1/11/2020

Iris Classification

DEBABRATA BHATTACHARYA

Table of Contents

[Iris Classification 2](#_Toc29649487)

[Aim of the Project 3](#_Toc29649488)

[Chapter 1: About the Project 4](#_Toc29649489)

[1.1 Project Steps 4](#_Toc29649490)

[1.2 Project Files 4](#_Toc29649491)

[Chapter 2: About the Iris Dataset 5](#_Toc29649492)

[2.1 Dataset Summarization 5](#_Toc29649493)

[2.1.1 Shape of the dataset (instance, attribute) 5](#_Toc29649494)

[2.1.2 First 20 instances 5](#_Toc29649495)

[2.1.3 Statistical summary 6](#_Toc29649496)

[2.1.4 Class Distribution 6](#_Toc29649497)

[2.2 Data Visualization & analysis 7](#_Toc29649498)

[2.2.1 Box and whisker 7](#_Toc29649499)

[2.2.2 Histogram 9](#_Toc29649500)

[2.2.3 Scatter matrix 10](#_Toc29649501)

[Chapter 3: Model Creation 11](#_Toc29649502)

[3.1 Spot Checking 11](#_Toc29649503)

[3.1.1 Results 11](#_Toc29649504)

[3.2 Creating the model. 11](#_Toc29649505)

[Chapter 4: Results 12](#_Toc29649506)

[4.1 Results of Testing on Validation Dataset 12](#_Toc29649507)

[4.2 Results of Testing on Entire Dataset 12](#_Toc29649508)

[Chapter 5: Tests 14](#_Toc29649509)

[5.1 Test Results: 14](#_Toc29649510)

[5.2 Test Status 14](#_Toc29649511)

[Chapter 6: Project Status 15](#_Toc29649512)

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.

# 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.

## 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

## Project Files

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

# About the Iris Dataset

The repository is hosted at [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/machine-learning-databases/iris/)

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

## Dataset Summarization

### Shape of the dataset (instance, attribute)

(150, 5)

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

### 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

### 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.

### Class Distribution

class

Iris-setosa 50

Iris-versicolor 50

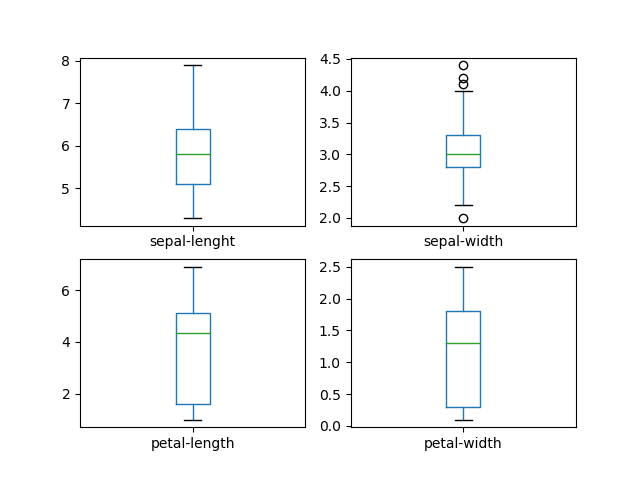
Iris-virginica 50

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.

## Data Visualization & analysis

### 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**

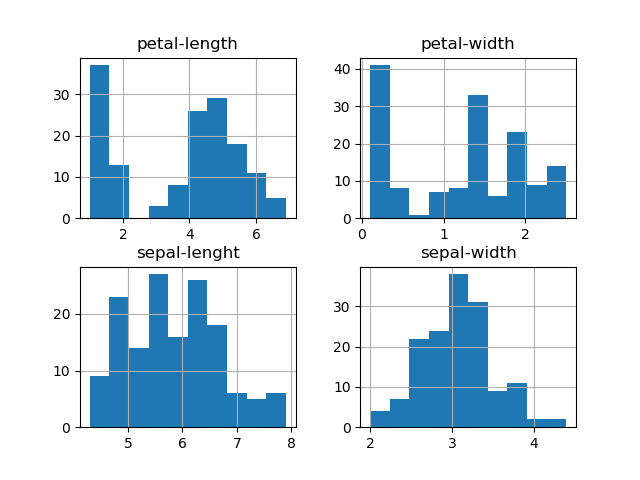
Again, the data is very much skewed towards the 25% quartile. The minimum value is quite far from the mean.

**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'.

### Histogram

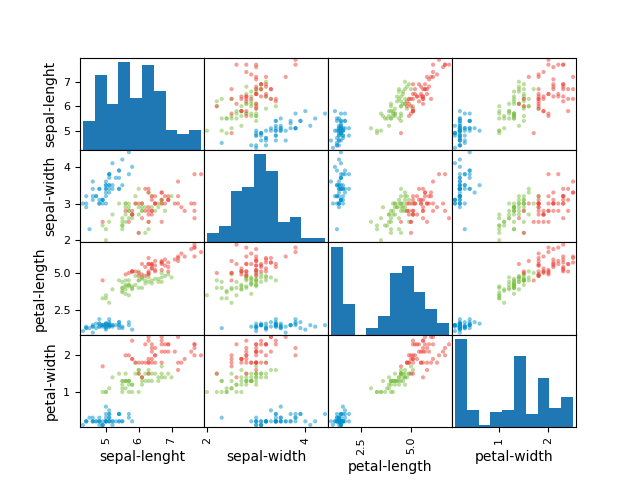


As expected, petal length and width are both heavily skewed to the left. You could draw a diagonal line from the left to the right across the Maxima of the petal width data.

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

Overall, the data seems very interesting.

### 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.

# 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

## Spot Checking

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

### 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.

## Creating the model.

We have created the model using the KNN function.

# Results

## 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

## 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

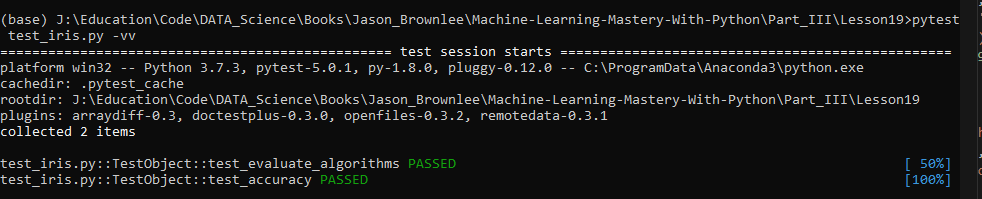
Model is accurate

# Tests

The following things were tested:

* Loading and partitioning of data
* Accuracy of finalized model

## Test Results



## Test Status

All tests have been successfully passed.

# Project Status

Project has been successfully completed.

# Git log

commit 56d176f159f9be7b55a4bf199fa1445e80b83082 (HEAD -> master, origin/master)

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Sun Jan 12 13:38:12 2020 -0500

Created Slide deck for project

commit 9a437e438d1d89c3e084a01a5fd68628d81a2276

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Sat Jan 11 15:40:03 2020 -0500

Created Project Report

commit 1a3c0d6b52a1813d7ab269bdea8b8f164f4b95e9

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Sat Jan 11 14:54:23 2020 -0500

Updated README.md with project information

README.md has been updated with all the information from the project

such as the output, images, porject status, etc.

This will now be used to create a project report.

commit f4f5f42daca1e7151b6e5eead28f2ed1998444e3

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Sat Jan 11 14:15:42 2020 -0500

Testing the saved model

We used joblib.load to load the saved iris model and made predictions

using it. These predictions were tested for accuracy using the

accuracy\_score function from sklearn.metrics.accuracy\_score.

It is necessary to test a saved model, to ensure it is operating

correctly.

commit 0ddc4ff57f68a6e3cbd43cf5844cf7c46fc63454

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Sat Jan 11 13:58:00 2020 -0500

Modfied function finalize\_iris\_model()

The function was modified to add code that recalculates results.

commit f68da50e1436c3387f6df47306f3de516a9618c1

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Sat Jan 11 13:39:00 2020 -0500

Saving the Pipelined Project

We have used joblib's dum to save the finalized model

commit c65d238006750eaf759c1354907bde7c3751768c

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Sat Jan 11 13:29:02 2020 -0500

Making Predictions using Selected Model and Summarization of Results

We have fit the KNN model to the training data and made predictions

using it on the test data. After that we have scored the accuracy,

created a confusion matrix and a classification report using the

validation dataset.

To ensure that we are getting as accurate a model as possible, we test

the model using a validation dataset. After that, we can use various

tools to record the results.

commit 4005ff2ee0d2abb6b09fbf116893154313f83b35

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Mon Jan 6 15:08:50 2020 -0500

Model Selection

We have created a 10k Kfold Test harness to spot check models for their

accuracy. After evaluating the models and summarizing and visualization

of results, we find the SVM Classifier model to have the highest

accuracy score.

Spot checking models is an important step in selecting a suitable model.

It has the following benefits:

- We find out which models perform well on our data

- We can find out which models might suit our problem before proceeding

with predictions

- We can trial a number of models before committing to them

Here, we find that KNN and SVC have the highest accuracy score.

commit 9ceb4963a7767204f6d15a9cbfedd0d9c450e9e4

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Sat Jan 4 14:37:58 2020 -0500

Partitioning of dataset into Training dataset and Validation dataset

The data set has been transformed into an array of values, seperated

into x and y sets, and split into training and validation sets.

This will allow us to use the available dataset very efficiently, by

allowing us to test future models on unseen data.

commit 5d4768e2e00218512bf525d8cc742be0547183a2

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Sat Jan 4 13:46:07 2020 -0500

Data Visualization-iris dataset

We have created visual representations of the data in the form of

box and whisker plots, histograms, and scatter matrices.

We have gained a further understanding of the data, including but not

limited to the following:

- 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'.

- The data has some slight correlation.

commit b707bad336a9732bcfb46a4b583aeb2f64a82e26

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Fri Dec 27 13:55:57 2019 -0500

Create 2 sub functions under summarize\_iris\_data()

2 sub functions were created under summarize\_iris\_data() as

summarize\_iris\_data\_stats() and summarize\_iris\_data\_visualtization().

The code now calls these 2 sub functions from the hiher level function,

to fulfil the 2 goals of summarize\_iris\_data():

- Descriptive statistics such as summaries.

- Data visualizations such as plots with Matplotlib, ideally using

convenience functions from Pandas.

This change was made to grant better narrative flow to the program,

while keeping the output constant.

commit 7c11c7c15a679a3ca356eb08cb3d5c5918fcfb62

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Fri Dec 27 13:37:39 2019 -0500

Changed summarize\_iris\_data() to summarize\_iris\_data\_stats()

commit 9dae4e833bdf1b663d5f3c64523310eabe0209a2

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Tue Dec 24 14:02:05 2019 -0500

Modified output at the end of the file for function summarize\_data()

commit 72146e82a1c6f363ba47f55691bb8ff3b0f724dc

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Tue Dec 24 13:22:07 2019 -0500

Data Summarization

A statistical analyis of the data has been conducted and the results

summarized and printed to stdout.

To create an accurate model of the data, we must take a good look at

the data and create a statistical summary of it. We now know the

characteristics of the data and will be able to visualize it for

further understanding.

commit 0fd79bebdfa7d6215ee2b17527fe8928f60d218c

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Tue Dec 24 12:40:20 2019 -0500

Moved datasets under datasets/iris

commit 290d20f5cd29ce8ab0b48fb472b542faac20a097

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Mon Dec 23 15:01:39 2019 -0500

Tick off 'Data Loading' in Project Steps in README.md

commit 489f75490b4237d7ae4810b6188df34878a11821

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Mon Dec 23 14:52:50 2019 -0500

Data Loading

I created a file called 'iris.py' to hold all the functional code for

the project. The file contains a class 'iris' and it will be used as

the primary object of this project.

The data has been loaded from the file 'iris.data' into a pandas

dataframe in the class \_\_init\_\_ function.

commit 2f9e0f23bc7c96a880154e81c2d9ed9837550f22

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Sat Dec 21 13:54:02 2019 -0500

Fix typos in README.md and update list of

completed project tasks

Some typos were fixed in the README.md file. The list of completed

project steps was updated to check off 'Data Download'.

commit ff6b6236cb7f6047fe5b6318779d41c08991e3b7

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Sat Dec 21 13:49:44 2019 -0500

Add section 'Project Status' to README.md

I have added a section entitled 'Project Status' to the README.md file.

This section will be used to track the current status of the project.

commit 37142783fa7c1afa5efbf63ce7fea411ca6daefa

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Sat Dec 21 13:42:45 2019 -0500

Download Dataset

The dataset iris flowers has been downloaded and saved at

Part\_III\Lesson19\datasets.

This dataset will be used to train and validate the model

for this project. The floder contains 2 datasets, an index,

and a names file

commit 7c5b878c029a9c930f74e8e1a59bc126c9ce2ef6

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Sat Dec 21 13:32:59 2019 -0500

Create README.md

I have created a README.md file for this project. This will serve as the primary documentation and reference as thee rpoject progresses.

The file conforms to GitHub Flavored Markdown Spec (Version 0.29-gfm).

It contains the following sections about the respository, the dataset, steps of the project, project files, future steps and images.

Part\_III\Lesson19\datasets.

This dataset will be used to train and validate the model

for this project. The floder contains 2 datasets, an index,

and a names file

commit 7c5b878c029a9c930f74e8e1a59bc126c9ce2ef6

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Sat Dec 21 13:32:59 2019 -0500

Create README.md

I have created a README.md file for this project. This will serve as the primary documentation and reference as thee rpoject progresses.

The file conforms to GitHub Flavored Markdown Spec (Version 0.29-gfm).

It contains the following sections about the respository, the dataset, steps of the project, project files, future steps and images.

I hope to frequently update the README.md file throughout the project.

commit a332344747b0bacc1ffa9e933b9b1304ed91b73c

Author: D-Bhatta <dbhatta1232@gmail.com>

Date: Wed Dec 18 14:23:12 2019 -0500