iris-flower-classification

August 30, 2023

[1]:
```
import numpy as np
import pandas
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
```

[2]:
```
dataset = pandas.read_csv("IRIS.csv")
dataset.head()
```

[2]:

	sepal_length	sepal_width	petal_length	petal_width	species
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

[3]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

[4]:
```
dataset['species'].unique()
```

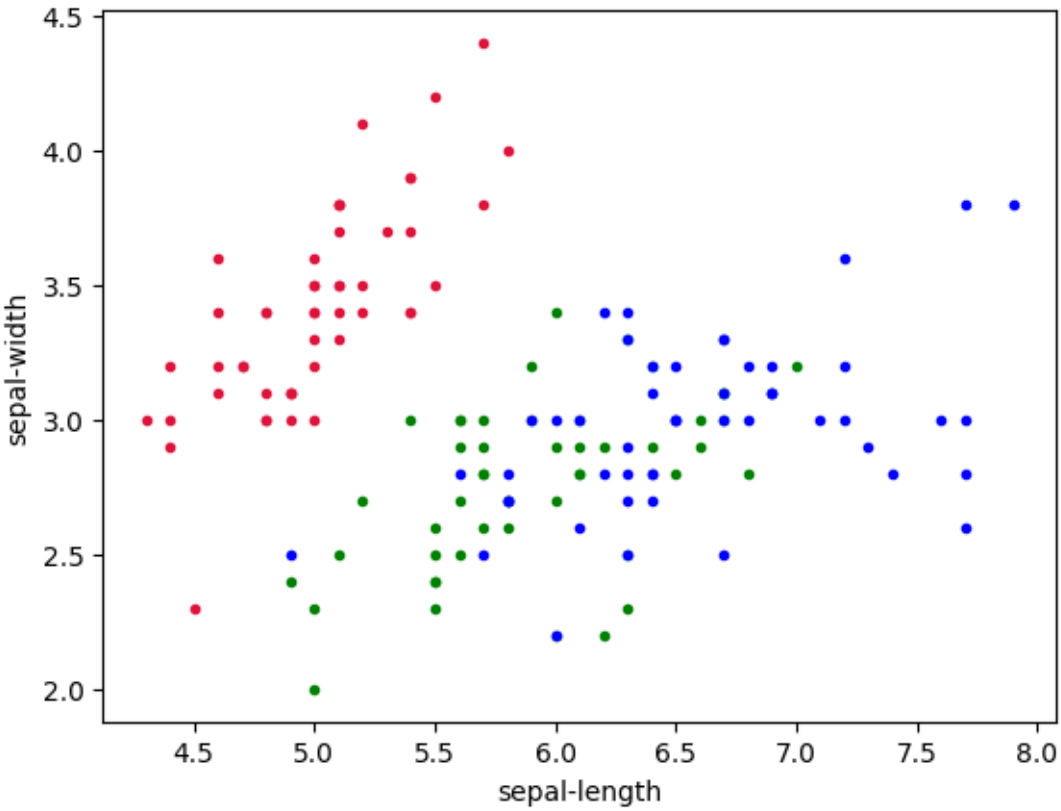
[4]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

[5]:
```
x1 = dataset.loc[dataset['species'] == 'Iris-setosa', 'sepal_length']
y1 = dataset.loc[dataset['species'] == 'Iris-setosa', 'sepal_width']

x2 = dataset.loc[dataset['species'] == 'Iris-versicolor', 'sepal_length']
y2 = dataset.loc[dataset['species'] == 'Iris-versicolor', 'sepal_width']

x3 = dataset.loc[dataset['species'] == 'Iris-virginica', 'sepal_length']
y3 = dataset.loc[dataset['species'] == 'Iris-virginica', 'sepal_width']

plt.plot(x1, y1, '.', color = 'crimson')
plt.plot(x2, y2, '.', color = 'green')
plt.plot(x3, y3, '.', color = 'blue')
plt.xlabel('sepal-length')
plt.ylabel('sepal-width')
plt.show()
```

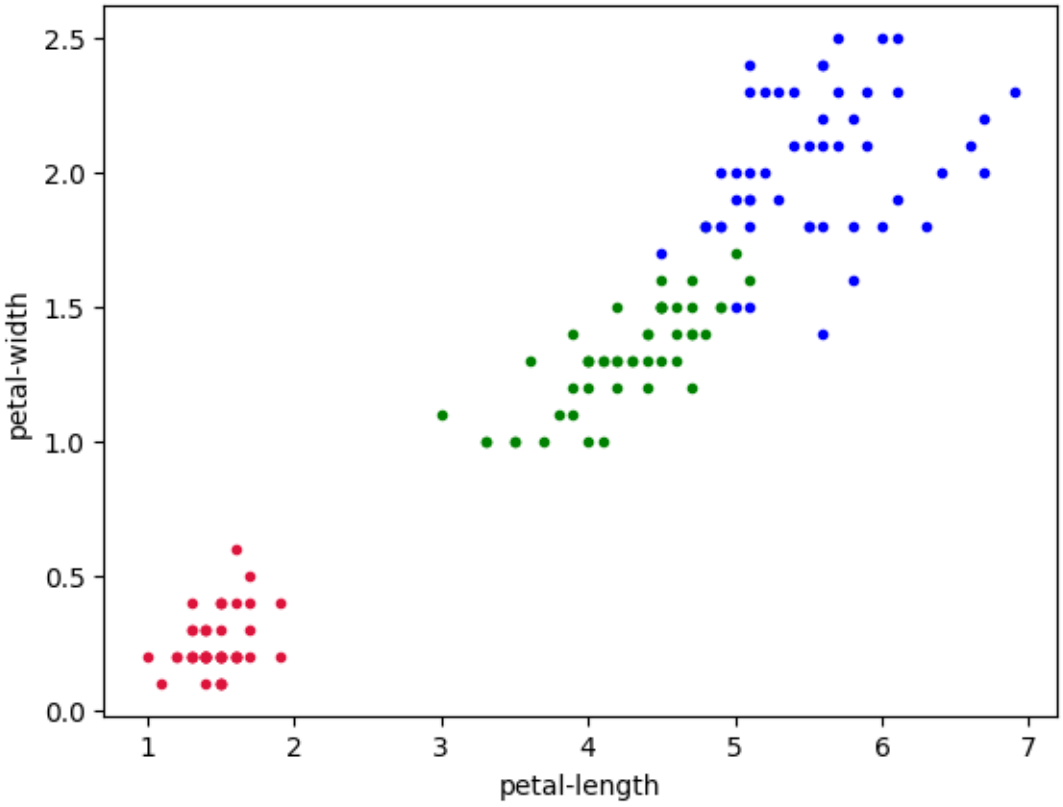


[6]:
```
x1 = dataset.loc[dataset['species'] == 'Iris-setosa', 'petal_length']
y1 = dataset.loc[dataset['species'] == 'Iris-setosa', 'petal_width']

x2 = dataset.loc[dataset['species'] == 'Iris-versicolor', 'petal_length']
y2 = dataset.loc[dataset['species'] == 'Iris-versicolor', 'petal_width']
```

```
x3 = dataset.loc[dataset['species'] == 'Iris-virginica', 'petal_length']
y3 = dataset.loc[dataset['species'] == 'Iris-virginica', 'petal_width']

plt.plot(x1, y1, '.', color = 'crimson')
plt.plot(x2, y2, '.', color = 'green')
plt.plot(x3, y3, '.', color = 'blue')
plt.xlabel('petal-length')
plt.ylabel('petal-width')
plt.show()
```



[7]:
```
X = dataset.iloc[:, :-1]
y = dataset.iloc[:, -1]
```

[8]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25,␣
↪random_state = 6)
```

[9]:
```
knn = KNeighborsClassifier(n_neighbors = 3, weights = 'distance')
dt = DecisionTreeClassifier()
lr = LogisticRegression(solver = 'liblinear')
acc = {}
```

[10]:
```
knn.fit(X_train, y_train)
lr.fit(X_train, y_train)
dt.fit(X_train, y_train)
```

[10]:
```
DecisionTreeClassifier()
```

[11]:
```
a,b,c = dt.score(X_test, y_test), lr.score(X_test, y_test), knn.score(X_test,␣
 ↪y_test)
acc = pandas.DataFrame({'models' :['DecisionTree', 'LogisticRegression',␣
 ↪'KNN'], 'accuracy': [a, b, c]})
acc
```

[11]:

	models	accuracy
0	DecisionTree	0.921053
1	LogisticRegression	0.973684
2	KNN	0.947368