**WEKA TOOL**

C.DEVI PRIYA

192011041

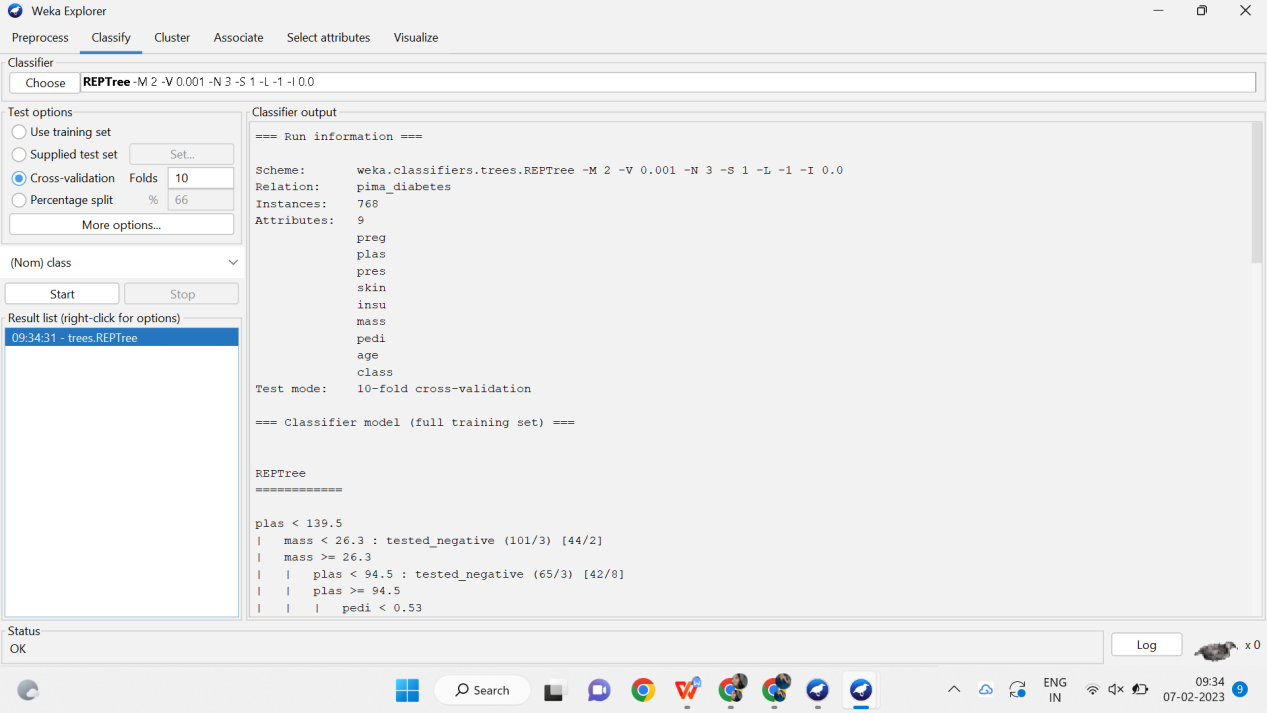
1. **DECISION TREE:**

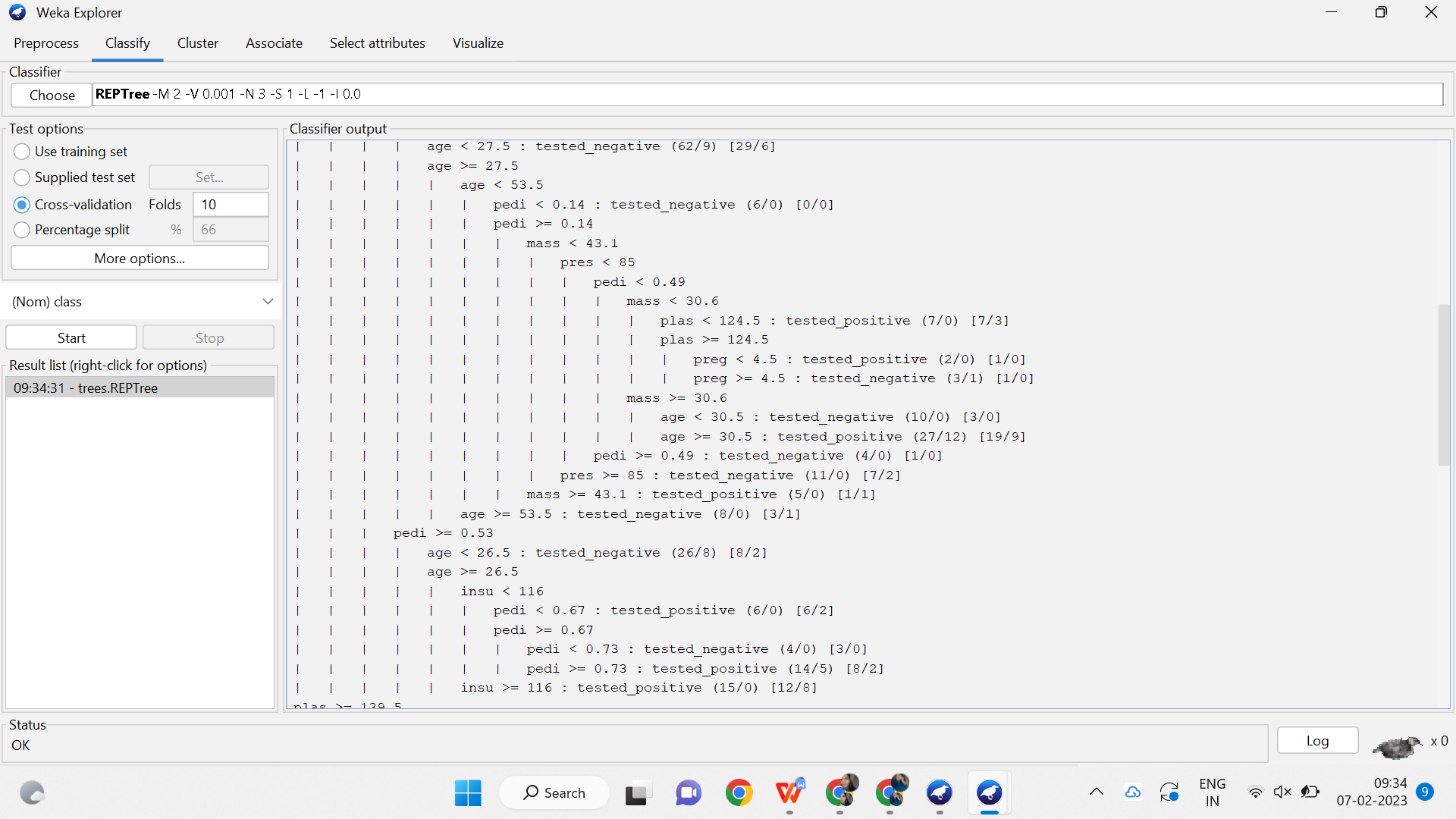
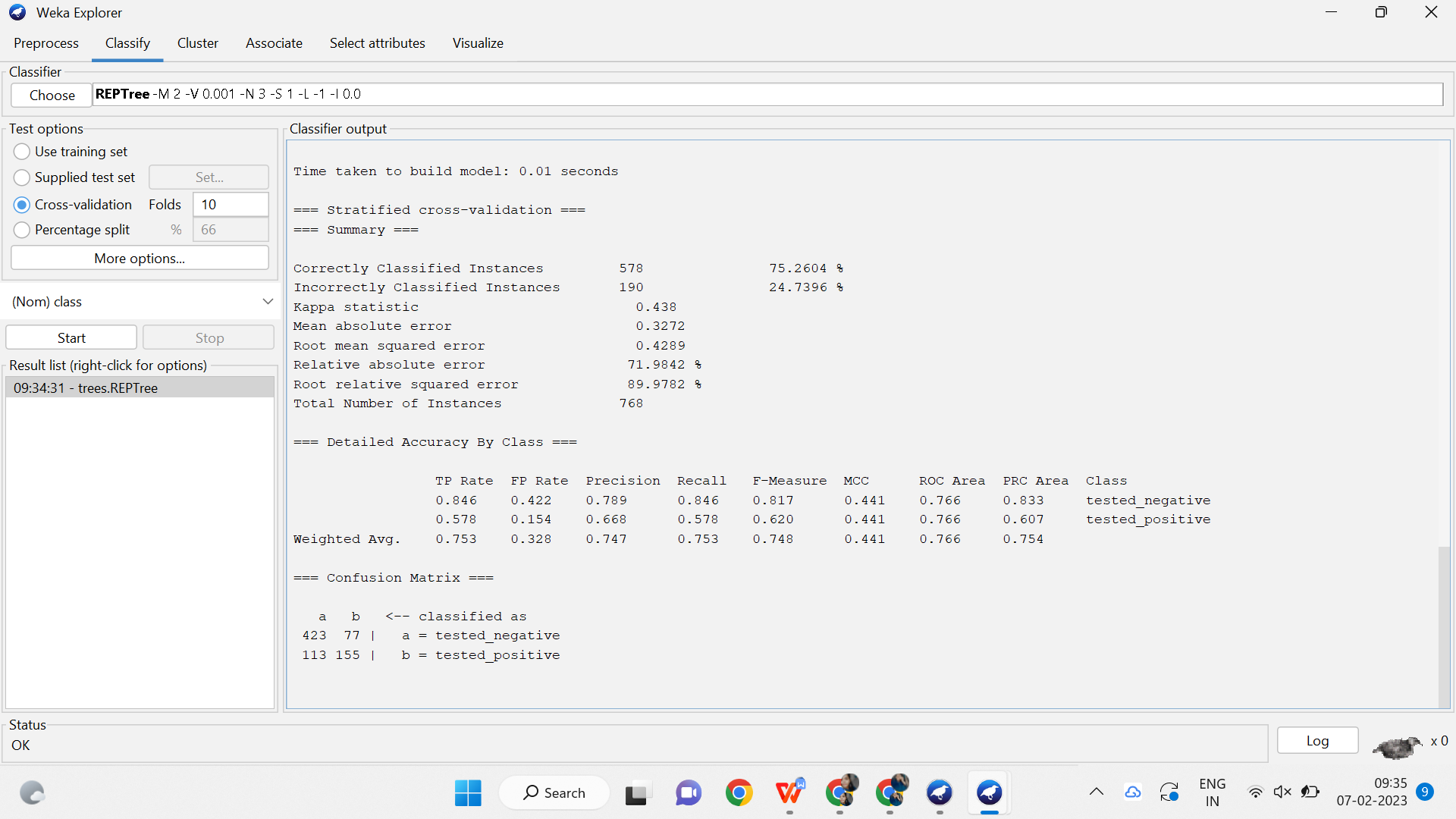
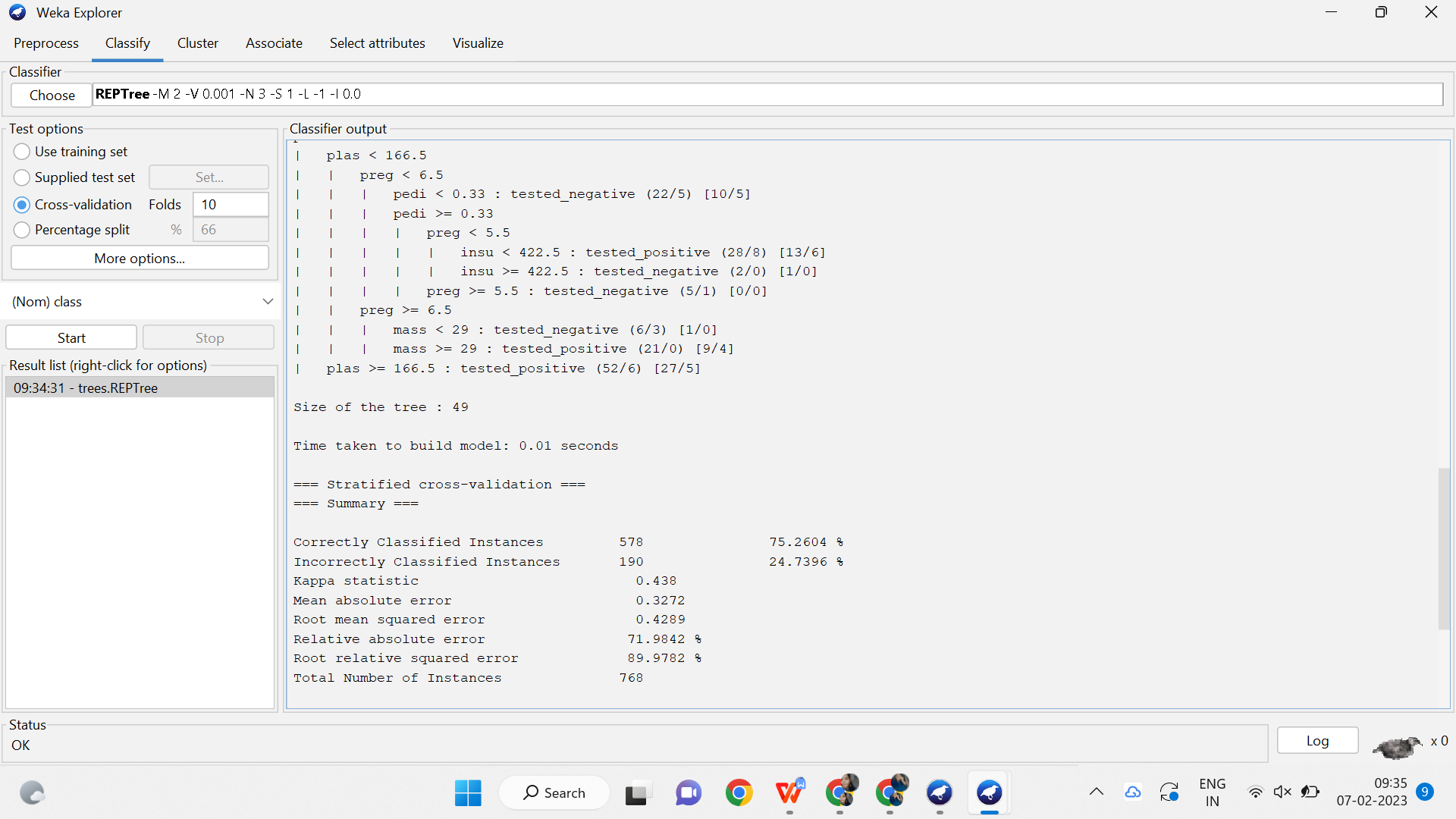
**DATA SET:**diabetes

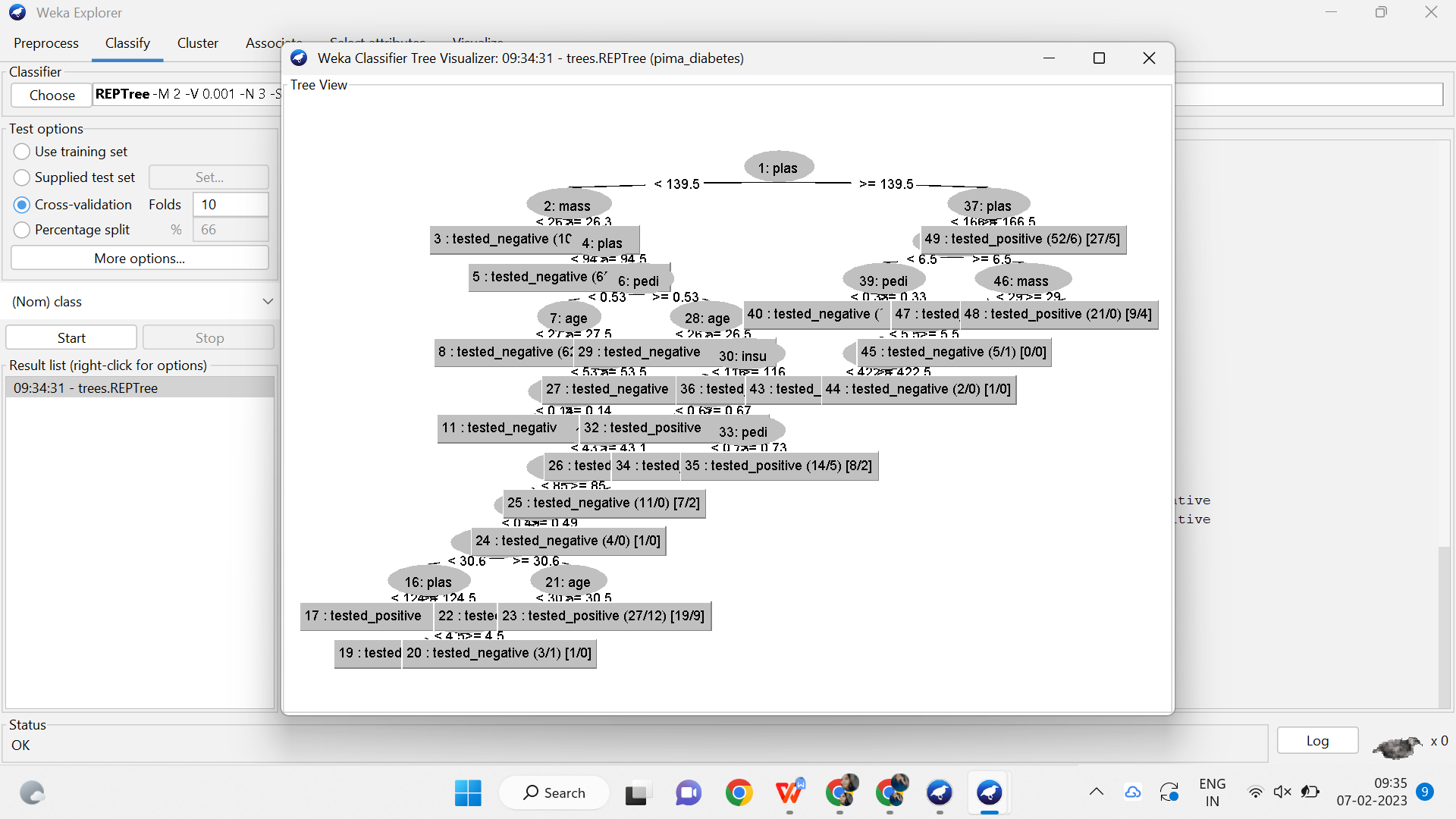
**ALGORITHM:**

1. Determine the root node.
2. Calculate the entropy of classes.
3. Calculate the entropy of the split of the attribute.
4. Calculate the information gain.
5. Perform split.
6. Perform further split.
7. Compute decision tree.
8. Entropy= -Σ pi log2 pi
9. Information gain=entropy of parent node-sum of weights of entropy of child node.

OUTPUT:





1. **APRIORI ALGORITHM:**

**DATASET:** supermarket

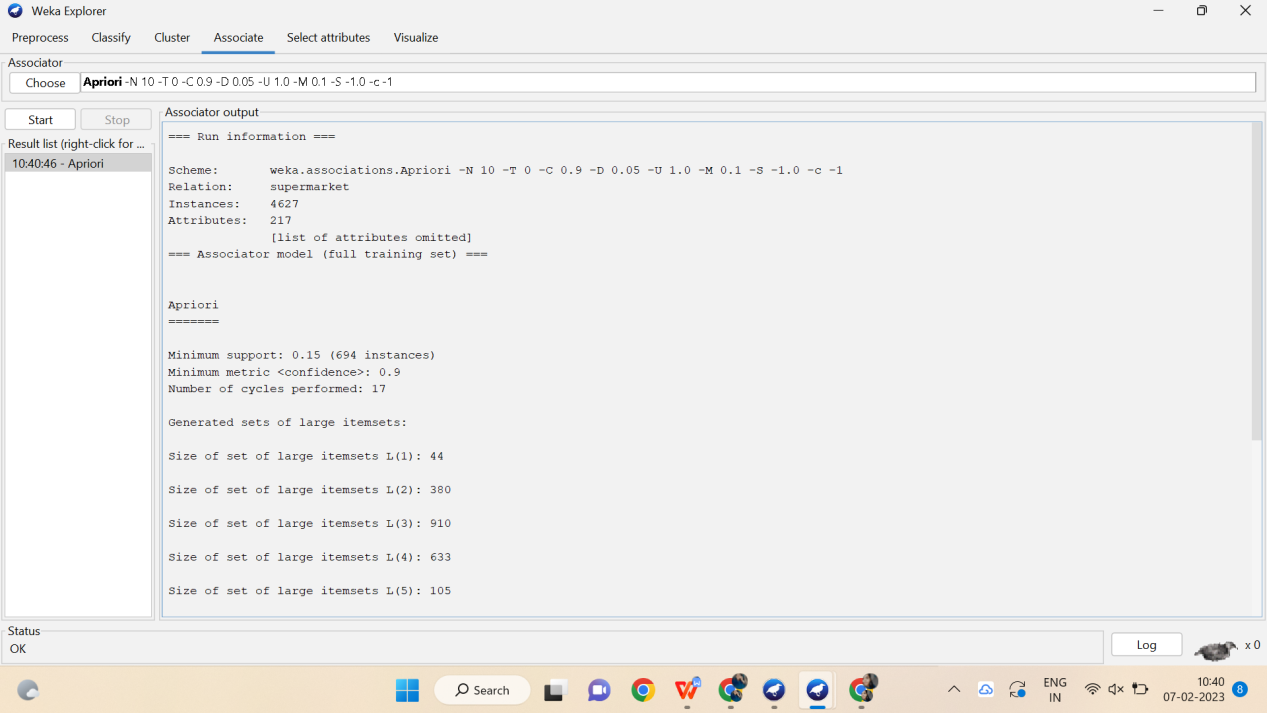
**ALGORITHM:**

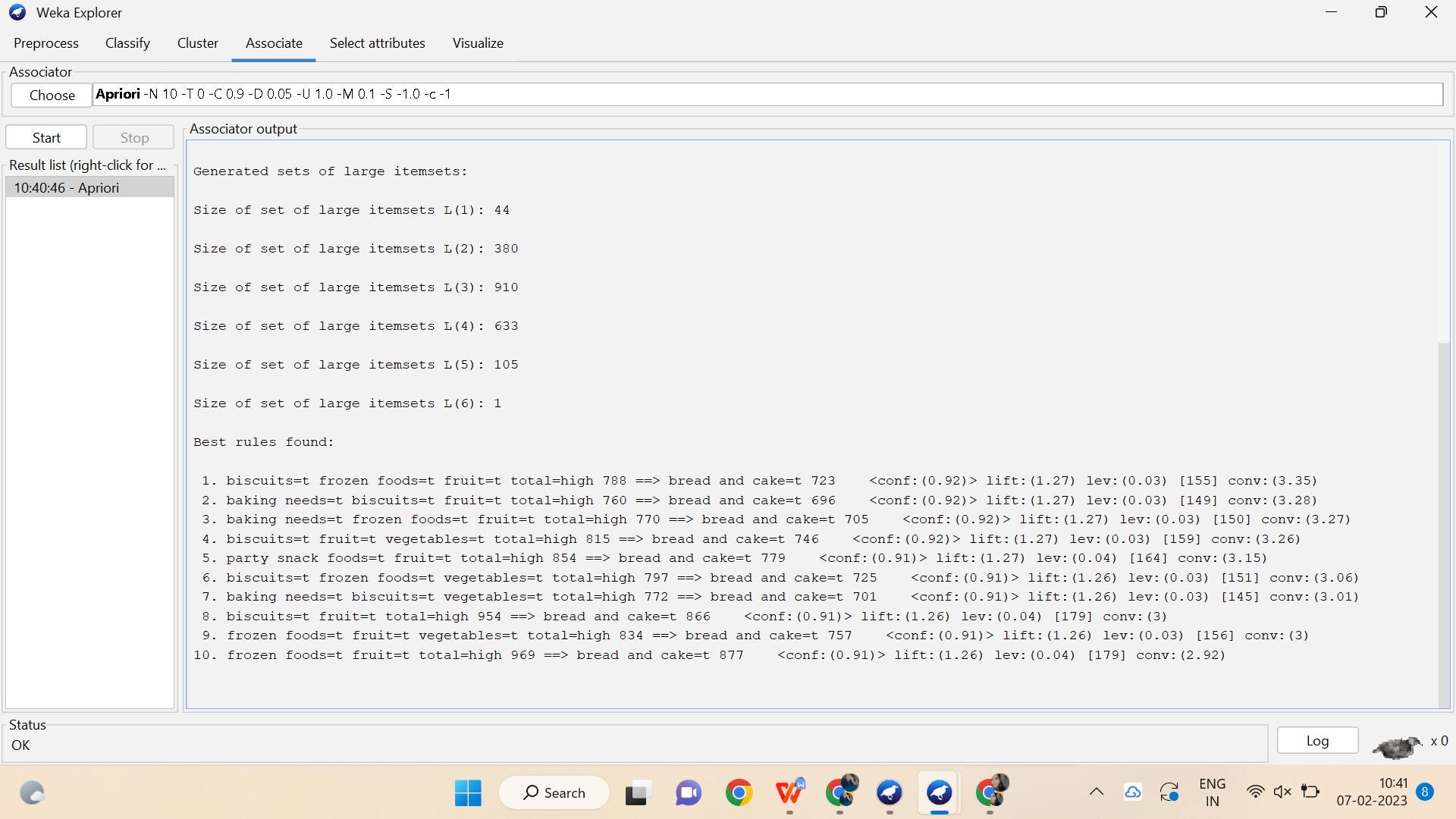
1. firstly,convert the given transactional database into an frequency table.
2. Assign any minimum support to the frequency table,in which contains item sets and support count.
3. The item sets and support count is combinely called as candidate set.
4. Now,check the support count with the minimum support.
5. Remove the support count which is less than minimum support and write the remaining item sets in descending order.
6. Again checking by combining two item sets.
7. iterate the steps until the support count should be equal to minimum support.

Confidence=support(A∩B)/support(A)

1. calculate the confidence and convert it into percentage.
2. Finally, check which is more efficient.

OUTPUT:

****

****

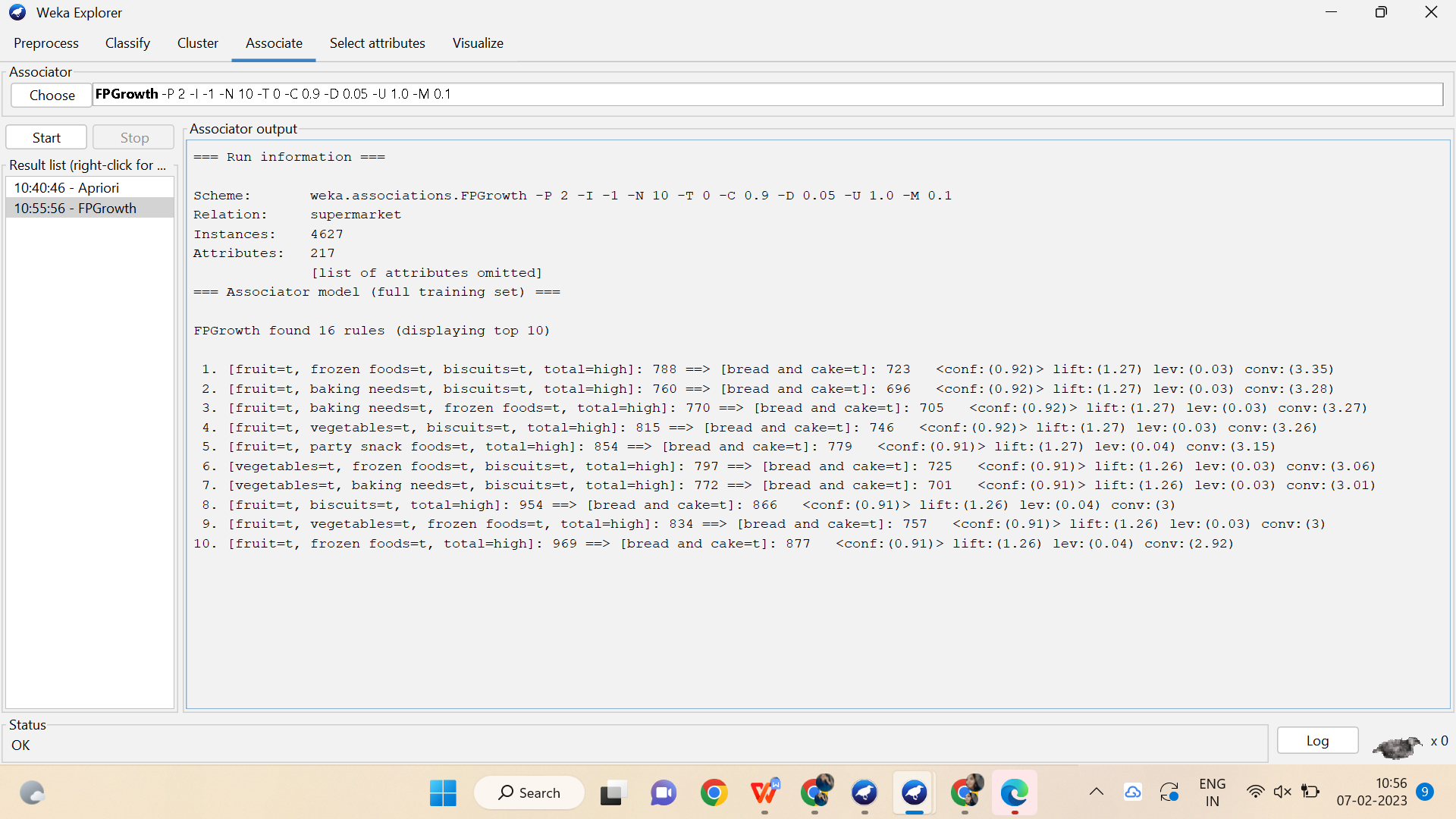
1. **FP GROWTH ALGORITHM:**

**DATASET:** supermarket

**ALGORITHM:**

1. Firstly, convert the given transactional database into an frequency table.
2. Assign any minimum support to the frequency table, in which contains item sets and support count.
3. The item sets and support count is combine called as candidate set.
4. Now, check the support count with the minimum support.
5. Remove the support count which is less than minimum support and write remaining items in descending order.
6. Find the ordered item set using frequency table.
7. Construct the FP growth using the ordered item set.
8. Then compute the conditionally pattern using FP growth.
9. Again find the conditionally frequency pattern.
10. Finally compute the FP growth algorithm.

OUTPUT:

****

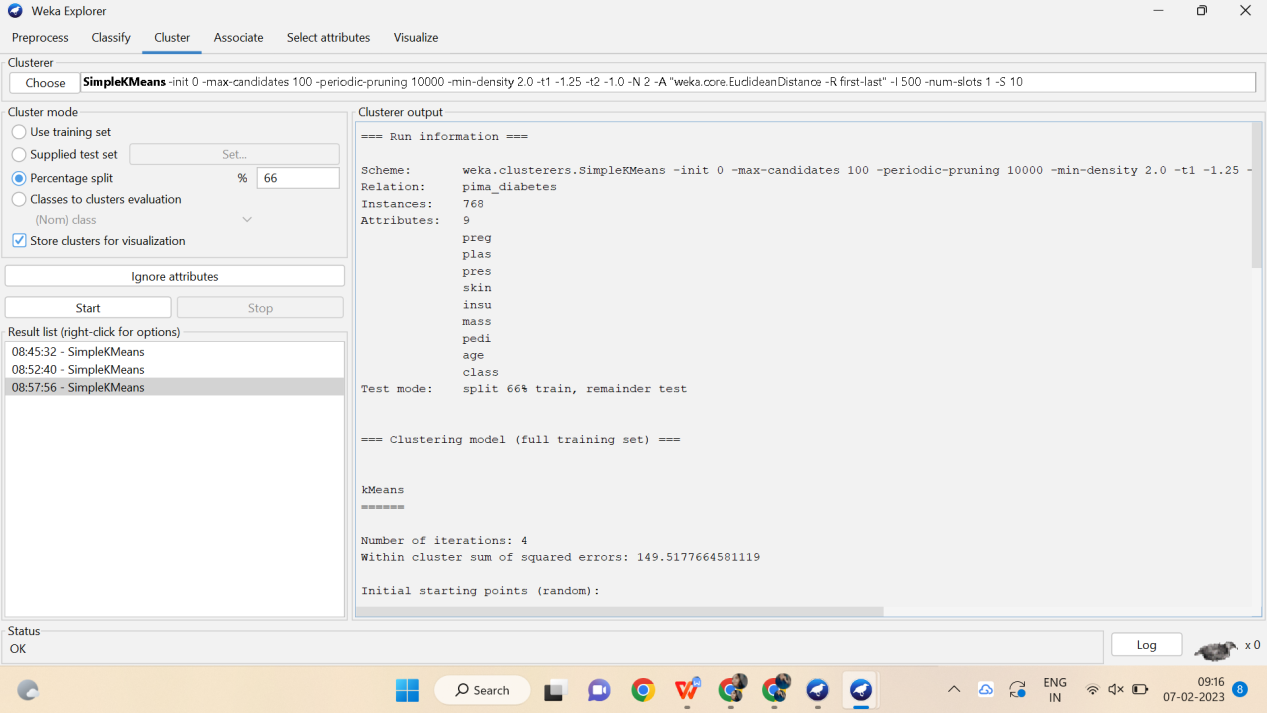
1. **K MEANS ALGORITHM:**

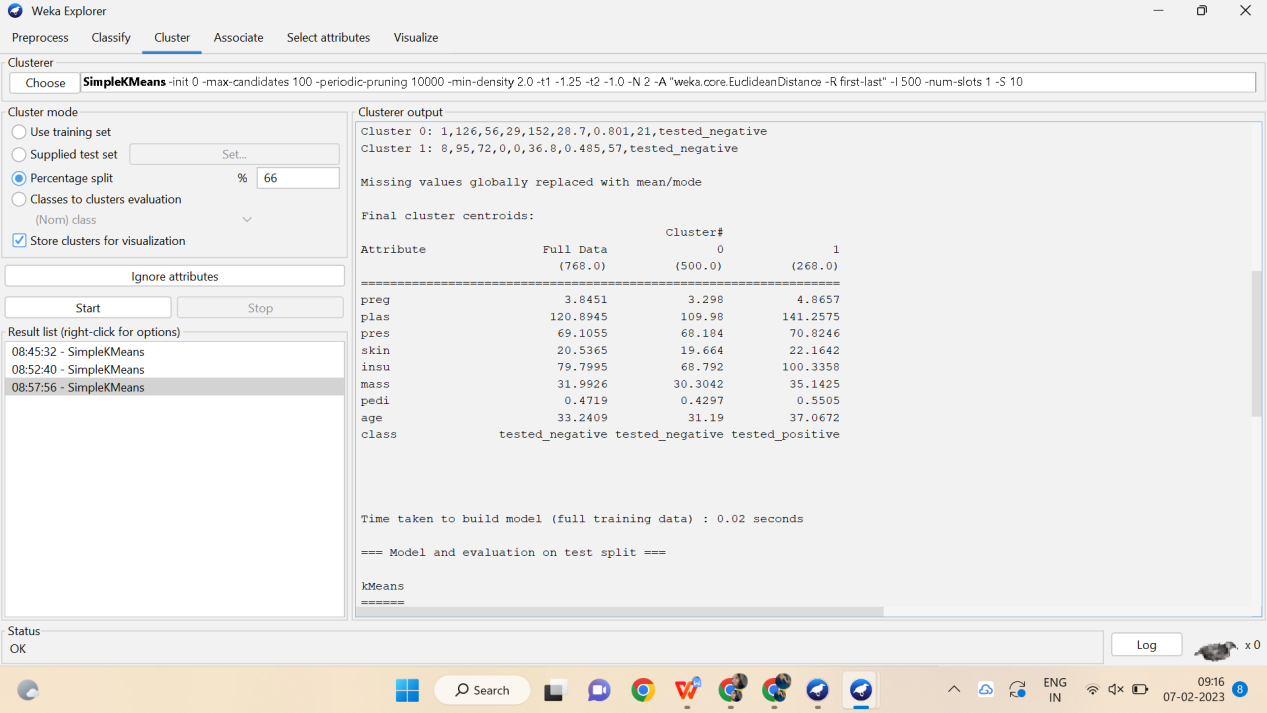
**DATA SET:** diabetes

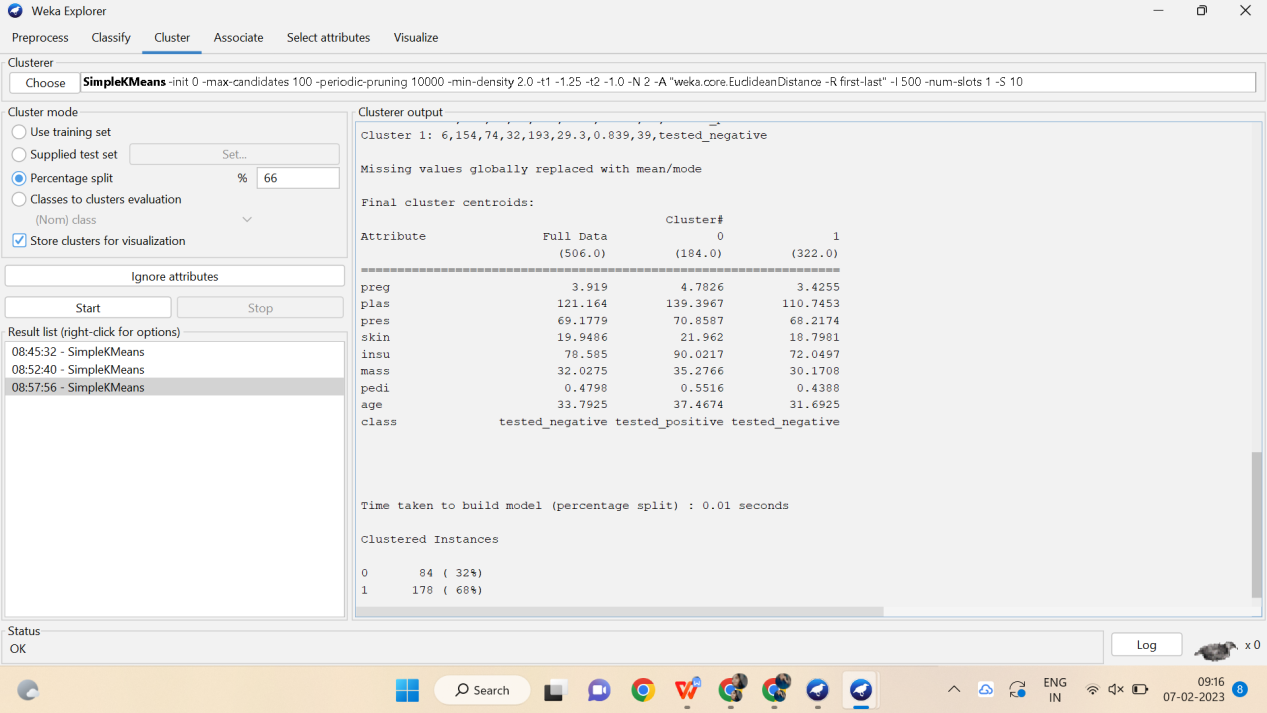
**ALGORITHM:**

1. The k means clustering algorithm computes the centroids and repeats until optimal centroid is found.
2. Firstly,provide k number of clusters.
3. Choose k data points and assign to each clusters and divide data based on data points.
4. The cluster centroids will be constructed.
5. Iterate steps until ideal centroid is found.
6. The sum of squared distances between data points and clusters should be find.
7. Allocate each data point to cluster which is closest to centroid.
8. Construct the centroids for clusters by averaging all data points of clusters.

OUTPUT:

****

****

****

1. **BAYESIAN CLASSIFICATION:**

**DATASET:**diabetes

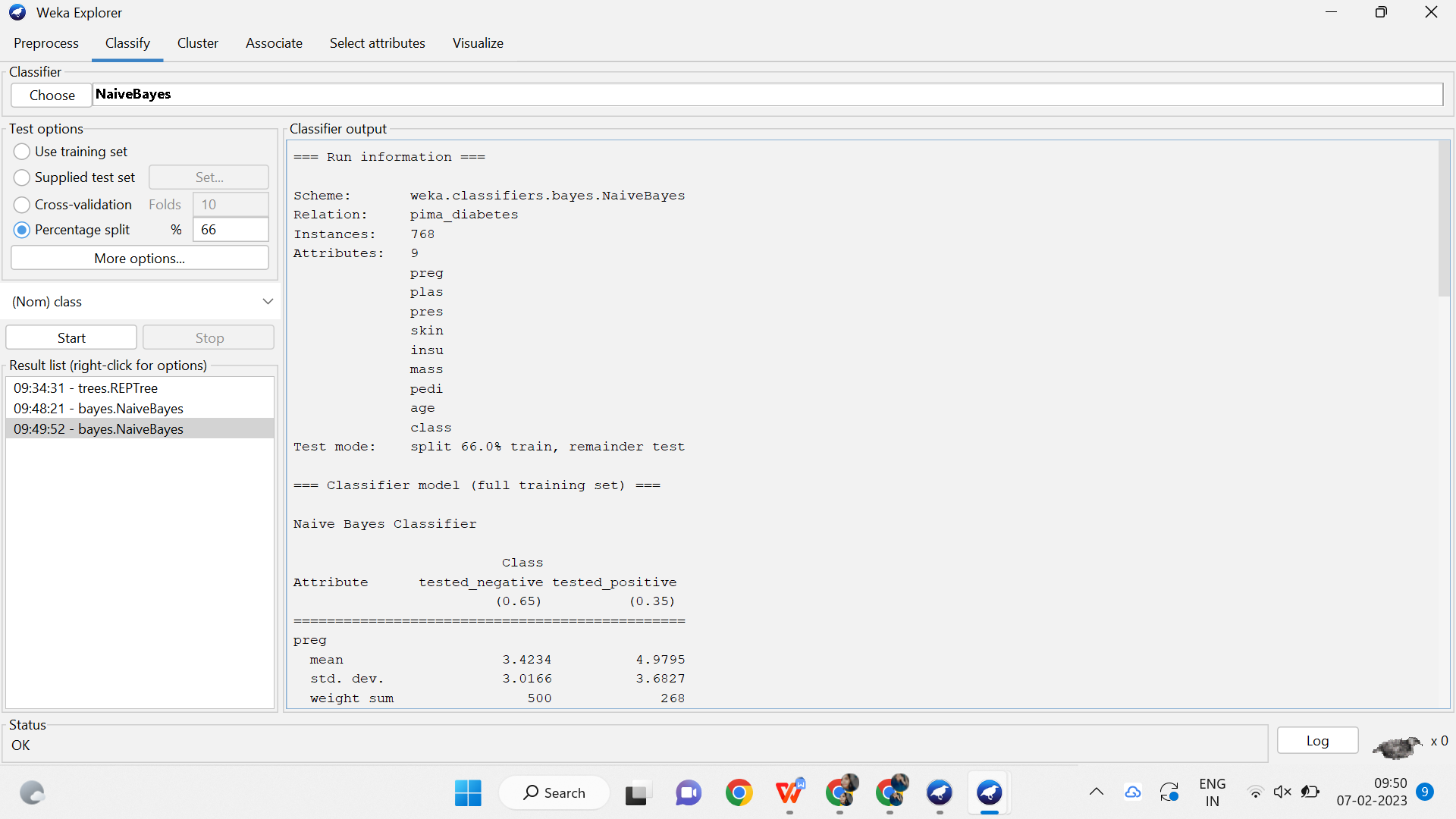
**ALGORITHM:**

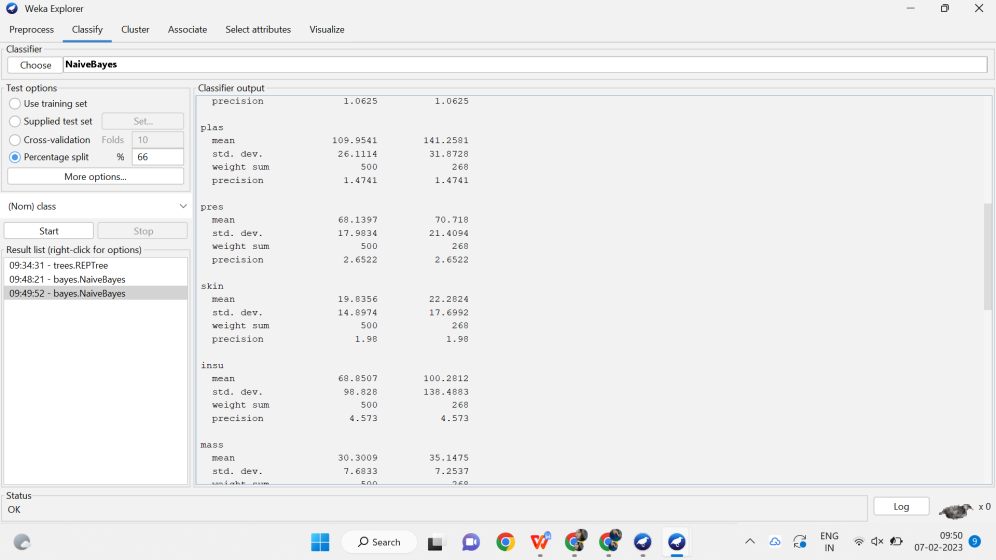
1. convert given dataset into frequency table.
2. Construct livelihood tables by calculating the probabilities.
3. Use the bayes formula for calculating probabilities.

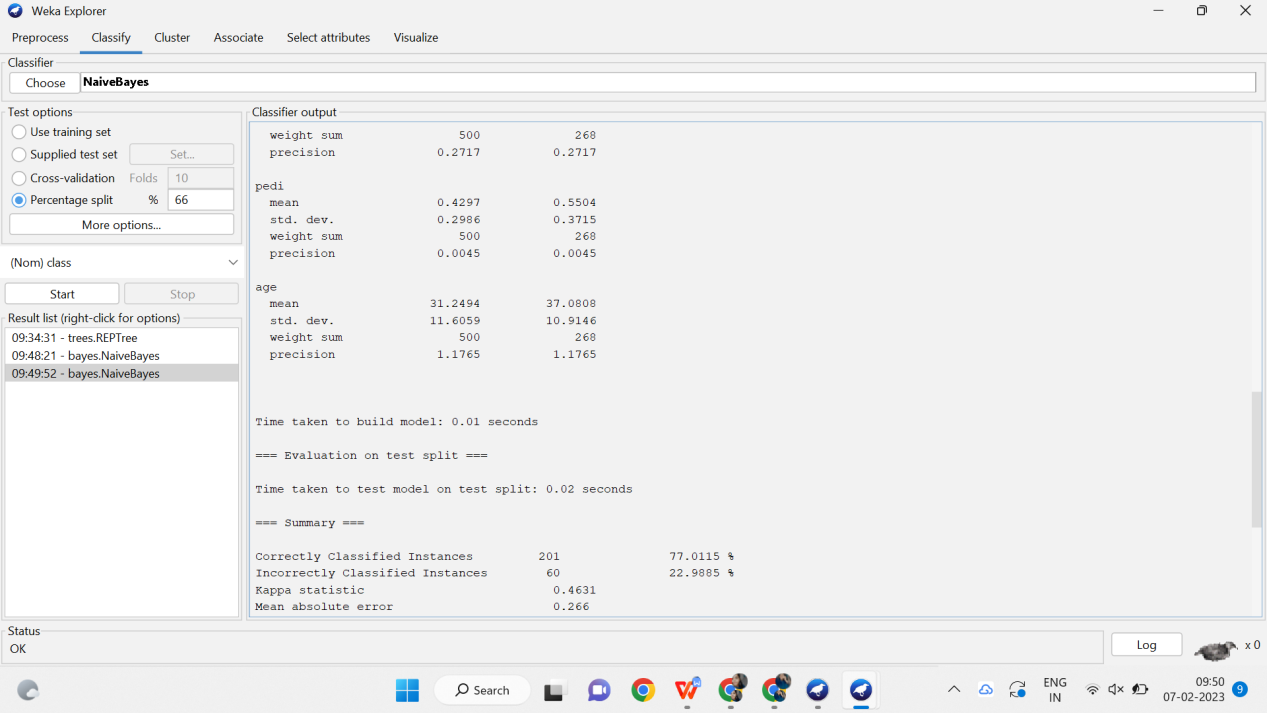
P(A|B) = [P(B|A) P(A)]/ P(B), where P(B) ≠ 0

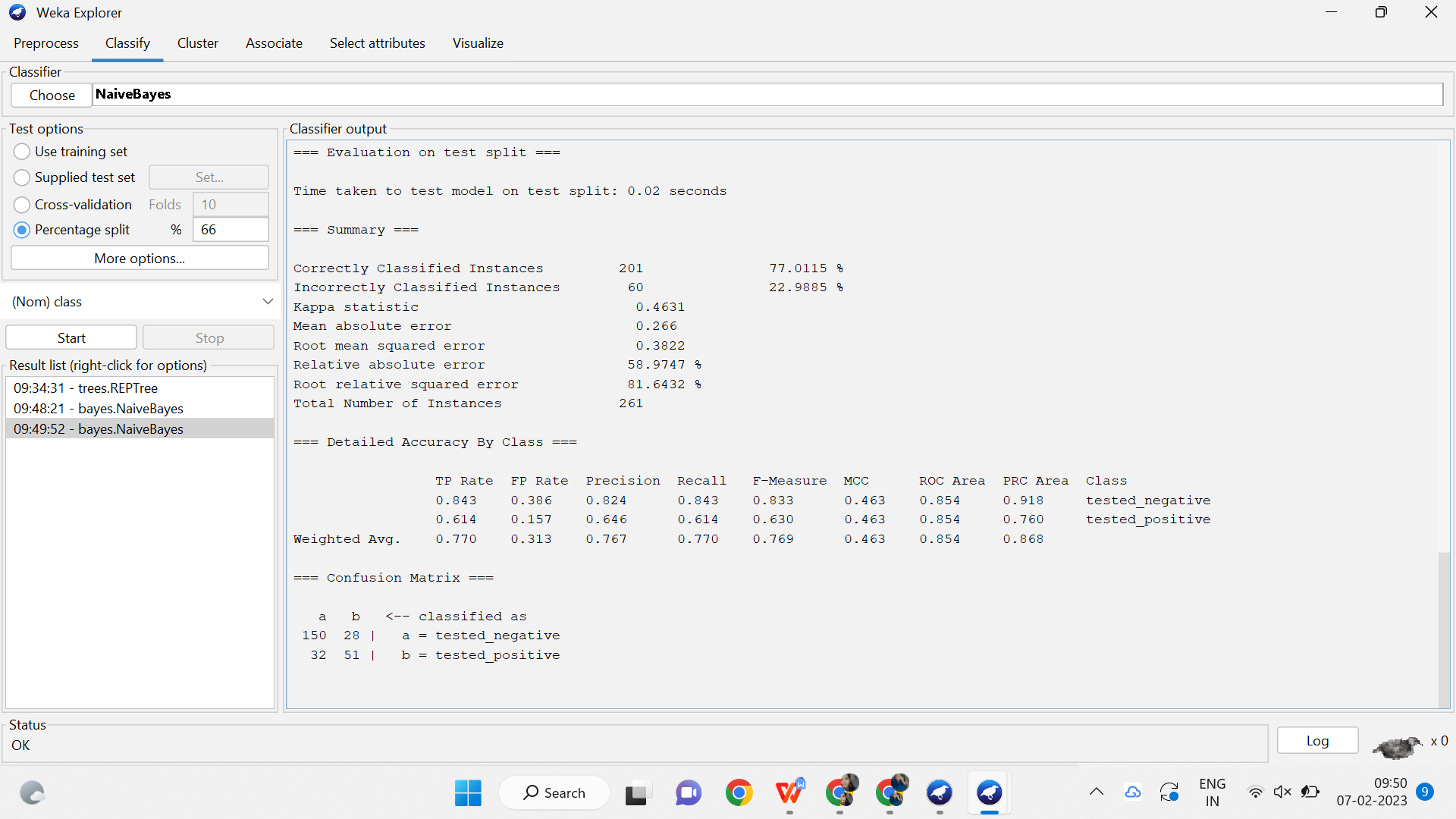
1. now calculate the probability for all possible choices.
2. Then compare all the outputs.
3. Determine the probability which is more efficient by checking outputs.
4. Finally,compute the probability using bayesian classification.

**OUTPUT:**

****

****

****

****