**CS773-HOMEWORK-1**

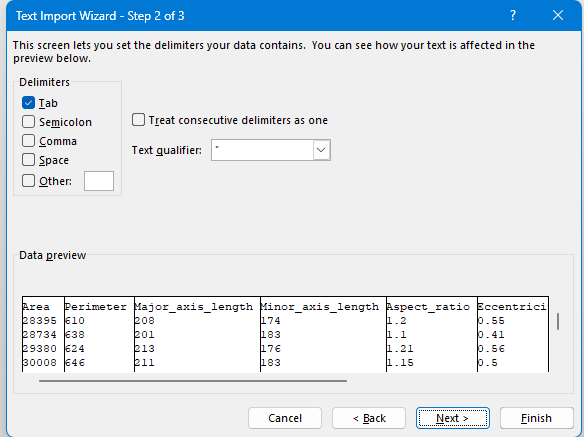
**Step 1:** Create an ARFF file bean\_training\_data.arff with the same 9 fields.

**Solution**

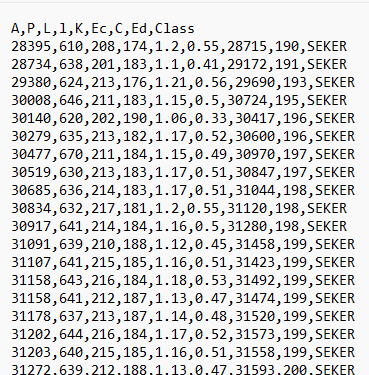
**(a)**Since the input file is in tab delimited, we need to first convert it into CSV so that Weka can read it. Open bean\_training\_data.txt with Microsoft Excel or R and convert the file into .CSV.

**(b)**Read the bean\_training\_data.csv into Weka and convert it into bean\_training\_data.arff as shown below. We can also change the field names wile exporting to Weka in ArffViewer.

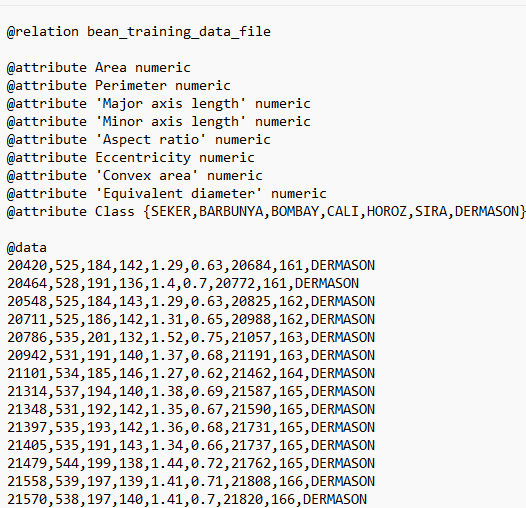
**Screenshots**

****

**Input CSV file**

****

**Output Arff File**

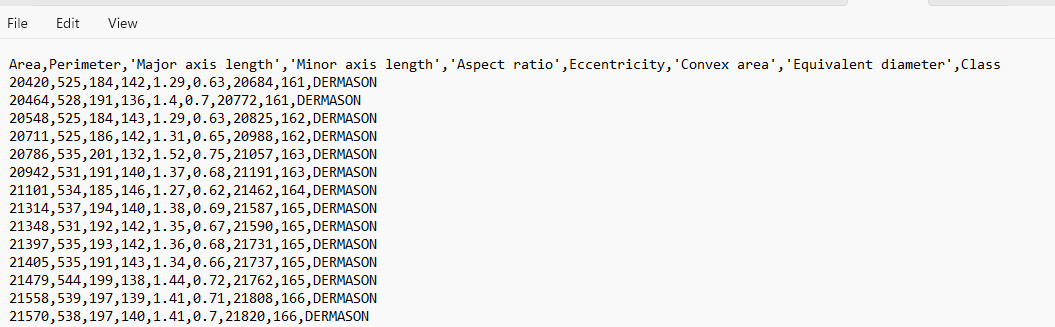
****

**Step 2.** Create a corresponding csv file bean\_training\_data.csv

**Solution**

Either we can use Excel or R to convert tab delimited to .CSV file or we can use previously created bean\_training\_data.arff file into CSV using Arffviewer in Weka.



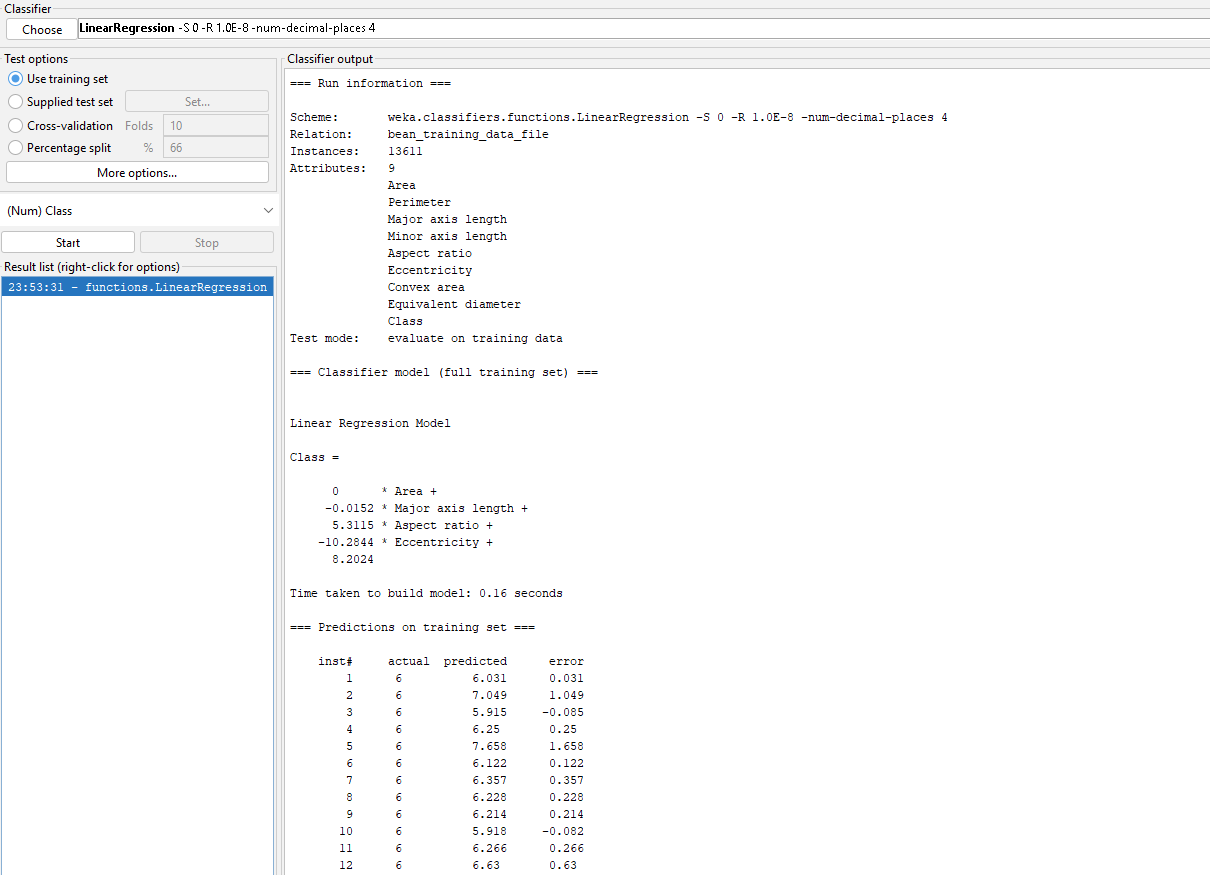


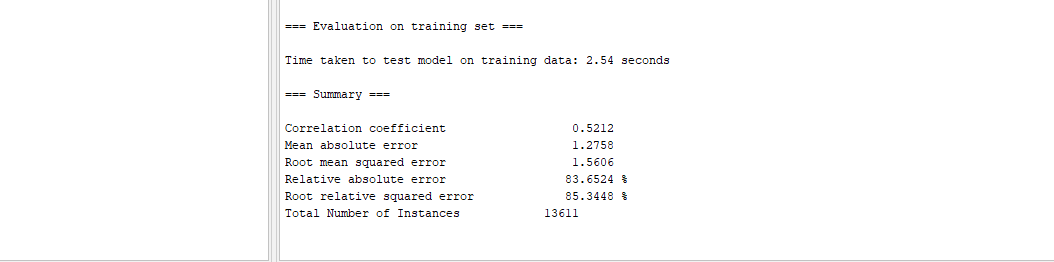
**Step 3.** Using Weka, with Output set to Plain text (under more options), runbean\_training\_data.csv file using (i) Linear Regression (ii) MultilayerPerceptron classifiers(available under function under classify). Show the predicted bean class for each of the 9 records in the test file (bean\_test\_file.txt or related bean\_test\_file.csv) along with the actual class shown

in the test file. You need to convert .txt to the corresponding formats.

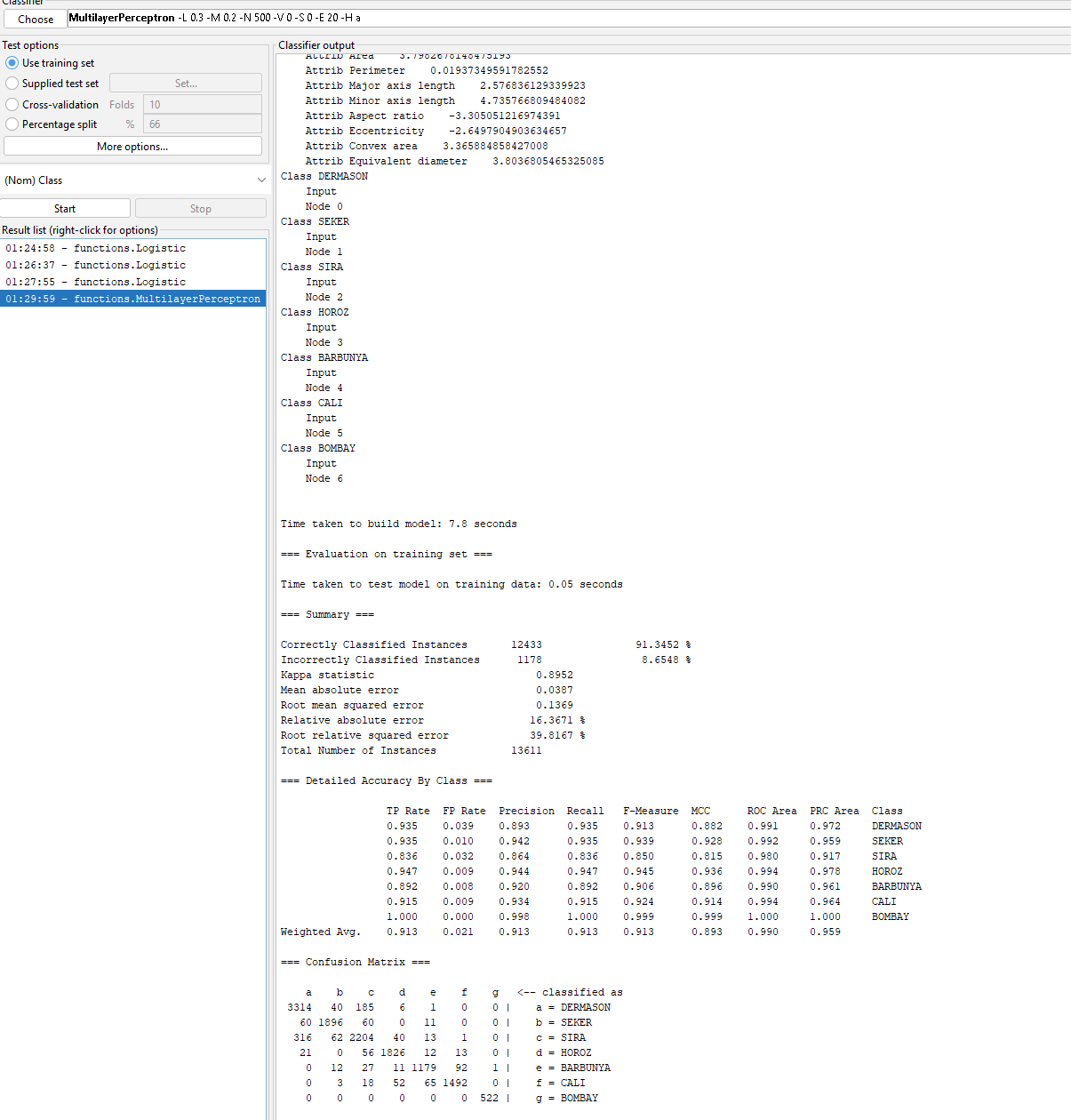
**Solution**

**(a).Linear Regression**





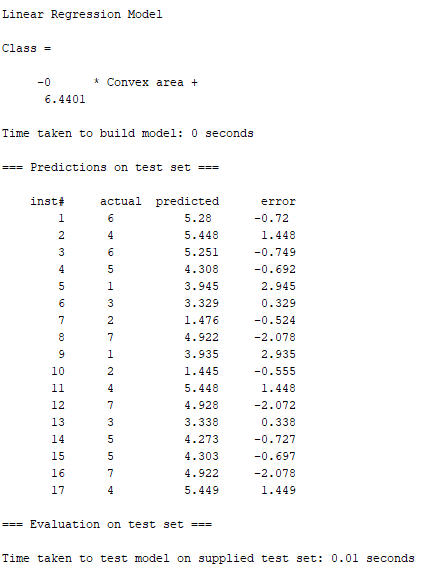
**(b). MultilayerPerceptron**

****

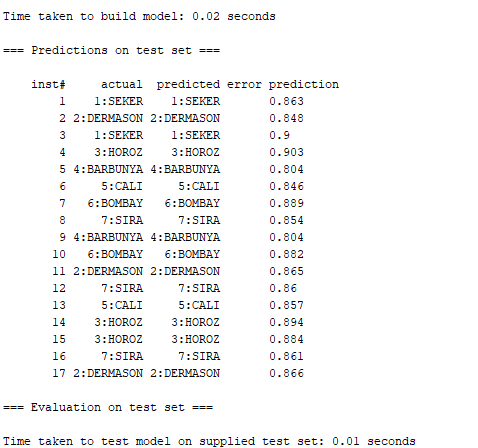
Convert bean\_test\_file.txt to bean\_test\_file.csv either using Excel or R

We need to save model for making prediction in Weka, save log.model and mlp.model for LogisticRegression and MLP.

**(a)Prediction with Linear Regression**

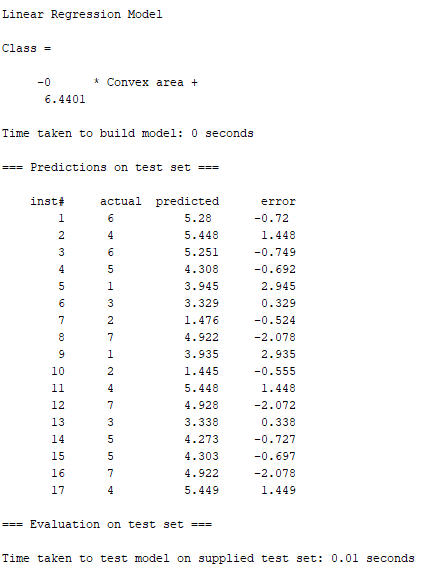
****

**(b)MLP Prediction**

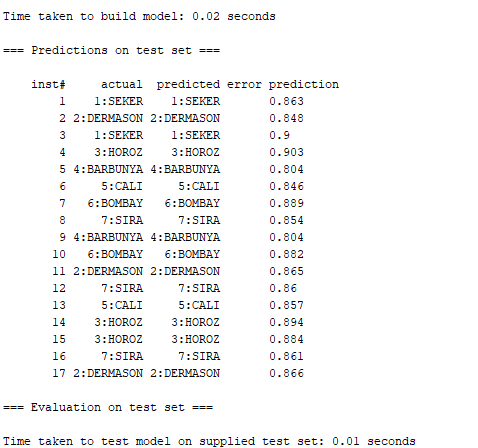
****

Step 4. Repeat step 3 with bean\_training\_data.arff file. You should get exactly the same results as in Step 3

**(a)Prediction with Linear Regression**

****

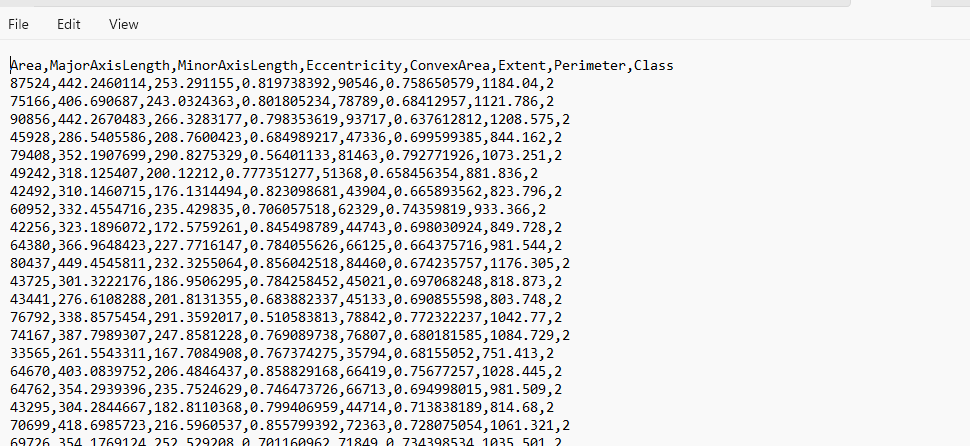
**(b)MLP Prediction**

****

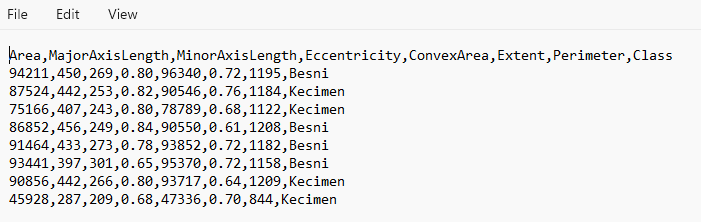
**Step 5.** Consider the raisin data (https://archive.ics.uci.edu/ml/datasets/Raisin+Dataset) predicting the type of raisins. You are provided the training set (Raisin\_data\_set) and the test set (raisin\_test\_data) for this data. By applying “Linear Regression” and “Gaussian Processes” options, predict the output for the test cases. Since output class is categorical, for linear regression, convert the output class to a numeric by mapping each class to an integer 1-7.Similarly, map the regression output back to an actual class based on nearest integer value.

**Solution:**  Since the training and test datasets are in .xls format we need to convert them first to .csv

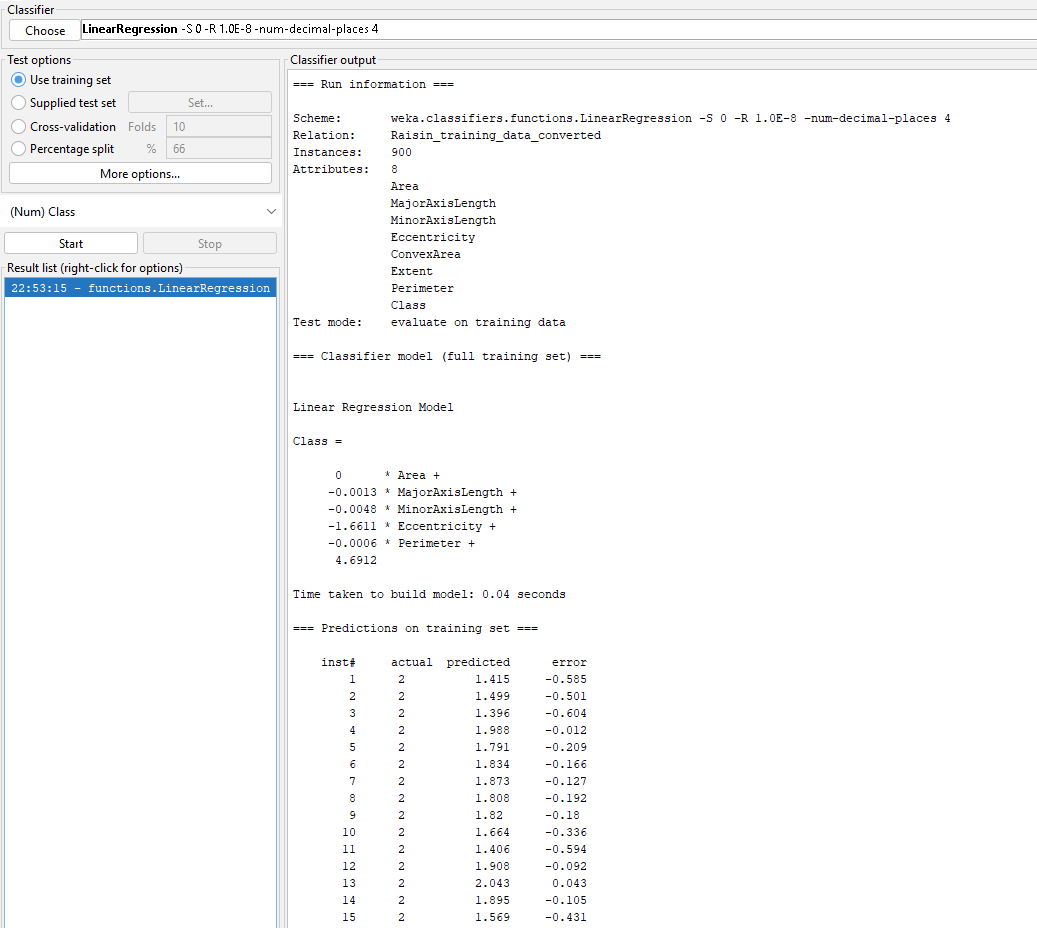
**Csv Training data**



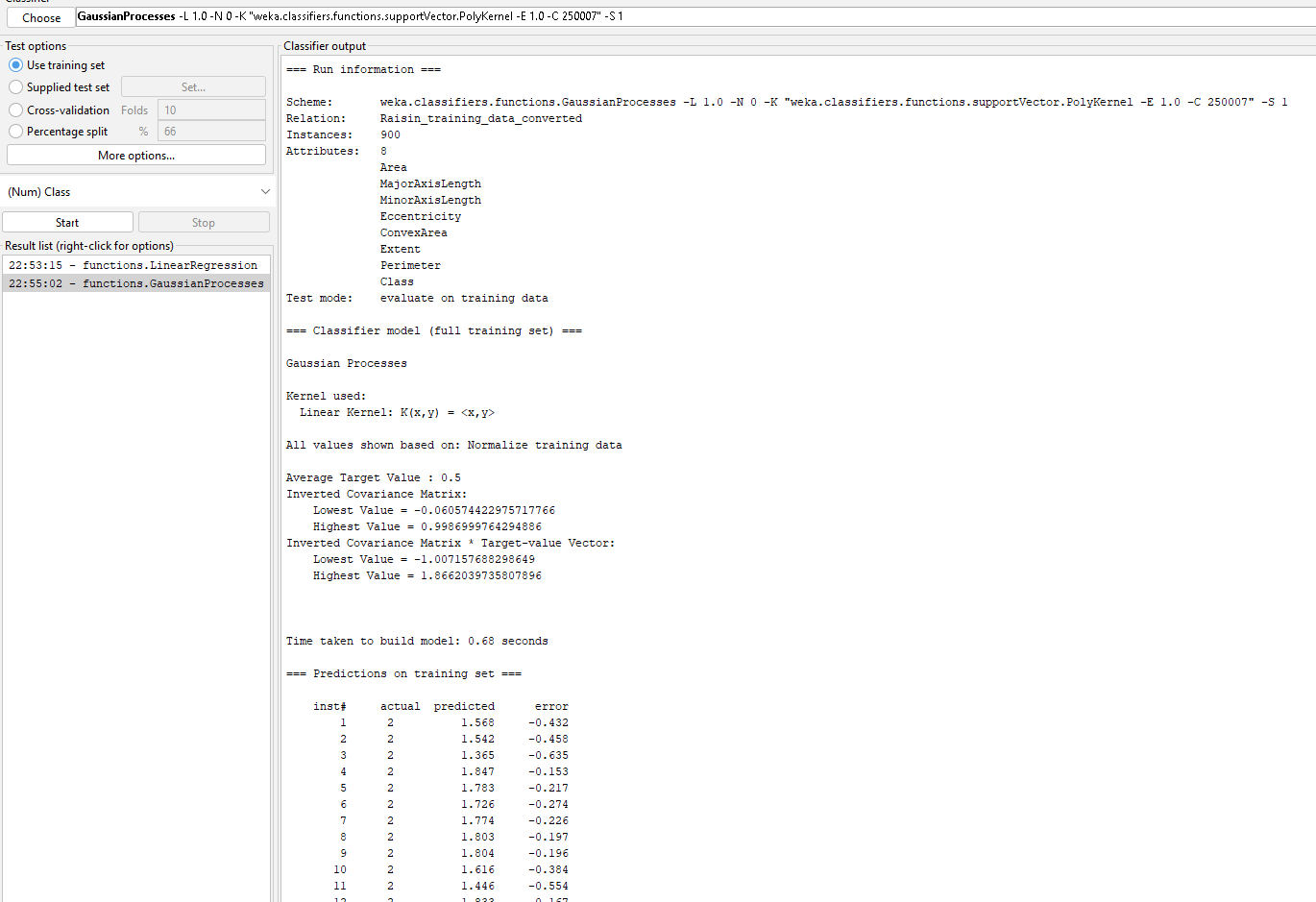
**Csv Test Data**



**Linear Regression on Train Data**



**Gaussian Processes on Train Data**



**Prediction with Linear Regression**

inst# actual predicted round-off r actual\_class predicted\_class

1 1 1.366 1 Besni Besni

2 2 1.559 2 Kecimen Kecimen

3 2 1.68 2 Kecimen Kecimen

4 1 1.608 2 Besni Kecimen

5 1 1.317 1 Besni Besni

6 1 0.977 1 Besni Besni

7 1 1.402 1 Besni Besni

8 2 2.092 2 Kecimen Kecimen

**Prediction with GaussianProcess**

inst# actual predicted roundoff actual\_class predicted\_class

1 1 1.437 1 Besni Besni

2 2 1.582 2 Kecimen Kecimen

3 2 1.54 1 Kecimen Besni

4 1 1.418 1 Besni Besni

5 1 1.42 1 Besni Besni

6 1 1.191 1 Besni Besni

7 2 1.355 1 Kecimen Besni

8 2 1.658 2 Kecimen Kecimen