Data Mining Experiments – Steps & Viva Scripts

Experiment 1 – Distance Matrix

Steps:

• Create ARFF file with coordinates (distance_example.arff).

• Load file in WEKA ’n Preprocess.

• Go to Visualize tab to inspect points.

• Manually compute Euclidean and Manhattan distances for all pairs.

• Interpret distance matrix values and symmetry.

Viva Script:

I computed distances between all objects using Euclidean and Manhattan formulas. Smaller distances indicate higher similarity. The diagonal is zero because distance(object, object) = 0. Then I compared results with WEKA visualization.

Experiment 2 – K-Means

Steps:

• Create CSV file with numeric attributes (kmeans.csv).

• Open WEKA ’n Preprocess and load CSV.

• Go to Cluster ’n SimpleKMeans.

• Click the options field next to SimpleKMeans and set numClusters = 2.

• Click Start to generate clusters and view centroids.

Viva Script:

I selected k=2, initialized centroids (farthest points), assigned points to nearest centroid using Euclidean distance, recomputed centroids and iterated until stable. WEKA provided cluster assignments and centroids which matched the manual solution up to initialization differences.

Experiment 3 – Preprocessing

Steps:

• Load preprocess.arff in WEKA.

• Select 'Select Attributes' tab.

• Choose CfsSubsetEval (Attribute Evaluator) and BestFirst (Search Method).

• Click Start to obtain selected attributes.

• Apply ReplaceMissingValues filter to handle missing entries.

Viva Script:

I performed attribute selection to remove irrelevant features and replaced missing numeric values with mean and categorical with mode using WEKA's ReplaceMissingValues filter. This makes the data ready for modeling.

Experiment 4 – Naive Bayes (Small)

Steps:

• Load naive_small.arff in WEKA.

• Open Classify tab.

• Choose NaiveBayes classifier.

• Ensure the class attribute is the last attribute (Buys).

• Set Test options to 'Use training set' for small datasets and click Start.

Viva Script:

I calculated prior probabilities and conditional likelihoods manually for the test instance. Multiplying likelihoods with priors gave posterior probabilities. The class with the higher posterior was chosen and confirmed using WEKA.

Experiment 5 – Decision Tree

Steps:

• Load decision_small.arff in WEKA.

• Go to Classify tab and choose J48.

• Click Start to build the tree.

• Right-click the result ′n Visualize Tree to see structure.

Viva Script:

I computed the entropy of the dataset and information gain for attributes. The attribute with highest gain (Outlook) became the root. The splits matched the J48 output from WEKA.

Experiment 6 – Apriori

Steps:

• Load apriori_small.arff in WEKA.

• Go to Associate tab.

• Choose Apriori algorithm.

• Set upperBoundMinSupport = 0.4 (this is an upper bound parameter in WEKA) and set minMetric (confidence) = 0.6.

• Click Start to generate association rules.

Viva Script:

I computed supports of itemsets and derived rules by calculating confidence. Strong rules like Bread’n Butter and Milk’n Bread were confirmed in WEKA. Note: in WEKA, Apriori uses 'upperBoundMinSupport' parameter which controls the highest support allowed for candidate generation; set it to 0.4 as required.

Experiment 7 – Hierarchical Clustering

Steps:

• Load hierarchical_small.arff in WEKA.

• Select Cluster tab ’n HierarchicalClusterer.

• Open options ’n set numClusters = 1 to view full dendrogram and set LinkType = SINGLE.

• Click Start and visualize the dendrogram.

Viva Script:

I manually computed pairwise Euclidean distances and merged the closest clusters iteratively (single-link). The dendrogram showed (A,B) merging first, (C,D) next, then final merge, matching WEKA output.

Experiment 8 – FP-Growth

Steps:

• Load fpgrowth_small.arff in WEKA.

• Go to Associate tab.

• Choose FPGrowth algorithm.

• Set lowerBoundMinSupport = 0.4 (this acts as the minimum support threshold in WEKA for FP-Growth).

• Click Start to mine frequent itemsets and rules.

Viva Script:

I explained FP-tree construction and conditional trees conceptually and showed that FP-Growth produces the same frequent itemsets and rules as Apriori but without candidate generation. WEKA's FPGrowth produced the expected rules.