```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.datasets import load_iris
```

```
iris=pd.read_csv("/content/IRIS.csv")
```

```
iris
```

```
     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
...           ...          ...           ...          ...             ...
145           6.7          3.0           5.2          2.3  Iris-virginica
146           6.3          2.5           5.0          1.9  Iris-virginica
147           6.5          3.0           5.2          2.0  Iris-virginica
148           6.2          3.4           5.4          2.3  Iris-virginica
149           5.9          3.0           5.1          1.8  Iris-virginica
150 rows × 5 columns
```

```
iris.head()
```

	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

```
iris.head(2)
```

	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

```
iris.tail()
```

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

```
iris.shape
```

```
(150, 5)
```

```
iris.size
```

```
750
```

```
iris.ndim
```

```
2
```

```
dir(iris)
```

```
 'select_dtypes',
 'sem',
 'sepal_length',
 'sepal_width',
 'set_axis',
 'set_index',
 'shape',
 'shift',
 'size',
 'skew',
 'slice_shift',
 'sort_index',
 'sort_values',
 'species',
 'squeeze',
 'stack',
 'std',
 'style',
 'sub',
 'subtract',
 'sum',
 'swapaxes',
 'swaplevel',
 'tail',
 'take',
 'to_clipboard',
 'to_csv',
 'to_dict',
 'to_excel',
 'to_feather',
 'to_gbq',
 'to_hdf',
 'to_html',
 'to_json',
 'to_latex',
 'to_markdown',
 'to_numpy',
 'to_parquet',
 'to_period',
 'to_pickle',
 'to_records',
 'to_sql',
 'to_stata',
 'to_string',
 'to_timestamp',
 'to_xarray',
 'transform',
 'transpose',
 'truediv',
 'truncate',
 'tz_convert',
 'tz_localize',
 'unstack',
 'update',
 'value_counts',
 'values',
 'var',
 'where',
 'xs']
```

```
iris=load_iris()
```

```
dir(iris)
```

```
['DESCR', 'data', 'feature_names', 'filename', 'target', 'target_names']
```

```
df = pd.DataFrame(iris.data,columns=iris.feature_names)
df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

```
df['Flower ID']=iris.target
df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Flower ID
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

```
df.isnull().sum()
```

```
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
Flower ID            0
dtype: int64
```

```
df.shape
```

```
(150, 5)
```

```
df.dtypes
```

```
sepal length (cm)    float64
sepal width (cm)     float64
petal length (cm)    float64
petal width (cm)     float64
Flower ID              int64
dtype: object
```

```
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 (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
 4   Flower ID          150 non-null    int64  
dtypes: float64(4), int64(1)
memory usage: 6.0 KB
```

```
df.describe()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Flower ID
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333	1.000000
std	0.828066	0.435866	1.765298	0.762238	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000

```
df.duplicated().sum()
```

```
1
```

75%	6.400000	3.300000	5.100000	1.800000	2.000000

```
df[df.duplicated()]
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Flower ID
142	5.8	2.7	5.1	1.9	2

```
df=df.drop(142)
```

```
df.skew()
```

```
sepal length (cm)    0.312826
sepal width (cm)     0.307149
petal length (cm)   -0.263101
petal width (cm)    -0.090076
Flower ID            0.012434
dtype: float64
```

```
df.kurt()
```

```
sepal length (cm)   -0.569006
sepal width (cm)     0.226236
petal length (cm)   -1.408270
petal width (cm)    -1.339953
```

```
Flower ID           -1.504770
dtype: float64
```

```
df.shape
```

```
(149, 5)
```

```
iris.target_names
```

```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

```
iris.target
```

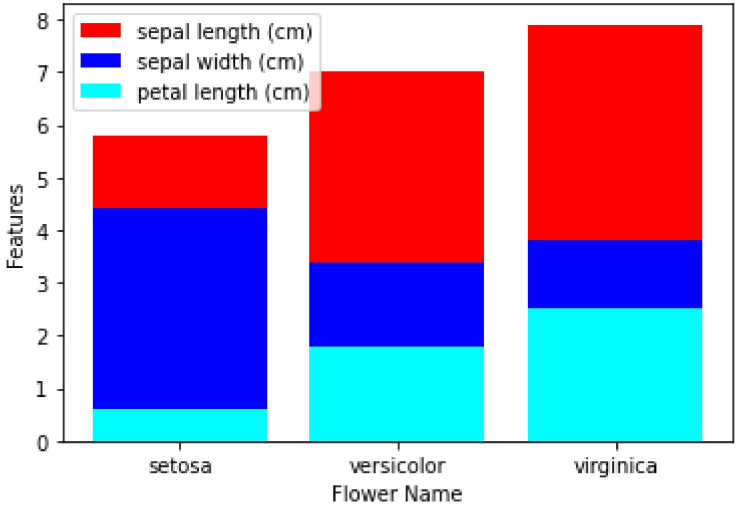
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
mapping={
    0:'setosa',
    1:'versicolor',
    2:'virginica'
}
df['Flower Name']=df['Flower ID'].map(mapping)
df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Flower ID	Flower Name
0	5.1	3.5	1.4	0.2	0	setosa
1	4.9	3.0	1.4	0.2	0	setosa

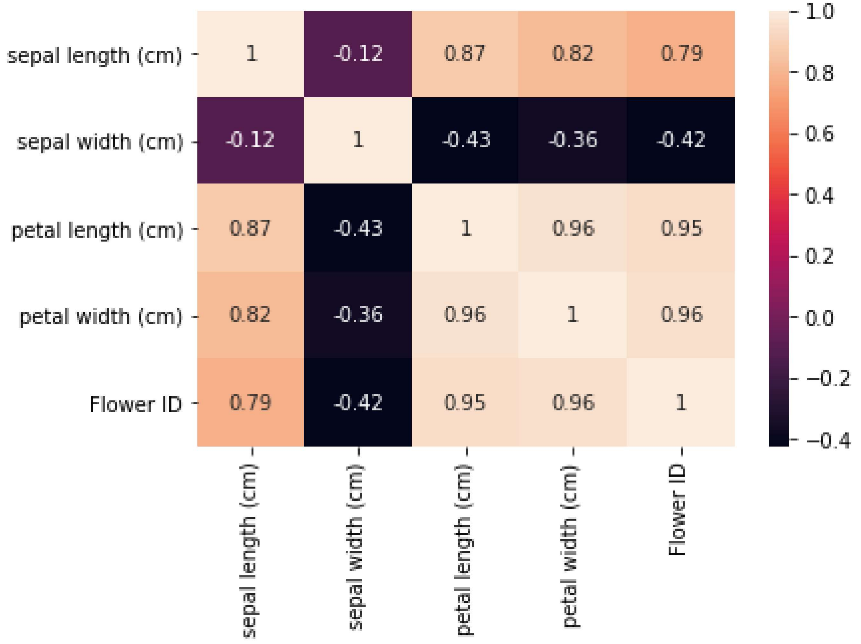
```
plt.xlabel('Flower Name')
plt.ylabel('Features')
plt.bar(df['Flower Name'],df['sepal length (cm)'],color='red')
plt.legend()
plt.bar(df['Flower Name'],df['sepal width (cm)'],color='blue')
plt.legend()
# plt.bar(df['Flower Name'],df['petal width (cm)'],color='brown')
plt.bar(df['Flower Name'],df['petal width (cm)'],color='cyan')
plt.legend(df.columns,loc=2)
```

```
No handles with labels found to put in legend.
No handles with labels found to put in legend.
<matplotlib.legend.Legend at 0x7f88fd2dbcd0>
```



```
sns.heatmap(df.corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f88fd119a10>
```



```
df.describe(include='all')
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Flower ID	Flower Name
count	149.000000	149.000000	149.000000	149.000000	149.000000	149
freq	NaN	NaN	NaN	NaN	NaN	50

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
```

```
df1=scaler.fit_transform(df.drop(columns=['Flower ID','Flower Name']))
```

```
scaled_df=pd.DataFrame(df1,columns=iris.feature_names)
scaled_df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	0.222222	0.625000	0.067797	0.041667
1	0.166667	0.416667	0.067797	0.041667
2	0.111111	0.500000	0.050847	0.041667
3	0.083333	0.458333	0.084746	0.041667
4	0.194444	0.666667	0.067797	0.041667

```
scaled_df.describe(include='all')
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
count	149.000000	149.000000	149.000000	149.000000
mean	0.428784	0.441555	0.465931	0.456096
std	0.230792	0.181809	0.299626	0.317759
min	0.000000	0.000000	0.000000	0.000000

```
from sklearn.preprocessing import StandardScaler
std_scaler=StandardScaler()
```

75%	0.583333	0.541667	0.694915	0.708333

```
std_scaled_value=std_scaler.fit_transform(df.drop(columns=['Flower Name','Flower ID']))
std_scaled_df=pd.DataFrame(std_scaled_value,columns=iris.feature_names)
std_scaled_df.head()
```

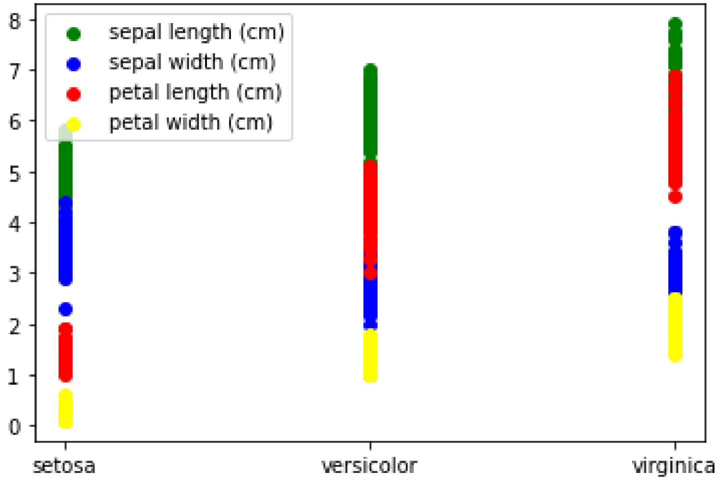
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	-0.898033	1.012401	-1.333255	-1.308624
1	-1.139562	-0.137353	-1.333255	-1.308624
2	-1.381091	0.322549	-1.390014	-1.308624
3	-1.501855	0.092598	-1.276496	-1.308624
4	-1.018798	1.242352	-1.333255	-1.308624

```
std_scaled_df.describe(include='all')
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
count	1.490000e+02	1.490000e+02	1.490000e+02	1.490000e+02
mean	-2.303247e-15	-1.951273e-15	-1.377720e-15	-1.186970e-15
std	1.003373e+00	1.003373e+00	1.003373e+00	1.003373e+00
min	-1.864149e+00	-2.436862e+00	-1.560289e+00	-1.440192e+00

```
plt.scatter(df['Flower Name'],df['sepal length (cm)'],color='green')
plt.scatter(df['Flower Name'],df['sepal width (cm)'],color='blue')
plt.scatter(df['Flower Name'],df['petal length (cm)'],color='red')
plt.scatter(df['Flower Name'],df['petal width (cm)'],color='yellow')
plt.legend(df.columns,loc=2)
```

```
<matplotlib.legend.Legend at 0x7f88f30abd10>
```



Splitting Train data and test data

```
dummies=pd.get_dummies(df['Flower Name'])
dummies
```

	setosa	versicolor	virginica
0	1	0	0
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
...	...	...	...
145	0	0	1
146	0	0	1
147	0	0	1
148	0	0	1
149	0	0	1

149 rows × 3 columns

```
mapping={
    0:'setosa',
    1:'versicolor',
    2:'virginica'
}
df['Flower Name']=df['Flower ID'].apply(lambda x: mapping[x])
```

```
df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Flower ID	Flower Name
0	5.1	3.5	1.4	0.2	0	setosa
1	4.9	3.0	1.4	0.2	0	setosa

```
mapping={
    0:'setosa',
    1:'versicolor',
    2:'virginica'
}
new_df['Flower Name']=df['Flower ID'].apply(lambda x: mapping[x])
```

```
new_df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	setosa	versicolor	virginica	Flower Name
0	0.222222	0.625000	0.067797	0.041667	1.0	0.0	0.0	setosa
1	0.166667	0.416667	0.067797	0.041667	1.0	0.0	0.0	setosa
2	0.111111	0.500000	0.050847	0.041667	1.0	0.0	0.0	setosa
3	0.083333	0.458333	0.084746	0.041667	1.0	0.0	0.0	setosa
4	0.194444	0.666667	0.067797	0.041667	1.0	0.0	0.0	setosa

```

new_df=pd.concat([scaled_df,dummies],axis=1)
new_df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	setosa	versicolor	virginica
0	0.222222	0.625000	0.067797	0.041667	1.0	0.0	0.0
4	0.194444	0.666667	0.067797	0.041667	1.0	0.0	0.0

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(new_df.drop(columns=['setosa','versicolor','virginica','Flower ID','Flower Name']),new
```

```
len(x_train)
```

```
len(y_train)
```

```
len(x_test)
```

```
len(y_test)
```