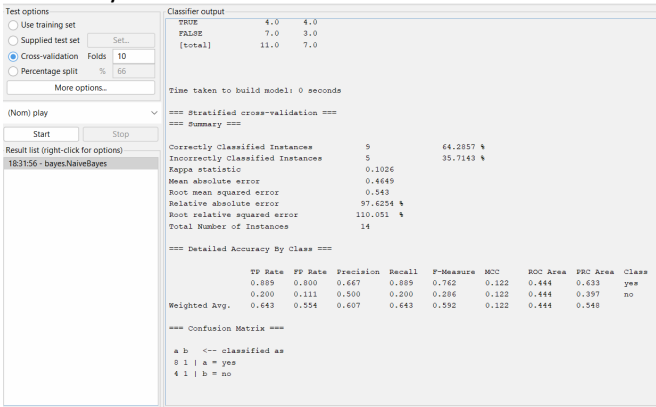


Slips	Question and Answer
Slip 1 Q. 1 Slip 3 Q. 2 Slip 4 Q. 1 Slip 5 Q. 1 Slip 6 Q. 2	<p>Using any open source software such as WEKA and its datasets, perform classification using Naïve Bayes classifier, note accuracy. Describe the output of confusion matrix & state the formula required for calculation of accuracy</p>
Ans.	<p>Step 1: Load Dataset</p> <ol style="list-style-type: none"> 1. Open WEKA and select the "Explorer" interface. 2. Load a dataset (for example, the "Weather.numeric" dataset) <p>Step 2: Select Naïve Bayes Classifier</p> <ol style="list-style-type: none"> 1. Go to the "Classify" tab in WEKA. 2. Under "Classifier," click "Choose," and select NaiveBayes from the list of classifiers. <p>Step 3: Perform Classification</p> <ol style="list-style-type: none"> 1. Choose a testing option (e.g., 10-fold cross-validation). 2. Click "Start" to run the classifier. <p>Step 4: Analyze Results</p> <p>After running the Naïve Bayes classifier, WEKA provides detailed output, including a confusion matrix and accuracy as follows,</p> <p>Output : Navie Bayes on “Weather.numeric” data</p>  <p>Analysis of the Naïve Bayes Classifier Output in WEKA :</p> <p>1. Confusion Matrix</p> <p>The confusion matrix generated as follows:</p> <pre> a b <-- classified as 8 1 a = yes 4 1 b = no </pre> <p>This confusion matrix shows:</p> <ul style="list-style-type: none"> • True Positives (TP) for class "yes" = 8 (instances correctly classified as "yes"). • False Negatives (FN) for class "yes" = 1 (instances that were actually "yes" but classified as "no"). • False Positives (FP) for class "no" = 4 (instances that were actually "no" but classified as "yes").

	<ul style="list-style-type: none"> • True Negatives (TN) for class "no" = 1 (instances correctly classified as "no"). <p>2. Accuracy Calculation Accuracy is calculated as:</p> $\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Instances}} = \frac{8 + 1}{14} = \frac{9}{14} \approx 64.29\%$ <p>3. Conclusion: The Naïve Bayes model correctly classified 64.29% of instances, indicating it performs moderately well on this dataset but not correctly classifying instances of the "no" class.</p>
Slip 1 Q2 Slip 2 Q1 Slip 3 Q1 Slip 4 Q2 Slip 5 Q2 Slip 6 Q2	<p>Using any open source software such as WEKA and its datasets, perform classification using Neural network classifier. Using any available attribute selection algorithm in WEKA note the accuracy and compare with it. Elaborate the working of NN classifier.</p>
Ans.	<p>Step 1: Load the Dataset</p> <ol style="list-style-type: none"> 1. Open WEKA and go to the "Explorer" interface. 2. Load a dataset, such as the "Weather" dataset. <p>Step 2: Apply the Neural Network Classifier</p> <ol style="list-style-type: none"> 1. Go to the "Classify" tab. 2. Under the "Classifier" menu, select Multilayer Perceptron from the functions group. 3. Set the desired parameters (e.g., learning rate, number of epochs, and hidden layers). 4. Choose the evaluation method (such as 10-fold cross-validation) and click "Start" to run the classifier. 5. Record the accuracy from the "Correctly Classified Instances" percentage. <p>Step 3: Perform Attribute Selection</p> <ol style="list-style-type: none"> 1. Go to the "Select attributes" tab. 2. Choose an Attribute Evaluator (e.g., CfsSubsetEval, InfoGainAttributeEval) and a Search Method (e.g., BestFirst, Ranker). 3. Click "Start" to perform attribute selection, which will identify the most relevant features. 4. Record the selected attributes (e.g. outlook and windy). 5. Once selected, go back to the "Classify" tab and perform following step. <p>Step 4: Apply Neural Network with Selected Attributes</p> <ol style="list-style-type: none"> 1. Filter Data to Use Selected Attributes: After identifying the selected attributes (e.g., outlook and windy), remove unselected attributes (in this case, temperature and humidity) through WEKA's Preprocess tab. Ensuring that only the play attribute (the target variable) and the selected attributes are available for training. <p>Step 5: Repeat Step 2</p> <ol style="list-style-type: none"> 1. Run the model to generate classification results using only the selected attributes. <p>Step 6: Compare Results</p> <ol style="list-style-type: none"> 1. Compare the accuracy with and without attribute selection to see if removing less relevant attributes improved the model's performance. <p>Output: Step 2</p>

Classifier

Choose **MultilayerPerceptron** 1.0.3-M 0.2-N 500-V 0.5-G 4-20-H s

Test options

☐ Use training set

☐ Supplied test set

☒ Cross-validation

☐ Percentage split

Set...

Folds: 10

%: 66

More options...

(Nom) play

Start

Stop

Result list (right-click for options)

183156 - bayes.NaiveBayes

192231 - functions.MultilayerPerceptron

Classifier output

Class no

Input

Mode 1

Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===

=== Summary ===

Currently Classified Instances

11

78.5714 %

Incorrectly Classified Instances

3

21.4286 %

Kappa statistic

0.5116

Mean absolute error

0.245

Root mean squared error

0.4627

Relative absolute error

55.6497 %

Root relative squared error

93.7923 %

Total Number of Instances

14

=== Detailed Accuracy By Class ===

TP Rate

FP Rate

Precision

Recall

F-Measure

MCC

ROC Area

PRC Area

Class

0.889

0.400

0.800

0.689

0.842

0.519

0.733

0.857

yes

0.600

0.111

0.750

0.600

0.667

0.519

0.733

0.589

no

Weighted Avg.

0.786

0.237

0.782

0.786

0.779

0.519

0.733

0.761

=== Confusion Matrix ===

a b <-- classified as

0 1 | a = yes

2 3 | b = no

Output: Step 3

Attribute Evaluator

Choose **ChSubsetEval** # 1-E 1

Search Method

Choose **BestFirst** 0 1-N 5

Attribute Selection Mode

☒ Use full training set

☐ Cross-validation

Folds: 10

Seed: 1

No class

Start

Stop

Result list (right-click for options)

192207 - BestFirst - ChSubsetEval

Attribute selection output

=== Run information ===

Evaluator: weka.attributeSelection.ChSubsetEval -P 1 -S 1

Search: weka.attributeSelection.BestFirst -S 1 -P 5

Relation: weather

Instances: 14

Attributes: 5

outlook

temperature

humidity

windy

play

Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:

Best first.

Search uses no attributes

Search direction: forward

Shake search after 0 node expansions

Total number of subsets evaluated: 11

Merit of best subset found: 0.196

Attribute Subset Evaluator (supervised, Class (nominal): 5 play):

CFR Subset Evaluator

Including locally predictive attributes

Selected attributes: 1,4 : 2

outlook

windy

Output: Step 4

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Open file...

Open URL...

Open DB...

Generate...

Undo

Edit...

Save...

Filter

Choose

Name

Apply

Stop

Current relation

Relation weather:weka.attributeSelection.attributeRemoval.R2...

Instances: 14

Attributes: 5

Selected attribute

Name: outlook

Sum of weights: 14

Weight: 0.786

Unselected attribute

Name: temperature

Sum of weights: 0

Weight: 0

Attributes

No.

Name

Invert

Pattern

0.

outlook

2.

windy

3.

play

Class plot (None)

Visualize All

outlook

no

yes

temperature

no

yes

humidity

no

yes

windy

no

yes

play

no

yes

Output: Step 5

Classifier

Choose **MultilayerPerceptron** 1.0.3-M 0.2-N 500-V 0.5-G 4-20-H s

Test options

☐ Use training set

☐ Supplied test set

☒ Cross-validation

☐ Percentage split

Set...

Folds: 10

%: 66

More options...

(Nom) play

Start

Stop

Result list (right-click for options)

183156 - bayes.NaiveBayes

192231 - functions.MultilayerPerceptron

192835 - meta.AttributeSelectedClassifier

193112 - functions.MultilayerPerceptron

194156 - functions.MultilayerPerceptron

Classifier output

Class yes

Input

Mode 0

Class no

Input

Mode 1

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Currently Classified Instances

10

71.4286 %

Incorrectly Classified Instances

4

28.5714 %

Kappa statistic

0.3778

Mean absolute error

0.3088

Root mean squared error

0.4638

Relative absolute error

64.5627 %

Root relative squared error

91.9758 %

Total Number of Instances

14

=== Detailed Accuracy By Class ===

TP Rate

FP Rate

Precision

Recall

F-Measure

MCC

ROC Area

PRC Area

Class

0.778

0.400

0.778

0.778

0.778

0.378

0.778

0.916

yes

0.600

0.222

0.600

0.600

0.600

0.378

0.778

0.563

no

Weighted Avg.

0.714

0.337

0.724

0.714

0.714

0.378

0.778

0.791

=== Confusion Matrix ===

a b <-- classified as

7 2 | a = yes

2 5 | b = no

Output: Step 6

Accuracy Comparison

- **Full Attribute Set Accuracy: 78.57%**
- **Selected Attribute Set Accuracy: 71.43%**

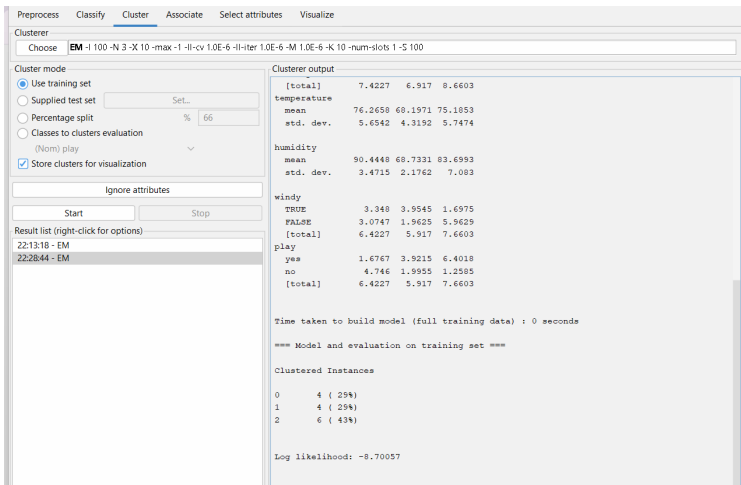
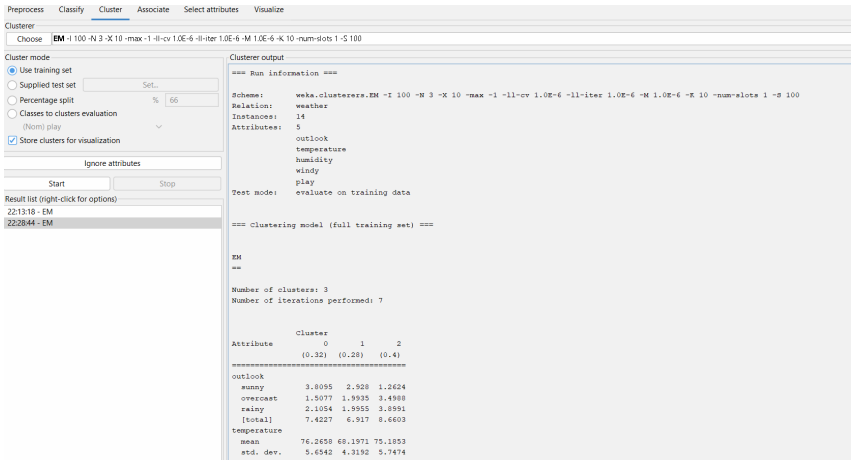
The accuracy of the model decreased when using the selected attributes (outlook and windy) compared to using the full set of attributes (outlook, temperature, humidity,

windy). It concludes that while attribute selection can simplify the model, it may also remove important information that contributes to accurate predictions.

Slip 1 Q. 2
Slip 2 Q. 2 **Using any open source software such as WEKA and its datasets, perform clustering using ‘EM’ algorithm. State about the generation of confusion matrix & its accuracy.**

- Step 1: Load the Dataset**
1. Open WEKA and go to the "Explorer" interface.
 2. Load a dataset, such as the "Weather" dataset.
- Step 2: Select the EM Algorithm:**
- Click on the "**Cluster**" tab.
 - In the clustering algorithms list, select "**EM**" (Expectation-Maximization).
 - **Set** the parameters for the EM algorithm, such as the number of clusters(right click on EM and set the parameters) or leave the default settings.
 - Click on the "**Start**" button to run the EM algorithm.
 - Once it completes, you will see the clustering results in the output area.
 - View Results about the clusters generated by the EM algorithm.

Output:



Confusion Matrix and Accuracy

categorize the clusters based on the majority class in each, the confusion matrix could be constructed as follows

	<table><tr><th></th><th>Predicted Yes</th><th>Predicted No</th></tr><tr><td>Actual Yes</td><td>8</td><td>1</td></tr><tr><td>Actual No</td><td>3</td><td>2</td></tr></table> <p>Calculate Accuracy:</p> <ul style="list-style-type: none">• Total Predictions: 14• Correct Predictions: Count of the true positives and true negatives.• Calculate accuracy as $\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}} = \frac{8 + 2}{14} \approx 0.714 \text{ or } 71.4\%$		Predicted Yes	Predicted No	Actual Yes	8	1	Actual No	3	2
	Predicted Yes	Predicted No								
Actual Yes	8	1								
Actual No	3	2								
Slip2 Q.2 Slip4 Q.2	<p>Implement K-Means clustering/ hierarchical clustering on sales_data_sample.csv dataset. Determine the number of clusters using the elbow method. Elaborate the output with respective clustering method.</p> <p>Dataset link : https://www.kaggle.com/datasets/kyanyoga/sample-sales-data</p>									
	<p>Step 1: Open WEKA and Load the Dataset</p> <ol style="list-style-type: none">1. Start WEKA: Open WEKA and go to the Explorer interface.2. Load the Dataset:<ul style="list-style-type: none">○ Click Open file... and select sales_data_sample.csv to load the dataset into WEKA.○ If the dataset is not in ARFF format, you might need to convert it to ARFF first or preprocess it within WEKA to ensure all values are numerical, as clustering requires numeric data. <p>Step 2: Preprocess the Data</p> <ol style="list-style-type: none">1. Data Preparation:<ul style="list-style-type: none">○ Use Select Attributes to choose the relevant numerical features for clustering (e.g., Sales Amount, Quantity Ordered, etc.).○ Ensure that non-numeric attributes (e.g., categorical data like Customer Name) are excluded or transformed using filters if needed.2. Standardize the Data:<ul style="list-style-type: none">○ In the Preprocess tab, go to Filters > Unsupervised > Attribute > Standardize.○ Apply this filter to normalize the features, which improves clustering results. <p>Step 3: Implement K-Means Clustering</p> <ol style="list-style-type: none">1. Go to the Cluster Tab:<ul style="list-style-type: none">○ Switch to the Cluster tab.○ Click Choose and select SimpleKMeans from the list of clustering algorithms.2. Set K-Means Parameters:<ul style="list-style-type: none">○ Click on SimpleKMeans to open its parameters.○ Set the number of clusters to an initial value (e.g., 3).○ Check Display standard deviations to better interpret cluster characteristics.○ Leave other parameters at their default settings for the first run.3. Run K-Means:<ul style="list-style-type: none">○ Click Start to run the K-Means clustering.○ WEKA will display the results in the output panel, including information on each cluster's centroid and the number of instances in each cluster.4. Use the Elbow Method to Determine Optimal Clusters:									

	<ul style="list-style-type: none"> ○ To use the Elbow Method, run K-Means multiple times with different values for k (e.g., from 1 to 10). ○ For each run, note the Sum of Squared Errors (SSE) in the output. ○ Plot these SSE values (manually, in a spreadsheet, or using another software) to identify the "elbow" point, where the decrease in SSE slows down significantly. <p>Interpretation: The optimal number of clusters is where this "elbow" appears. Set k to this value in your final K-Means model in WEKA.</p> <p>Step 4: Implement Hierarchical Clustering</p> <ol style="list-style-type: none"> Choose Hierarchical Clustering: <ul style="list-style-type: none"> ○ In the Cluster tab, click Choose and select HierarchicalClusterer. ○ Open the parameter options by clicking on HierarchicalClusterer. Set Parameters for Hierarchical Clustering: <ul style="list-style-type: none"> ○ Select the Link type (e.g., Ward's method, Single Linkage, Complete Linkage). ○ Specify the number of clusters (if known), or leave it blank to allow WEKA to determine a default based on the dendrogram. Run Hierarchical Clustering: <ul style="list-style-type: none"> ○ Click Start to run the hierarchical clustering. ○ WEKA's output will include cluster details, typically showing how clusters are grouped at each step. Visualize Using Dendrogram (Optional): <ul style="list-style-type: none"> ○ WEKA does not directly support dendrogram visualization in the Explorer interface. However, you can try exporting the data and use Python, R, or specialized software to generate the dendrogram and determine the number of clusters by visually inspecting where clusters naturally split. <p>Step 5: Analyze the Output</p> <ul style="list-style-type: none"> ● K-Means Clustering Output: <ul style="list-style-type: none"> ○ WEKA will show the centroids of each cluster, providing insight into the average feature values in each cluster. ○ You'll see the number of instances in each cluster, which helps evaluate if clusters are well-balanced or skewed. ○ Based on the elbow method, the optimal k can give you a balanced clustering solution, avoiding too few or too many clusters. ● Hierarchical Clustering Output: <ul style="list-style-type: none"> ○ Hierarchical clustering output shows which data points or clusters are grouped together in a stepwise manner. ○ Interpreting this can reveal patterns where smaller sub-clusters merge into larger clusters, helpful for identifying natural groupings in the data. <p>Interpretation</p> <ul style="list-style-type: none"> ● K-Means: The final model provides distinct, flat clusters. This method is effective if you need a set number of clusters and works well with large datasets. ● Hierarchical Clustering: Hierarchical clustering provides a nested view of clusters. It's useful when you want to understand data structure at multiple levels, especially in smaller datasets.
Slip 3 Q. 2	Using any open source software such as WEKA and its datasets, perform classification using C4.5 – the decision tree classifier. Describe the output of confusion matrix & state the formula required for calculation of accuracy.
Slip 5	Using any open source software such as WEKA and its datasets, perform classification

<p>Q. 2 Slip6 Q. 1</p>	<p>using C4.5 – the decision tree classifier. Using both attribute and instance selection algorithm in WEKA and note the accuracy and compare with it. Elaborate the working of Decision tree classifier.</p>
	<p>Steps for Classification in WEKA using C4.5 (J48)</p> <ol style="list-style-type: none"> 1. Open WEKA: Start WEKA and choose the "Explorer" option. 2. Load Dataset: <ul style="list-style-type: none"> ○ Click on "Open file" in the Preprocess tab. ○ Load a dataset in ARFF, CSV, or other supported formats. WEKA provides sample datasets like "iris" or "weather," which you can use for testing. 3. Set Up Classification: <ul style="list-style-type: none"> ○ Go to the "Classify" tab. ○ Select the classifier by clicking "Choose" and navigating to trees > J48. J48 is WEKA's implementation of the C4.5 algorithm. 4. Select the Target Attribute: <ul style="list-style-type: none"> ○ Set the class attribute (the attribute to predict) by choosing the relevant target variable (often selected by default). 5. Run the Classification: <ul style="list-style-type: none"> ○ Click "Start" to run the classification. WEKA will build the decision tree using J48 and show the results in the "Classifier output" panel. <p>Understanding the Output: Confusion Matrix and Accuracy</p> <p>Confusion Matrix</p> <p>The confusion matrix shows the performance of the classification model, comparing actual versus predicted class labels.</p>
<p>Slip 7 Q.1</p>	<p>Find the missing values from “<i>weather.arff</i>” dataset and replace that values using WEKA. Analyze how it will affect the dataset</p>
	<p>Steps to Identify and Replace Missing Values in WEKA</p> <ol style="list-style-type: none"> 1. Open WEKA: Launch WEKA and select the “Explorer” option. 2. Load the Dataset: <ul style="list-style-type: none"> ○ Click on the “Open file” button in the Preprocess tab. ○ Select and load the weather.arff dataset. This dataset typically includes missing values, which can be identified and imputed. 3. Identify Missing Values: <ul style="list-style-type: none"> ○ In the Preprocess tab, scroll through the dataset attributes. ○ Missing values will be indicated with a “?” symbol in WEKA's attribute statistics or summary, often next to specific attributes showing incomplete data. 4. Replace Missing Values: <ul style="list-style-type: none"> ○ In the Preprocess tab, click on “Choose” under the “Filter” section. ○ Select filters > unsupervised > attribute > ReplaceMissingValues. ○ This filter automatically replaces missing values for each attribute with a suitable substitution: <ul style="list-style-type: none"> ▪ For numeric attributes, the mean value of the attribute is used. ▪ For categorical (nominal) attributes, the most frequent (mode) value is used. 5. Apply the Filter: <ul style="list-style-type: none"> ○ After selecting the filter, click “Apply” to implement the changes. The missing values will be replaced according to the substitution strategy. ○ You can check if the missing values are filled by reviewing the dataset in the Preprocess tab.

Slip 7 Q.2	Scale all features between 0 and 1 of <i>"iris.arff"</i> dataset using WEKA. Compare the statistics of one feature before and after normalization.
	<p>Steps to Normalize Features in WEKA</p> <ol style="list-style-type: none"> 1. Open WEKA: Launch WEKA and choose the "Explorer" option. 2. Load the Dataset: <ul style="list-style-type: none"> ○ In the "Preprocess" tab, click "Open file." ○ Select the iris.arff dataset and load it into WEKA. 3. Check Original Statistics: <ul style="list-style-type: none"> ○ Before normalization, review the original statistics for a feature. ○ In the Preprocess tab, click on an attribute (e.g., <i>petal length</i> or <i>sepal width</i>) in the list of attributes. ○ The summary statistics for this attribute (e.g., minimum, maximum, mean, and standard deviation) will be displayed on the right side. 4. Apply Normalization: <ul style="list-style-type: none"> ○ In the "Filter" section under the Preprocess tab, click on "Choose." ○ Select filters > unsupervised > attribute > Normalize. ○ This filter scales all numeric attributes to a specified range. By default, it normalizes them to a range between 0 and 1. 5. Set the Normalization Range (Optional): <ul style="list-style-type: none"> ○ Click on the "Normalize" filter to open its options if you need to adjust the range. By default, it is set to [0,1], so no changes are necessary here. 6. Apply the Filter: <ul style="list-style-type: none"> ○ Click "Apply" to implement normalization. WEKA will rescale all numeric attributes to fall between 0 and 1. 7. Compare the Statistics: <ul style="list-style-type: none"> ○ After normalization, select the same attribute you noted in Step 3 to observe the updated statistics. ○ You should see the minimum value now as 0 and the maximum value as 1 (or very close, due to precision), with other statistics adjusted accordingly. <p>Example Comparison of Statistics for a Feature</p> <p>Suppose you chose the <i>petal length</i> attribute. Before normalization, it might have statistics such as:</p> <ul style="list-style-type: none"> • Minimum: 1.0 • Maximum: 6.9 • Mean: 3.8 <p>After normalization, the statistics for <i>petal length</i> should look similar to:</p> <ul style="list-style-type: none"> • Minimum: 0.0 • Maximum: 1.0 • Mean: (scaled to the new range)
Slip 7 Q.2	Use the Apriori algorithm to find association rules on <i>weather.nominal.arff</i> dataset using WEKA. Interpret top 3 association rules and their confidence values.
	<p>Steps to Apply the Apriori Algorithm in WEKA</p> <ol style="list-style-type: none"> 1. Open WEKA: Start WEKA and select the "Explorer" option. 2. Load the Dataset: <ul style="list-style-type: none"> ○ Click on the "Open file" button in the Preprocess tab. ○ Load the weather.nominal.arff dataset, which contains nominal attributes suited for association rule mining. 3. Set Up the Apriori Algorithm: <ul style="list-style-type: none"> ○ Go to the "Associate" tab.

	<ul style="list-style-type: none"> Under “Choose,” select associations > Apriori to use the Apriori algorithm for finding association rules. <ol style="list-style-type: none"> Configure Apriori Settings (Optional): <ul style="list-style-type: none"> Click on the Apriori algorithm name to open its configuration options. Here, you can adjust parameters like minMetric (minimum confidence threshold), numRules (number of rules to find), and lowerBoundMinSupport (minimum support threshold). For example, you might set numRules to a higher number if you want to see more than the default (10) rules. Run Apriori: <ul style="list-style-type: none"> Click “Start” to execute the Apriori algorithm. WEKA will generate association rules based on the dataset and display them in the output window. Examine the Results: <ul style="list-style-type: none"> In the output, you will see a list of association rules with metrics like support and confidence. Note down the top three rules and their respective confidence values for interpretation. <p>Understanding Confidence</p> <ul style="list-style-type: none"> Confidence is a measure of the rule's reliability, indicating how often the rule's outcome is true when its conditions are met. A higher confidence value suggests a stronger rule. In the examples above, the rule with outlook overcast → play=yes has the highest confidence, suggesting it is the most reliable among the three.
Slip 8 Q.1	Apply the J48 algorithm on “ <i>iris.arff</i> ” dataset and classify the data using WEKA. Report the accuracy and visualize the resulting decision tree.
	<p>Steps to Apply J48 in WEKA</p> <ol style="list-style-type: none"> Open WEKA: Start WEKA and choose the “Explorer” option. Load the Dataset: <ul style="list-style-type: none"> In the “Preprocess” tab, click “Open file.” Load the iris.arff dataset, which contains features for classifying iris species. Set Up the J48 Classifier: <ul style="list-style-type: none"> Go to the “Classify” tab. Under the “Choose” button, select trees > J48. J48 is WEKA's implementation of the C4.5 decision tree algorithm. Configure J48 Settings (Optional): <ul style="list-style-type: none"> If you want to fine-tune the model, click on the J48 classifier name. Options include setting the confidence factor for pruning and minimum number of instances per leaf. However, the default settings work well for the iris.arff dataset. Select Evaluation Method: <ul style="list-style-type: none"> In the “Test options” section, choose the evaluation method. Select “Use training set” for training accuracy, or choose “Cross-validation” (e.g., 10-fold cross-validation) for a more generalized accuracy estimate. Run the Classifier: <ul style="list-style-type: none"> Click “Start” to run the J48 classifier on the dataset. WEKA will output the classification results, including accuracy and detailed performance metrics. <p>Reporting the Accuracy</p>

	<ul style="list-style-type: none"> In the output, look for the Correctly Classified Instances percentage under “Summary” in the "Classifier output" section. This percentage is the model’s accuracy. For example, if the accuracy is 96%, it means the J48 classifier correctly predicted the iris species 96% of the time on the dataset. <p>Visualize the Decision Tree</p> <ol style="list-style-type: none"> View the Tree: <ul style="list-style-type: none"> In the “Result list” section, right-click on the result you just generated and select Visualize tree. This will open a window displaying the decision tree structure created by the J48 algorithm. Interpret the Tree Visualization: <ul style="list-style-type: none"> The visualization shows the decision nodes based on the features (e.g., petal length, sepal width) used to classify each iris species. Each branch represents a decision path, and leaf nodes represent the final classification outcomes (e.g., Iris-setosa, Iris-versicolor, or Iris-virginica).
Slip 8 Q.2	<p>Use the Naive Bayes classifier on <i>weather.nominal.arff</i> dataset and classify the data using WEKA. Note the accuracy of the model</p>
	<ul style="list-style-type: none"> Steps: <ul style="list-style-type: none"> Open the <i>weather.nominal.arff</i> dataset. In the Classify tab, choose Classifier → bayes → NaiveBayes. Click Start to run the classification. <p>View the results in the Classifier output panel for accuracy</p> <p>Steps to Apply Naive Bayes in WEKA</p> <ol style="list-style-type: none"> Open WEKA: Start WEKA and select the “Explorer” option. Load the Dataset: <ul style="list-style-type: none"> In the “Preprocess” tab, click on “Open file.” Load the <i>weather.nominal.arff</i> dataset. Set Up the Naive Bayes Classifier: <ul style="list-style-type: none"> Go to the “Classify” tab. Click on “Choose” under “Classifier,” and select bayes > NaiveBayes to choose the Naive Bayes classifier. Select Evaluation Method: <ul style="list-style-type: none"> In the “Test options” section, choose an evaluation method: <ul style="list-style-type: none"> Use training set: Evaluates the model’s accuracy on the training data. Cross-validation: Select 10-fold cross-validation for a more generalized accuracy measurement. Percentage split: Splits the data (e.g., 70% for training and 30% for testing). Cross-validation is typically recommended for a reliable accuracy estimate. Run the Classifier: <ul style="list-style-type: none"> Click “Start” to run the Naive Bayes classifier on the dataset. WEKA will display the classification results in the “Classifier output” section. <p>Reporting the Model’s Accuracy</p> <ul style="list-style-type: none"> After the classifier has finished running, look under the “Summary” section in the output window for Correctly Classified Instances. This shows the percentage of instances correctly classified by the Naive Bayes model, representing the model’s

	<p>accuracy.</p> <ul style="list-style-type: none"> For example, if the output shows “Correctly Classified Instances: 85%,” this indicates that the Naive Bayes classifier achieved an 85% accuracy on the weather.nominal.arff dataset.
Slip 8 Q.2	<p>Apply the KNN classifier with $k=3$ on “iris.arff” dataset. Compare the performance with $k=5$ and explain which value of k performs better and why using WEKA.</p> <p>Steps to Apply KNN in WEKA with $k=3$</p> <ol style="list-style-type: none"> Open WEKA: Start WEKA and select the “Explorer” option. Load the Dataset: <ul style="list-style-type: none"> In the “Preprocess” tab, click on “Open file.” Load the iris.arff dataset. Set Up the KNN Classifier: <ul style="list-style-type: none"> Go to the “Classify” tab. Click on “Choose” under the “Classifier” section. Select lazy > IBk, which is WEKA’s implementation of the K-Nearest Neighbors (KNN) algorithm. Configure $k=3$: <ul style="list-style-type: none"> Click on “IBk” to open its configuration options. In the kNN parameter, set $k = 3$. You can also set the distance metric (default is Euclidean distance). Select Evaluation Method: <ul style="list-style-type: none"> In the “Test options” section, choose an evaluation method: <ul style="list-style-type: none"> Cross-validation (e.g., 10-fold cross-validation) is recommended for a reliable performance comparison. Alternatively, use “Percentage split” (e.g., 70% for training and 30% for testing) if desired. Run the Classifier with $k=3$: <ul style="list-style-type: none"> Click “Start” to run the classifier. The “Classifier output” section will display performance metrics, including accuracy. <p>Steps to Apply KNN with $k=5$</p> <ol style="list-style-type: none"> Set $k=5$: <ul style="list-style-type: none"> Click on “IBk” again to open its configuration options. Change the kNN parameter to $k = 5$. Run the Classifier with $k=5$: <ul style="list-style-type: none"> Click “Start” again to classify with $k=5$. The output will display the results for $k=5$, including the accuracy and other performance metrics. <p>Comparing the Performance of $k=3$ and $k=5$</p> <p>After running both models, compare the Correctly Classified Instances (accuracy) for $k=3$ and $k=5$. Additionally, look at other performance metrics like precision, recall, and F-measure if available.</p> <p>Determining Which k Value Performs Better</p> <ul style="list-style-type: none"> Higher accuracy generally indicates better performance. Observe which k value gives higher accuracy. Effect of k on Bias and Variance: <ul style="list-style-type: none"> A smaller k (e.g., $k=3$) tends to fit the model more closely to the training data, capturing local patterns but may introduce more variance (sensitivity to noise). A larger k (e.g., $k=5$) considers more neighbors, which can smooth the decision boundary and reduce the effect of noise but may reduce

	sensitivity to finer distinctions.
Slip 9 Q.1	Use the J48 classifier and perform 10-fold cross-validation on the “ <i>cancer.arff</i> ” Dataset using WEKA. Note the accuracy, and analyze how does it compare to training on the full dataset.
	<p>Steps to Apply the J48 Classifier with 10-Fold Cross-Validation</p> <ol style="list-style-type: none"> Open WEKA: <ul style="list-style-type: none"> Launch WEKA and choose the Explorer option. Load the Dataset: <ul style="list-style-type: none"> In the Preprocess tab, click on Open file. Load the cancer.arff dataset. Set Up the J48 Classifier: <ul style="list-style-type: none"> Go to the Classify tab. Click Choose under the "Classifier" section. Select trees > J48. This is WEKA’s implementation of the C4.5 decision tree algorithm. Set Evaluation Method to 10-Fold Cross-Validation: <ul style="list-style-type: none"> In the Test options section, select Cross-validation and set 10 for the number of folds. This will perform 10-fold cross-validation to evaluate the model’s performance. Run the Classifier with Cross-Validation: <ul style="list-style-type: none"> Click the Start button to run the J48 classifier with 10-fold cross-validation. WEKA will display the results in the Classifier output section, including performance metrics such as accuracy, precision, recall, and F-measure. <p>Reporting Accuracy and Comparing with Full Dataset</p> <ol style="list-style-type: none"> View Accuracy from Cross-Validation: <ul style="list-style-type: none"> In the Classifier output, look for the Correctly Classified Instances section. The percentage listed here represents the accuracy of the classifier using 10-fold cross-validation (e.g., “Correctly Classified Instances: 95%”). Compare Performance on Full Dataset: <ul style="list-style-type: none"> Now, we will train the classifier on the full dataset and compare its accuracy. Training on the Full Dataset: <ul style="list-style-type: none"> In the Test options section, choose Use training set. This will train the model on the entire dataset without cross-validation. Click Start to run the classifier and display the results. Again, check the Correctly Classified Instances to see the accuracy for the full dataset. <p>Analyzing the Results</p> <ol style="list-style-type: none"> Accuracy from Cross-Validation: <ul style="list-style-type: none"> Cross-validation gives an estimate of model performance on unseen data, which is often lower than training on the full dataset. It helps reduce overfitting and gives a more generalized performance metric. Accuracy from Full Dataset: <ul style="list-style-type: none"> Training on the full dataset often yields higher accuracy because the model is evaluated on the same data it was trained on, which can lead to overfitting (i.e., the model may perform well on the training set but poorly on unseen data). Comparing the Two:

	<ul style="list-style-type: none"> ○ If the accuracy from 10-fold cross-validation is slightly lower than training on the full dataset, this is expected and indicates that the model is generalizing well. ○ If the accuracy from cross-validation is significantly lower, it might suggest that the model is overfitting to the training data when trained on the full dataset.
Slip 9 Q.2	<p>Train a Random Forest classifier and J48 classifier on the "<i>iris.arff</i>" dataset using WEKA. Compare performance of the both the classifier.</p>
	<p>Steps to Train the Random Forest Classifier in WEKA</p> <ol style="list-style-type: none"> 1. Open WEKA: <ul style="list-style-type: none"> ○ Launch WEKA and select the Explorer option. 2. Load the Dataset: <ul style="list-style-type: none"> ○ In the Preprocess tab, click Open file and load the iris.arff dataset. 3. Select the Random Forest Classifier: <ul style="list-style-type: none"> ○ Go to the Classify tab. ○ Click Choose under the "Classifier" section. ○ Select trees > RandomForest. This is WEKA's implementation of the Random Forest classifier. 4. Configure Random Forest (Optional): <ul style="list-style-type: none"> ○ You can adjust settings such as the number of trees (default is 100) and the number of features to consider when splitting nodes (default is the square root of the number of features). ○ You can also adjust other parameters, but for now, using the default settings will work for the comparison. 5. Set Evaluation Method: <ul style="list-style-type: none"> ○ In the Test options section, select Cross-validation and set 10 for the number of folds. This will perform 10-fold cross-validation to evaluate the Random Forest model. 6. Run the Random Forest Classifier: <ul style="list-style-type: none"> ○ Click the Start button to train and evaluate the Random Forest classifier. ○ WEKA will display the performance metrics, including accuracy, in the Classifier output section. <p>Steps to Train the J48 Classifier in WEKA</p> <ol style="list-style-type: none"> 1. Select the J48 Classifier: <ul style="list-style-type: none"> ○ In the Classify tab, click Choose under the "Classifier" section. ○ Select trees > J48, which is WEKA's implementation of the C4.5 decision tree algorithm. 2. Set Evaluation Method: <ul style="list-style-type: none"> ○ As with Random Forest, choose Cross-validation and set 10 for the number of folds to perform 10-fold cross-validation. 3. Run the J48 Classifier: <ul style="list-style-type: none"> ○ Click Start to train and evaluate the J48 classifier. ○ The Classifier output will show performance metrics for the J48 classifier, including accuracy. <p>Comparing the Performance of Both Classifiers</p> <ol style="list-style-type: none"> 1. Check Accuracy: <ul style="list-style-type: none"> ○ After running both classifiers, check the Correctly Classified Instances percentage in the Classifier output for both models. ○ This gives the accuracy for both the Random Forest and J48 classifiers. 2. Compare Other Metrics:

	<ul style="list-style-type: none"> ○ In the Classifier output, you will also see the Confusion Matrix, Precision, Recall, and F-measure. ○ These metrics provide more details about the performance of each classifier, especially when dealing with imbalanced datasets or to assess model robustness. <p>3. Visual Comparison:</p> <ul style="list-style-type: none"> ○ Compare the accuracy from both classifiers. The Random Forest classifier typically has better generalization performance because it builds multiple decision trees and averages their results. ○ J48, being a single decision tree, may overfit or underperform compared to Random Forest in certain cases. <p>Interpreting Results</p> <ul style="list-style-type: none"> ● Random Forest: <ul style="list-style-type: none"> ○ Random Forest aggregates the results of multiple decision trees, making it more robust to overfitting, especially for datasets with noise. ○ It typically performs better on datasets with complex relationships or noisy features. ● J48: <ul style="list-style-type: none"> ○ J48 creates a single decision tree and might overfit on the training data if the tree is too deep. ○ It may have a higher variance compared to Random Forest and can perform worse if there is noise or outliers in the dataset. <p>Conclusion</p> <ul style="list-style-type: none"> ● By comparing accuracy and other metrics such as precision and recall, you can assess which classifier performs better on the iris.arff dataset. ● Random Forest usually performs better due to its ensemble nature, while J48 might be faster but could have slightly lower performance depending on the dataset.
Slip 9 Q.2	Apply k-Means clustering with k=3 on <i>iris.arff</i> dataset using WEKA. Find out centroids of the clusters
	<p>Steps to Apply k-Means Clustering in WEKA</p> <ol style="list-style-type: none"> 1. Open WEKA: <ul style="list-style-type: none"> ○ Start WEKA and choose the Explorer option. 2. Load the Dataset: <ul style="list-style-type: none"> ○ In the Preprocess tab, click Open file. ○ Load the iris.arff dataset, which contains the data you will cluster. 3. Select k-Means Algorithm: <ul style="list-style-type: none"> ○ Go to the Cluster tab. ○ Click Choose under the "Clusterer" section. ○ Select weka > clusters > SimpleKMeans to apply the k-Means clustering algorithm. 4. Set the Number of Clusters (k=3): <ul style="list-style-type: none"> ○ Click on SimpleKMeans to open the options. ○ In the options dialog, set numClusters to 3. This specifies that you want to create 3 clusters. 5. Run the k-Means Clustering Algorithm: <ul style="list-style-type: none"> ○ Once you've set k=3, click OK to apply the changes. ○ Click Start to run the k-Means clustering on the iris.arff dataset. 6. View the Results: <ul style="list-style-type: none"> ○ After the algorithm finishes running, the Cluster output will appear in

	<p>the Result list.</p> <ul style="list-style-type: none"> ○ In the output, you will see the cluster centroids for each of the 3 clusters. <p>Finding the Centroids of the Clusters</p> <ol style="list-style-type: none"> 1. Centroid Information: <ul style="list-style-type: none"> ○ The Cluster output section will show the centroids for each of the 3 clusters. The centroids are represented as the mean values of the attributes for the instances within each cluster. For example: <ul style="list-style-type: none"> ▪ Centroid for Cluster 1: The mean value for each attribute (e.g., sepal length, sepal width, petal length, petal width) for all instances assigned to Cluster 1. ▪ Similarly, the centroids for Cluster 2 and Cluster 3 will be displayed. 2. Centroid Format: <ul style="list-style-type: none"> ○ The centroids will be listed for each cluster, showing the values of each attribute (e.g., sepal length, sepal width, petal length, and petal width) for the center of each cluster. <p>Interpreting the Results</p> <ul style="list-style-type: none"> • The centroid values represent the average value of each attribute for the data points assigned to each cluster. • These centroids are useful for understanding the general characteristics of each cluster. <ul style="list-style-type: none"> ○ For example, Cluster 1 might be mostly made up of Iris-setosa flowers (small sepal and petal measurements). ○ Cluster 2 might represent Iris-versicolor (medium-sized flowers), and Cluster 3 might correspond to Iris-virginica (larger flowers). <p>By applying k-Means clustering with $k=3$, WEKA groups the data into three clusters based on the similarities in the attributes, and the centroids represent the center points of these clusters.</p>
Slip 10 Q.1	<p>Apply the hierarchical clustering algorithm on <i>iris.arff</i> dataset using WEKA. Interpret the pattern and formation of the cluster using dendrogram.</p>
	<p>Steps to Apply Hierarchical Clustering in WEKA</p> <ol style="list-style-type: none"> 1. Open WEKA: <ul style="list-style-type: none"> ○ Launch WEKA and select the Explorer option. 2. Load the Dataset: <ul style="list-style-type: none"> ○ In the Preprocess tab, click Open file and load the iris.arff dataset. 3. Select the Hierarchical Clustering Algorithm: <ul style="list-style-type: none"> ○ Go to the Cluster tab. ○ Click Choose under the "Clusterer" section. ○ Select weka > clusters > HierarchicalClusterer to apply the hierarchical clustering algorithm. 4. Configure Hierarchical Clustering (Optional): <ul style="list-style-type: none"> ○ Click on the HierarchicalClusterer to open its options. ○ By default, the Agglomerative method is selected, which is a bottom-up approach where individual instances are merged into clusters. ○ You can adjust other settings, such as the distance function (Euclidean, Manhattan, etc.) and the linkage method (single, complete, or average linkage). The default setting is usually fine for this analysis. 5. Run the Hierarchical Clustering Algorithm: <ul style="list-style-type: none"> ○ Once the settings are configured, click Start to run the hierarchical

	<p>clustering algorithm.</p> <ul style="list-style-type: none"> WEKA will process the data and generate the cluster results. <p>6. View the Dendrogram:</p> <ul style="list-style-type: none"> In the Result list section, you will see the output for the HierarchicalClusterer. To visualize the clusters, click on the visualize button (located in the result list) to generate a dendrogram. <p>Interpreting the Dendrogram</p> <p>1. Understanding the Dendrogram:</p> <ul style="list-style-type: none"> The dendrogram is a tree-like diagram that shows the arrangement of clusters based on their similarity. The x-axis of the dendrogram represents the individual data points (instances). The y-axis represents the distance (or dissimilarity) at which clusters are merged. Initially, each instance starts as its own cluster. As you move up the dendrogram, clusters merge based on their similarity, with the height of the merge indicating how similar the clusters are. <p>2. Reading the Dendrogram:</p> <ul style="list-style-type: none"> Lower part of the dendrogram: Shows individual data points as separate clusters (at the bottom of the tree). Higher part of the dendrogram: As you go higher, clusters are merged based on their similarity. The branches show the sequence of cluster merges, with the height of the branches indicating the dissimilarity between the clusters. The point at which branches connect indicates the distance (or dissimilarity) at which two clusters merge. <p>3. Interpreting the Cluster Formation:</p> <ul style="list-style-type: none"> Cluster Merging: If two clusters merge at a very low height (distance), it means they are very similar. Conversely, if they merge at a higher distance, they are less similar. In the iris dataset, for example, you might observe that the clusters corresponding to Iris-setosa, Iris-versicolor, and Iris-virginica merge at different heights. This indicates the degree of similarity between the different species.
Slip 10 Q.2	<p>Identify the top 3 most significant attributes from “Cancer.arff” dataset using WEKA. Find which attributes were selected, and why they are important?</p>
	<p>Steps to Identify the Top 3 Most Significant Attributes Using WEKA</p> <p>1. Open WEKA:</p> <ul style="list-style-type: none"> Launch WEKA and choose the Explorer option. <p>2. Load the Dataset:</p> <ul style="list-style-type: none"> In the Preprocess tab, click Open file and load the Cancer.arff dataset (make sure the dataset contains the necessary attributes for classification, such as features related to cancer diagnosis). <p>3. Go to Attribute Selection:</p> <ul style="list-style-type: none"> Switch to the Select attributes tab at the top of the WEKA Explorer window. This tab allows you to apply attribute selection methods to identify the most significant features. <p>4. Choose Attribute Selection Method:</p>

- In the **Attribute Evaluator** section, click on the **Choose** button.
- Select an evaluator based on the method you want to use:
 - **InfoGainAttributeEval** (Information Gain)
 - **CfsSubsetEval** (Correlation-based Feature Selection)
 - **ChiSquaredAttributeEval** (Chi-squared test)

For this example, **CfsSubsetEval** is commonly used, as it evaluates subsets of features based on how well they correlate with the class label, taking feature interactions into account.

5. **Choose Search Method:**

- In the **Search Method** section, click on the **Choose** button and select a search method. You can use:
 - **Ranker** (if you want to rank features individually based on their importance)
 - **GeneticSearch** (if you want to use a genetic algorithm to search for optimal feature subsets)

For a simple ranking of features, select **Ranker** as the search method.

6. **Configure the Search Method:**

- After selecting **Ranker**, you can adjust the search settings if needed. By default, it ranks the attributes based on the score assigned by the evaluator you selected (e.g., **Information Gain** or **CFS**).

7. **Start Attribute Selection:**

- Click the **Start** button to perform the attribute selection.
- WEKA will compute the importance of each attribute and list them in order of significance.

8. **View Results:**

- After running the attribute selection, the output will show the ranked list of attributes.
- The **top 3 attributes** will be listed at the top, based on their importance score.
- WEKA will also provide details about the evaluator (e.g., **Information Gain** value or **CFS score**) and the attributes that were selected.

Interpreting the Results

• **Attributes Selection:**

- The **top 3 attributes** are those with the highest importance scores.
- These attributes are considered to be the most **relevant** and **informative** with respect to the target class (e.g., cancer diagnosis).

• **Why These Attributes are Important:**

- **Information Gain (InfoGain):** Measures how much knowing the value of an attribute reduces uncertainty about the class label. Higher values indicate more significant attributes.
- **Correlation-based Feature Selection (CFS):** Evaluates subsets of attributes that are highly correlated with the class but less correlated with each other. It selects attributes that together provide the most information.
- **Chi-Squared Test:** Measures how much an attribute differs from the expected distribution based on its association with the class.

This means that **Feature_1**, **Feature_3**, and **Feature_4** are the top 3 most significant attributes based on **Information Gain**.

- These attributes are likely to have a strong relationship with the class variable (e.g., cancer diagnosis).
- By using these features in your model, the classifier will have access to the most

	<p>informative data, potentially improving the model's performance.</p> <p>Conclusion:</p> <ul style="list-style-type: none"> • Attribute Selection in WEKA helps to identify the most relevant attributes for building predictive models. • The top 3 significant attributes are selected based on metrics such as Information Gain, CFS, or Chi-square, and they are important because they have the highest ability to distinguish between different classes in the dataset. • Using these significant attributes can improve the performance and interpretability of machine learning models.
Slip 10 Q.2	<p>Use the Apriori algorithm to find association rules on <i>weather.nominal.arff</i> dataset using WEKA and change minimum support value to 0.5. Interpret how change in support affect the number of rules generated</p>
	<p>Steps in WEKA</p> <ol style="list-style-type: none"> 1. Open WEKA: Start the WEKA GUI, then select the "Explorer" interface. 2. Load Dataset: <ul style="list-style-type: none"> ○ Click on the Open file... button. ○ Navigate to your <i>weather.nominal.arff</i> file, select it, and load it into WEKA. 3. Choose the Apriori Algorithm: <ul style="list-style-type: none"> ○ In the "Explorer" window, go to the Associations tab. ○ Select Choose and choose Apriori from the list of association algorithms. 4. Set Parameters: <ul style="list-style-type: none"> ○ In the Apriori settings, find the Minimum support parameter and set it to 0.5 (50%). ○ Adjust any other parameters as needed, but leave them at their default if you're just exploring the effect of minimum support. 5. Run the Algorithm: <ul style="list-style-type: none"> ○ Click Start to run Apriori on the dataset with the modified support threshold. ○ WEKA will display the association rules generated based on the chosen support level. 6. Analyze the Results: <ul style="list-style-type: none"> ○ After the algorithm finishes, review the generated rules in the output panel. ○ Note the number of rules generated. <p>Changing the Support Value and Observing Effects</p> <ol style="list-style-type: none"> 1. Change Minimum Support: Lower the minimum support value to a lower threshold, such as 0.2 or 0.1. 2. Run the Algorithm Again: Click Start to run Apriori again with the new support threshold. 3. Compare the Results: Notice how the number of rules changes as you adjust the support level. <p>Interpretation of Results</p> <ul style="list-style-type: none"> • Higher Minimum Support (0.5): When you set the minimum support to 0.5, only those rules that apply to at least 50% of the data instances are considered. This often results in fewer rules, as higher support restricts the algorithm to more frequent itemsets. • Lower Minimum Support (e.g., 0.2 or 0.1): When you reduce the minimum support, the algorithm considers less frequent itemsets, which usually leads to generating a larger number of association rules. Lower support values can reveal

	patterns that are less common in the data but might still be interesting.
Slip 11 Q.1	<p>Apply the Apriori algorithm on <i>supermarket.arff</i> dataset to discover frequent itemsets using WEKA. Interpret the potential usefulness of the rules for retail business.</p> <ul style="list-style-type: none"> ○ Steps: <ul style="list-style-type: none"> ▪ Open the <i>supermarket.arff</i> dataset. ▪ In the Associate tab, choose Apriori. ▪ Run the algorithm and analyze the top rules generated in the Associator output. <p>Steps to Apply the Apriori Algorithm in WEKA</p> <ol style="list-style-type: none"> 1. Open WEKA: <ul style="list-style-type: none"> ○ Launch WEKA and select the Explorer option. 2. Load the Dataset: <ul style="list-style-type: none"> ○ In the Preprocess tab, click Open file and load the supermarket.arff dataset. This dataset typically contains transactional data from a supermarket, with items bought together in different transactions. 3. Go to the Associate Tab: <ul style="list-style-type: none"> ○ After loading the dataset, click on the Associate tab at the top of the WEKA Explorer window. ○ This tab is used for association rule mining, including applying the Apriori algorithm. 4. Choose the Apriori Algorithm: <ul style="list-style-type: none"> ○ In the Associate tab, click on the Choose button under the "Associate" section. ○ From the list of algorithms, select assoc > Apriori. This will apply the Apriori algorithm to the dataset. 5. Set Parameters for the Apriori Algorithm: <ul style="list-style-type: none"> ○ You can configure several parameters for the Apriori algorithm: <ul style="list-style-type: none"> ▪ Confidence: The minimum confidence for the rules (typically between 0.5 and 1.0). A higher value ensures that the rule has a stronger predictive power. ▪ Support: The minimum support for the itemsets. Support measures the frequency of an itemset occurring in the dataset. A higher support means that the itemset appears more frequently in transactions. ▪ Lift: This can be adjusted to help refine the rules based on how much more likely items are to be bought together than by chance. ▪ Search Method: You can choose the search method (e.g., Best First Search) to identify frequent itemsets efficiently. 6. Run the Apriori Algorithm: <ul style="list-style-type: none"> ○ Once you've set the parameters, click the Start button to apply the Apriori algorithm. ○ WEKA will then generate the frequent itemsets and the corresponding association rules. 7. View the Output: <ul style="list-style-type: none"> ○ After the algorithm runs, the Result list will display the discovered frequent itemsets and association rules. ○ The output will include details such as: <ul style="list-style-type: none"> ▪ Support: How often the itemset appears in the dataset. ▪ Confidence: The likelihood of the rule being true when the antecedent is true.

- **Lift:** The increase in the likelihood of the consequent occurring when the antecedent occurs, compared to random chance.

Key Points to Interpret:

1. **Support:** This measures how frequently an itemset appears in the dataset.
 - For example, if {Bread} => {Butter} has a **support of 0.2**, it means that 20% of all transactions in the dataset contain both bread and butter. Higher support suggests that the items are frequently bought together.
2. **Confidence:** This measures how likely it is that the consequent item is bought when the antecedent item is bought.
 - For example, if {Bread} => {Butter} has **confidence of 0.8**, it means that 80% of the time when bread is bought, butter is also bought. A higher confidence suggests a stronger relationship between the items.
3. **Lift:** This measures the strength of the rule relative to the chance of the items being bought together by random chance.
 - For example, if {Bread} => {Butter} has a **lift of 1.2**, it means that bread and butter are bought together **1.2 times more often** than if the items were bought independently. A lift greater than 1 indicates a positive association, meaning the items are more likely to be bought together than by chance.

Example Rules and Their Usefulness for Retail Business

1. **Rule 1: {Bread} => {Butter} (Support: 0.2, Confidence: 0.8, Lift: 1.2)**
 - **Interpretation:** Customers who buy bread are likely to also buy butter. This rule has a **high confidence** (80%) and a **positive lift** (1.2), indicating that promoting bread alongside butter can lead to more sales.
 - **Usefulness:** Retailers can place bread and butter near each other on the shelves or offer promotions like "Buy bread and get butter at a discount" to increase sales.
2. **Rule 2: {Milk, Bread} => {Butter} (Support: 0.15, Confidence: 0.75, Lift: 1.3)**
 - **Interpretation:** If a customer buys both milk and bread, there is a 75% chance they will also purchase butter. The **lift value of 1.3** suggests a strong relationship between milk, bread, and butter.
 - **Usefulness:** Retailers can create bundled promotions for milk, bread, and butter, or recommend butter to customers buying milk and bread, increasing the likelihood of cross-selling.
3. **Rule 3: {Eggs} => {Milk} (Support: 0.1, Confidence: 0.7, Lift: 1.1)**
 - **Interpretation:** Customers buying eggs often buy milk. The **support** of 0.1 indicates that 10% of the transactions contain both items, and a **confidence** of 70% suggests a good likelihood of customers purchasing both.
 - **Usefulness:** Retailers can use this information to target promotions for eggs and milk together, such as discounts or recipes that use both items.

Business Implications:

- **Cross-Selling and Up-Selling:** The association rules help in identifying which items are frequently purchased together, enabling the business to create effective cross-selling and up-selling strategies. For example, if butter is frequently bought with bread, the store could promote butter when customers pick up bread.
- **Shelf Space Optimization:** Retailers can use the rules to optimize store layout by placing frequently bought together items near each other, making it easier for customers to find them and potentially increasing sales.
- **Targeted Promotions:** By identifying items that are often bought together,

	<p>supermarkets can offer discounts or bundle deals that encourage customers to buy more products. For example, a bundle deal for milk, bread, and butter could increase overall sales for these items.</p> <p>Conclusion: The Apriori algorithm is a powerful tool for discovering relationships in transactional data, and the association rules derived from it are extremely useful for retail businesses. By understanding which items are frequently bought together, businesses can optimize product placement, offer targeted promotions, and enhance customer experience, all of which can lead to increased sales and customer satisfaction.</p>
Slip 11 Q.2	<p>Apply the Naive Bayes classifier on <i>cancer.arff</i> dataset and generate the confusion Matrix using WEKA. Interpret the precision and recall.</p>
	<p>Steps to Apply the Naive Bayes Classifier and Generate the Confusion Matrix in WEKA</p> <ol style="list-style-type: none"> Open WEKA: <ul style="list-style-type: none"> Launch WEKA and choose the Explorer option. Load the Dataset: <ul style="list-style-type: none"> In the Preprocess tab, click Open file and load the cancer.arff dataset. This dataset typically contains attributes related to cancer diagnosis (e.g., tumor size, shape, etc.) with a class attribute indicating whether the tumor is benign or malignant. Select Naive Bayes Classifier: <ul style="list-style-type: none"> Go to the Classify tab. Click Choose under the "Classifier" section. From the list of classifiers, select bayes > NaiveBayes to apply the Naive Bayes classifier. Set Parameters (Optional): <ul style="list-style-type: none"> You can adjust the parameters of Naive Bayes if needed (e.g., setting the distribution option for continuous attributes). For most cases, the default settings should work fine. Train the Model: <ul style="list-style-type: none"> Click the Start button to train the Naive Bayes model using the dataset. WEKA will output the results, including the accuracy, precision, recall, and confusion matrix. Generate the Confusion Matrix: <ul style="list-style-type: none"> After the model is trained, the Result list section will display the output. To view the confusion matrix: <ul style="list-style-type: none"> Scroll down to the output and locate the Confusion Matrix. The confusion matrix will show the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) values. <p>Interpretation of Precision and Recall:</p> <ul style="list-style-type: none"> Precision tells us how many of the instances predicted as a particular class are actually correct. A high precision means that when the model predicts an instance as Malignant or Benign, it is likely to be correct. Recall tells us how many of the actual instances of a particular class have been correctly identified. A high recall means the model is good at identifying all instances of that class. <p>For a retail or medical application like cancer diagnosis:</p> <ul style="list-style-type: none"> High precision for Malignant means that when the system flags a tumor as malignant, it is more likely to be correctly identified, which is crucial for minimizing false alarms.

	<ul style="list-style-type: none"> High recall for Malignant ensures that most of the actual malignant cases are detected, which is critical in a medical diagnosis context to avoid missing any malignant tumors (false negatives). <p>Conclusion: By using the Naive Bayes classifier in WEKA, we can easily calculate and interpret the confusion matrix, precision, and recall to evaluate the performance of a classifier. In this context, high precision and recall for detecting malignant tumors are desirable because they help minimize false positives (misclassifying benign tumors as malignant) and false negatives (missing malignant tumors).</p>
Slip 11 Q.2	<p>Apply the J48 classifier and generate the ROC curve on <i>weather.nominal.arff</i> dataset using WEKA. Analyze ROC curve indicate about the model's performance in terms of sensitivity and specificity?</p>
	<p>Steps to Apply the J48 Classifier and Generate the ROC Curve in WEKA</p> <ol style="list-style-type: none"> Open WEKA: <ul style="list-style-type: none"> Launch WEKA and select the Explorer option. Load the Dataset: <ul style="list-style-type: none"> In the Preprocess tab, click Open file and load the <i>weather.nominal.arff</i> dataset. Select J48 Classifier: <ul style="list-style-type: none"> Go to the Classify tab. In the Classifier section, click Choose and select trees > J48 (J48 is WEKA's implementation of the C4.5 decision tree algorithm). Set Parameters (Optional): <ul style="list-style-type: none"> You can modify the J48 parameters by clicking the box next to the Choose button. However, the default settings should be sufficient for basic classification. Generate the ROC Curve: <ul style="list-style-type: none"> To generate the ROC curve, select the Cross-validation option (or Percentage Split for testing on a holdout set). Then, click Start to train the model and perform evaluation. Enable ROC Curve Output: <ul style="list-style-type: none"> After training, in the Result list, you will see the results of the classification, including the ROC curve. If you cannot see the ROC curve directly, you may need to use the Visualize threshold curve option, which displays the ROC curve. <ul style="list-style-type: none"> Click on the Visualize threshold curve button to view the ROC curve. <p>From this output:</p> <ul style="list-style-type: none"> Sensitivity (True Positive Rate) = 0.86 (86% of "Yes" cases are correctly identified). Specificity = $1 - 0.1 = 0.9$ (90% of "No" cases are correctly identified). <p>This indicates that the model is doing a good job of identifying both positive and negative instances, with high sensitivity and specificity.</p> <p>Conclusion:</p> <ul style="list-style-type: none"> ROC curve analysis helps you assess the trade-off between sensitivity (recall) and specificity in classification tasks. Sensitivity measures the model's ability to detect positive instances, while specificity measures its ability to detect negative instances. A good classifier will produce a ROC curve that is as close to the top-left corner as possible, indicating high sensitivity and specificity.

	<ul style="list-style-type: none">The AUC value can also help quantify the model’s performance, with a higher AUC indicating better overall classification performance.												
Slip 12 Q.1	Using WEKA compare the accuracy and ROC area of the J48, Naive Bayes, and Random Forest classifiers when applied to the <i>iris.arff</i> dataset. Analyze which classifier performs best												
	<ul style="list-style-type: none">Steps:<ul style="list-style-type: none">Open the <i>iris.arff</i> dataset.Run the J48, NaiveBayes, and RandomForest classifiers from the Classify tab.Record and compare their accuracy and ROC area from the Classifier output. <p>Example Analysis: Assume the output from WEKA shows the following:</p> <table><tr><th>Classifier</th><th>Accuracy (%)</th><th>ROC Area (AUC)</th></tr><tr><td>J48</td><td>96.00</td><td>0.97</td></tr><tr><td>Naive Bayes</td><td>94.00</td><td>0.94</td></tr><tr><td>Random Forest</td><td>98.00</td><td>0.99</td></tr></table> <p>Interpretation:</p> <ul style="list-style-type: none">Random Forest has the highest accuracy (98%) and AUC (0.99), indicating that it is the most effective classifier for this dataset.J48 is slightly lower in accuracy (96%) but still has a high AUC (0.97), which shows that it performs well but may be more prone to overfitting compared to Random Forest.Naive Bayes has the lowest accuracy (94%) and AUC (0.94), suggesting that it is the least effective classifier in this case.	Classifier	Accuracy (%)	ROC Area (AUC)	J48	96.00	0.97	Naive Bayes	94.00	0.94	Random Forest	98.00	0.99
Classifier	Accuracy (%)	ROC Area (AUC)											
J48	96.00	0.97											
Naive Bayes	94.00	0.94											
Random Forest	98.00	0.99											
Slip 12 Q.2	Create a scatter plot for the <i>iris.arff</i> dataset using WEKA’s visualization tool. Analyze patterns observed in the plot, and the classes distributed												
	<ul style="list-style-type: none">Steps:<ul style="list-style-type: none">Open the <i>iris.arff</i> dataset.Go to the Visualize tab.Create a scatter plot by selecting two attributes for the x-axis and y-axis.Analyze and interpret the visualization. <p>Steps to Create a Scatter Plot in WEKA:</p> <ol style="list-style-type: none">Open WEKA:<ul style="list-style-type: none">Launch WEKA and select the Explorer option.Load the Dataset:<ul style="list-style-type: none">In the Preprocess tab, click Open file and load the <i>iris.arff</i> dataset. This dataset contains four numerical attributes (sepal length, sepal width, petal length, and petal width) and the class attribute (species: Setosa, Versicolor, Virginica).Select Attributes for Plotting:<ul style="list-style-type: none">In the Preprocess tab, you can view the attributes of the dataset. To create a scatter plot, you need to select which two attributes to plot against each other.For example, select petal length (1st attribute) and petal width (2nd attribute) as the X and Y axes for the scatter plot.Open the Visualization Tool:												

	<ul style="list-style-type: none"> ○ In the Preprocess tab, click the Visualize button located at the top of the attribute list. ○ A window will appear showing a 2D scatter plot with two axes. By default, it may show the first two attributes. <p>5. Customize the Scatter Plot:</p> <ul style="list-style-type: none"> ○ If you want to compare specific attributes, select them from the dropdown menus for X-axis and Y-axis. For example, choose petal length for the X-axis and petal width for the Y-axis. <p>6. Coloring by Class:</p> <ul style="list-style-type: none"> ○ In the scatter plot window, you can color the data points according to their class (species). This will help differentiate between the three classes (Setosa, Versicolor, Virginica). ○ Look for the class attribute in the scatter plot options, and make sure that the points are colored by their class values (species). <p>7. View and Analyze the Plot:</p> <ul style="list-style-type: none"> ○ The scatter plot will show a distribution of the instances with the selected attributes. The data points will be grouped based on their species (Setosa, Versicolor, Virginica) and represented by different colors or shapes. ○ The Visualize tool will allow you to zoom in or out, hover over points to see their details, and observe how the different species are separated in the plot. <p>Example of the Scatter Plot Analysis:</p> <p>In a scatter plot of petal length vs. petal width for the iris.arff dataset, the points might be distributed as follows:</p> <ul style="list-style-type: none"> ● Setosa: Data points concentrated in the bottom-left corner, with small petal lengths and widths. ● Versicolor: Data points spread out with moderate petal lengths and widths, overlapping with Virginica. ● Virginica: Data points spread out in the top-right corner, with larger petal lengths and widths. <p>Conclusion:</p> <ul style="list-style-type: none"> ● Setosa is typically the easiest to distinguish from the other two species due to its distinct cluster in the scatter plot. ● Versicolor and Virginica show some overlap and are harder to separate based solely on these two attributes. ● The scatter plot helps in visualizing how well the attributes separate the classes and provides insights into the features that could be useful for classification algorithms. ● You may need to use more advanced classification techniques or additional attributes to distinguish Versicolor and Virginica more effectively, as they tend to overlap in simple scatter plots.
Slip 12 Q.2	<p>Train a model using the J48 algorithm on the <i>cancer.arff</i> dataset using WEKA, save the model to disk. Analyze how this model can be reloaded in WEKA for future predictions.</p>
	<ul style="list-style-type: none"> ○ Steps: <ul style="list-style-type: none"> ▪ Open the <i>breast-cancer.arff</i> dataset. ▪ In the Classify tab, select Classifier → trees → J48. ▪ After running the classifier, click Save model... to save the model to disk. ▪ To reload the model, go to Classify tab and click Load model...

Train the Model:

- **Choose classifier:** trees > J48
- **Set options:** 10-fold cross-validation or train-test split.
- **Start** training to generate the model.

Save the Model:

- Right-click on the model in the **Result list** > **Save model**.
- Save as cancer_J48_model.model file.

Reload the Model for Future Predictions:

- **Open model** > Browse to cancer_J48_model.model > **Load**.
- **Apply to test data** by selecting a test dataset and clicking **Start**.

Benefits of Reloading the Model:

- **Consistency:** By saving and reloading the model, you can ensure that future predictions are made consistently using the same trained model.
- **Time Efficiency:** Reloading models eliminates the need to retrain them every time you want to make predictions, saving processing time and computational resources.
- **Future Predictions:** The model can be used in the future for predictions on new data (e.g., predicting cancer diagnosis for new test instances).

Conclusion:

By following the steps above, you can train a **J48** classifier on the **cancer.arff** dataset in WEKA, save the model, and reload it for future predictions. This allows for easy reusability of the model without needing to retrain it each time you want to apply it to new data.