

```
In [ ]: import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split, KFold, cross_val_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
```

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

```
Out[ ]:
```

	sepal.length	sepal.width	petal.length	petal.width	variety
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
...
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3.0	5.1	1.8	Virginica

150 rows × 5 columns

Standardising Columns

```
In [ ]: for i in range(4):
    x = df.iloc[:, i]
    df.iloc[:, i] = (x - np.mean(x)) / np.std(x)
df
```

```
Out[ ]:
```

	sepal.length	sepal.width	petal.length	petal.width	variety
0	-0.900681	1.019004	-1.340227	-1.315444	Setosa
1	-1.143017	-0.131979	-1.340227	-1.315444	Setosa
2	-1.385353	0.328414	-1.397064	-1.315444	Setosa
3	-1.506521	0.098217	-1.283389	-1.315444	Setosa
4	-1.021849	1.249201	-1.340227	-1.315444	Setosa
...
145	1.038005	-0.131979	0.819596	1.448832	Virginica
146	0.553333	-1.282963	0.705921	0.922303	Virginica
147	0.795669	-0.131979	0.819596	1.053935	Virginica
148	0.432165	0.788808	0.933271	1.448832	Virginica
149	0.068662	-0.131979	0.762758	0.790671	Virginica

150 rows × 5 columns

For 75% training and 25% testing

```
In [ ]: X = df.iloc[:, :4]
Y = df.iloc[:, 4]
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
                                                    random_state=104,
                                                    test_size=0.25,
                                                    shuffle=True)
```

Bayesian

```
In [ ]: model = GaussianNB()
model.fit(X_train, Y_train)
preds = model.predict(X_test)
accuracy = accuracy_score(preds, Y_test)
print(f"Accuracy : {accuracy}\n")
```

Accuracy : 0.9736842105263158

Decision Tree

```
In [ ]: model = DecisionTreeClassifier()
model.fit(X_train, Y_train)
preds = model.predict(X_test)
accuracy = accuracy_score(preds, Y_test)
print(f"Accuracy : {accuracy}\n")
```

Accuracy : 0.9736842105263158

KNN with k = 4

```
In [ ]: model = KNeighborsClassifier(n_neighbors=4)
        model.fit(X_train , Y_train)
        preds = model.predict(X_test)
        accuracy = accuracy_score(preds, Y_test)
        print(f"Accuracy : {accuracy}\n")
```

Accuracy : 0.9736842105263158