LOTI.05.046 PATTERN RECOGNITION

HOMEWORK 8 - PCA LDA WITH DECISION TREE

May 14, 2019

Quazi Saimoon Islam
Msc. Robotics and Computer Engineering

1 INTRODUCTION

A decision tree is a flowchart structure where an internal node represents features, the branch represents a decision rule and each leaf node represents the outcome. It learns to partition on the basis of the attribute value. It partitions by recursive partitioning. This flowchart-like structure helps to make decisions that best suite the partitioning of the input features. It's visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.

In this assignment, a decision tree classifier was used to classify the Fisher data. This was then repeated to find performance when feature reduction was performed using PCA and LDA.

2 METHODOLOGY

The given methodology was followed and are as follows:

1. Split the data in 80-20% for training and test dataset respectively.
2. Perform Decision Tree classification on the training data and test it using the test data.
3. Evaluate the confusion matrix to compute the accuracy, precision, specificity and sensitivity.
4. Perform PCA to reduce the feature space dimensions and re-classify the data and test with appropriate dataset.
5. Similarly, evaluate confusion matrix to compute the accuracy, precision, specificity and sensitivity.
6. Perform LDA to reduce the feature space dimensions and re-classify the data and test with appropriate dataset.
7. Also evaluate confusion matrix to compute the accuracy, precision, specificity and sensitivity.

3 RESULTS

3.1 Classification using decision tree

Using appropriate tools from the Sklearn library, it was quite easy to split the data into a 80% training set and 20% test set to perform the classification. The Decision Tree evaluated on just the raw features is as portrayed below:

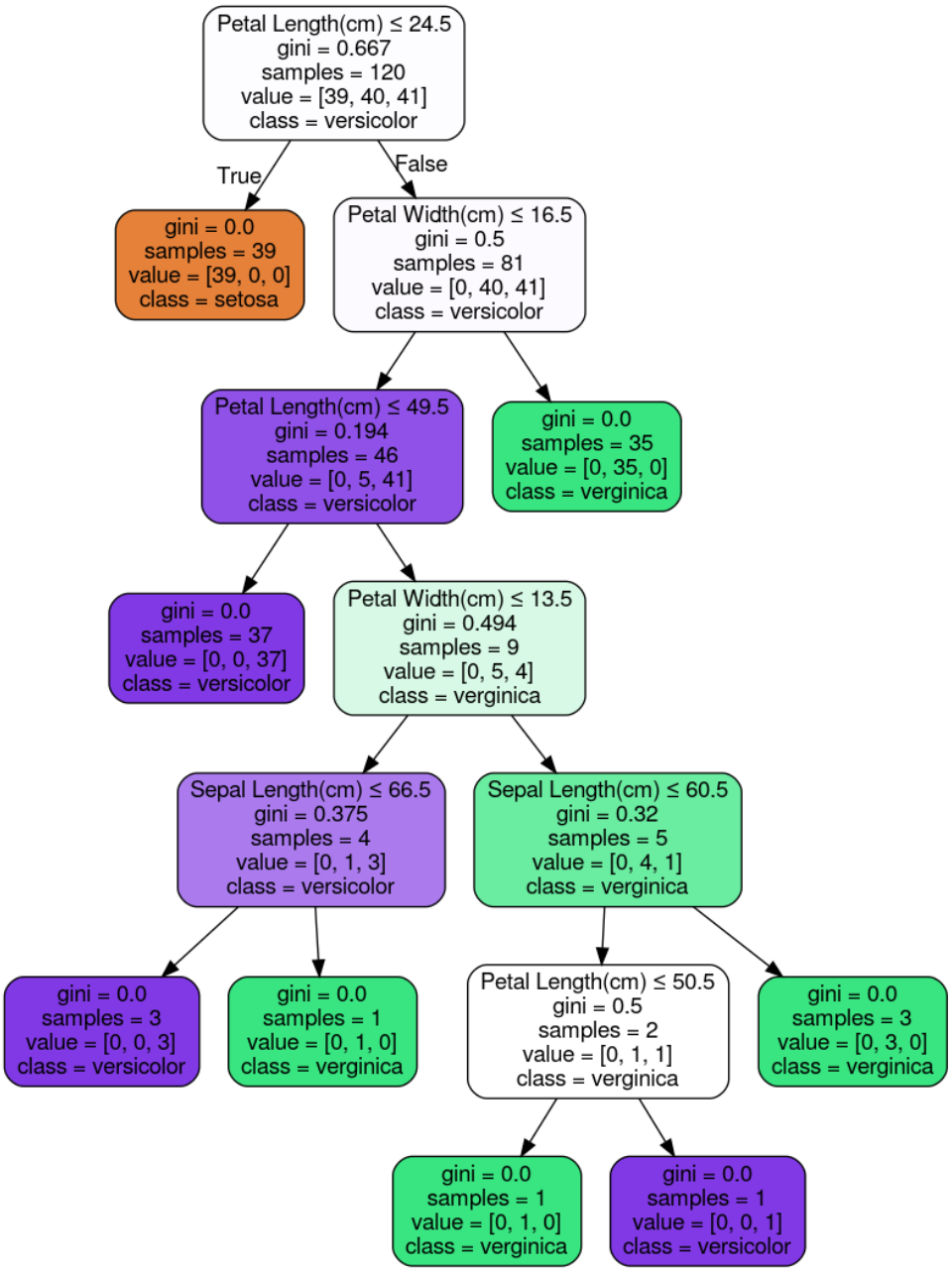

Figure 1: Decision Tree Classifier on Features only

As can be seen above, the input data is quite well classified using a rather short tree and correspondingly the following shows the evaluated confusion matrix based on the decision tree classifier alone.

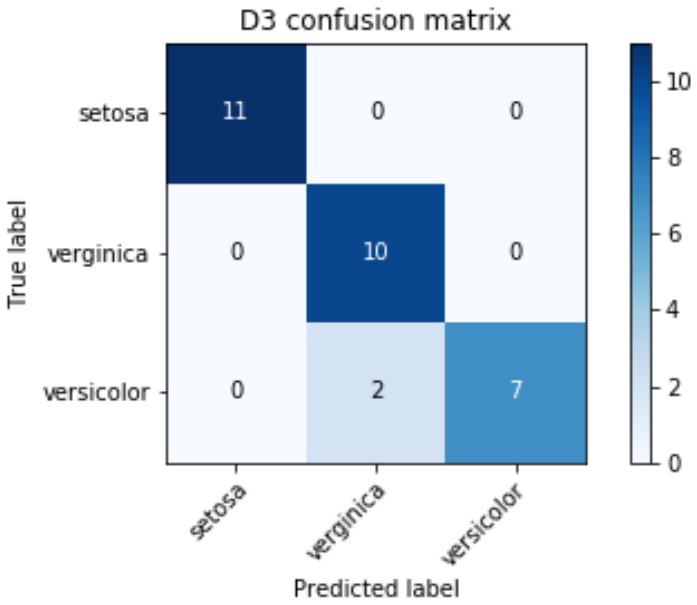

Figure 2: Confusion Matrix for Decision Tree Classifier

The confusion matrix can further be analyzed to compute the precision, specificity and sensitivity of the classifier. The accuracy score is pretty easy to compute as well either by comparing the predictions directly, or by using dedicated function in the Sklearn library.

1. The Decision Tree Classifier Accuracy is : 93.333333%
2. The Decision Tree Classifier Predicted Precision is: 94.444444%
3. The Decision Tree Classifier Predicted Specificity is: 96.666667%
4. The Decision Tree Classifier Predicted Sensitivity is: 92.592593%

3.2 Classification using Decision Tree with PCA

Similarly the Decision Tree can be used together with PCA to reduce the feature space. Doing so is quite rudimentary with Sklearn libraries but a good common practice is to scale the feature data before running PCA. The PCA was conducted taking only 2 components into effect. This was due to the fact that upon evaluation of the expected variance, it as found that

the highest variance lied within the first two principal components. This was done and the following implementation of the classifier, the following confusion matrix was generated (it should be noted that the same training data and test data were used for this classification as well):

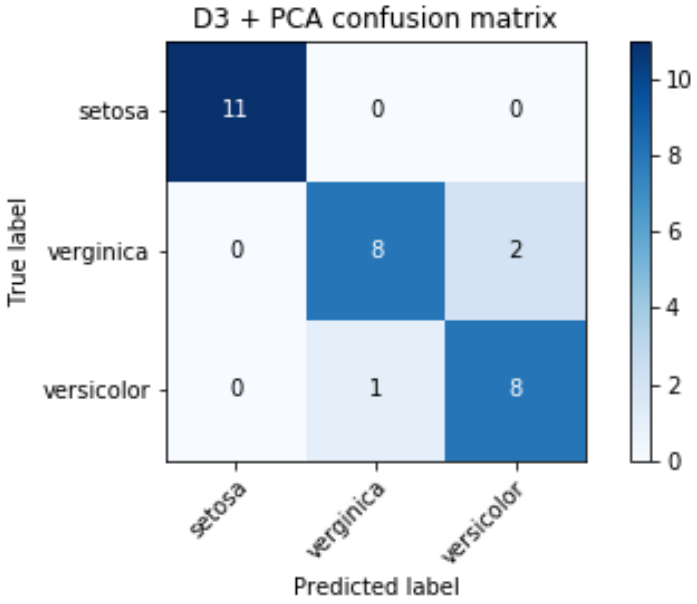

Figure 3: Confusion Matrix for Decision Tree Classifier with PCA

Similarly, the confusion matrix was used to evaluate the precision, specificity and sensitivity of the classifier with the following results:

1. The Decision Tree Classifier Accuracy with PCA is: 90.000000%
2. The PCA + Decision Tree Classifier Predicted Precision is: 89.629630%
3. The PCA + Decision Tree Classifier Predicted Specificity is: 95.158730%
4. The PCA + Decision Tree Classifier Predicted Sensitivity is: 89.629630%

3.3 Classification using Decision Tree with LDA

Finally, the Decision Tree classifier was conducted with LDA to reduce the feature space. Once again, only 2 principal components were used for this part as it performed the best. Following the training and testing, the follwing confusion matrix was obtained:

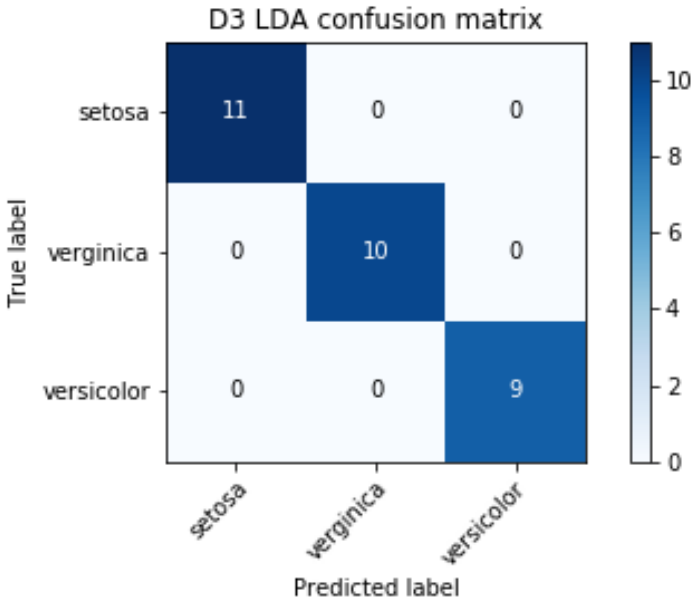

Figure 4: Confusion Matrix for Decision Tree Classifier with LDA

Similarly, the confusion matrix was used to evaluate the precision, specificity and sensitivity of the classifier with the following results:

1. The Decision Tree Classifier Accuracy with LDA is: 100.000000%
2. The Decision Tree Classifier Predicted Precision is: 100.000000%
3. The LDA + Decision Tree Classifier Predicted Specificity is: 100.000000%
4. The LDA + Decision Tree Classifier Predicted Sensitivity is: 100.000000

4 Discussion

It is quite evident from the results that the Decision Tree with LDA was the best classifier for the given data as it achieved the best accuracy, specificity, precision and sensitivity. Surprisingly, just a bare bones decision tree classifier performed better than the implementation with PCA, and that could be down to small variances in the data in non-principal components. However, the test procedure here was not perfect, as the same training and test data was used for all three. The Random State selected for the data splitting also seems to have a strong effect on the performance, leading one to believe that the classifier is definitely sensitive to some data, but not as much to others.

REFERENCES

[1] Sci-Kit Learn Documentation *sklearn.decomposition.PCA* [Online]. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html> [Accessed 3 May 2019].

[2] Sci-Kit Learn Documentation *1.10. Decision Trees* [Online]. Available at: <https://scikit-learn.org/stable/modules/tree.html#decision-trees> [Accessed 3 May 2019].

[3] Stack Abuse *Implementing PCA in Python with Scikit-Learn* [online]. Available at: <https://stackabuse.com/implementing-pca-in-python-with-scikit-learn/> [Accessed 3 May 2019].