```
// <NaiveBayes rollNo=8390 name='Joyson' />
1
 2
 3
     #include <iostream>
 4
     #include <vector>
 5
     #include <string>
 6
     #include <algorithm>
 7
     #include <iterator>
8
     #include <map>
9
10
     using namespace std;
11
12
     auto countFreq(vector<string> vect, int n) {
13
         map<string, float> mp;
14
15
         for(auto x : vect)
16
              mp[x]++;
17
18
         return mp;
19
     }
20
     int main() {
21
         int n,m,i,j,e;
22
23
         string temp, temp1, temp2;
24
         cout<<"Enter number of tuples\n";</pre>
25
         cin>>n;
26
         cout<<"Enter number of attributes\n";</pre>
27
         cin>>m;
28
         vector<string> b, newtuple, classes, raw_list;
29
30
         vector<float> probabilities;
         map<string, float> frequency, prob_match;
31
32
         vector<vector<string>> a;
33
         a.resize(n);
         for(i=0; i<a.size(); i++)
34
35
              a[i].resize(m);
36
37
         cout<<"Enter attribute labels (Restrict label names within 7 alphabets)\n";</pre>
38
         for(i=0;i<m;i++) {
39
              cin>>temp;
40
              b.push_back(temp);
         }
41
42
43
44
         for(i=0;i<n;i++) {
              cout<<"Enter tuple "<<i+1<<" (Restrict attribute values within 7 alphabets)</pre>
45
              \n";
              for(j=0;j<m;j++) {
46
47
                  cin>>temp;
48
                  a[i][j]=temp;
49
              }
         }
50
51
         cout<<"\n\n||\t\t||";
52
         for(i=0;i<m;i++) {
53
              cout<<"\t"<<b[i]<<"\t||";
54
55
         cout<<"\n";
56
57
         for(i=0;i<20*m;i++)
              cout<<"_";
58
         cout<<"\n";</pre>
59
60
         for(i=0;i<n;i++) {
61
62
              cout<<"||\t"<<ii+1<<"\t||";
63
              for(j=0;j<m;j++) {
```

```
cout<<"\t"<<a[i][j]<<"\t||";
 64
 65
              cout<<"\n";</pre>
 66
 67
          }
 68
          cout<<"\n\nEnter new tuple to be classified (Restrict attribute values within 7
 69
          alphabets) \n";
          for(i=0;i<m-1;i++) {
 70
 71
              cin>>temp;
 72
              newtuple.push_back(temp);
 73
          }
 74
          for(i=0;i<n;i++) {
 75
              temp = a[i][n-1];
 76
 77
              bool present = binary_search(classes.begin(), classes.end(), temp);
              if(!present)
 78
 79
                   classes.push_back(temp);
          }
 80
 81
          for(i=0;i<n;i++) {
 82
 83
               for(j=0;j<m;j++) {
                   raw_list.push_back(a[i][j]);
 84
 85
              }
 86
          }
 87
 88
          e = sizeof(raw_list)/sizeof(raw_list[0]);
          frequency = countFreq(raw_list, e);
 89
 90
          for(auto x : classes) {
 91
              float prod = 1;
 92
               for(auto y : newtuple) {
 93
 94
                   float count = 0;
                   for( auto z : a) {
 95
                       bool contains1 = 0, contains2 = 0;
96
 97
                       if(std::find(z.begin(), z.end(), x)!=z.end())
                           contains1 = 1;
 98
 99
                       if(std::find(z.begin(), z.end(), y)!=z.end())
100
                           contains2 = 1;
101
                       if(contains1 && contains2)
102
                       {
103
                           count++;
                       }
104
105
                   }
106
                   prod = prod * ((float)count/(float)frequency[x]);
107
              prod = prod * ((float)frequency[x]/(float)n);
108
109
              prob_match[x]=prod;
          }
110
111
112
          float sum = 0;
113
          for (auto x : prob_match)
114
              sum+=x.second;
115
          for (auto &x : prob_match)
116
117
              x.second = x.second/sum;
118
119
          for(auto x : prob_match)
120
              cout<<"\n"<<x.first<<" : "<<x.second*100<<"%"<<endl;</pre>
121
          return 0;
122
      }
123
124
125
```

126

```
/*
128
      Enter number of
      tuples
129
      Enter number of
130
      attributes
131
      Enter attribute labels (Restrict label names within 7
132
      alphabets)
      Outlook Temp Humidty Windy
133
      Enter tuple 1 (Restrict attribute values within 7
134
      alphabets)
      Rainy Hot High False
135
      Enter tuple 2 (Restrict attribute values within 7
136
      alphabets)
      Rainy Hot High True
137
      Enter tuple 3 (Restrict attribute values within 7
138
      alphabets)
139
      Ovrcast Hot High False
      Enter tuple 4 (Restrict attribute values within 7
140
      alphabets)
      Sunny Mild High False
141
      Enter tuple 5 (Restrict attribute values within 7
142
      alphabets)
      Sunny Cool Normal False
143
      Yes
144
145
                                    146
      П
              П
                  Outlook ||
                               Temp
                                         Humidty ||
                                                      Windy
                                                              Ш
                                                                  Play
                                                                       - 11
147
148
      П
          1
              Ш
                  Rainy
                           Ш
                               Hot
                                          High
                                                  Ш
                                                      False
                                                              Ш
                                                                  No
                                                                        Ш
149
          2
                  Rainy
                               Hot
                                          High
                                                      True
                                                                  No
      Ш
              Ш
                           Ш
                                      Ш
                                                  Ш
                                                              Ш
                                                                        Ш
                                                      False
          3
                               Hot
                                          High
                                                                  Yes
150
      Ш
                  Ovrcast ||
                                      Ш
              П
                                                  Ш
                                                                        Ш
151
          4
              Ш
                  Sunny
                           Ш
                               Mild
                                          High
                                                      False
                                                              Ш
                                                                  Yes
                                                                        Ш
152
      П
          5
              П
                  Sunny
                           \prod
                               Cool
                                     Ш
                                          Normal
                                                  Ш
                                                      False
                                                              Ш
                                                                  Yes
                                                                        Ш
153
154
      Enter new tuple to be classified (Restrict attribute values within 7
155
      alphabets)
      Rainy Hot High
156
      True
157
      No:
158
      100%
159
      Yes : 0%
160
161
      */
```

127