Task_1

July 8, 2022

[1]:
```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
```

[2]:
```
iris_data=pd.read_csv("iris_data.csv")
```

[3]:
```
print(iris_data)
```

```
     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
..            …            …             …            …             …
145           6.7          3.0           5.2          2.3  Iris-virginica
146           6.3          2.5           5.0          1.9  Iris-virginica
147           6.5          3.0           5.2          2.0  Iris-virginica
148           6.2          3.4           5.4          2.3  Iris-virginica
149           5.9          3.0           5.1          1.8  Iris-virginica

[150 rows x 5 columns]
```

[4]:
```
print(iris_data.keys())
```

```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'Class'],
dtype='object')
```

[5]:
```
iris_data.head()
```

[5]:

	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

[6]:
```
iris_data.describe()
```

[6]:
```
       sepal_length  sepal_width  petal_length  petal_width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.057333      3.758000     1.199333
std        0.828066     0.435866      1.765298     0.762238
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
```

[7]:
```
iris_data['ID'] = "01"
iris_data.head()
```

[7]:

	sepal_length	sepal_width	petal_length	petal_width	Class	ID
0	5.1	3.5	1.4	0.2	Iris-setosa	01
1	4.9	3.0	1.4	0.2	Iris-setosa	01
2	4.7	3.2	1.3	0.2	Iris-setosa	01
3	4.6	3.1	1.5	0.2	Iris-setosa	01
4	5.0	3.6	1.4	0.2	Iris-setosa	01

[8]:
```
iris_data = iris_data.drop(columns = ['ID'])
iris_data.head()
```

[8]:

	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

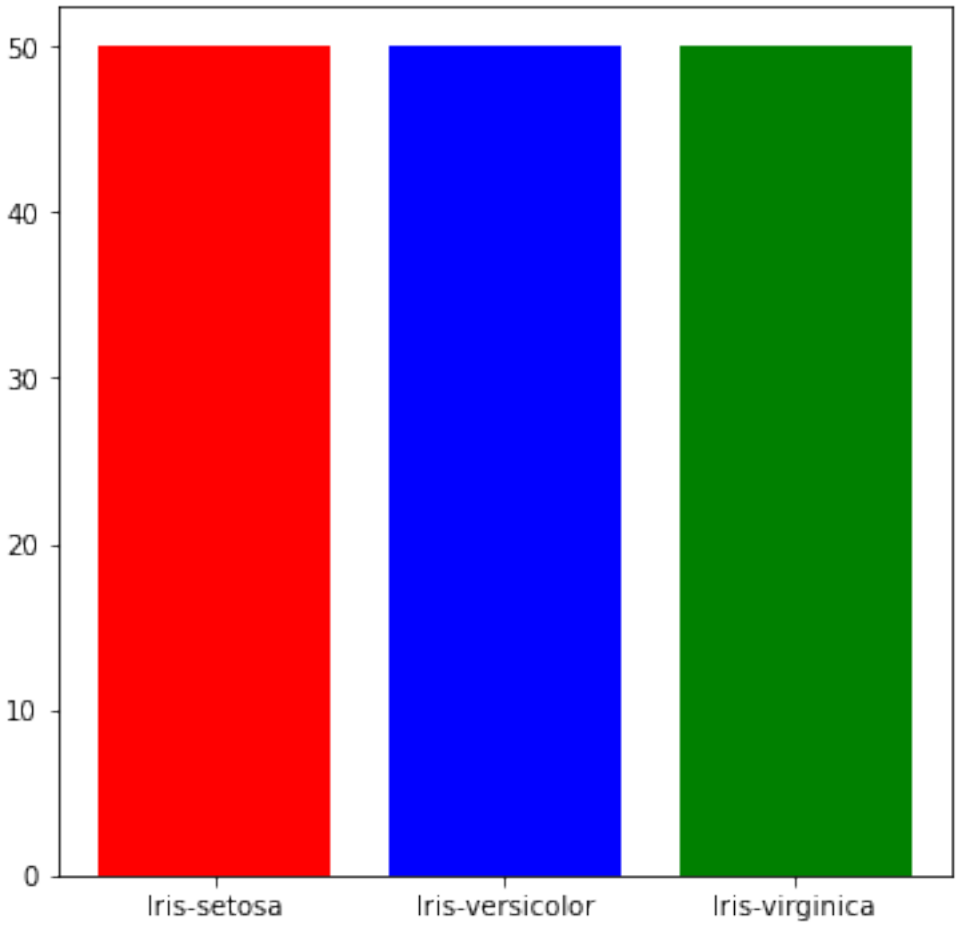
[9]:
```
iris_data['Class'].value_counts()
```

[9]:
```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Class, dtype: int64
```

[10]:
```
iris_data['Class'].value_counts().keys()
```

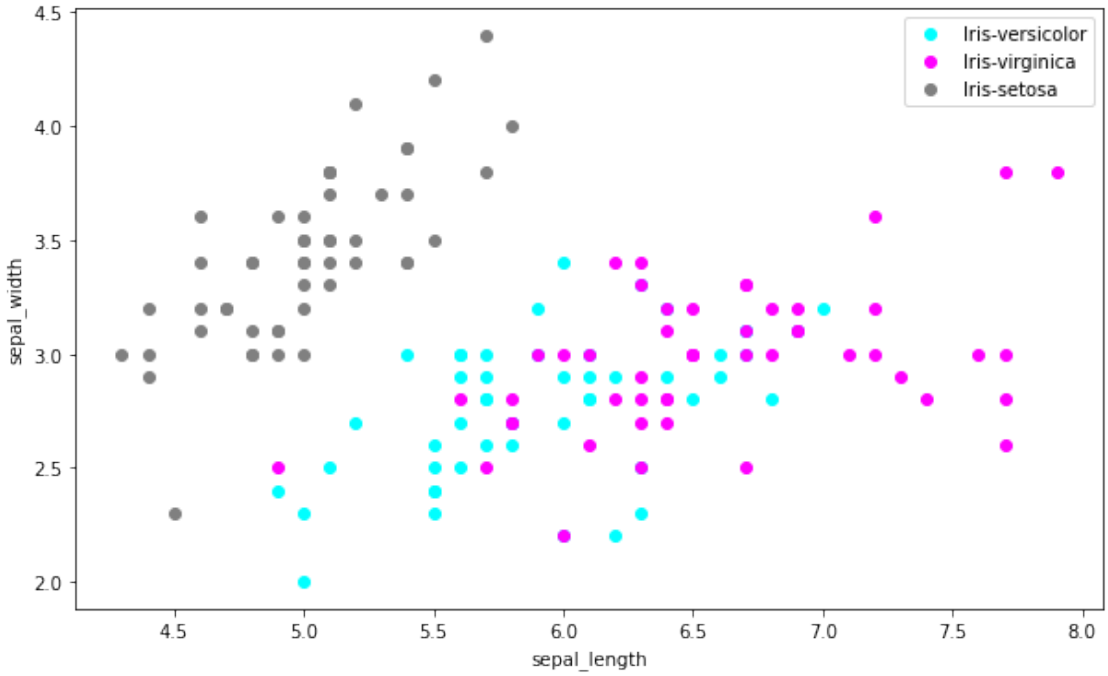
[10]:
```
Index(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype='object')
```

[11]:
```
plt.figure(figsize=(6,6))
plt.bar(list(iris_data['Class'].value_counts().keys()),list(iris_data['Class'].
 ↪value_counts()),color=["r","b","g"])
plt.show()
```



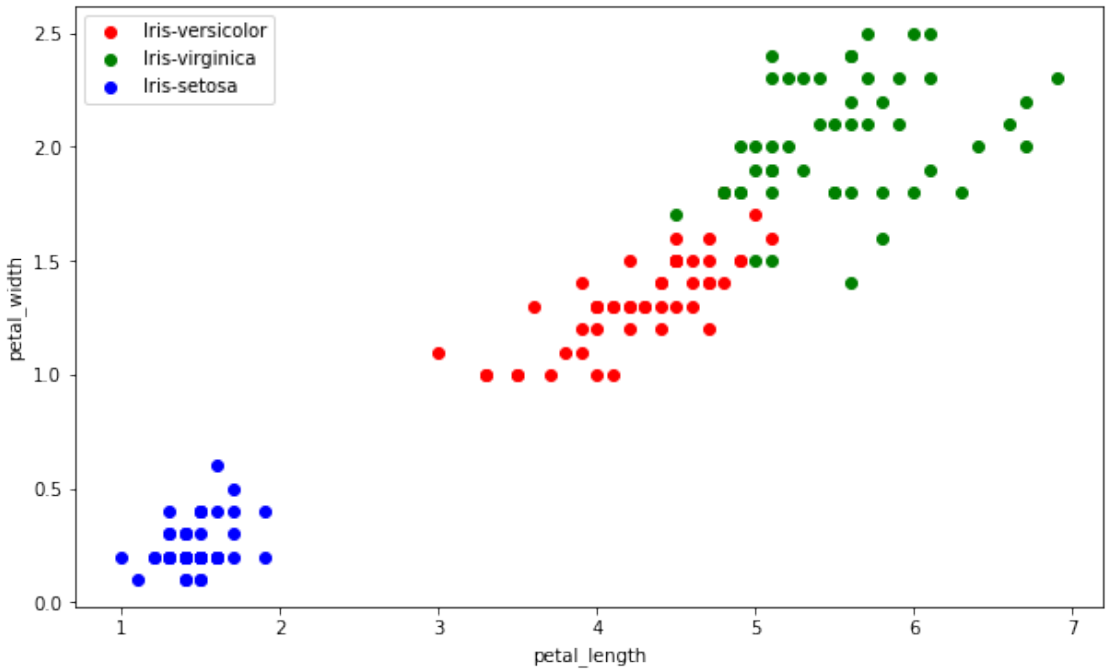
[12]:
```
Class = ['Iris-versicolor','Iris-virginica','Iris-setosa']
colors = ['cyan', 'magenta', 'gray']
plt_1 = plt.figure(figsize=(10, 6))
for i in range(3):
    x = iris_data[iris_data['Class'] == Class[i]]
    plt.scatter(x['sepal_length'], x['sepal_width'], c = colors[i],␣
 ↪label=Class[i])
plt.xlabel("sepal_length")
plt.ylabel("sepal_width")
plt.legend()
```

[12]:
```
<matplotlib.legend.Legend at 0x22ad4e56670>
```



[13]:
```
Class = ['Iris-versicolor','Iris-virginica','Iris-setosa']
colors = ['red', 'green', 'blue']
plt_1 = plt.figure(figsize=(10, 6))
for i in range(3):
    x = iris_data[iris_data['Class'] == Class[i]]
    plt.scatter(x['petal_length'], x['petal_width'], c = colors[i],␣
 ↪label=Class[i])
plt.xlabel("petal_length")
plt.ylabel("petal_width")
plt.legend()
```

[13]:
```
<matplotlib.legend.Legend at 0x22ad4f0c100>
```



[14]:
```
#importing sklearn
from sklearn.preprocessing import LabelEncoder
label = LabelEncoder()
iris_data['Class'] = label.fit_transform(iris_data['Class'])
iris_data.head()
#it removes alpha numeric values
```

[14]:

	sepal_length	sepal_width	petal_length	petal_width	Class
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

[15]:
```
#splitting test and train datasets
from sklearn.model_selection import train_test_split
```

[16]:
```
a = iris_data.drop(columns=['Class'])
b = iris_data['Class']
```

[17]:
```
a_train, a_test, b_train, b_test = train_test_split(a, b, test_size=0.3,␣
↪random_state=1)
```

[18]:
```
from sklearn.linear_model import LogisticRegression
logmodel = LogisticRegression(solver="liblinear")
```

[19]:
```
logmodel.fit(a_train , b_train)
```

[19]:
```
LogisticRegression(solver='liblinear')
```

[20]:
```
pred=logmodel.predict(a_test)
```

[21]:
```
from sklearn.metrics import classification_report
classification_report(b_test,pred)
```

[21]:
```
'              precision    recall  f1-score   support\n\n           0
1.00      1.00      1.00        14\n           1       1.00      0.72      0.84
18\n           2       0.72      1.00      0.84        13\n\n    accuracy
0.89        45\n   macro avg       0.91      0.91      0.89        45\nweighted
avg       0.92      0.89      0.89        45\n'
```

[22]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(b_test,pred)
```

[22]:
```
array([[14,  0,  0],
       [ 0, 13,  5],
       [ 0,  0, 13]], dtype=int64)
```

[23]:
```
from sklearn.metrics import accuracy_score
print("Accuracy of Logistic Regression:", accuracy_score(b_test,pred)*100)
```

```
Accuracy of Logistic Regression: 88.88888888888889
```

[ ]:
```

```