Name: Muhammad Usman

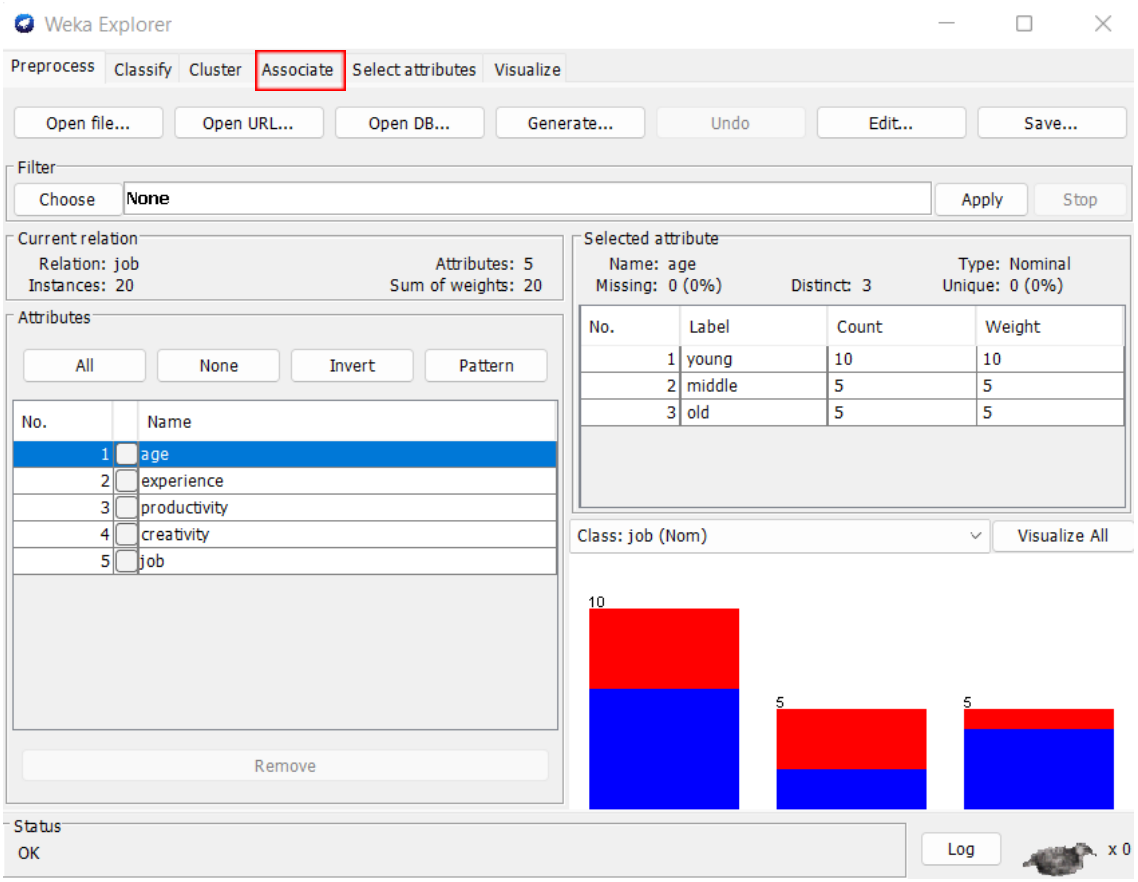
Roll no: 19P-0116

ML Assignment

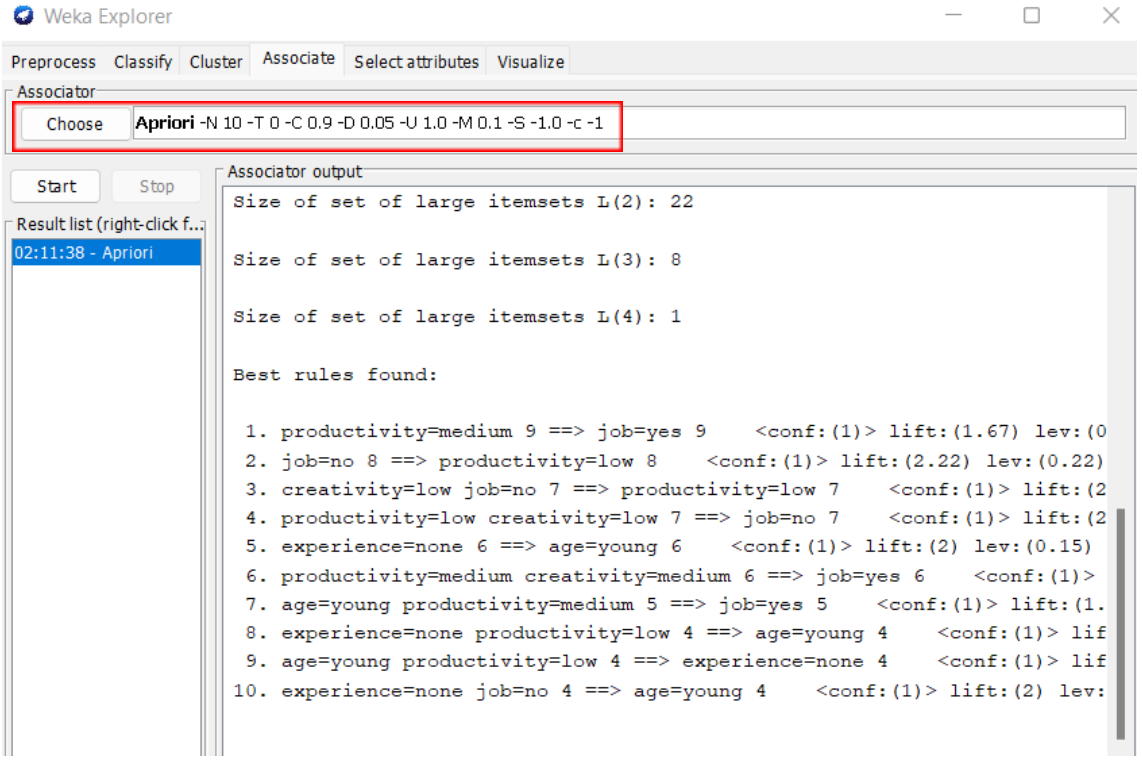
ARFF File:

```
% Job Interview Dataset

@RELATION job
@ATTRIBUTE age {young, middle, old}
@ATTRIBUTE experience {none, low, medium, high}
@ATTRIBUTE productivity {low, medium, high}
@ATTRIBUTE creativity {low, medium, high}
@ATTRIBUTE job {yes, no}

@DATA
young, none, medium, medium, yes
young, low, medium, low, yes
young, medium, medium, medium, yes
young, none, low, low, no
young, low, medium, low, yes
young, medium, high, high, yes
old, high, medium, medium, yes
middle, medium, low, low, no
old, medium, low, low, no
young, none, medium, medium, yes
middle, medium, medium, low, yes
old, medium, low, medium, yes
young, none, low, low, no
old, high, high, medium, yes
middle, low, low, low, no
young, none, low, medium, no
middle, medium, medium, medium, yes
old, high, medium, medium, yes
young, none, low, low, no
middle, medium, low, low, no
```

Now, open the arff file in Weka.



Using Apriori Algorithm to mine rules.



Association Rule Mining:

```
Apriori
=======

Minimum support: 0.2 (4 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 16

Generated sets of large itemsets:

Size of set of large itemsets L(1): 11

Size of set of large itemsets L(2): 22

Size of set of large itemsets L(3): 8

Size of set of large itemsets L(4): 1

Best rules found:

 1. productivity=medium 9 ==> job=yes 9    <conf:(1)> lift:(1.67) lev:(0.18) [3] conv:(3.6)
 2. job=no 8 ==> productivity=low 8    <conf:(1)> lift:(2.22) lev:(0.22) [4] conv:(4.4)
 3. creativity=low job=no 7 ==> productivity=low 7    <conf:(1)> lift:(2.22) lev:(0.19) [3] conv:(3.85)
 4. productivity=low creativity=low 7 ==> job=no 7    <conf:(1)> lift:(2.5) lev:(0.21) [4] conv:(4.2)
 5. experience=none 6 ==> age=young 6    <conf:(1)> lift:(2) lev:(0.15) [3] conv:(3)
 6. productivity=medium creativity=medium 6 ==> job=yes 6    <conf:(1)> lift:(1.67) lev:(0.12) [2] conv:(2.4)
 7. age=young productivity=medium 5 ==> job=yes 5    <conf:(1)> lift:(1.67) lev:(0.1) [2] conv:(2)
 8. experience=none productivity=low 4 ==> age=young 4    <conf:(1)> lift:(2) lev:(0.1) [2] conv:(2)
 9. age=young productivity=low 4 ==> experience=none 4    <conf:(1)> lift:(3.33) lev:(0.14) [2] conv:(2.8)
10. experience=none job=no 4 ==> age=young 4    <conf:(1)> lift:(2) lev:(0.1) [2] conv:(2)
```

Selected Rules:

1. productivity = medium □ job = yes (1)
2. productivity = low, creativity = low □ job = no (4)
3. productivity = medium, creativity = medium □ job = yes (6)
4. age = young, productivity = medium □ job = yes (7)

Classifier:

The same data is used to test the classifier.

```
job_df.loc[(job_df['productivity'] == 'medium'), 'class'] = 'yes'

job_df.loc[(job_df['productivity'] == 'low') & (job_df['creativity'] == 'low'), 'class'] = 'no'

job_df.loc[(job_df['productivity'] == 'medium') & (job_df['creativity'] == 'medium'), 'class'] = 'yes'

job_df.loc[(job_df['age'] == 'young') & (job_df['productivity'] == 'medium'), 'class'] = 'yes'
```

Predicted classed are in column named ‘class’.

	age	experience	productivity	creativity	class
0	young	low	medium	low	yes
1	young	medium	medium	medium	yes
2	young	none	low	low	no
3	young	low	medium	low	yes
4	young	medium	high	high	NaN
5	old	high	medium	medium	yes
6	middle	medium	low	low	no
7	old	medium	low	low	no
8	young	none	medium	medium	yes
9	middle	medium	medium	low	yes
10	old	medium	low	medium	NaN
11	young	none	low	low	no
12	old	high	high	medium	NaN
13	middle	low	low	low	no
14	young	none	low	medium	NaN
15	middle	medium	medium	medium	yes
16	old	high	medium	medium	yes
17	young	none	low	low	no
18	middle	medium	low	low	no

Accuracy Calculation:

```
original_labels = np.array(labels)
predicted_labels = job_df['class'].to_numpy()

correct = (original_labels == predicted_labels)

accuracy = (correct.sum() / correct.size)*100

print("Accuracy: ",accuracy,"%")
```

```
Accuracy:  78.94736842105263 %
```