

Data Analytics III

May 2, 2022

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

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

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

```
[3]:   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
```

```
[4]: iris.isnull().sum()
```

```
[4]: sepal_length    0
sepal_width        0
petal_length       0
petal_width        0
species            0
dtype: int64
```

```
[5]: from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
iris['species'] = pd.DataFrame(le.fit_transform(iris['species']))
iris
```

```
[5]:   sepal_length  sepal_width  petal_length  petal_width  species
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
..         ...         ...         ...         ...         ...
145        6.7         3.0         5.2         2.3         2
```

146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

[150 rows x 5 columns]

```
[6]: x = iris.loc[:,['sepal_length','sepal_width']]
     y = iris['species']
```

```
[7]: from sklearn.model_selection import train_test_split
     x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25,
     ↪random_state=0)
```

```
[8]: from sklearn.naive_bayes import GaussianNB
     gaussian = GaussianNB()
     gaussian.fit(x_train,y_train)
```

```
[8]: GaussianNB()
```

```
[9]: y_pred = gaussian.predict(x_test)
```

```
[10]: from sklearn.metrics import
     ↪accuracy_score,precision_score,recall_score,confusion_matrix
     accuracy = accuracy_score(y_test,y_pred)
     print(accuracy)
```

```
0.7631578947368421
```

```
[11]: precision = precision_score(y_test,y_pred,average='micro')
     print(precision)
```

```
0.7631578947368421
```

```
[12]: recall = recall_score(y_test,y_pred,average='micro')
     print(recall)
```

```
0.7631578947368421
```

```
[13]: confusion = confusion_matrix(y_test,y_pred)
     print(confusion)
```

```
[[13  0  0]
 [ 0 12  4]
 [ 0  5  4]]
```