Comparing Scikit-learn Classifiers with Local Feature Selector Classifier

Chukwuebuka Amaefula

August 2020

Department of Computer Science

Memorial University of Newfoundland

1 Datasource

Given samples of two species of Iris (Iris setosa, Iris virginica) I will compare the binary classification performance of three scikit-learn classifier and third party classification algorithms:

- K-Nearest Neighbor
- Gaussian Bayes
- Decision Tree
- Local Feature Selection (LFS)

The dataset is publicly available at `https://owlya.s3.us-east-2.amazonaws.com/iris.csv`

1.1 Attribute Information

- Sepal length
- Sepal width
- Petal length
- Petal width
- Species

Dataset Descripion

	sepal_length	sepal_width	petal_length	petal_width
count	100.0	100.0	100.0	100.0
mean	5.471000000000001	3.094	2.8620000000000005	0.7849999999999998
std	0.6416983463254116	0.4760570375809006	1.4485645977792616	0.5662877521029553
min	4.3	2.0	1.0	0.1
25%	5.0	2.8	1.5	0.2
50%	5.4	3.05	2.45	0.8
75%	5.9	3.4	4.324999999999999	1.3
max	7.0	4.4	5.1	1.8

Figure 1: Dataset description after running df.describe()

2 Workflow Analysis

For the classification problem I compared four algorithms.

2.1 K-Nearest Neighbor

k-NN is a non-parametric classification method, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor[4]

2.2 Gaussian Bayes

Gaussian Bayes classifier belongs to the Naive Bayes family that uses a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set[3]

2.3 Decision Tree

Decision tree learning is one of the predictive modelling approaches used in statistics, data mining and machine learning. It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves).[2]

2.4 Local Feature Selection (LFS)

Localized feature selection (LFS) is the third party classifier, it is a supervised machine learning approach for embedding localized feature selection in classification. The sample space is partitioned into overlapping regions, and subsets of features are selected that are optimal for classification within each local region. As the size and membership of the feature subsets can vary across regions, LFS is able to adapt to local variation across the entire sample space[1].

3 Preprocessing Steps

3.1 Conversion to binary dataset

The first preprocessing step was to convert our dataset from a multiclass dataset to a binary dataset. We selected only two kinds of species from the provided 3.

```
clean_df = df.query('species_==”setosa”_or_species==”versicolor”')
```

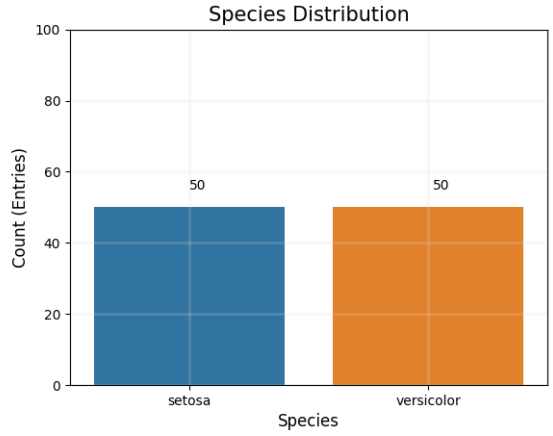

Figure 2: Species Distribution

3.2 Data splitting into training and test set

I splitted the dataset into training and dataset with the ratio of 8:2. Below figure show the specie distribution when plot against Sepal Length and Width

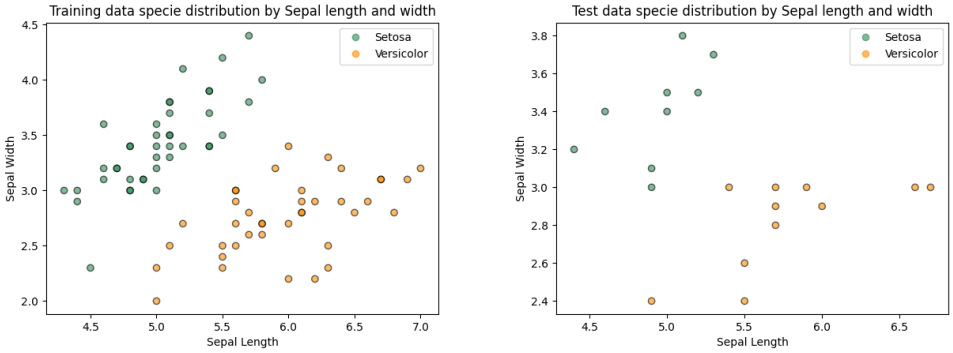

Figure 3: Training an Test dataset

4 Comparison

Having been able to make the third party software work, I proceeded to the main problem which was to compare the performance of each classifier on our new binary dataset.

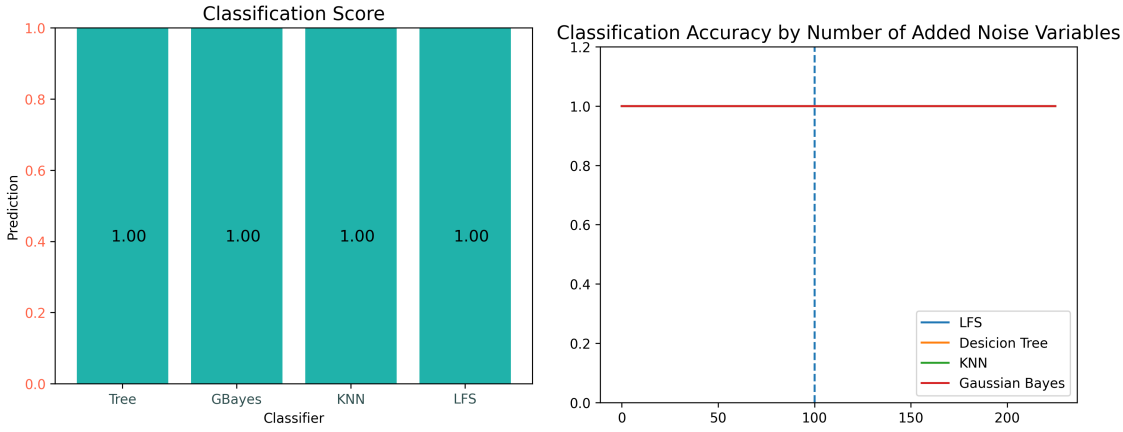

Figure 4: Comparison

The four classifiers performed brilliantly on the clean dataset. I went ahead to add noise to the dataset do another comparison. Despite the introduction of noise the four classifiers perform equally well.

5 Conclusion

As much as LFS did perform well in this comparison, I noticed that it only works with certain dataset and can only perform binary classification. For such a promising package these limitations would negatively affect how far it could go.

References

[1] Dhindsa et al., (2020). LFSpy: A Python Implementation of Local Feature Selection for Data Classification with scikit-learn Compatibility. Journal of Open Source Software, 5(49), 1958, https://doi.org/10.21105/joss.01958

[2] @MISC title = Decision tree learning, howpublished=`https://en.wikipedia.org/wiki/Decision_tree_learning`

[3] @MISC title = Naive Bayes classifier, howpublished=`https://en.wikipedia.org/wiki/Naive_Bayes_classifier#Gaussian_na%C3%AFve_Bayes`

[4] @MISC title = k-nearest neighbors algorithm, howpublished=`https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm`