true	false	false	false	c0
17	true	true	true	false	false	true	true	true	false	true	c1
18	true	false	false	true	false	true	false	true	false	false	c0
19	true	false	true	false	false	false	false	false	true	true	c0
20	false	false	false	true	false	true	true	false	false	true	c1
21	false	true	false	true	true	true	false	true	true	false	c1
22	false	true	false	true	true	false	false	false	false	true	c0
23	false	true	true	true	false	false	false	false	true	true	c0
24	false	true	false	false	true	false	false	false	false	false	c0
25	true	true	true	false	false	true	false	false	true	true	c0
26	true	false	true	true	false	false	true	false	true	false	c0
27	true	false	false	true	false	true	false	true	true	true	c0
28	true	false	true	true	true	true	true	true	false	false	c1
29	true	true	true	true	true	true	true	true	true	false	c0
30	true	true	false	true	false	false	true	false	false	true	c0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
MultiFilter -F "weka.filters.AllFilter " -S 1
Apply
Stop
Current relation
Relation: weka.datagenerators.classifiers.classification.RDG1-S_1_-n_100_-a_10_-c_2_-N_0_-I_0_-...
Attributes: 11
Instances: 100
Sum of weights: 100
Attributes
All
None
Invert
Pattern
No.
Name
1 a0
2 a1
3 a2
4 a3
5 a4
6 a5
7 a6
8 a7
9 a8
10 a9
11 class
Remove
Selected attribute
Name: a0
Type: Nominal
Missing: 0 (0%)
Distinct: 2
Unique: 0 (0%)
No.
Label
Count
Weight
1 false 46 46
2 true 54 54
Class: class (Nom)
Visualize All
46
54
Status


All attributes
a0
46
54
a1
51
49
a2
58
42
a3
39
61
a4
54
46
a5
44
56
a6
55
45
a7
50
50
a8
52
48
a9
49
51
class
66
34



Conclusion:

Weka facilitates crucial data preprocessing tasks such as transformation and integration. These tasks ensure data quality and coherence, enhancing the effectiveness of subsequent analyses. Weka's comprehensive approach, encompassing preprocessing alongside advanced analytics, solidifies its position as a versatile and indispensable tool for data mining and analysis.