Classification of Iris flowers using Machine Learning

University of Texas at Dallas - MS Business Analytics

BUAN 6341.501 - Applied Machine Learning - S23

Assignment 1: Building a test model - Douglas Obeng

Notebook submitted for Assignment 1 of the Applied Machine Learning Course at The University of Texas at Dallas, Spring 2023

In this project, machine learning is used to model 3 species of iris flowers using some characteristics of the flowers.

The goal is to build a simple test model that can be used to predict unknown species of iris flowers.

Assignment Specifics

- Create a data set of about 15 to 20 records and three to four features.
- Load your data into a Dataframe
- Split your data into two tables – One with features and one with outcomes (only one column – with a binary outcome)
- Clean your data set (Use a CSV file and it will be easier to clean)
- Build a model based
- Use the Predict method to predict a value that is not in your outcomes
- Due Date – Tuesday, January 31, 2022 by Midnight.
- You will upload your assignment in the dropbox .

Introduction

The iris is a common flowering plant (genus *iris*) mainly found in the temperate zones of the Northern hemisphere. There are about 310 species that come in various sizes, shapes and colours.

Three of the most common species are the *iris setosa*, *iris versicolor* and *iris virginica*. These 3 look similar in color (bluish purple) and shape but may be distinguished from one another using the dimensions of their flower petals and sepals.

The original Iris flower data set, also known as Fisher's Iris data or Anderson's Iris data set was used by British Biologist Statistician Ronald Fisher in his 1936 paper *The use of multiple measurements in taxonomic problems*. The data was collected by Edgar Anderson and consists of 50 samples of each of the 3 species (150 total).

References

Neuraldesigner - Iris flower classification

Analyticsvidhya - Iris flower classification using ML

Kaggle - Iris flower dataset

Wikipedia - Iris flower

Wikipedia - Iris flower data set

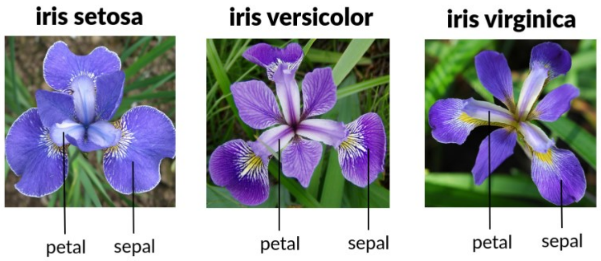
Data

The data for this project was downloaded from the Kaggle website - Iris Flower Dataset.

The data has been saved in my github repository for direct online access within this notebook: DO Github - complete iris dataset.

NOTE - For the purpose of this assignment, the original dataset has been modified to include only 20 records with 2 possible outcomes (iris setosa and iris versicolor). Two of the records have also been given null values for illustration of data cleaning. There is the option to use the complete data though

The modified data has been saved in my github repository for direct online read in the notebook: DO Github - modified iris dataset



The 4 independent features are Sepal Length, Sepal Width, Petal Length, and Petal Width. All the lengths are in centimeters. The dependent feature (outcome) is Species. Species refers to one of the species whose characteristics were measured.

Libraries and packages

```
#import libraries
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
```

Load dataset into dataframe

```
irisdata = 'https://raw.githubusercontent.com/obengdouglas/UTD-BUAN6341_AppliedML/main/Assignment1-Building_a_test_model/IRIS_modified.csv' #save url of modified datase
#irisdata = 'https://raw.githubusercontent.com/obengdouglas/UTD-BUAN6341_AppliedML/blob/main/Assignment1-Building_a_test_model/IRIS.csv' #uncomment this line to use com

iris.df = pd.read_csv(irisdata) #read data

#This works fine in colab. In case of any issues, read the included dataset from your local directory
#iris.df = pd.read_csv('Desktop/Github/UTD-BUAN6341_AppliedML/Assignment1-Building_a_test_model/IRIS_modified.csv') #replace with local directory of dataset
```

```
iris.df #view dataframe
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	4.9	3.1	1.5	0.1	Iris-setosa
1	5.4	3.9	1.3	0.4	Iris-setosa
2	5.7	3.8	1.7	0.3	Iris-setosa
3	5.1	3.8	1.5	0.3	Iris-setosa
4	5.4	3.4	1.7	NaN	Iris-setosa
5	5.1	3.7	1.5	0.4	Iris-setosa
6	4.6	3.6	1.0	0.2	Iris-setosa
7	4.7	3.2	1.6	0.2	Iris-setosa
8	4.8	3.1	1.6	0.2	Iris-setosa
9	5.4	3.4	1.5	0.4	Iris-setosa
10	7.0	3.2	4.7	1.4	Iris-versicolor
11	6.4	3.2	4.5	1.5	Iris-versicolor
12	6.9	3.1	4.9	NaN	Iris-versicolor
13	5.5	2.3	4.0	1.3	Iris-versicolor
14	6.5	2.8	4.6	1.5	Iris-versicolor
15	5.7	2.8	4.5	1.3	Iris-versicolor
16	6.3	3.3	4.7	1.6	Iris-versicolor
17	4.9	2.4	3.3	1.0	Iris-versicolor
18	6.6	2.9	4.6	1.3	Iris-versicolor
19	5.2	2.7	3.9	1.4	Iris-versicolor

Data Cleaning

Drop rows with null values

```
iris.isnull() #dataframe showing boolean results for the presence of null values. Index 5 and 6 have TRUE for column 'petal_width'
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	True	False
6	False	False	False	False	False
7	False	False	False	False	False
8	False	False	False	False	False
9	False	False	False	False	False
10	False	False	False	False	False
11	False	False	False	True	False
12	False	False	False	False	False
13	False	False	False	False	False
14	False	False	False	False	False
15	False	False	False	False	False

```
iris.df.isnull().sum() #number of null values for each column. Column petal_width has 2 null values
```

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     2
species         0
dtype: int64
```

```
iris_clean.df = iris.df.dropna(how="any") #drop all rows that have at least one null value (any)
```

```
iris_clean.df #view cleaned dataframe
```

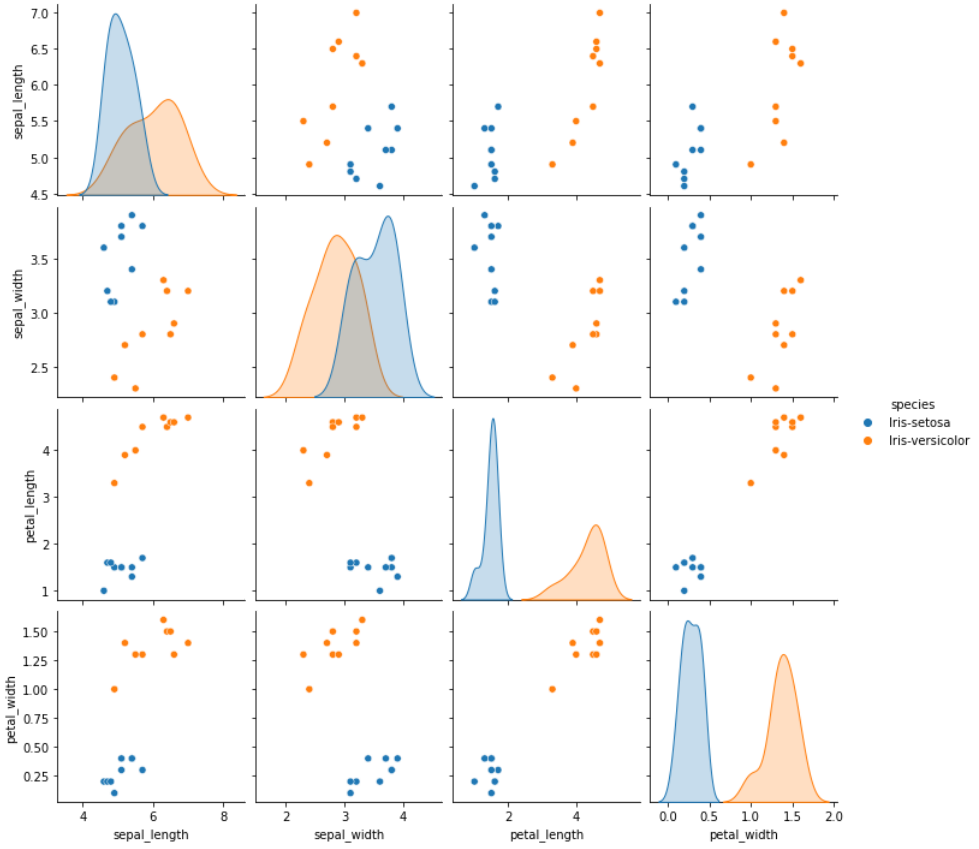
	sepal_length	sepal_width	petal_length	petal_width	species
0	4.9	3.1	1.5	0.1	Iris-setosa
1	5.4	3.9	1.3	0.4	Iris-setosa
2	5.7	3.8	1.7	0.3	Iris-setosa
3	5.1	3.8	1.5	0.3	Iris-setosa
5	5.1	3.7	1.5	0.4	Iris-setosa
6	4.6	3.6	1.0	0.2	Iris-setosa
7	4.7	3.2	1.6	0.2	Iris-setosa

Exploratory Data Analysis (EDA)

10	7.0	3.2	4.7	1.4	Iris-versicolor

```
import seaborn as sns
sns.pairplot(iris_clean.df, hue='species') #Use seaborn to plot pairwise relationship of independent features
```

```
<seaborn.axisgrid.PairGrid at 0x7fd555576820>
```



Split data into features and outcome

```
X = iris_clean.df.drop(columns = ['species']) #save features as X
y = iris_clean.df['species'] #save outcome as y
```

```
X #view features dataframe
```

	sepal_length	sepal_width	petal_length	petal_width
0	4.9	3.1	1.5	0.1
1	5.4	3.9	1.3	0.4
2	5.7	3.8	1.7	0.3
3	5.1	3.8	1.5	0.3
5	5.1	3.7	1.5	0.4
6	4.6	3.6	1.0	0.2
7	4.7	3.2	1.6	0.2
8	4.8	3.1	1.6	0.2
9	5.4	3.4	1.5	0.4
10	7.0	3.2	4.7	1.4
11	6.4	3.2	4.5	1.5
13	5.5	2.3	4.0	1.3
14	6.5	2.8	4.6	1.5
15	5.7	2.8	4.5	1.3
16	6.3	3.3	4.7	1.6
17	4.9	2.4	3.3	1.0
18	6.6	2.9	4.6	1.3
19	5.2	2.7	3.9	1.4

```
y #view outcome dataframe
```

```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
5         Iris-setosa
6         Iris-setosa
7         Iris-setosa
8         Iris-setosa
9         Iris-setosa
10    Iris-versicolor
11    Iris-versicolor
13    Iris-versicolor
14    Iris-versicolor
15    Iris-versicolor
16    Iris-versicolor
17    Iris-versicolor
18    Iris-versicolor
```

```
19    Iris-versicolor
Name: species, dtype: object
```

Model Training

```
model = DecisionTreeClassifier() #select type of model algorithm to use
model.fit(X,y) #fit model
```

```
DecisionTreeClassifier()
```

Model Prediction

```
predictions = model.predict([[6,3,1.5,0.3], [6,2.5,4,3.5]]) #predict the outcoe of 2unknown species of iris flowers with the specified feature dimensions in order
predictions #view prediction results
```

```
/usr/local/lib/python3.8/dist-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
array(['Iris-setosa', 'Iris-versicolor'], dtype=object)
```

Conclusion

The model has predicted the unknown species as *Iris*-setosa and Iris-versicolor respectively