

Answer to the question – 01(A)

Applying the Apriori algorithm to the transaction data to find all frequent itemsets with a minimum relative support threshold of 50%.

To find the frequent itemsets, we first calculate the support for each item and then the combinations of items.

Minimum support threshold = 50% → 0.50 \* 5 (total transactions) = 2.5 ~ 3 transactions.

Step 1: Count individual items:

* Book A: 4/5 = 80% (frequent)
* Book B: 3/5 = 60% (frequent)
* Book C: 3/5 = 60% (frequent)
* Book D: 4/5 = 80% (frequent)
* Book E: 3/5 = 60% (frequent)

Step 2: Generate 2-itemsets and calculate support:

* {Book A, Book B}: 2/5 = 40% (not frequent)
* {Book A, Book C}: 2/5 = 40% (not frequent)
* {Book A, Book D}: 3/5 = 60% (frequent)
* {Book A, Book E}: 2/5 = 40% (not frequent)
* {Book B, Book C}: 2/5 = 40% (not frequent)
* {Book B, Book D}: 2/5 = 40% (not frequent)
* {Book B, Book E}: 1/5 = 20% (not frequent)
* {Book C, Book D}: 2/5 = 40% (not frequent)
* {Book C, Book E}: 2/5 = 40% (not frequent)
* {Book D, Book E}: 3/5 = 60% (frequent)

Step 3: Generate 3-itemsets and calculate support:

* {Book A, Book B, Book C}: 1/5 = 20% (not frequent).
* {Book A, Book B, Book D}: 1/5 = 20% (not frequent).
* {Book A, Book B, Book E}: 0/5 = 0% (not frequent).
* {Book A, Book C, Book D}: 1/5 = 20% (not frequent).
* {Book A, Book C, Book E}: 1/5 = 20% (not frequent).
* {Book A, Book D, Book E}: 2/5 = 40% (not frequent).
* {Book B, Book C, Book D}: 1/5 = 20% (not frequent).
* {Book B, Book C, Book E}: 1/5 = 20% (not frequent).
* {Book B, Book D, Book E}: 1/5 = 20% (not frequent).
* {Book C, Book D, Book E}: 2/5 = 40% (not frequent).

Answer to the question – 01(B)

Calculating the confidence for the following association rules derived from the frequent itemsets:

1. Confidence for {Book A} → {Book B} = Support({Book A, Book B}) / Support({Book A}) = 2/4 = 50%.
2. Confidence for {Book A} → {Book D} = Support({Book A, Book D}) / Support({Book A}) = 3/4 = 75%.

Answer to the question – 01(C)

Support Calculation for {Book A, Book C, Book D}:

* Transactions containing {Book A, Book C, Book D}: Only Transaction 5 contains all three books together.
* Support of {Book A, Book C, Book D}: 1/5 = 20%

1. Closed Pattern: To be considered closed, the itemset must first be frequent, and then none of its supersets should have the same support. However, since {Book A, Book C, Book D} is not frequent (support of only 20%), it cannot be considered a closed pattern.
2. Max Pattern: Similarly, to be considered maximal, the itemset must be frequent, and none of its supersets should be frequent. Again, because {Book A, Book C, Book D} is not frequent, it cannot be considered a maximal pattern either.

Answer to the question – 02(A)

Step 1:

Calculate Gini Impurity for the full dataset

There are 5 records:

* 2 "Yes" (Loan Approved)
* 3 "No" (Loan Not Approved)

Gini impurity for the dataset is calculated as follows:

Gini\_total = 1 - (P(Yes)^2 + P(No)^2)

Gini\_total = 1 - ((2/5)^2 + (3/5)^2)

Gini\_total = 1 - (0.16 + 0.36) = 1 - 0.52 = 0.48

Step 2:

Calculate Gini Impurity for each feature

We will calculate the Gini impurity for each feature and determine the best split.

1. Split by Credit Score:

* High (2 records): 1 Yes, 1 No
* Gini\_High = 1 - ((1/2)^2 + (1/2)^2) = 1 - 0.5 = 0.5
* Medium (2 records): 1 Yes, 1 No
* Gini\_Medium = 1 – ((1/2)^2 + (1/2)^2) = 1 - 0.5 = 0.5
* Low (1 record): 0 Yes, 1 No
* Gini\_Low = 1 – ((0/1)^2 + (1/1)^2) = 1 - 1 = 0
* Weighted Gini for Credit Score:
* Gini\_CreditScore = (2/5)\*0.5 + (2/5)\*0.5 + (1/5)\*0 = 0.4

2. Split by Annual Income:

* High (2 records): 1 Yes, 1 No
* Gini\_High = 1 – ((1/2)^2 + (1/2)^2) = 1 - 0.5 = 0.5
* Medium (1 records): 1 Yes, 0 No
* Gini\_Medium = 1 – ((1/1)^2 + (0/1)^2) = 1 - 1 = 0
* Low (2 record): 0 Yes, 2 No
* Gini\_Low = 1 – ((0/2)^2 + (2/2)^2) = 1 - 1 = 0
* Weighted Gini for Annual Income:
* Gini\_AnnualIncome = (2/5)\*0.5 + (2/5)\*0 + (1/5)\*0 = 0.2

3. Split by Employment Status:

* Employed (3 records): 2 Yes, 1 No
* Gini\_Employed = 1 – ((2/3)^2 + (1/3)^2) = 1 - 0.56 = 0.44
* Unemployed (2 records): 0 Yes, 2 No
* Gini\_Unemployed = 1 – ((0/2)^2 + (2/2)^2) = 0
* Weighted Gini for Employment Status:
* Gini\_EmploymentStatus = (3/5)\*0.44 + (2/5)\*0 = 0.264

4. Split by Existing Debt:

* Low (1 record): 1 Yes, 0 No
* Gini\_Low = 0
* Medium (2 records): 1 Yes, 1 No
* Gini\_Medium = 0.5
* High (2 records): 0 Yes, 2 No
* Gini\_High = 0
* Weighted Gini for Existing Debt:
* Gini\_ExistingDebt = (1/5)\*0 + (2/5)\*0.5 + (2/5)\*0 = 0.2

Step 3:

Choose the best feature for the first split

Here are the Gini values:

* Credit Score Gini: 0.4
* Annual Income Gini: 0.2
* Employment Status Gini: 0.264
* Existing Debt Gini: 0.2

Step 4:

“Annual Income” and “Existing Debt” both have the same Gini impurity (0.2). Further split on “Annual Income”.

We split the data into three groups based on “Annual Income”:

- High Income: 2 records (1 Yes, 1 No)

- Medium Income: 1 record (Yes)

- Low Income: 2 records (No)

Gini Impurity for Each Group:

* High Income: 1 Yes, 1 No
* Gini\_High = 1 – ((1/2)^2 +(1/2)^2) = 0.5
* Medium Income: 1 Yes, 0 No
* Gini\_Medium = 1 – ((1/1)^2 + (0/1))^2 = 0
* Low Income: 0 Yes, 2 No
* Gini\_Low = 1 – ((0/2)^2 + (2/2)^2) = 0
* Weighted Gini Impurity for Annual Income:
* Gini\_Annual Income = (2/5)\*0.5 + (1/5)\*0 + (2/5)\*0 = 0.2

Further Split the “High Income” Group

The “High Income” group is impure, so we need to split it further. The group looks like this:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Application ID | Credit Score | Annual Income | Employment Status | Existing Debt | Loan Approval |
| 1 | High | High | Employed | Low | Yes |
| 5 | Medium | High | Unemployed | Medium | No |

We can split this based on another feature, such as “Employment Status”.

Split Based on “Employment Status”:

* Employed: 1 record (Yes)
* Unemployed: 1 record (No)

Both groups are pure, so this split is effective.

* Gini\_Employed = 1− ((1/1​)^2 + (1/0​)^2) = 0
* Gini\_Unemployed = 1− ((0/1​)^2 + (1/1​)^2) = 0

Step 5: Final Decision Tree

Here’s how the final decision tree looks with “Annual Income” as the root split:

Annual Income

/ | \

/ | \

High Medium Low

/ \ (Yes) (No)

/ \

Employed Unemployed

(Yes) (No)

* If “Annual Income” is “Medium”, the loan is approved (Yes).
* If “Annual Income” is “Low”, the loan is not approved (No).
* If “Annual Income” is “High”, we further split by “Employment Status”:
  + If “Employed”, the loan is approved (Yes).
  + If “Unemployed”, the loan is not approved (No).

Answer to the question – 02(B)

Confusion Matrix

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | 3 | 2 |
| Actual Negative | 1 | 4 |

* Accuracy = (TP + TN) / Total = (3 + 4) / 10 = 0.70 or 70%.
* Precision = TP / (TP + FP) = 3 / (3 + 1) = 0.75 or 75%.
* Recall = TP / (TP + FN) = 3 / (3 + 2) = 0.60 or 60%.
* F1-Score = 2 \* (Precision \* Recall) / (Precision + Recall) = 2 \* (0.75 \* 0.60) / (0.75 + 0.60) = 0.67 or 67%.
* Accuracy: 70%
* Precision: 75%
* Recall: 60%
* F1-Score: 67%