

Candidate 1-itemlist, C1

|  |  |  |
| --- | --- | --- |
| **Itemset** | **Count** | **Support** |
| Meat | 5 | 0.5 |
| Potato | 5 | 0.5 |
| Onion | 4 | 0.4 |
| Noodle | 4 | 0.4 |
| Spinach | 3 | 0.3 |
| Eggs | 2 | 0.2 |
| Salt | 2 | 0.2 |

1-itemset, L1

|  |  |  |
| --- | --- | --- |
| **Itemset** | **Count** | **Support** |
| Meat | 5 | 0.5 |
| Potato | 5 | 0.5 |
| Onion | 4 | 0.4 |
| Noodle | 4 | 0.4 |
| Spinach | 3 | 0.3 |

Candidate 2-itemlist, C2

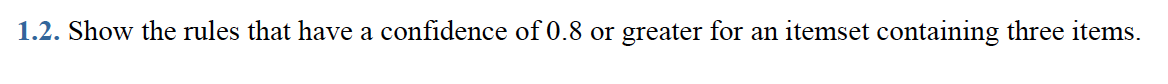
|  |  |  |
| --- | --- | --- |
| **Itemset** | **Count** | **Support** |
| Meat, Potato | 4 | 0.4 |
| Meat, Onion | 3 | 0.3 |
| Meat, Noodle | 1 | 0.1 |
| Meat, Spinach | 0 | 0.0 |
| Potato, Onion | 3 | 0.3 |
| Potato, Noodle | 1 | 0.1 |
| Potato, Spinach | 0 | 0.0 |
| Onion, Noodle | 1 | 0.1 |
| Onion, Spinach | 0 | 0.0 |
| Noodle, Spinach | 1 | 0.1 |

2-itemset, L2

|  |  |  |
| --- | --- | --- |
| **Itemset** | **Count** | **Support** |
| Meat, Potato | 4 | 0.4 |
| Meat, Onion | 3 | 0.3 |
| Potato, Onion | 4 | 0.4 |

Candidate 3-itemlist, C3 and 3-itemset, L3

|  |  |  |
| --- | --- | --- |
| **Itemset** | **Count** | **Support** |
| Meat, Potato, Onion | 3 | 0.3 |



1. Confidence {Meat, Onion} 🡪 {Potato}:

support (LHS U RHS) = 0.3

support (LHS) = 0.3

confidence = 1

1. Confidence {Potato, Onion} 🡪 {Meat}:

support (LHS U RHS) = 0.3

support (LHS) = 0.4

confidence = 0.75

1. Confidence {Meat, Potato} 🡪 {Onion}:

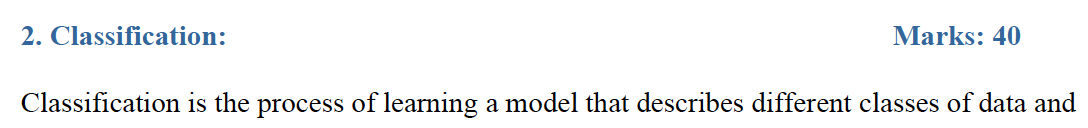
support (LHS U RHS) = 0.3

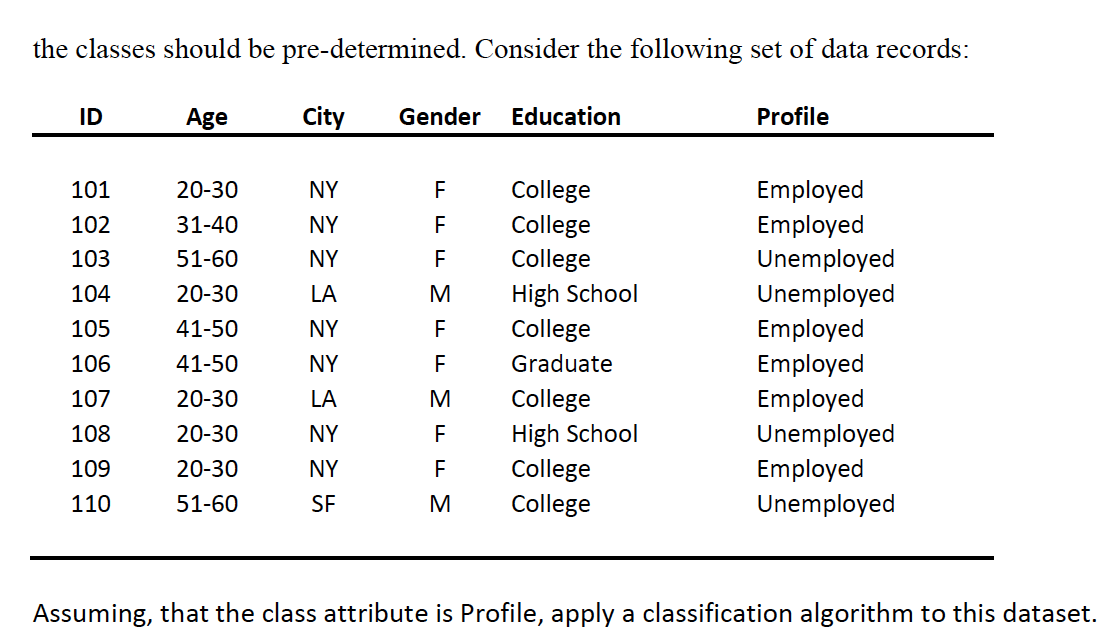
support (LHS) = 0.4

confidence = 0.75

Therefore, the rules that have a confidence of 0.8 or greater for an itemset containing three items are:

1. Confidence {Meat, Onion} 🡪 {Potato}





**Step 1: Determine the Decision Column**

Frequency Table for the class attribute:

|  |  |
| --- | --- |
| **Profile (10)** | |
| Employed | Unemployed |
| 6 | 4 |

**Step 2: Calculating Entropy for the classes (Profile)**

Entropy(Profile) = Entropy(6,4)

=

= 0.97095

**Step 3: Calculate Entropy for Other Attributes After Split**

Entropy(S,T) = E(S,T)

* E(Profile, Age)
* E(Profile, City)
* E(Profile, Gender)
* E(Profile, Education)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Profile (10) | |  |
| Employed | Unemployed | Total |
| Age | 20-30 | 3 | 2 | 5 |
| 31-40 | 1 | 0 | 1 |
| 41-50 | 2 | 0 | 2 |
| 51-60 | 0 | 2 | 2 |

E(Profile, Age) = P(20-30)E(20-30) + P(31-40)E(31-40) + P(41-50)E(41-50) +P(51-60)E(51-60)

= 0.5E(3,2) + 0.1E(1,0) + 0.2E(2,0) + 0.2(0,2)

= 0.5(0.97095) + 0 + 0 + 0

= 0.485475

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Profile (10) | |  |
| Employed | Unemployed | Total |
| City | NY | 5 | 2 | 7 |
| LA | 1 | 1 | 2 |
| SF | 0 | 1 | 1 |

E(Profile, City) = P(NY)E(NY) + P(LA)E(LA) + P(SF)E(SF)

= 0.7E(5,2) + 0.2E(1,1) + 0.1E(0,1)

= 0.7(0.86312) + 0.2 + 0

= 0.804184

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Profile (10) | |  |
| Employed | Unemployed | Total |
| Gender | M | 1 | 2 | 3 |
| F | 5 | 2 | 7 |

E(Profile, Gender) = P(M)E(M) + P(F)E(F)

= 0.3E(1,2) + 0.7E(5,2)

= 0.3(0.91829) + 0.7(0.86312)

= 0.879671

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Profile (10) | |  |
| Employed | Unemployed | Total |
| Education | College | 5 | 2 | 7 |
| High School | 0 | 2 | 2 |
| Graduate | 1 | 0 | 1 |

E(Profile, Education) = P(College)E(College) + P(HS)E(HS) + P(Graduate)E(Graduate)

= 0.7E(5,2) + 0.2E(0,2) + 0.1E(0,1)

= 0.7(0.86312) + 0 + 0

= 0.804184

= 0.604184

**Step 4: Calculating Information Gain for Each Split**

Gain(S,T) = Entropy(S) – Entropy(S,T)

Gain(Profile, Age) = E(Profile) – E(Profile, Age)

= 0.97095 - 0.485475 = 0.485475

Gain(Profile, City) = E(Profile) – E(Profile, City)

= 0.97095-0.804184 =0.166766

Gain(Profile, Gender) = E(Profile) – E(Profile, Gender)

= 0.97095-0.879671 = 0.091279

Gain(Profile, Education) = E(Profile) – E(Profile, Education)

= 0.97095-0.604184 = 0.366766

The attribute “Age” gives the highest information gain after the split, therefore; it will be the decision node of the decision tree.

**Step 5: Find the Second Node:**

Entropy(Age = 20-30) =

= 0.97095

Gain(Age = 20-30, Education) = E(Age = 20-30) – E(Age = 20-30, Education)

= 0.97095 – 0 – 0 – 0

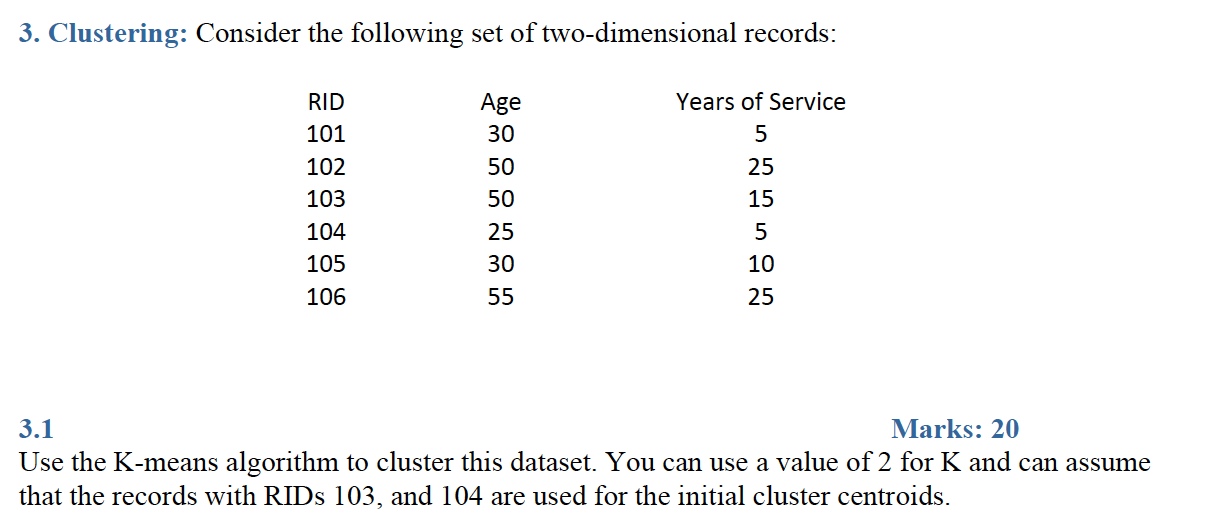
= 0.97095

Since 0.97095 is the highest possible value for the second information gain, Education is the second decision node of the decision tree.

**Step 6: Decision Tree**

Yellow – Decision node

Blue – Leaf nodes



Using Euclidean Distance:

Distance [(x,y), (a,b)] =

Initial cluster centroids: (50,15) and (25,5)

RID 101:

= 17.32050

= 5

RID 102:

= 10

= 32.01562

RID 103:

= 0

= 26.92582

RID 104:

= 26.92582

= 0

RID 105:

= 20.61552

= 7.07106

RID 106:

= 11.18033

= 36.551

Cluster 1: RID 101, 104, 105

Cluster 2: RID 102, 103, 106

**Location:**

Cluster 1: (28.33, 6.67)

Cluster 2: (51.67, 21.67)

**Iteration 2**

RID 1011:

= 2.36173

= 27.34004

RID 102:

= 28.38270

= 3.72529

RID 103:

= 23.21589

= 6.87588

RID 104:

= 3.72529

= 31.45119

RID 105:

= 3.72529

= 24.61255

RID 106:

= 32.36167

= 4.70933

Cluster 1: RID 101, 104, 105

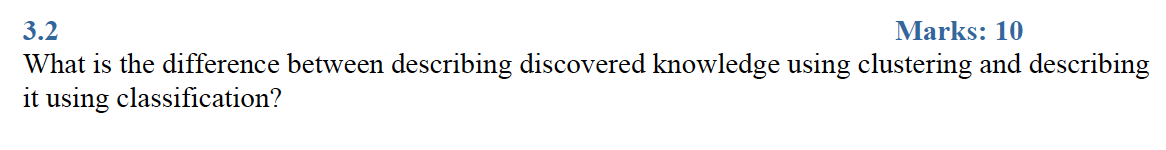
Cluster 2: RID 102, 103, 106

**Location:**

Cluster 1: (28.33, 6.67)

Cluster 2: (51.67, 21.67)

Since locations of two clusters do not change, K-means algorithm ends.



The purpose of clustering is to understand data and patterns in data set (descriptive method), whereas the goal of classification is to predict a value for categorical variable (predictive method).

Clustering is an unsupervised learning method that groups a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups. Discovered knowledge of clustering is the descriptions of the dataset.

On the other hand, classification is a supervised learning method that identifies which of a set of categories a new observation belongs to, on the basis of a training set. Discovered knowledge of classification is the prediction of the class variable for the test set.