Perfect 👍 — let’s go **step by step in detail** for each of the 11 questions from your image using the **Weka KnowledgeFlow interface** (the one shown in your screenshot).

This version assumes you already have the .arff file of your employee dataset ready (the one with attributes age, income, student, creditrate, buyscomp).

**🧠 Detailed Step-by-Step Solution Using Weka KnowledgeFlow**

**Step 1: Load ARFF file and explore**

1. Open **Weka → KnowledgeFlow**.
2. From the left sidebar, expand **DataSources**.
3. Drag **ArffLoader** into the workspace.
4. Right-click the **ArffLoader** box → **Configure**.
5. Click **Browse**, select your file employee.arff → click **Open** → **OK**.
6. To check data visually:
   * From **Visualization**, drag **DataVisualizer** into the workspace.
   * Right-click **ArffLoader → Connect → dataSet → DataVisualizer**.
7. Click the **Play (Run)** button on the toolbar (top left ▶️).
8. Look at the bottom panel (**Status** tab): it should say *Data loaded successfully*.

**Step 2: Check the status area**

* After clicking **Run**, check the **Status** panel below.
* You will see messages like:
* Component: ArffLoader
* Status: Data loaded successfully
* This confirms your dataset is connected and flowing properly.

**Step 3: Attribute selection (Feature selection)**

1. From **AttrSelection** in the left panel, drag **AttributeSelection** to workspace.
2. Right-click it → **Configure**.
3. In the configuration window:
   * Evaluator: choose **InfoGainAttributeEval** (measures information gain of each attribute).
   * Search method: choose **Ranker**.
4. Connect components:
   * **ArffLoader → AttributeSelection** (right-click → Connect → dataSet).
   * **AttributeSelection → TextViewer** (add from **DataSinks** → TextViewer).
5. Click **Run**.
6. In the **TextViewer**, you’ll see ranked attributes (like student highest).

**Step 4: Classification — J48 Decision Tree**

1. From **Classifiers**, drag a **Classifier** into workspace.
2. Right-click it → **Configure** → choose **J48** from the list → click **OK**.
3. Add a **ClassAssigner** (from **Evaluation** section).
   * Connect **ArffLoader → ClassAssigner**.
   * Right-click **ClassAssigner → Configure**, set class attribute = buyscomp.
4. Connect the rest:
   * **ClassAssigner → Classifier**.
   * **Classifier → TextViewer** (to see output).
5. Right-click **Classifier → Configure**:
   * Choose **Test options** → select **Cross-validation (10 folds)** or **Percentage split (70%)**.
6. Click **Run ▶️**.
7. Open **TextViewer** to see:
   * Accuracy percentage
   * Confusion matrix
   * Decision tree rules

✅ Example Output (you’ll see similar):

Correctly Classified Instances: 13 (92.8571 %)

Incorrectly Classified Instances: 1 (7.1429 %)

=== Confusion Matrix ===

a = yes, b = no

a b

8 1

0 5

**Step 5: Classification — Naive Bayes**

1. Copy your flow from Step 4 (or add new).
2. Right-click **Classifier → Configure → NaiveBayes**.
3. Keep same connections.
4. Click **Run**.
5. In **TextViewer**, note:
   * Accuracy
   * Precision
   * Recall
   * Confusion matrix

**Step 6: Classification — k-Nearest Neighbor (IBk)**

1. From **Classifiers**, drag another **Classifier**.
2. Right-click → **Configure → Choose → IBk**.
3. In IBk settings, set **k = 3**.
4. Use same class assigner connection:
   * **ArffLoader → ClassAssigner → Classifier (IBk)**.
   * **Classifier → TextViewer**.
5. Click **Run** → View accuracy results.

**Step 7: Generate Decision Matrix**

1. After you run classifiers, open **TextViewer** for each classifier.
2. Note down:
   * **True Positives (TP)**
   * **False Positives (FP)**
   * **True Negatives (TN)**
   * **False Negatives (FN)**
3. Compute manually:
   * **Accuracy = (TP + TN) / Total**
   * **Precision = TP / (TP + FP)**
   * **Recall = TP / (TP + FN)**
   * **F1-score = 2 × (Precision × Recall) / (Precision + Recall)**

**Step 8: Linear Regression**

1. From **Classifiers**, drag a **Classifier** → right-click → **Configure → LinearRegression**.
2. Connect:
   * **ArffLoader → ClassAssigner → Classifier (LinearRegression)**.
   * **Classifier → TextViewer**.
3. Click **Run ▶️**.
4. See the coefficients and R² value (works best if the target variable is numeric).

**Step 9: Association Rule (Apriori Algorithm)**

1. From **Associations**, drag **Associator** → right-click → **Configure → Choose → Apriori**.
2. Set:
   * minSupport = 0.1
   * minConfidence = 0.7
3. Connect:
   * **ArffLoader → Associator**.
   * **Associator → TextViewer**.
4. Click **Run ▶️**.
5. See rules like:
6. student=yes ==> buyscomp=yes (conf:1.0)
7. age=31-40 & income=low ==> buyscomp=yes (conf:0.8)

**Step 10: Clustering (K-Means)**

1. From **Clusterers**, drag **Clusterer** → right-click → **Configure → SimpleKMeans**.
2. Set **numClusters = 2**.
3. Connect:
   * **ArffLoader → Clusterer**.
   * **Clusterer → TextViewer**.
4. Run ▶️.
5. You’ll see output like:
6. Cluster 0: mostly buyscomp=yes
7. Cluster 1: mostly buyscomp=no

**Step 11: Test 70% Training / 30% Testing Split**

1. Right-click your **Classifier (e.g., J48)** → **Configure**.
2. Under **Test options**, choose **Percentage Split = 70**.
3. Run ▶️.
4. Observe:
   * Training: 70%
   * Testing: 30%
   * Output: new accuracy and confusion matrix.

**(Optional) Add Noise Test**

1. Open your .arff file in Notepad.
2. Flip 1–2 class values (yes → no).
3. Save file, reload in KnowledgeFlow.
4. Rerun J48 — observe slight accuracy drop.

✅ **Final Notes**

* Always connect outputs to **TextViewer** to see results.
* Save your workspace using **File → Save Flow** if you want to reuse later.

Would you like me to make this version into a **ready-to-print formatted document (Word/PDF)** with bullet-style Weka steps only (no explanations)?