THE SPARKS FOUNDATION

Batch : September2022

Varsha Reddy

Task2 :Prediction using Unsupervised ML

Dataset : https://bit.ly/3kXTdox

Importing Libraries for Data Manipulation

In [1]:
```
#importing the libraries required for the Data Manipulation
import numpy as np
import pandas as pd
```

Importing Libraries for Data Visualization

In [2]:
```
#importing the libraries required for the Data Visualization
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import plotly.express as px
import plotly.graph_objects as go
from collections import Counter
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
```

Importing the dataset

In [3]:
```
df=pd.read_csv("Iris.csv")
```

Preprocessing the data

In [4]:
```
df.head()
```

Out[4]:

	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 [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             150 non-null    int64  
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [7]:
```
df.columns
```

Out[7]:
```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

In [8]:
```
df.describe()
```

Out[8]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [9]:
```
df.drop(axis=0, inplace=True)
df.shape
```

Out[9]:
```
(150, 6)
```

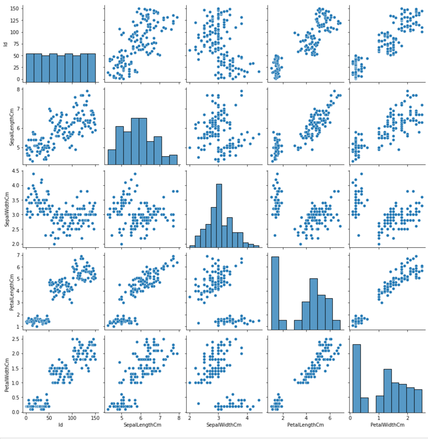
Re-Checking for Missing Data:

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             150 non-null    int64  
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object 
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

In [11]:
```
sns.pairplot(df)
```

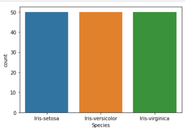
Out[11]:
```
<seaborn.axisgrid.PairGrid at 0x2587dfdb880>
```



In [12]:
```
df['Species'].value_counts()
```

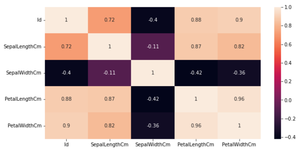
Out[12]:
```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

In [13]:
```
sns.countplot(df['Species']);
```



Visualizing the Data

In [14]:
```
correl=df.corr()
plt.figure(figsize=(10,5))
sns.heatmap(correl,annot=True)
plt.show()
```

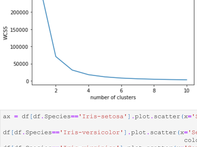


Applying KMeans

In [15]:
```
x=df.iloc[:,[0,1,2,3]].values
```

In [16]:
```
from sklearn.cluster import KMeans
wcss = []
for i in range(1,11):
    kmeans =KMeans(n_clusters = i,init = 'k-means++',max_iter=300,n_init = 10,random_state=0)
    kmeans.fit(x)
    wcss.append(kmeans.inertia_)
```

In [17]:
```
plt.plot(range(1,11),wcss)
plt.title('The elbow method')
plt.xlabel('number of clusters')
plt.ylabel('WCSS')
plt.show()
```



In [18]:
```
ax = df[df.Species=='Iris-setosa'].plot.scatter(x='SepalLengthCm', y='SepalWidthCm',
                                                color='red', label='Iris - Setosa')
df[df.Species=='Iris-versicolor'].plot.scatter(x='SepalLengthCm', y='SepalWidthCm',
                                               color='green', label='Iris - Versicolor', ax=ax)
df[df.Species=='Iris-virginica'].plot.scatter(x='SepalLengthCm', y='SepalWidthCm',
                                              color='blue', label='Iris - Virginica', ax=ax)
ax.set_title("Scatter Plot")
```

Out[18]:
```
Text(0.5, 1.0, 'Scatter Plot')
```

In [ ]:

In [ ]:

In [ ]: