

Task 3

Dobe by Cherukuri Krishna Lathvik

Importing the Libraries

```
In [7]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

Done by Cherukuri Krishna Lathvik

Loading the dataset

```
In [8]: df = sns.load_dataset('iris')
df.head()
```

```
Out[8]:
```

	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

```
In [9]: df.shape
```

```
Out[9]: (150, 5)
```

```
In [10]: df.describe()
```

```
Out[10]:
```

	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

In [9]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  -
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object  
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [14]: `df['species'].value_counts()`

```
Out[14]: setosa      50
versicolor  50
virginica    50
Name: species, dtype: int64
```

In [15]: `df.isnull().any()`

```
Out[15]: sepal_length    False
sepal_width     False
petal_length     False
petal_width      False
species          False
dtype: bool
```

In [12]: `X = df.drop('species', axis=1)`
`y = df['species']`

Splitting the data into Training and Testing sets

In [13]: `from sklearn.model_selection import train_test_split`

In [18]: `X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_s`

Training a KNN Classifier on training data

In [19]: `from sklearn.neighbors import KNeighborsClassifier`

In [20]: `classifier = KNeighborsClassifier(n_neighbors = 3)`
`classifier.fit(X_train, y_train)`

Out[20]: `KNeighborsClassifier`
`KNeighborsClassifier(n_neighbors=3)`

Predicting the species

In [21]: `y_pred = classifier.predict(X_test)`

```
In [24]: from sklearn.metrics import accuracy_score
```

Calculating the Accuracy of the model

```
In [25]: accuracy = accuracy_score(y_test, y_pred)
print("Accuracy of the model : ", accuracy)
```

Accuracy of the model : 1.0