1/11/2020

Iris Classification

DEBABRATA BHATTACHARYA

Table of Contents

Iris Classification

Aim of the Project

This project aims to build a model of the iris dataset. This model can be further used to classify unknown data.

Chapter 1: **About the Project**

This is the Iris classification project. The aim of this project is to build a model that can classify the iris dataset.

1.1 Project Steps

- Data Download
- Data Loading
- Data Summarization
- Data Visualization
- Partitioning of dataset into Training dataset and Validation dataset
- Model Creation
- Model Selection
    - Create test harness using K-Fold Cross Validation, with scoring set to 'accuracy'
    - Evaluation of models using test harness
    - Summarization, Visualization and Comparison of Results
    - Model Selection
- Making Predictions using Selected Model
- Summarization of Results
- Saving the Pipelined Project
- Testing the saved model

1.2 Project Files

- Dataset
- Python file (using template.py as the base)
- Model files
- Image files
- Documentation
- README.md
- Project Report
- Slide deck

Chapter 2: **About the Iris Dataset**

The repository is hosted at [UCI Machine Learning Repository]

The data set is multivariate and contains ratio(numerical) and nominal data. There are 150 instances and 4 attributes.

2.1 Dataset Summarization

2.1.1 Shape of the dataset (instance, attribute)

(150, 5)

We can see that there are 150 instances (or rows) and 5 attributes

2.1.2 First 20 instances

	sepal-length	sepal-width	petal-length	petal-width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa
10	5.4	3.7	1.5	0.2	Iris-setosa
11	4.8	3.4	1.6	0.2	Iris-setosa
12	4.8	3.0	1.4	0.1	Iris-setosa
13	4.3	3.0	1.1	0.1	Iris-setosa

14	5.8	4.0	1.2	0.2	Iris-setosa
15	5.7	4.4	1.5	0.4	Iris-setosa
16	5.4	3.9	1.3	0.4	Iris-setosa
17	5.1	3.5	1.4	0.3	Iris-setosa
18	5.7	3.8	1.7	0.3	Iris-setosa
19	5.1	3.8	1.5	0.3	Iris-setosa

A look at the first 20 rows shows us that the data X values are of ratio(float) type and the y values are categorical and nominal

2.1.3 Statistical summary

	sepal-length	sepal-width	petal-length	petal-width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

From the summary we can see that the data is of 150 count. The values lie between 0 and 8.

2.1.4 Class Distribution

class

Iris-setosa 50

Iris-versicolor 50

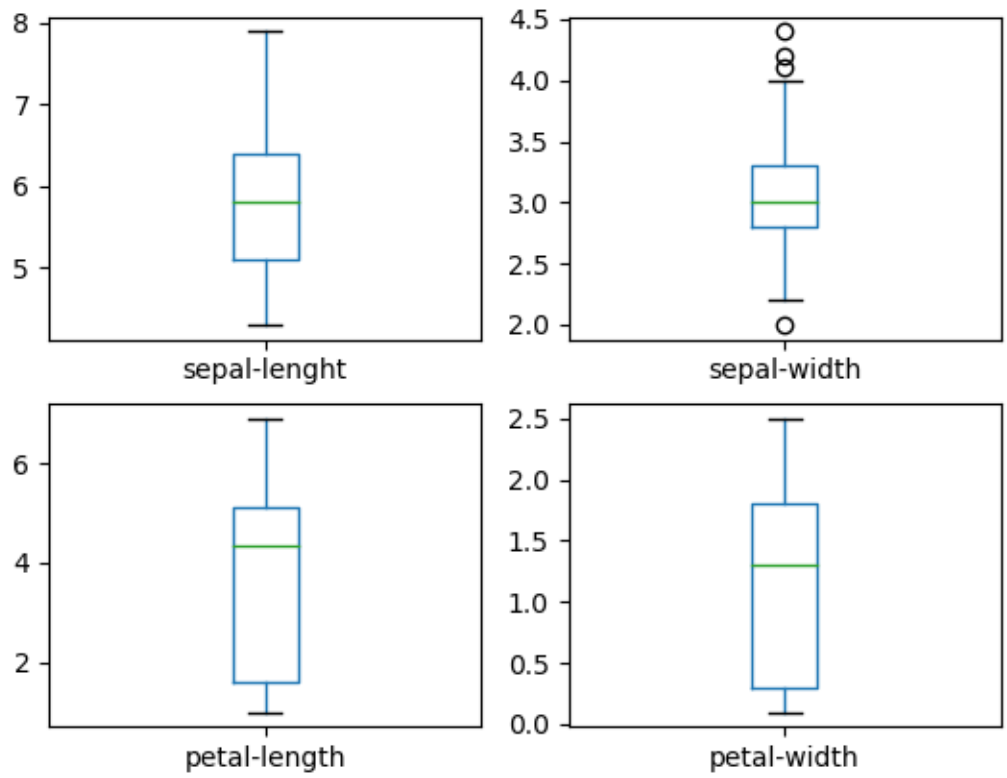
Iris-virginica 50

dtype: int64

We can see that the class distributions are well balanced, with each of the 3 classes comprising a neat third of the dataset.

2.2 Data Visualization & analysis

2.2.1 Box and whisker



Sepal length

We can see a well-balanced dataset. There is no visible skew. The max data point seems to be well above the 75% quartile.

Sepal width

We can see some outliers here, above the max point. There is slight skew towards the 75% quartile and, the data is probably skewed to the right.

Petal length

No outliers, but the data is very much skewed towards the 25% quartile. The 75% quartile is much closer to the mean than the 25% quartile. The minimum value is quite far from the mean.

Petal width

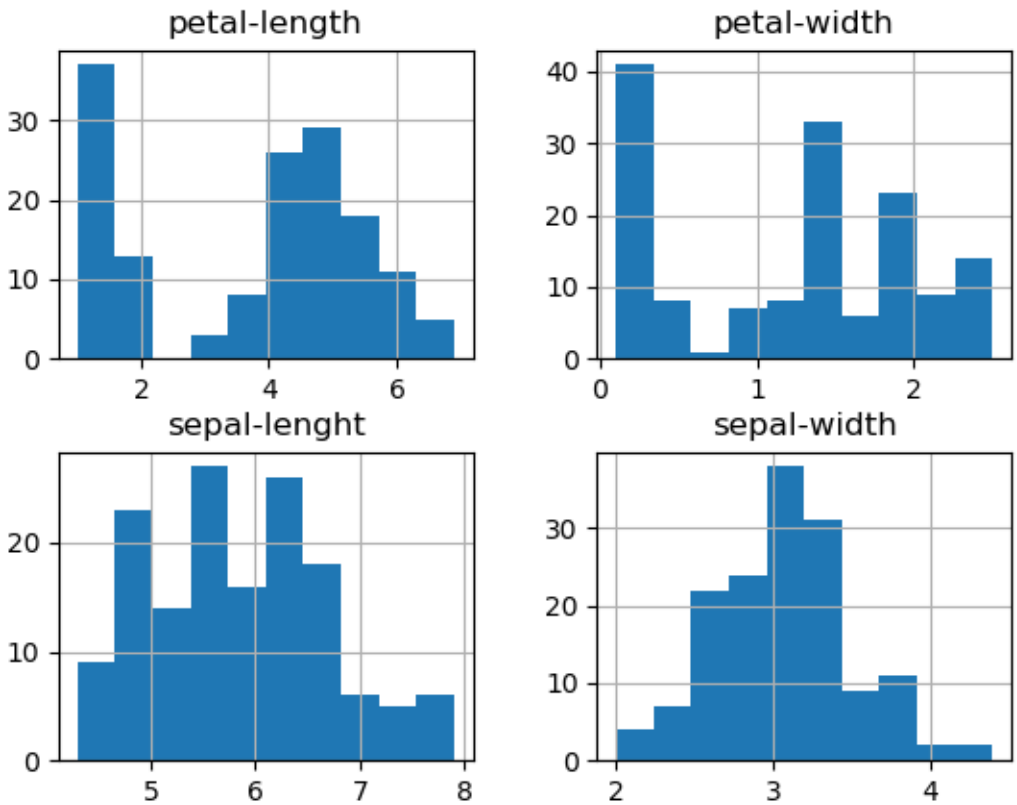
Again, the data is very much skewed towards the 25% quartile. The minimum value is quite far from the mean.

Conclusion

Petal length and width are both on the smaller side. Values in these 2 columns are skewed to the left. Very interesting.

In contrast, sepal length and width are much more 'normal'.

2.2.2 Histogram

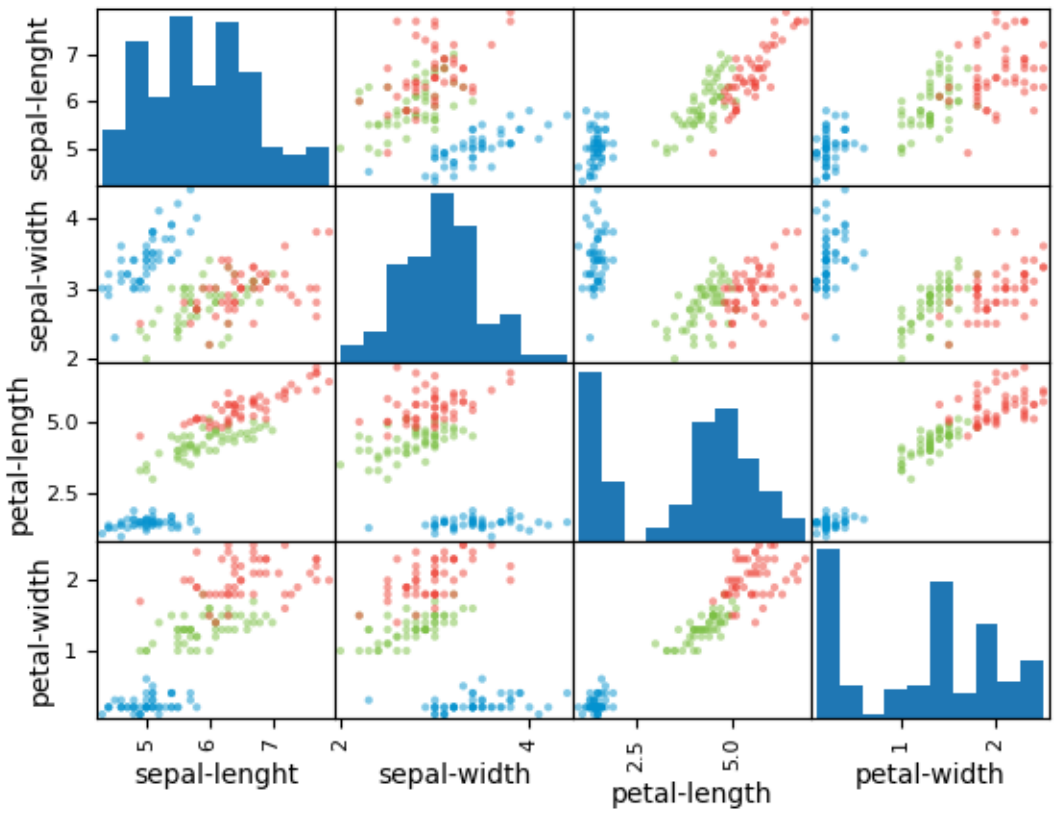


As expected, petal length and width are both heavily skewed to the left. You could draw a diagonal line from the left to the right across the Maxima of the petal width data.

Sepal length and width assume a very broken, but still imaginable bell curve.

Overall, the data seems very interesting.

2.2.3 Scatter matrix



There's a slight correlation between sepal length and sepal width for one of the classes. This is also the case for sepal length and petal length.

Petal length and width also have a correlation for a part of the data.

Conclusion

The data has some slight correlation.

Chapter 3: **Model Creation**

The following functions were considered to build the model:

1. Linear Regression
2. Linear Discriminant Analysis
3. K-Nearest Neighbors
4. CART
5. Gaussian Naïve Bayes
6. Support Vector Machine

3.1 Spot Checking

The models were spot checked on training dataset using a 10-k KFold Harness.

3.1.1 Results

```
The Cross Eval Scores using 10-kfold test harness is:
lr: 0.9666666666666666 (0.04082482904638632)
lda: 0.975 (0.03818813079129868)
knn: 0.9833333333333332 (0.03333333333333335)
cart: 0.975 (0.03818813079129868)
nb: 0.975 (0.053359368645273735)
svm: 0.9916666666666666 (0.025000000000000012)
```

From the figure we can see the nearly all the non-linear models reach near 1.00 accuracy.

SVM and KNN seem to have the highest estimated accuracy scores. We have chosen the KNN.

3.2 Creating the model.

We have created the model using the KNN function.

Chapter 4: **Results**

4.1 Results of Testing on Validation Dataset

Accuracy = 0.9

Confusion Matrix:

[[7 0 0]

[0 11 1]

[0 2 9]]

Classification report:

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	7
Iris-versicolor	0.85	0.92	0.88	12
Iris-virginica	0.90	0.82	0.86	11
accuracy			0.90	30
macro avg	0.92	0.91	0.91	30
weighted avg	0.90	0.90	0.90	30

4.2 Results of Testing on Entire Dataset

Accuracy = 0.9666666666666667

Confusion Matrix:

```
[[50 0 0]
[ 0 47 3]
[ 0 2 48]]

Classification report:
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        50
Iris-versicolor       0.96      0.94      0.95        50
 Iris-virginica       0.94      0.96      0.95        50

       accuracy                           0.97       150
      macro avg       0.97      0.97      0.97       150
   weighted avg       0.97      0.97      0.97       150
```

Model is accurate

Chapter 5: **Tests**

The following things were tested:

- Loading and partitioning of data
- Accuracy of finalized model

5.1 Test Results:

```
(base) J:\Education\Code\DATA_Science\Books\Jason_Brownlee\Machine-Learning-Mastery-With-Python\Part_III\Lesson19>pytest
 test_iris.py -vv
============================================= test session starts =============================================
platform win32 -- Python 3.7.3, pytest-5.0.1, py-1.8.0, pluggy-0.12.0 -- C:\ProgramData\Anaconda3\python.exe
cachedir: .pytest_cache
rootdir: J:\Education\Code\DATA_Science\Books\Jason_Brownlee\Machine-Learning-Mastery-With-Python\Part_III\Lesson19
plugins: arraydiff-0.3, doctestplus-0.3.0, openfiles-0.3.2, remotedata-0.3.1
collected 2 items

test_iris.py::TestObject::test_evaluate_algorithms PASSED                                                [ 50%]
test_iris.py::TestObject::test_accuracy PASSED                                                           [100%]
```

5.2 Test Status

All tests have been successfully passed.

Chapter 6: **Project Status**

Project has been successfully completed.