Reza Mahroo EE7600: Machine Learning Project 3

In this project, the classification is implemented for the load digits data set and the Olivetti faces data set using the support vector machine and random forest classifier.

1. Implement the SVM for these classification problems.

The SVM is implemented for both datasets, and the Training error and testing errors are calculated as follows. The SVM parameters are set to default at this step.

For the load digits dataset:

Training Error is: 0.27 percent

Testing Error is: 0.55 percent

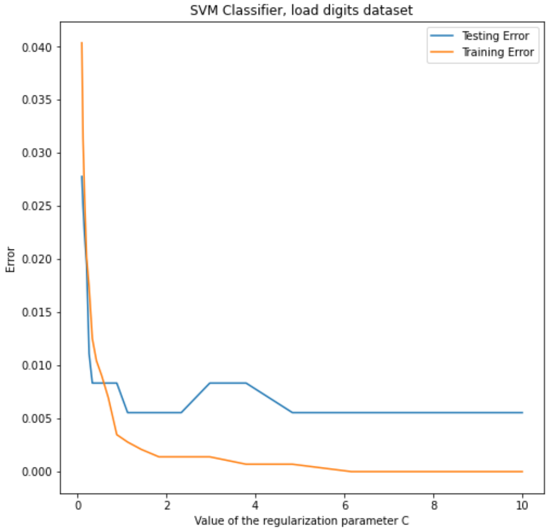
For the Olivetti faces dataset:

Training Error is: 0.31 percent

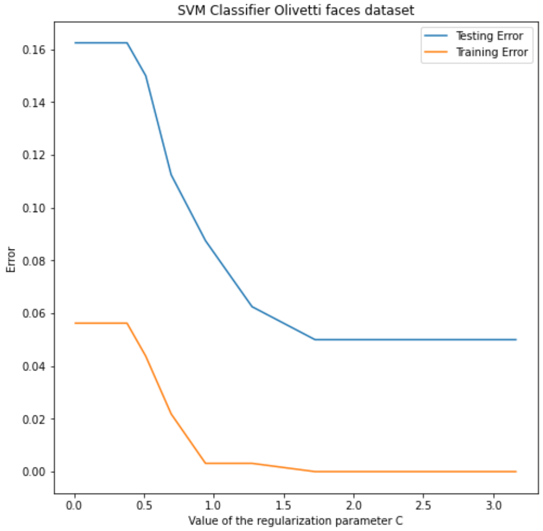
Testing Error is: 8.7 percent

2. plot the misclassification rate for both training and test sets vs. the value of the regularization parameter C.

For the load digits dataset: Parameter C changes from 0.1 to 10. It is obvious that when C increases, the training and testing error decrease, and it means we do not allow more misclassification by increasing the penalty term in the objective function.



For the Olivetti faces dataset: Parameter C changes from 0.01 to $\sqrt{10}$

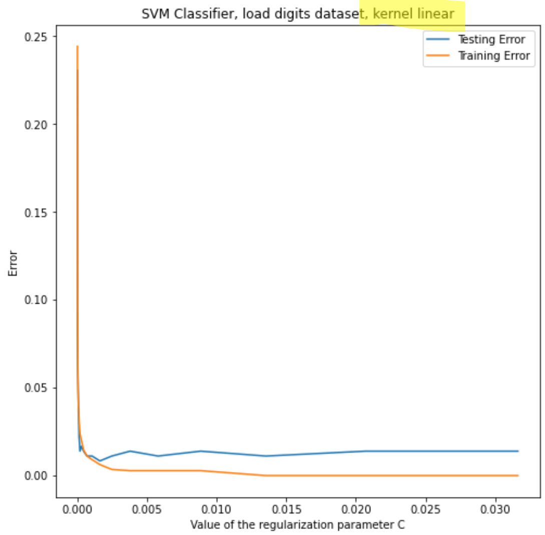


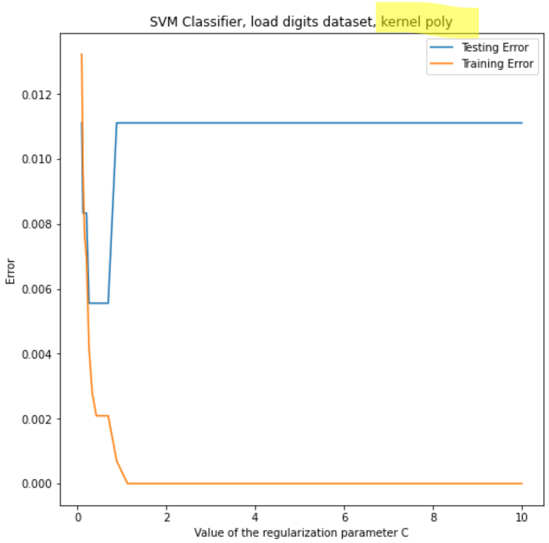
3. Find the misclassification rate for linear, polynomial, and RBF kernels.

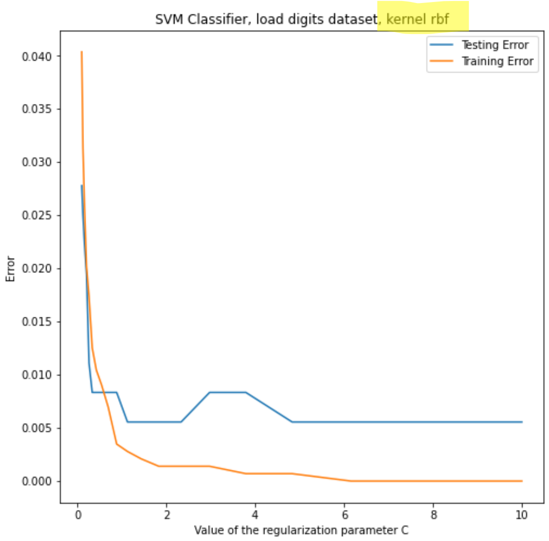
Everything is set to default here, but the kernel has changed, and the results are recorded.

		Linear	**Polynomial**	**RBF**
Load digits dataset	Training Error	0	0	0.27
	Testing Error	1.3	1.1	0.55
Olivetti faces dataset	Training Error	0	0	0.3
	Testing Error	2.5	2.5	8.7

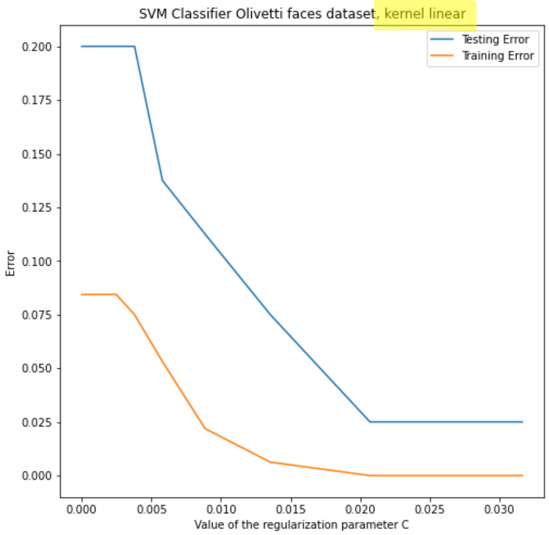
For the load digits dataset: Parameter C changes for every kernel.

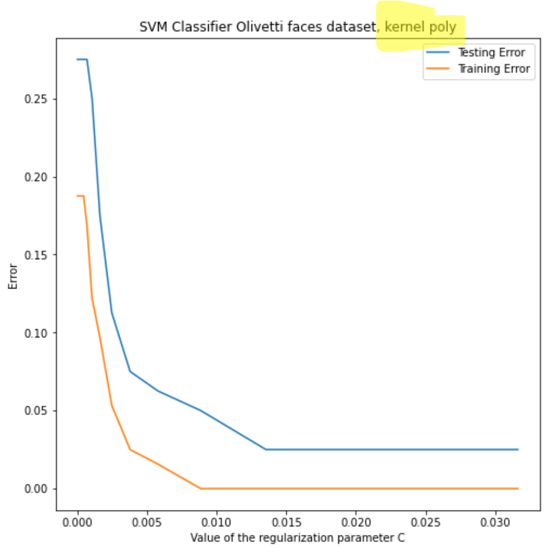


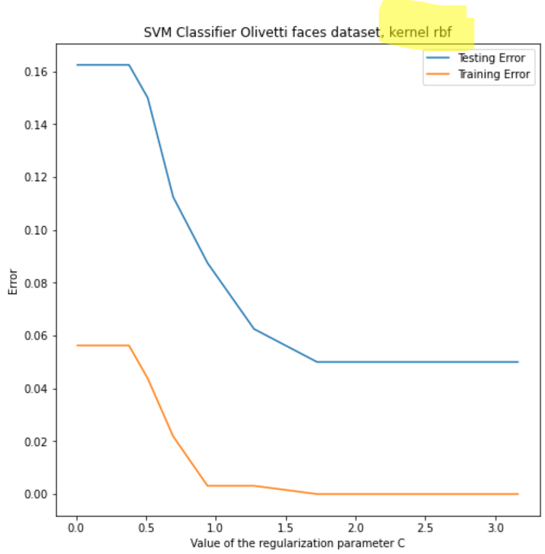




For the Olivetti faces dataset: Parameter C changes for every kernel.







1. Implement random forest classifier for these classification problems.

The random forest is implemented for both datasets, and the training error and testing errors are calculated as follows. The random forest parameters are set to default at this step.

For the load digits dataset:

Training Error is: 0 percent

Testing Error is: 1.1 percent

For the Olivetti faces dataset:

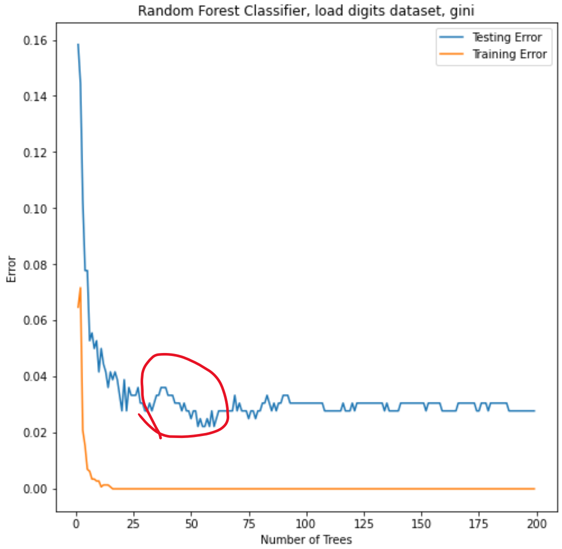
Training Error is: 0 percent

Testing Error is: 3.7 percent

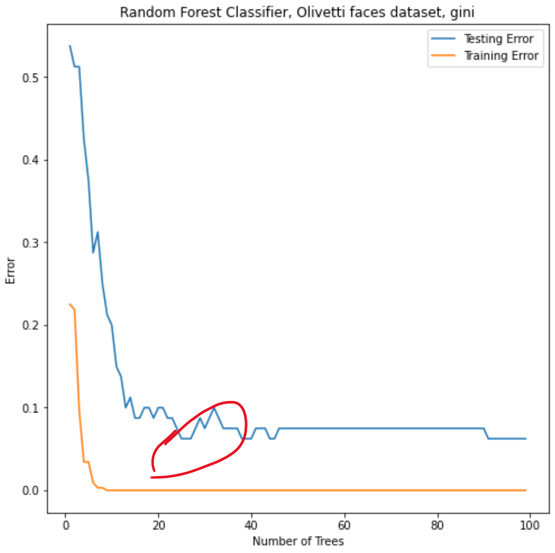
2. Plot the misclassification rate for both training and test sets vs. the number of trees.

Training errors settle down before the test error. The most favorable number of trees in the random forest is when the testing error is settling down (around the knee area of the testing error).

For the load digits dataset: The number of trees changes from 1 to 200.



For the Olivetti faces dataset: The number of trees changes from 1 to 100.



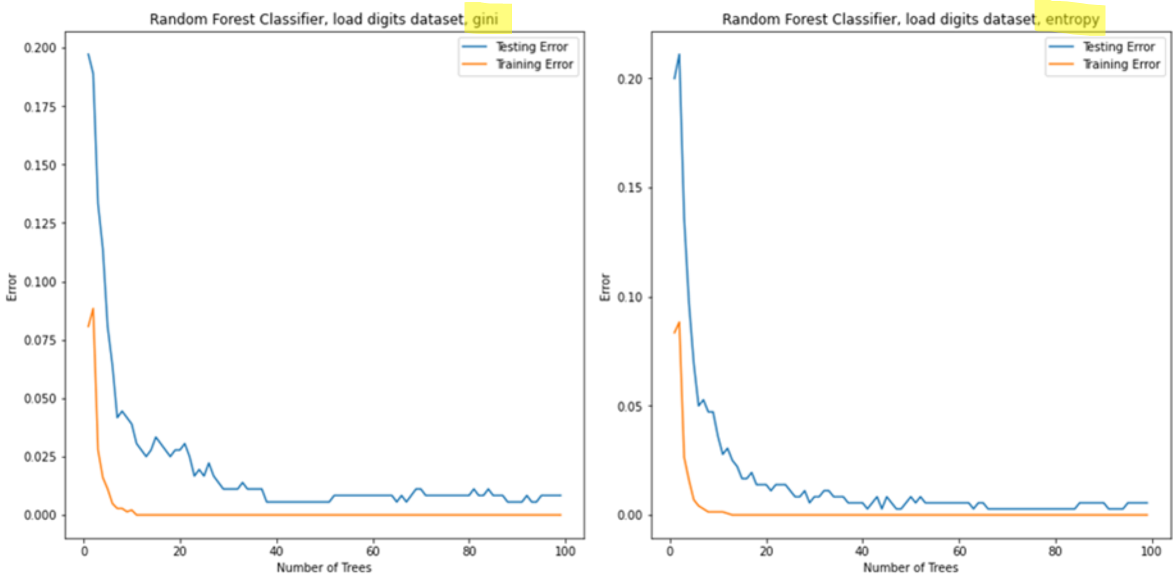
As the number of trees increases, both training and testing errors decrease.

3. Compare the performance for the following criteria.

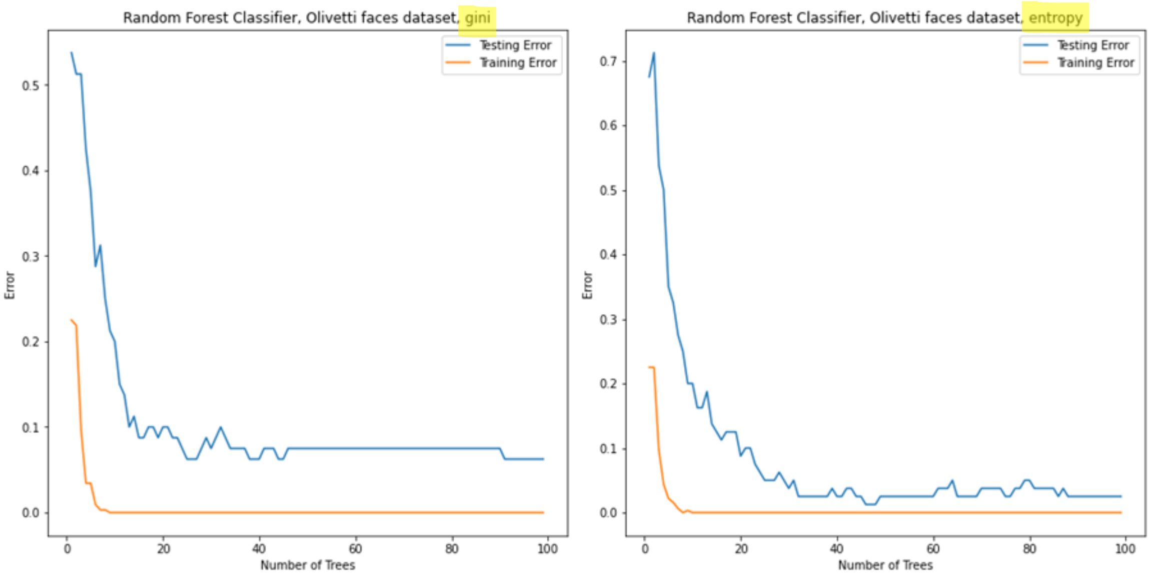
a) Gini vs entropy

The obtained results using the entropy criterion are slightly better. Computationally, entropy is more complex since it uses logarithms, and consequently, the calculation of the Gini Index will be faster.

For the load digits dataset: Everything is set to default, but the criterion changes.



For the Olivetti faces dataset: Everything is set to default, but the criterion changes.



b) Number of features selected for splitting (’sqrt’, ’log2’, ’None’).

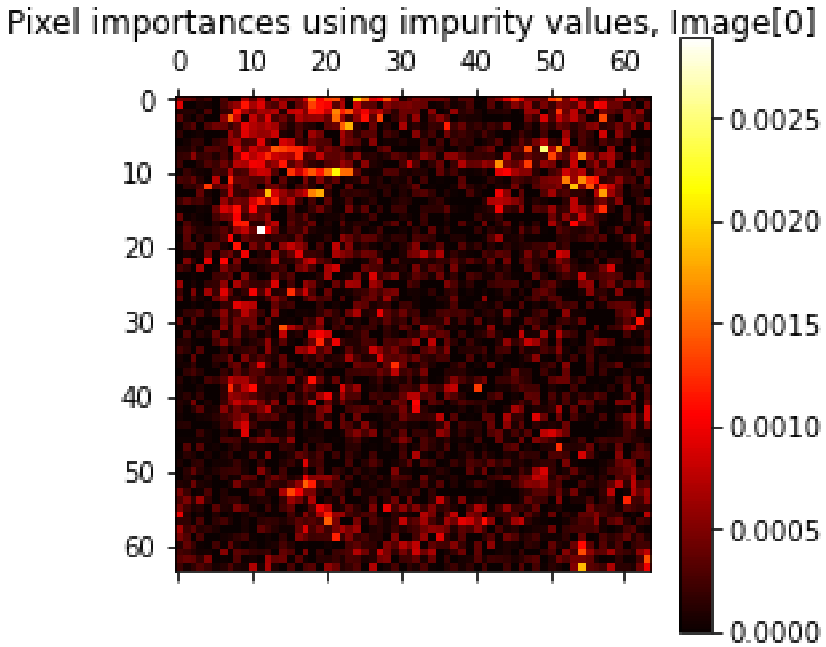
The number of selected features for splitting is an important index, especially when the number of features is high. It mainly makes a trade-off between accuracy and solution time.

In the load digits case, the obtained error for the splitting criteria (’($sqrt(n_{features})$’, $\log_2 n_{features}$’’, ’None’) is (0.8, 0.8, 3.3) percent. Since the number of all features is small (64 pixels), solution time does not show that much difference. The solution time is (0.34, 0.29, 1.23) seconds. In the load digits case, the obtained accuracy for the splitting criteria (’sqrt’, ’log2’, ’None’) is (0.8, 0.8, 3.3) percent.

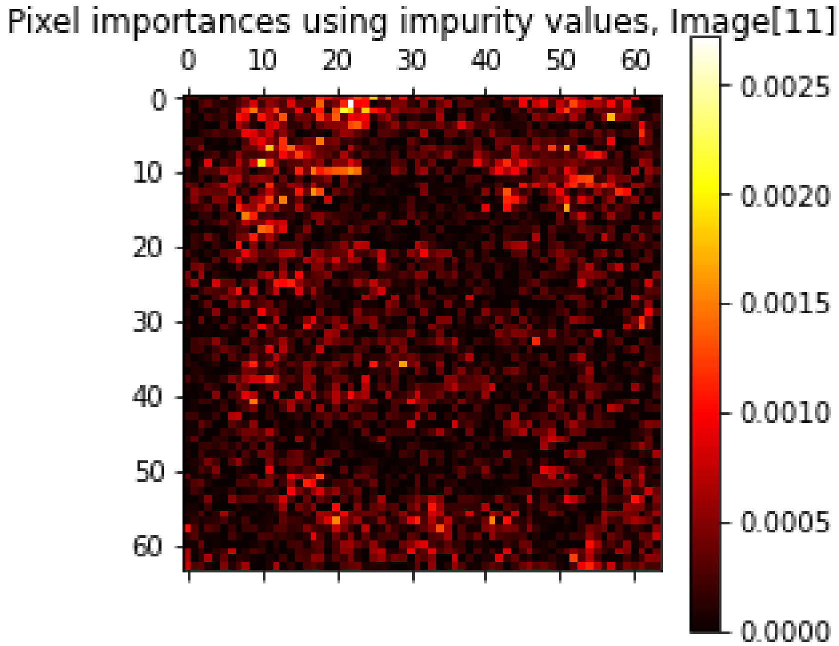
In the Olivetti faces dataset, the obtained error for the splitting criteria (’sqrt’, ’log2’, ’None’) is (3.7, 3.7, 8.7) percent, and the solution times are (2.29, 0.54, 142.7) seconds. In this case, since the number of features is high, 4096, the splitting criterion shows different performance (mainly in terms of solution time). Since the log2 criteria select fewer features (12 features), its solution time is less than others; however, its accuracy is comparable to others. On the other hand, if we do not put any criteria for splitting the features, the solution time significantly increases.

4. For a few images show the importance of the pixels in the image for the classification task (see the example in Scikit Learn).

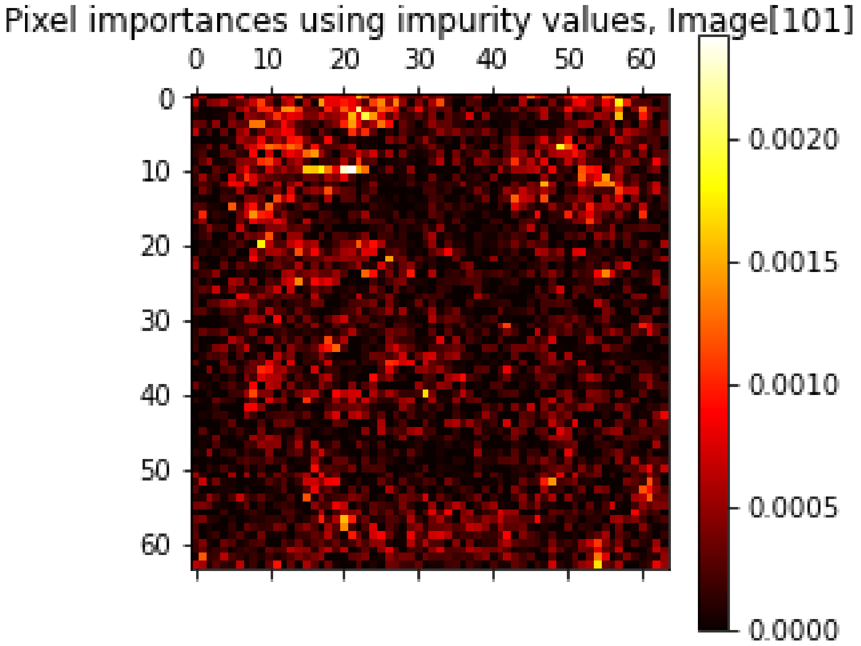
Feature importance is calculated based on the mean decrease in impurity. Pixels with a lighter color are more important. These



Pixel importances using impurity values, Image[0]



Pixel importances using impurity values, Image[11]


Pixel importances using impurity values, Image[101]
0
10
20
30
40
50
60
0
10
20
30
40
50
60
0.0020
0.0015
0.0010
0.0005
0.0000