siddhu6

April 15, 2025

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import pandas as pd
```

[2]:
```
kd = pd.read_csv('iris.csv')
```

[3]:
```
kd.head()
```

[3]:

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

[4]:
```
%matplotlib inline
```

[5]:
```
img=mpimg.imread('iris_types.jpg')
```

[6]:
```
plt.figure(figsize=(20,40))
plt.axis('off')
plt.imshow(img)
```

[6]:
```
<matplotlib.image.AxesImage at 0x136bbfed050>
```

[7]:
```
X = kd.iloc[:,:4].values
y = kd['species'].values
```

[8]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20,␣
 ↪random_state = 82)
```

[9]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

[10]:
```
from sklearn.naive_bayes import GaussianNB
nvclassifier = GaussianNB()
nvclassifier.fit(X_train, y_train)
```

[10]:
```
GaussianNB()
```

[11]:
```
y_pred = nvclassifier.predict(X_test)
print(y_pred)
```

```
['Iris-virginica' 'Iris-virginica' 'setosa' 'setosa' 'setosa'
 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'setosa' 'setosa'
 'setosa' 'setosa' 'Iris-virginica' 'Iris-versicolor' 'setosa'
 'Iris-versicolor' 'setosa' 'Iris-virginica' 'setosa' 'Iris-virginica'
 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica' 'setosa'
 'Iris-virginica' 'Iris-versicolor']
```

[12]:
```
y_compare = np.vstack((y_test,y_pred)).T
```

[13]:
```
y_compare[:5,:]
```

[13]:
```
array([['Iris-virginica', 'Iris-virginica'],
       ['Iris-virginica', 'Iris-virginica'],
       ['setosa', 'setosa'],
       ['setosa', 'setosa'],
       ['setosa', 'setosa']], dtype=object)
```

[14]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[ 8  1  0]
 [ 1  9  0]
```

```
 [ 0  0 11]]
```

[15]:
```
a = cm.shape
corrPred = 0
falsePred = 0
```

[21]:
```
corrPred = 0
falsePred = 0

for row in range(a[0]):
    for c in range(a[1]):
        if row == c:
            corrPred += cm[row, c]
        else:
            falsePred += cm[row, c]

print('Correct predictions:', corrPred)
print('False predictions:', falsePred)
print('\n\nAccuracy of the Naive Bayes classification is:', corrPred / cm.sum())
```

```
Correct predictions: 28
False predictions: 2


Accuracy of the Naive Bayes classification is: 0.9333333333333333
```

[ ]:

[ ]: