Assignment 1 – Data Mining

Jugal Chauhan 0755856

Introduction

In machine learning classification is a prevalent method of identifying objects from a pool of predefined categories. It is a supervised learning which is applied to predict a categorical response variable. Classification algorithms are used for many different applications, usually to separate or detect specific kind of data from the original dataset. It has applications in healthcare, fraud detection, speech recognition, image classification etc. In this report four different classification algorithms are discussed on Iris dataset to classify the species of Iris flower.

Dataset Description

Iris dataset consists of 6 attributes and 150 records, it describes the length and width of Sepal and Petal of a species of Iris flower. The species attribute, which is the target variable, consists of 3 labels ‘Virginica’, ‘Setosa’, and ‘Versicolor’ species. Classification algorithms will be trained using the length and width attributes of sepal and petal to accurately distinguish between the species of Iris data. A detailed dataset summary is provided in Table 1.

Sr No	Name	Description	Data Type	Non-Null Count	Mean	Standard Deviation
1	Id	Index	int	150	-	-
2	SepalLengthCm	Length of Sepal in cm	float	150	5.84	0.82
3	SepalWidthCm	Width of Sepal in cm	float	150	3.05	0.43
4	PetalLengthCm	Length of Petal in cm	float	150	3.75	1.76
5	PetalWidthCm	Width of Petal in cm	float	150	1.19	0.76
6	Species	Name of the Species	object	150	-	-

Table 1: Brief Overview of Iris Dataset

The dataset consists of 150 records with 50 records for each species, this balanced nature of the Iris dataset makes it suitable training and testing classification algorithms. The dataset is widely used in statistics and machine learning, and it provides a very good basis to discuss classification algorithms.

Method

To begin with data preprocessing, the first step is to check for null values in the data. Null values can affect the accuracy of classification algorithms and needs to properly addressed before conducting the analysis. Luckily, in Iris dataset there are no null values and the data is well balanced. The second step involves checking for duplicate values, three duplicated rows are found in the dataset (Row 34, 37, and 142). Ideally duplicate values should not be considered in the analysis, however, given that there are only three of them and removing it can cause the data to be unbalanced, the duplicated rows are retained in dataset. In the next step any presence of outliers in data is investigated using box plots in Figure 1.

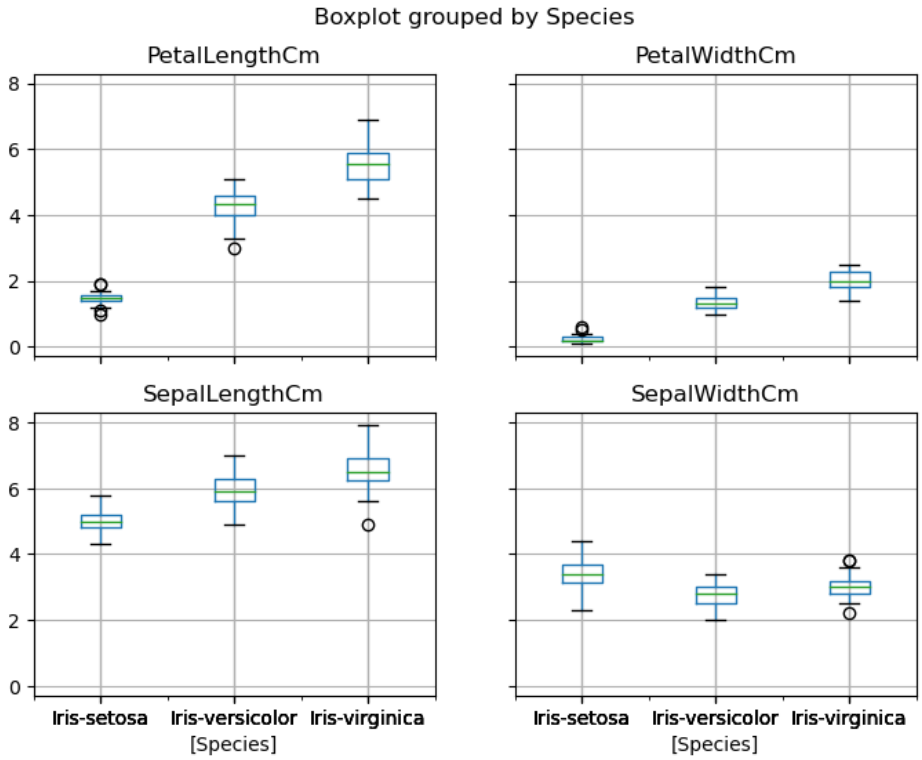


Figure 1: Box Plot Grouped by Species

The boxplot suggests there are few outliers in each of the three class labels, however, the outliers are not extreme, and again removing them would unbalance the dataset. Additionally, to understand the distribution of data, violin chart is plotted in Figure 2. It can

be observed from the violin plots that distribution of data for attributes PetalLengthCm and PetalWidthCm clearly differentiates Setosa class label from other two labels.

Data standardization is not considered as the data is not largely varied and is under the same unit, thus we want the original scale of the data to be preserved for classification algorithms. Finally, a train-test split is performed using python's scikit-learn module, the 'Id' column in not included in this data. With this, the preprocessing steps are concluded and the data is ready to be used for classification. Four classification algorithms discussed are: 1. Decision Trees 2. Support Vector Machine 3. K Nearest Neighbour and 4. Random Forest. The result section discusses the performance of each algorithm.

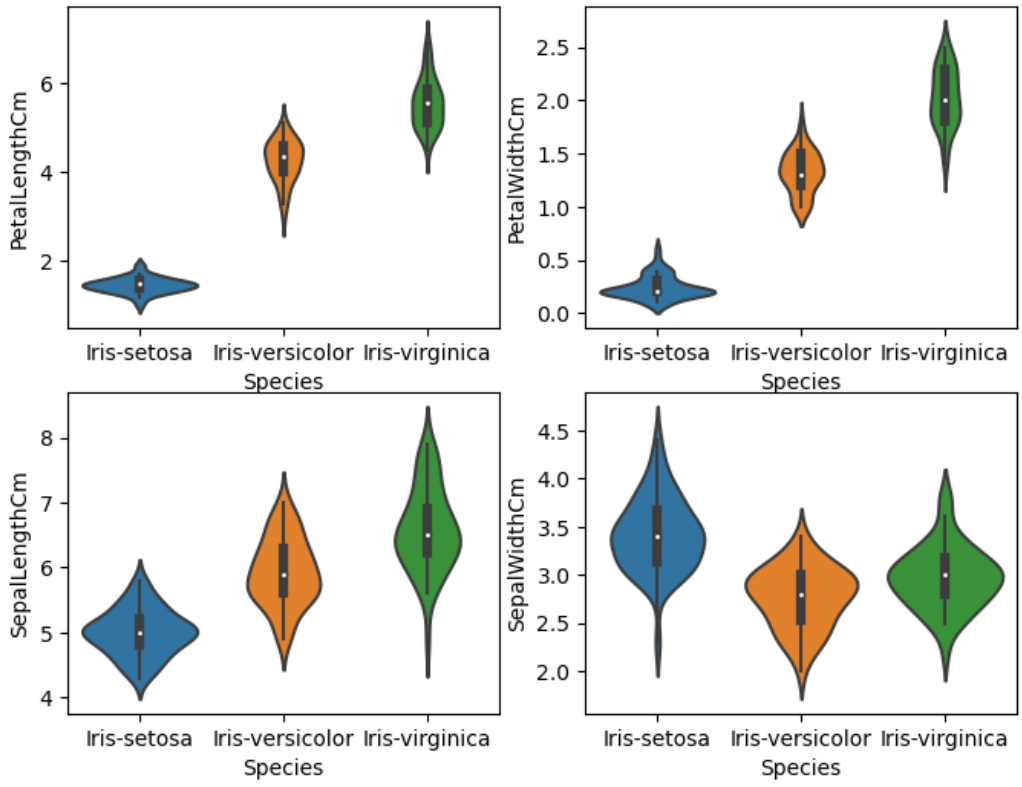


Figure 2: Observing distribution of attributes using violin plot

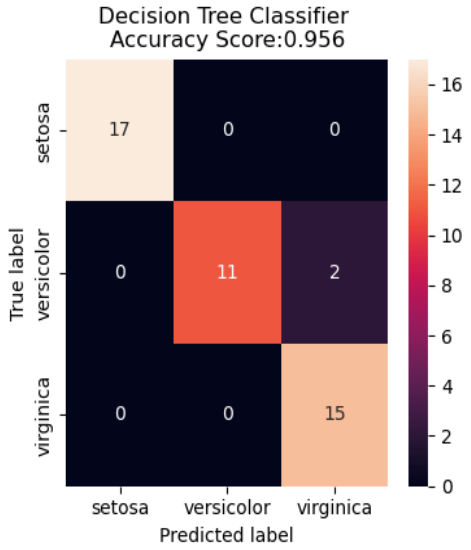
Results

1) Decision Trees: In this case the decision tree algorithm used is from scikit-learn module which include regression or classification trees. Decision tree is a flowchart like structure that breaks the data into smaller units until its successfully classifies a class label. The decision of splitting the data is based off attribute value.

   In case of Iris data, the decision tree classifier produces an accuracy of 0.956, that is, 95.6% of test data instances were correctly classified. The precision score of 1 means

each true positive class label was correctly identified by the classifier. Similarly, recall score means how many true positive values are captured, it is one if every true positive is identified by the classifier.

Now, by looking at the confusion matrix of decision tree classifier (Figure 3), it is observed that it wrongly classifies two instances of the true label 'Versicolor' species as 'Virginica' species. This is reflected in the recall and precision scores of respective classes from Table 2.



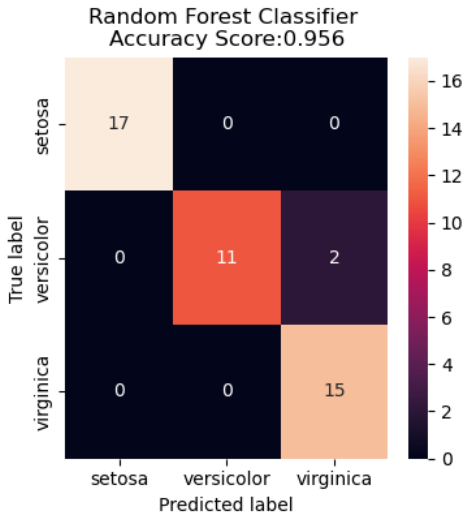
	Precision	Recall	F1-score	support
Iris-setosa	1.00	1.00	1.00	17
Iris-versicolor	1.00	0.85	0.92	13
Iris-virginica	0.88	1.00	0.94	15
accuracy			0.96	45
Macro avg	0.96	0.95	0.95	45
Weighted avg	0.96	0.96	0.96	45

Figure 3: Confusion Matrix for Decision Tree Classifier

Table 2: Classification Report for Decision Tree

2) Random Forest Classification: An improvement on decision tree algorithm is the Random Forest algorithm that employs multiple decision trees using random subsets of data. The final prediction is made by taking the most popular result.

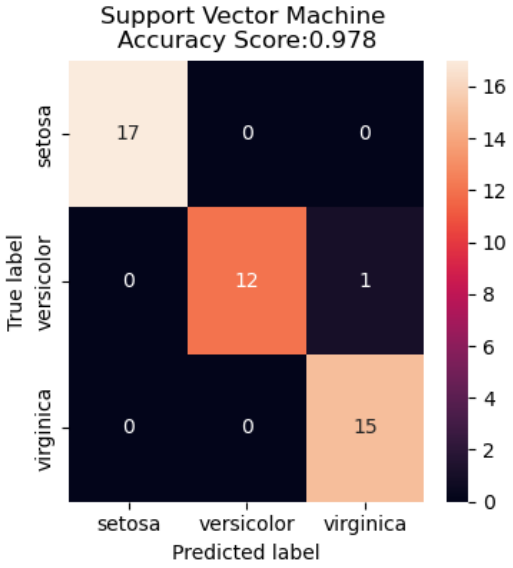
Random Forest Classifier is implemented on the Iris dataset and the result obtained is similar to that of decision tree classifier. The accuracy score obtained is 0.956 and the classification report states that the Random Forest algorithm does not improve on the Decision Tree classifier.



	Precision	Recall	F1-score	support
Iris-setosa	1.00	1.00	1.00	17
Iris-versicolor	1.00	0.85	0.92	13
Iris-virginica	0.88	1.00	0.94	15
accuracy			0.96	45
Macro avg	0.96	0.95	0.95	45
Weighted avg	0.96	0.96	0.96	45

Figure 4: Confusion Matrix for Random Forest Algorithm

3) Support Vector Machine: This classification algorithm separates the data using a hyperplane that contains the largest margin. It is also known as a discriminative classifier. The base idea of this algorithm is to obtain Maximum Marginal Hyperplane that gives the best division of data.



	Precision	Recall	F1-score	support
Iris-setosa	1.00	1.00	1.00	17
Iris-versicolor	1.00	0.92	0.96	13
Iris-virginica	0.94	1.00	0.97	15
accuracy			0.98	45
Macro avg	0.98	0.97	0.98	45
Weighted avg	0.98	0.98	0.98	45

Figure 5: Confusion Matrix for Support Vector Machine

Table 4: Classification Report for Support Vector Machine

As seen in the confusion matrix, SVM improves on the previous two classification algorithm and provides an accuracy score of 0.978. The precision and recall scores are also improved. Only one class label is incorrectly identified by SVM.

SVM performs better on the iris data because of the clear difference in distribution of class labels in PentalLengthCm and PetalWidthCm attributes. SVM is able to effectively separate data points located on different hyperplanes and effectively classify the class labels.

Discussion

The highest accuracy score obtained is 0.978 by SVM, the model can be considered successful as it can identify about 97% of test data accurately. While Decision Trees and Random Forest algorithm also provided about 95% accuracy, from the classification reports it can be said that the algorithm most suited for Iris data is SVM.

There are several reasons why SVM is better that the other classifier one of which is already discussed, and can be elaborated further with the help of a pair plot. By looking at the scatterplot for PetalLengthCm vs PetalWidthCm we can see that class labels are easily separable from each other, they almost lay on different hyperplane. This kind of distribution suits to SVM and hence it is more effective than Decision Tree and Random Forest algorithm.

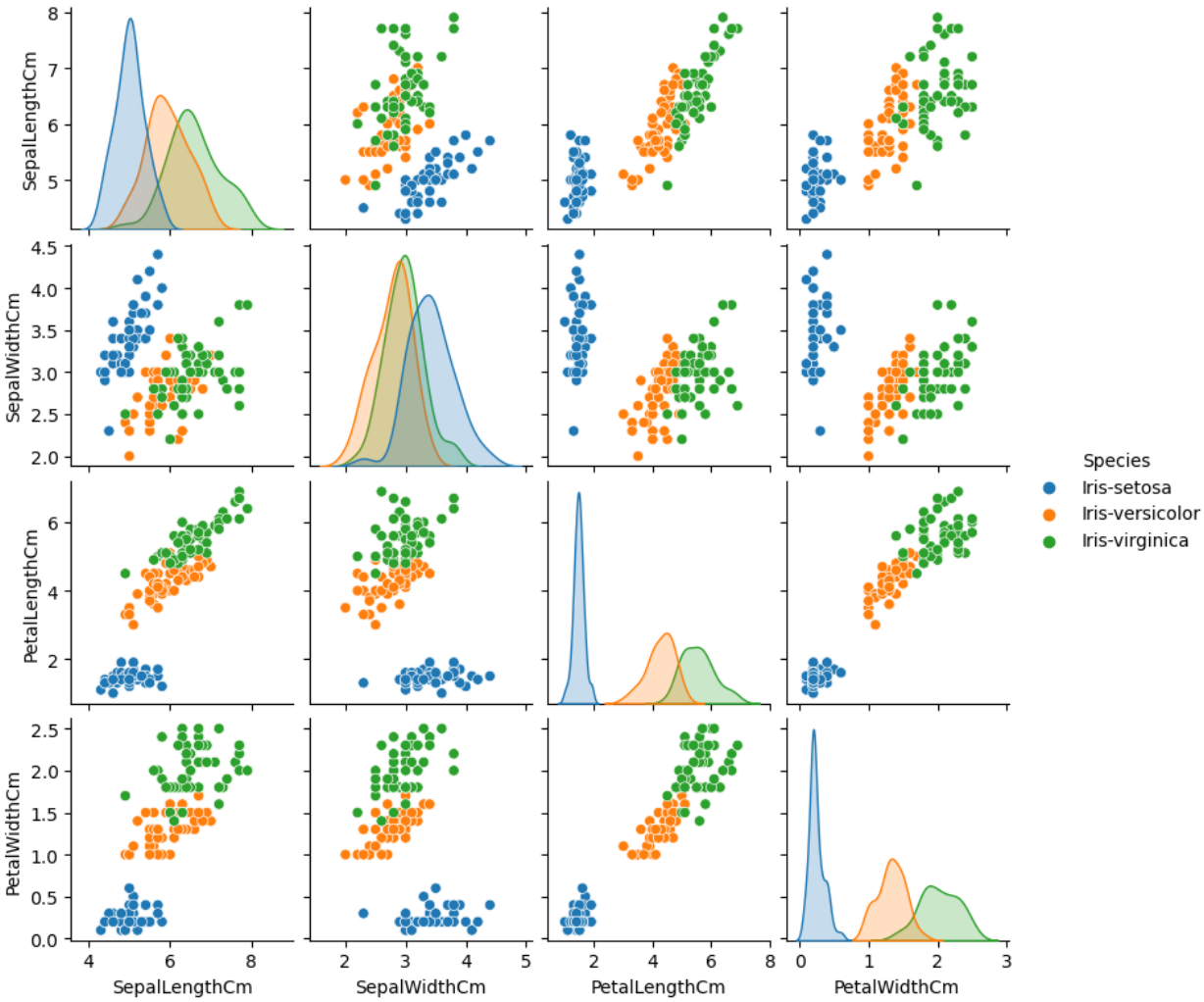


Figure 6: Pair plot for Iris data

Conclusion

Three different classification algorithms are discussed and their performance is evaluated on the Iris dataset. All three algorithms accurately predict the test data with SVM having the highest accuracy of 98%. Further evaluation metrics like precision, recall and f1-scores are used to test the performance of each algorithm. It is also found that attributes PetalLengthCm and PetalWidthCm attributes provide more accurate classification of class labels than other attributes.