CS521 Assignment 2: Data Analytics with the Fisher Iris Dataset

Ashwin Goyal
2018meb1214@iitrpr.ac.in

Indian Institute of Technology
Ropar, Punjab

Abstract

This assignment is based on the famous **Fisher Iris dataset** which includes 3 classes each corresponding to the species of the Iris flower(Iris Setosa, Iris Virginica and Iris Versicolor). This dataset has 4 features, namely sepal width, sepal length, petal width, petal length, has 150 datapoints and the labels corresponding to each datapoint.

1 Introduction

In this assignment, I have carried out some exploratory data analysis, followed by classification of data using both supervised(Gaussian Naïve Bayes and logistic regression) and unsupervised(K-Means clustering) learning algorithms. All this has given me a deep insight into the dataset and has helped me visualise and understand the essence of data science through using real world datasets. Also it has helped me compare the use and performance of the 3 models.

1.1 Importing libraries, loading the dataset and Data preprocessing

Libraries imported are, *Numpy, Pandas, Seaborn, and Matplotlib, SKlearn*. After imported the libraries, I've imported the Iris dataset**and created a Pandas dataframe wherein, I have set the columns as the feature names and the last column as the labels**. The dataframe is shown in Figure 1.

After this, I've preprocessed the dataset through scaling the features. This is done by dividing by the standard deviation and subtracting the mean for each data point. I've even checked if some values in the dataset are missing, using df.isnull().sum(), and this shows that there is no missing value.

1.2 Splitting the data into testing and training sets

I have split data in an 80-20 fashion (80 % of the data for training set and 20 % data for testing set). I have used a five-fold cross validation technique for each of my models. For this I had to import KFold function from model selection package of sklearn library. This technique means splitting the dataset into five parts, and using four parts for training and one for testing.

© 2020. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

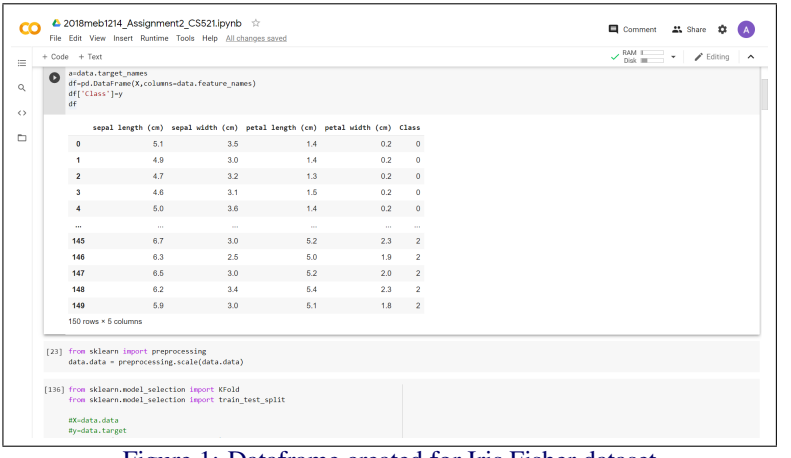

Figure 1: Dataframe created for Iris Fisher dataset

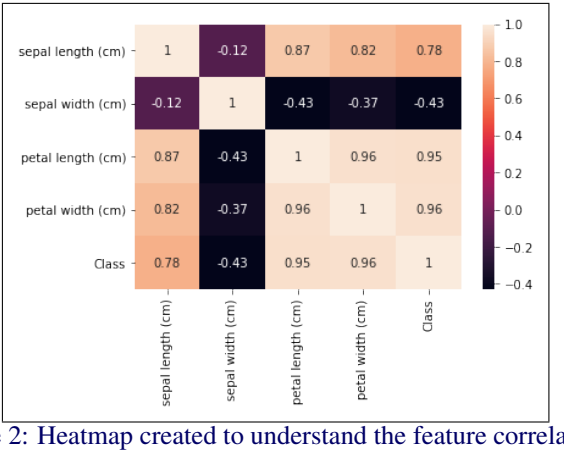

Figure 2: Heatmap created to understand the feature correlations

2 DATA EXPLORATION

2.1 Data Visualization

To begin with, I have plotted a heatmap of the correlations of my features. I have rounded the correlation of each pair of the features to 2 decimal places and created a correlation matrix which has been shown through heatmap (using in built functionality of Seaborn). Through this heatmap, I was able to identify that petal length and petal width have high correlation (0.95 and 0.96 respectively) with the class(species) label. The heatmap is shown in figure 2.

Next, as asked in the assignment, the distribution of Sepal Width (y) by grouping Sepal Length (x) has been plotted using matplotlib functionality. We can notice from this plot that sepal width is maximum for flowers having a mean sepal length of 5.5. We have close to 8 as the maximum sepal length and around 4 as the minimum. The maximum sepal width is around 4.5. On plotting a scatter plot, **if we colour the points according the their species, we'll notice that in general, the maximum sepal width is of Iris Setosa (shown in dark**

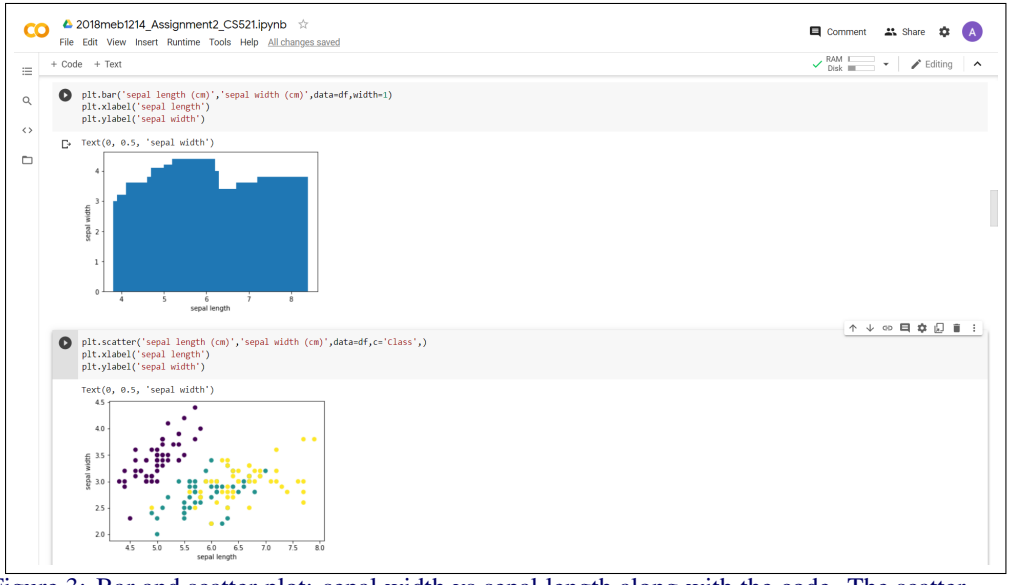

Figure 3: Bar and scatter plot: sepal width vs sepal length along with the code. The scatter plot is coloured according to the different species.

blue in scatter plot in figure 3), and sepal length of Virginica is maximum). ***(Figure 3)*** **Also, I've plotted a joinplot histogram of sepal width vs sepal length with around 10 bins.** *(Figure 4)*

This inference is reaffirmed through plotting the Facet Grid plots of sepal width and sepal length using Seaborn. Also I have plotted a bar plot of sepal width vs sepal length for easy understanding of the sepal data(length and width) with the hue as the species. The corresponding figure is figure 5.

Similarly, on plotting the Petal Length (x) and Petal Width (y) using a scatter plot and coloring based to class(species), we notice that in general Iris Virginica(shown in yellow in graph) has the maximum petal length as well as petal width whereas, Iris Setosa has both minimum. *(refer Figure 6)*

A very useful plot is the boxplot which has been plotted next. The outliers are represented as hollow circles and are present only in case of sepal width. This plot tells me that *the interquartile range for petal length is the maximum and the median of sepal length is the maximum among all 4 features.* (Figure 7)

[EXTRA] Finally, I've plotted a pairplot to represent the entire dataset into one, very useful graph with coloring based on the three different classes. *(refer Colab Notebook)*

3 DATA CLASSIFICATION

3.1 Supervised Learning

We have employed two supervised learning techniques here, to classify the data into the 3 classes (species of Iris flower).

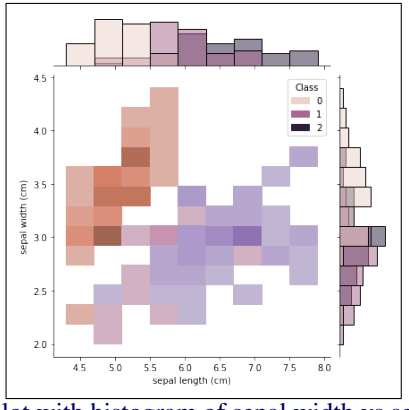

Figure 4: Jointplot with histogram of sepal width vs sepal length (cm)

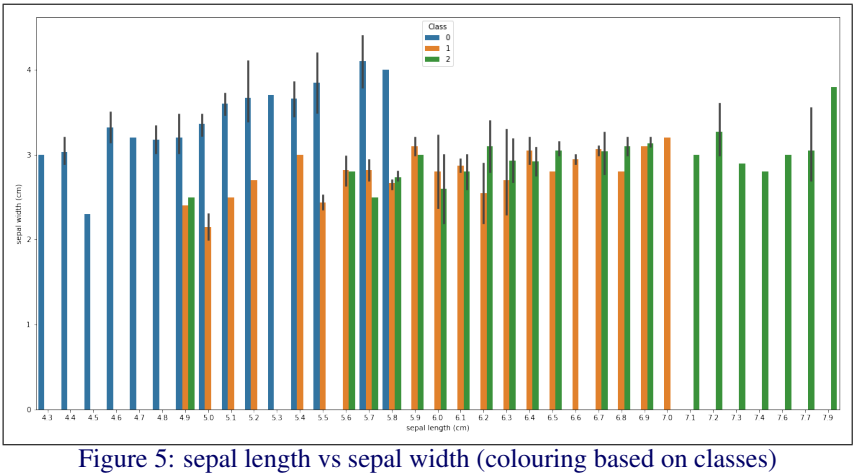

Figure 5: sepal length vs sepal width (colouring based on classes)

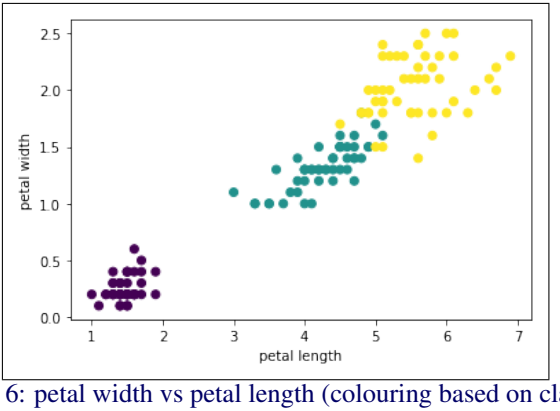

Figure 6: petal width vs petal length (colouring based on classes)

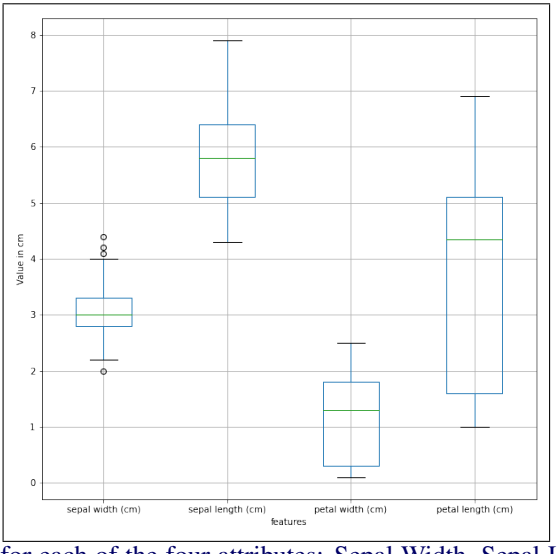

Figure 7: Boxplots for each of the four attributes: Sepal Width, Sepal Length, Petal Width and Petal Length.

	Predicted setosa	Predicted versicolor	Predicted Virginica
Real setosa	8	0	0
Real versicolor	0	10	1
Real virginica	0	0	11

Table 1: Confusion Matrix for Logistic Regression model

1. ***Logistic Regression***

2. ***Gaussian Naïve Bayes***

3.1.1 Logistic Regression

I have used the Logistic Regression model (imported from the linear model package of sklearn) on the train and test data. I have fitted on the training set and predicted for the test set and obtained good results. These results are: **Classification report:** *Average precision, Recall and F1-Score are all*: =0.9733 **Confusion Matrix: [[8 0 0] [0 10 1] [0 0 11]]** (Table 1 above)

As we can see from this matrix, only 1 result has been classified wrongly based on the test set (1 Versicolor flower has been predicted as Virginica), out of 30 data points in the testing set.

After this, for cross validation, I have used 5 fold cross validation using KFold Cross Validation. This has been implemented by me using cross_val_score functionality of model_selection package. In this functionality, I've done the scoring using the **accuracy** measure. The results are impressive, with the resulting **Cross validation mean as 0.966 and Cross validation standard deviation as 0.016**.

	Predicted setosa	Predicted versicolor	Predicted Virginica
Real setosa	8	0	0
Real versicolor	0	10	1
Real virginica	0	2	9

Table 2: Confusion Matrix for Gaussian Naïve Bayes model

3.1.2 Gaussian Naïve Bayes

I have imported this through the naive_bayes package of sklearn. Similar, to the Logistic Regression case, here also, I've fitted on the train data(4/5 of the dataset) and predicted for the test data(1/5 of the dataset). The results obtained through this approach are: **Classification report:** *Weighted Average precision, Recall and F1-Score are all*: =0.90 and *macro Average precision, Recall and F1-Score are* = 0.91

Confusion Matrix: [[8 0 0] [0 10 1] [0 2 9]] *(Table 2 Above)* In total, 3 results have been wrongly classified out of 30 in the test set.

After this, for cross validation, I have used 5 fold cross validation using KFold Cross Validation. This has been implemented by me using cross_val_score functionality of model_selection package. In this functionality, I've done the scoring using the **accuracy** measure. The results are impressive, with the resulting **Cross validation mean as 0.966 and Cross validation standard deviation as 0.031.**

As we can see, we've got somewhat bad results in Naive Bayes as compared to logistic regression, and a lesser precision, recall and f1 score and more number of false predictions. Also, as seen after cross validation, the standard deviation is higher although the mean is the same. All this confirms that in classification, naive Bayes has typically a higher error than logistic regression. As seen from practice, it converges quicker and hence may work better than logistic regression for smaller datasets, but in general Logistic regression gives better results.

3.2 Unsupervised Learning

Here, I have used K-Means once on the entire dataset and the second time on a subset of the dataset (with only the more useful features). Let's see how well the classification fares.

3.2.1 K-Means

To classify the data, K-Means is one good approach that is usually taken, when we do not have the ground truth labels. It is an unsupervised learning algorithm as it does not require the data labels. I have imported the KMeans functionality from the cluster package of sklearn. I have defined the number of clusters as 3 as we know that there are 3 distinct classes(species) asmentioned in the assignment. Then, I have fitted the model on X (the entire dataset) as we don't need to make the train/test split on our data set while implementing unsupervised learning models. Also, the predicted labels can be seen from KMeans.labels_. I've stored the predicted labels in a new feature which I've used to plot a scatter plot of the classification done using kmeans.

From the scatter plot, we can see that although the results are good, there are a few points which are classified wrongly. Iris Versicolor, and Iris Virginica have been wrongly classified for some data points. Tabulating the confusion matrix and classification report: **1.**

	Predicted setosa	Predicted versicolor	Predicted Virginica
Real setosa	50	0	0
Real versicolor	0	48	2
Real virginica	0	14	36

Table 3: Confusion Matrix for K-Means model

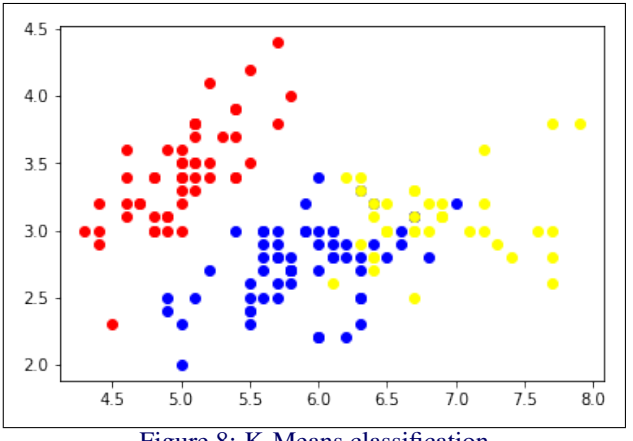

Figure 8: K-Means classification

Classification report: Average Precision: 0.91, Average Recall: 0.89, Average F1-Score: 0.89 These are significantly less than logistic regression and comparable to Gaussian Naive Bayes. **2. Confusion Matrix: [[50 0 0] [0 48 2] [0 14 36]]** *(table 3 above)*

As we can see from these results, the performance is not as good as the supervised algorithms (logistic regression and Gaussian Naive Bayes) but is still appreciable. Around 16 (14+2=16) data points out of 150 have been classified wrongly/falsely as seen from the confusion matrix. In general, the supervised learning models will always give better results. *(The results are plotted in figure 8)*

3.2.2 Improved K-Means: attempting to plot the mis-classified data points

To improve the classification plot, I remember that I had plotted the correlations of the various features with the class label, using a heatmap(Figure 2). The features: petal length and petal width have high correlation(0.95 and 0.96 respectively) with the class(species) label. Hence, I now run KMeans using these two features only: a subset of the original dataset. On running KMeans successfully on this subset, I plot the predictions on a scatter plot. *(FIgure 9)* As expected, I get a near-perfect classification: almost all points in the scatter plot are accurately being classified into the 3 classes. This is confirmed by comparing the **actual plot**(*refer figure 10*) of the 3 classes with the **plot obtained after the modified KMeans**.

Hence, by reducing the features in the dataset to only highly correlated features (more useful features), I was able to improve the classification and was able to plot all the misclassified points from the previous plot. **The misclassifications in the previous plot were due to the 2 features: sepal width and sepal length wrongly directing the algorithm to classify class 2 as class 3 and vice versa. These features do not have a strong relation**

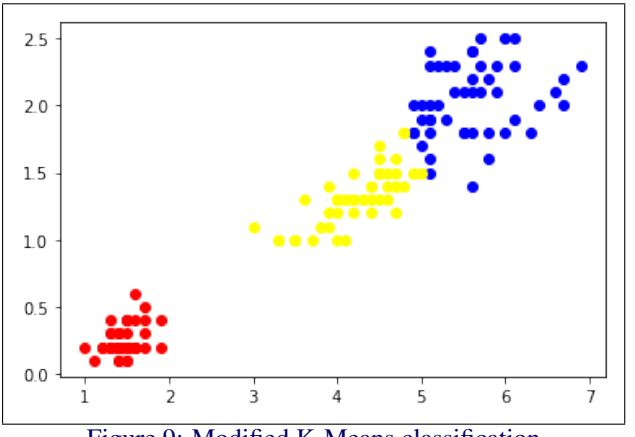

Figure 9: Modified K-Means classification

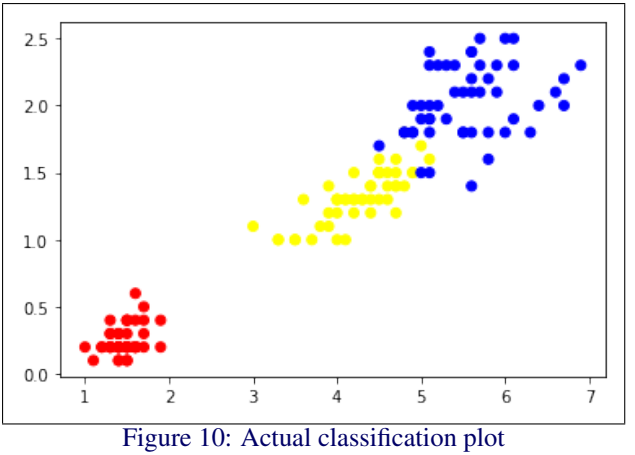

Figure 10: Actual classification plot

(a high correlation) with the class species and as they are given the same weightage as petal length and width, they can cause the algorithm to classify falsely, which it had done in the previous case. In the previous case, there were some misclassifications, but in the case after removing the not-so-useful features, I was able to classify almost all those points correctly(150 data-points). This clearly explains the misclassifications.[1]

(After running this modified model, I didn't need to go for PCA as I had already achieved the desired classification results!)

4 Conclusion and Inferences

The Fisher Iris dataset gives us deep insight into the usefulness of various classification algorithms: both supervised and unsupervised. **Logistic Regression is a better classifier in general as compared to Naive Bayes which in turn is still better than KMeans.** Also, the

[1] Answer to the question: explain the misclassifications

Learning Model	F1 Score
Logistic Regression	0.97
Gaussian Naive Bayes Classifier	0.91
K-Means	0.89

Table 4: F1 Score Results for the 3 learning models

F1 scores from the 3 classification algorithms used in this assignment have been tabulated above in Table 4. KMeans is affected by the less correlated features in a not-so-useful manner and this leads to some misclassifications sometimes, hence using only the useful features, i.e. features that relate more to the species type and are indicative of the species, will help to increase the model accuracy in classification. Hence, after removing those features, I was able to achieve better results.

As far as the Iris species is concerned this analysis helped me understand which species has the maximum sepal length, width and petal length, width. *For example, Setosa has the minimum petal length and petal width and Iris Virginica has the maximum petal length as well as petal width.* ***Also, the Setosa class is, as seen in the plots, very much separate from the other 2 classes, i.e. its properties are a bit different (and distinct) from the rest of the 2 classes and is appearing separately in all the plots.*** This can be affirmed by seeing the pair plot that I've plotted in the end of the data exploration section. ***Hence, most of the ML algorithms will be able to easily separate it from the other 2 classes.*** Through this dataset, I also learnt that the differences between the petal and sepal sizes are indeed indicative of the species and hold information of high value.

This exercise helped me understand and achieve the following:

1. **Use of Data Exploration and Data Visualisation** for understanding real world datasets.
2. **In-depth understanding of Machine Learning models:** Logistic Regression, Naive Bayes and K-Means, their application and performance comparison.
3. **Very good classification results obtained after modifying the K-Means algorithm** and even better results with supervised learning models.

5 References

[1] Department of Computer Science - University of Houston, http://www2.cs.uh.edu/~ceick/UDM/DA_Tan.pdf

[2] Orielly, https://www.oreilly.com/library/view/python-data-science/978149 ch04.html

[3] Andrew NG, Logistic Regression, http://www2.cs.uh.edu/~ceick/UDM/DA_Tan.pdf

[4] Stack Exchange

[5] Documentation of Python scikit learn library

[6] Towards Data Science (Medium Blog): Cross Validation in Python, https://towardsdatascien com/complete-guide-to-pythons-cross-validation-with-examples-a9676b5ca