

Aryaman Gautam

J001

```
In [18]: import numpy as np
import pandas as pd
```

```
In [21]: data=pd.read_csv('Iris.csv')
data.head()
```

```
Out[21]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
In [22]: Species = list(set(data['Species']))
Specie1 = data[data['Species']==Species[0]]
Specie2 = data[data['Species']==Species[1]]
Specie3 = data[data['Species']==Species[2]]
```

```
In [23]: req_data = data.iloc[:,1:]
req_data.head()
```

```
Out[23]:
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
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

```
In [24]: shuffle_index = np.random.permutation(req_data.shape[0])
req_data = req_data.iloc[shuffle_index]
req_data.head()
```

```
Out[24]:
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
69	5.6	2.5	3.9	1.1	Iris-versicolor
9	4.9	3.1	1.5	0.1	Iris-setosa
25	5.0	3.0	1.6	0.2	Iris-setosa
81	5.5	2.4	3.7	1.0	Iris-versicolor
144	6.7	3.3	5.7	2.5	Iris-virginica

```
In [25]: train_size = int(req_data.shape[0]*0.7)
```

```
In [26]: train_df = req_data.iloc[:train_size,:]
test_df = req_data.iloc[train_size:,:]
train = train_df.values
test = test_df.values
y_true = test[:, -1]
print('Train_Shape: ', train_df.shape)
print('Test_Shape: ', test_df.shape)
```

```
Train_Shape: (105, 5)
Test_Shape: (45, 5)
```

```
In [27]:
```

```

from math import sqrt
def euclidean_distance(x_test, x_train):
    distance = 0
    for i in range(len(x_test)-1):
        distance += (x_test[i]-x_train[i])**2
    return sqrt(distance)

```

```

In [28]: def get_neighbors(x_test, x_train, num_neighbors):
    distances = []
    data = []
    for i in x_train:
        distances.append(euclidean_distance(x_test,i))
        data.append(i)
    distances = np.array(distances)
    data = np.array(data)
    sort_indexes = distances.argsort()
    data = data[sort_indexes]
    return data[:num_neighbors]

```

```

In [29]: def prediction(x_test, x_train, num_neighbors):
    classes = []
    neighbors = get_neighbors(x_test, x_train, num_neighbors)
    for i in neighbors:
        classes.append(i[-1])
    predicted = max(classes, key=classes.count)           #taking the most repeated class
    return predicted

```

```

In [30]: def accuracy(y_true, y_pred):
    num_correct = 0
    for i in range(len(y_true)):
        if y_true[i]==y_pred[i]:
            num_correct+=1
    accuracy = num_correct/len(y_true)
    return accuracy

```

```

In [31]: y_pred = []
    for i in test:

```

```
y_pred.append(prediction(i, train, 5))  
y_pred
```

```
Out[31]: ['Iris-setosa',  
          'Iris-versicolor',  
          'Iris-virginica',  
          'Iris-virginica',  
          'Iris-virginica',  
          'Iris-virginica',  
          'Iris-versicolor',  
          'Iris-versicolor',  
          'Iris-versicolor',  
          'Iris-versicolor',  
          'Iris-virginica',  
          'Iris-versicolor',  
          'Iris-setosa',  
          'Iris-setosa',  
          'Iris-setosa',  
          'Iris-virginica',  
          'Iris-virginica',  
          'Iris-virginica',  
          'Iris-versicolor',  
          'Iris-virginica',  
          'Iris-setosa',  
          'Iris-versicolor',  
          'Iris-virginica',  
          'Iris-virginica',  
          'Iris-versicolor',  
          'Iris-setosa',  
          'Iris-setosa',  
          'Iris-virginica',  
          'Iris-setosa',  
          'Iris-virginica',  
          'Iris-versicolor',  
          'Iris-virginica',  
          'Iris-versicolor',  
          'Iris-versicolor',  
          'Iris-versicolor',  
          'Iris-setosa',  
          'Iris-virginica',  
          'Iris-versicolor',  
          'Iris-versicolor',  
          'Iris-versicolor',  
          'Iris-versicolor']
```

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

```
In [32]: accuracy = accuracy(y_true, y_pred)
```

```
In [33]: accuracy
```

```
Out[33]: 0.9333333333333333
```

```
In [34]: test_df.insert(5, 'Predicted_Species', y_pred, False)
```

```
In [35]: test_df.sample(5)
```

```
Out[35]:
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species	Predicted_Species
123	6.3	2.7	4.9	1.8	Iris-virginica	Iris-virginica
82	5.8	2.7	3.9	1.2	Iris-versicolor	Iris-versicolor
24	4.8	3.4	1.9	0.2	Iris-setosa	Iris-setosa
134	6.1	2.6	5.6	1.4	Iris-virginica	Iris-virginica
105	7.6	3.0	6.6	2.1	Iris-virginica	Iris-virginica

```
In [ ]:
```