IMPORTING PYTHON LIBRARIES

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder,MinMaxScaler
from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.feature_selection import chi2
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.metrics import
classification_report,confusion_matrix,ConfusionMatrixDisplay,accuracy
_score
```

LOADING DATASET

```
df=pd.read_csv('/home/anusha/Desktop/collab/IRIS.csv')
df

     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 x 5 columns]
```

DATA PREPROCESSING

```
#Printing first five rows
df.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
```

_

```
#Printing last five rows
df.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
```

_

```
#Printing datatype
df.dtypes
```

```
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
species          object
dtype: object
```

_

```
#Checking for missing value
df.isna().sum()
```

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
```

```
species         0
dtype: int64
```

_

```
#Number of rows and columns
df.shape
```

```
(150, 5)
```

_

```
#Describing Dataset
df.describe()
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
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

_

```
df.nunique()
```

```
sepal_length    35
sepal_width     23
petal_length    43
petal_width     22
species          3
dtype: int64
```

_

```
df['species'].value_counts()
```

```
species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

This is a balanced dataset

DATA VISUALISATION

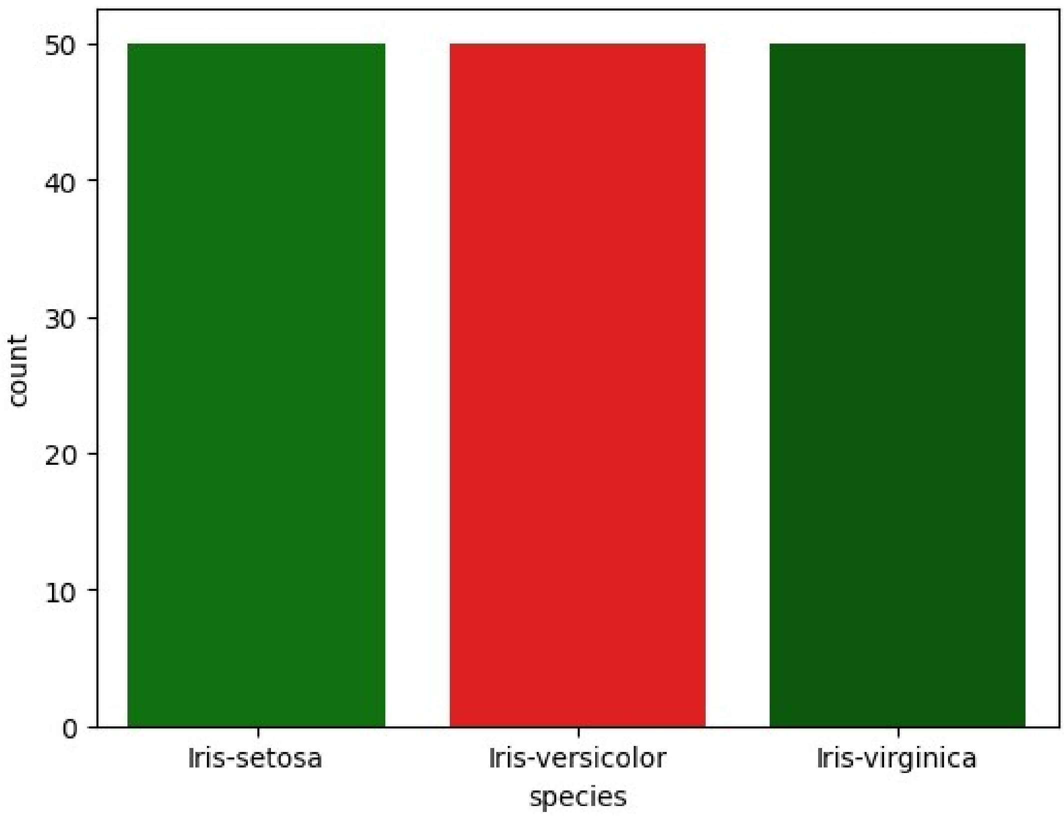
```
custom_palette=['green','red','darkgreen']
sns.countplot(x='species',data=df,palette=custom_palette)
```

```
/tmp/ipykernel_8167/2449826961.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.countplot(x='species',data=df,palette=custom_palette)
```

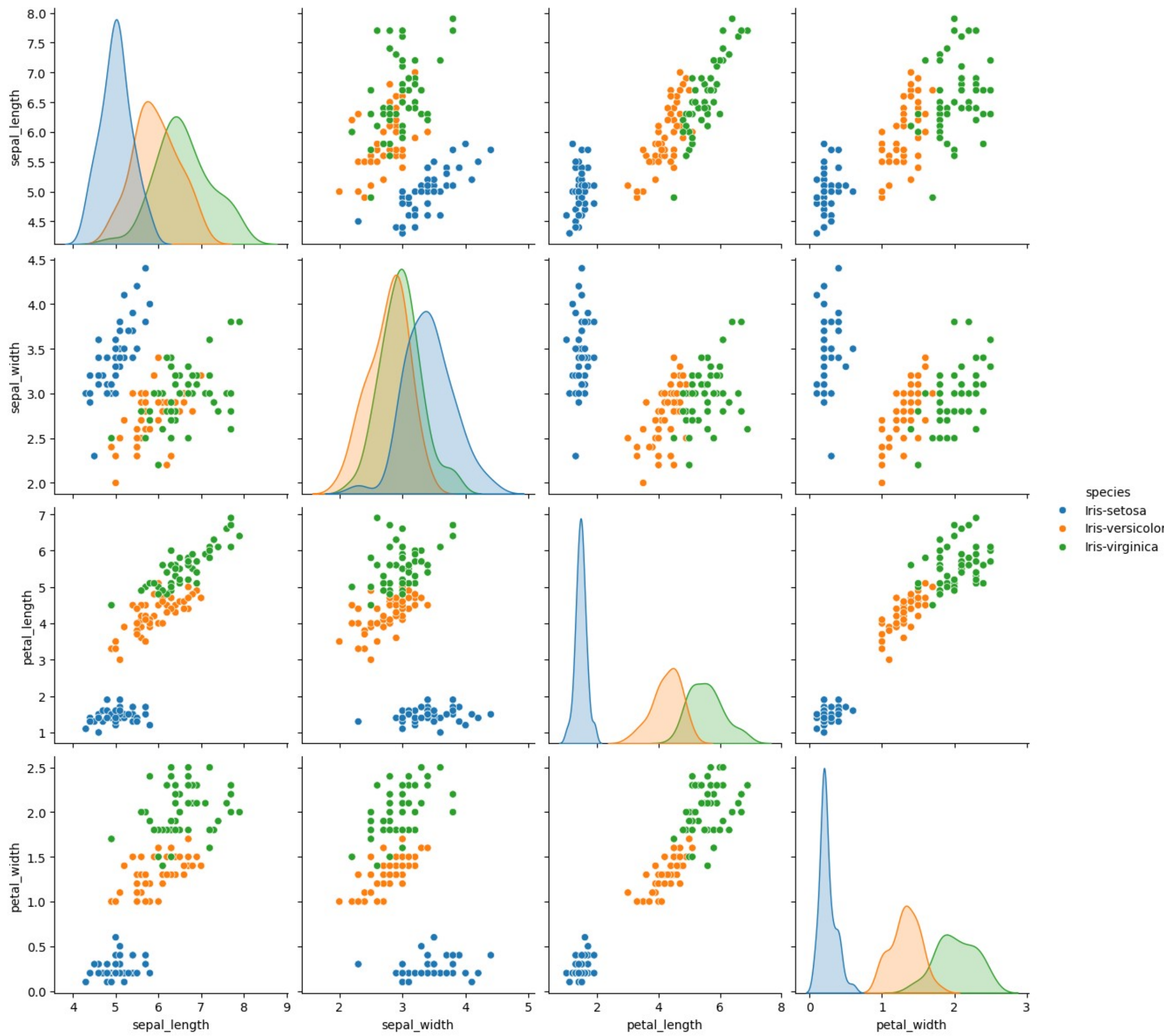
```
<Axes: xlabel='species', ylabel='count'>
```



_

```
sns.pairplot(data=df,hue='species',height=3)
```

```
<seaborn.axisgrid.PairGrid at 0x7f9a9cee9c70>
```



SEPERATING X AND Y

```
x=df.iloc[:,:-1].values
x
```

```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
```

```
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
       [5.7, 3.8, 1.7, 0.3],
       [5.1, 3.8, 1.5, 0.3],
       [5.4, 3.4, 1.7, 0.2],
       [5.1, 3.7, 1.5, 0.4],
       [4.6, 3.6, 1. , 0.2],
       [5.1, 3.3, 1.7, 0.5],
       [4.8, 3.4, 1.9, 0.2],
       [5. , 3. , 1.6, 0.2],
       [5. , 3.4, 1.6, 0.4],
       [5.2, 3.5, 1.5, 0.2],
       [5.2, 3.4, 1.4, 0.2],
       [4.7, 3.2, 1.6, 0.2],
       [4.8, 3.1, 1.6, 0.2],
       [5.4, 3.4, 1.5, 0.4],
       [5.2, 4.1, 1.5, 0.1],
       [5.5, 4.2, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5. , 3.2, 1.2, 0.2],
       [5.5, 3.5, 1.3, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [4.4, 3. , 1.3, 0.2],
       [5.1, 3.4, 1.5, 0.2],
       [5. , 3.5, 1.3, 0.3],
       [4.5, 2.3, 1.3, 0.3],
       [4.4, 3.2, 1.3, 0.2],
       [5. , 3.5, 1.6, 0.6],
       [5.1, 3.8, 1.9, 0.4],
       [4.8, 3. , 1.4, 0.3],
       [5.1, 3.8, 1.6, 0.2],
       [4.6, 3.2, 1.4, 0.2],
       [5.3, 3.7, 1.5, 0.2],
       [5. , 3.3, 1.4, 0.2],
       [7. , 3.2, 4.7, 1.4],
       [6.4, 3.2, 4.5, 1.5],
       [6.9, 3.1, 4.9, 1.5],
       [5.5, 2.3, 4. , 1.3],
       [6.5, 2.8, 4.6, 1.5],
       [5.7, 2.8, 4.5, 1.3],
       [6.3, 3.3, 4.7, 1.6],
       [4.9, 2.4, 3.3, 1. ],
       [6.6, 2.9, 4.6, 1.3],
       [5.2, 2.7, 3.9, 1.4],
       [5. , 2. , 3.5, 1. ],
```

```
       [5.9, 3. , 4.2, 1.5],
       [6. , 2.2, 4. , 1. ],
       [6.1, 2.9, 4.7, 1.4],
       [5.6, 2.9, 3.6, 1.3],
       [6.7, 3.1, 4.4, 1.4],
       [5.6, 3. , 4.5, 1.5],
       [5.8, 2.7, 4.1, 1. ],
       [6.2, 2.2, 4.5, 1.5],
       [5.6, 2.5, 3.9, 1.1],
       [5.9, 3.2, 4.8, 1.8],
       [6.1, 2.8, 4. , 1.3],
       [6.3, 2.5, 4.9, 1.5],
       [6.1, 2.8, 4.7, 1.2],
       [6.4, 2.9, 4.3, 1.3],
       [6.6, 3. , 4.4, 1.4],
       [6.8, 2.8, 4.8, 1.4],
       [6.7, 3. , 5. , 1.7],
       [6. , 2.9, 4.5, 1.5],
       [5.7, 2.6, 3.5, 1. ],
       [5.5, 2.4, 3.8, 1.1],
       [5.5, 2.4, 3.7, 1. ],
       [5.8, 2.7, 3.9, 1.2],
       [6. , 2.7, 5.1, 1.6],
       [5.4, 3. , 4.5, 1.5],
       [6. , 3.4, 4.5, 1.6],
       [6.7, 3.1, 4.7, 1.5],
       [6.3, 2.3, 4.4, 1.3],
       [5.6, 3. , 4.1, 1.3],
       [5.5, 2.5, 4. , 1.3],
       [5.5, 2.6, 4.4, 1.2],
       [6.1, 3. , 4.6, 1.4],
       [5.8, 2.6, 4. , 1.2],
       [5. , 2.3, 3.3, 1. ],
       [5.6, 2.7, 4.2, 1.3],
       [5.7, 3. , 4.2, 1.2],
       [5.7, 2.9, 4.2, 1.3],
       [6.2, 2.9, 4.3, 1.3],
       [5.1, 2.5, 3. , 1.1],
       [5.7, 2.8, 4.1, 1.3],
       [6.3, 3.3, 6. , 2.5],
       [5.8, 2.7, 5.1, 1.9],
       [7.1, 3. , 5.9, 2.1],
       [6.3, 2.9, 5.6, 1.8],
       [6.5, 3. , 5.8, 2.2],
       [7.6, 3. , 6.6, 2.1],
       [4.9, 2.5, 4.5, 1.7],
       [7.3, 2.9, 6.3, 1.8],
       [6.7, 2.5, 5.8, 1.8],
       [7.2, 3.6, 6.1, 2.5],
```

```
       [6.5, 3.2, 5.1, 2. ],
       [6.4, 2.7, 5.3, 1.9],
       [6.8, 3. , 5.5, 2.1],
       [5.7, 2.5, 5. , 2. ],
       [5.8, 2.8, 5.1, 2.4],
       [6.4, 3.2, 5.3, 2.3],
       [6.5, 3. , 5.5, 1.8],
       [7.7, 3.8, 6.7, 2.2],
       [7.7, 2.6, 6.9, 2.3],
       [6. , 2.2, 5. , 1.5],
       [6.9, 3.2, 5.7, 2.3],
       [5.6, 2.8, 4.9, 2. ],
       [7.7, 2.8, 6.7, 2. ],
       [6.3, 2.7, 4.9, 1.8],
       [6.7, 3.3, 5.7, 2.1],
       [7.2, 3.2, 6. , 1.8],
       [6.2, 2.8, 4.8, 1.8],
       [6.1, 3. , 4.9, 1.8],
       [6.4, 2.8, 5.6, 2.1],
       [7.2, 3. , 5.8, 1.6],
       [7.4, 2.8, 6.1, 1.9],
       [7.9, 3.8, 6.4, 2. ],
       [6.4, 2.8, 5.6, 2.2],
       [6.3, 2.8, 5.1, 1.5],
       [6.1, 2.6, 5.6, 1.4],
       [7.7, 3. , 6.1, 2.3],
       [6.3, 3.4, 5.6, 2.4],
       [6.4, 3.1, 5.5, 1.8],
       [6. , 3. , 4.8, 1.8],
       [6.9, 3.1, 5.4, 2.1],
       [6.7, 3.1, 5.6, 2.4],
       [6.9, 3.1, 5.1, 2.3],
       [5.8, 2.7, 5.1, 1.9],
       [6.8, 3.2, 5.9, 2.3],
       [6.7, 3.3, 5.7, 2.5],
       [6.7, 3. , 5.2, 2.3],
       [6.3, 2.5, 5. , 1.9],
       [6.5, 3. , 5.2, 2. ],
       [6.2, 3.4, 5.4, 2.3],
       [5.9, 3. , 5.1, 1.8]])
```

_

```
y=df.iloc[:,-1].values
y
```

```
array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
```

```
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-
versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica'], dtype=object)
```

TRAIN TEST SPLIT

A train_test_split function is used for spliting the datasets into a training set and a testing set. The training set is used for training the model, and the testing set is used to testing the model.

This allows us to train the models on the training set, and then test their accuracy on the unseen testing set.

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,rand
om_state=42)
```

_

```
x_train
```

```
array([[5.5, 2.4, 3.7, 1. ],
       [6.3, 2.8, 5.1, 1.5],
       [6.4, 3.1, 5.5, 1.8],
       [6.6, 3. , 4.4, 1.4],
       [7.2, 3.6, 6.1, 2.5],
       [5.7, 2.9, 4.2, 1.3],
       [7.6, 3. , 6.6, 2.1],
       [5.6, 3. , 4.5, 1.5],
       [5.1, 3.5, 1.4, 0.2],
       [7.7, 2.8, 6.7, 2. ],
       [5.8, 2.7, 4.1, 1. ],
       [5.2, 3.4, 1.4, 0.2],
       [5. , 3.5, 1.3, 0.3],
       [5.1, 3.8, 1.9, 0.4],
       [5. , 2. , 3.5, 1. ],
       [6.3, 2.7, 4.9, 1.8],
       [4.8, 3.4, 1.9, 0.2],
       [5. , 3. , 1.6, 0.2],
       [5.1, 3.3, 1.7, 0.5],
       [5.6, 2.7, 4.2, 1.3],
       [5.1, 3.4, 1.5, 0.2],
       [5.7, 3. , 4.2, 1.2],
       [7.7, 3.8, 6.7, 2.2],
       [4.6, 3.2, 1.4, 0.2],
       [6.2, 2.9, 4.3, 1.3],
       [5.7, 2.5, 5. , 2. ],
       [5.5, 4.2, 1.4, 0.2],
       [6. , 3. , 4.8, 1.8],
       [5.8, 2.7, 5.1, 1.9],
       [6. , 2.2, 4. , 1. ],
       [5.4, 3. , 4.5, 1.5],
       [6.2, 3.4, 5.4, 2.3],
       [5.5, 2.3, 4. , 1.3],
       [5.4, 3.9, 1.7, 0.4],
       [5. , 2.3, 3.3, 1. ],
       [6.4, 2.7, 5.3, 1.9],
       [5. , 3.3, 1.4, 0.2],
       [5. , 3.2, 1.2, 0.2],
       [5.5, 2.4, 3.8, 1.1],
```

```
       [6.7, 3. , 5. , 1.7],
       [4.9, 3.1, 1.5, 0.1],
       [5.8, 2.8, 5.1, 2.4],
       [5. , 3.4, 1.5, 0.2],
       [5. , 3.5, 1.6, 0.6],
       [5.9, 3.2, 4.8, 1.8],
       [5.1, 2.5, 3. , 1.1],
       [6.9, 3.2, 5.7, 2.3],
       [6. , 2.7, 5.1, 1.6],
       [6.1, 2.6, 5.6, 1.4],
       [7.7, 3. , 6.1, 2.3],
       [5.5, 2.5, 4. , 1.3],
       [4.4, 2.9, 1.4, 0.2],
       [4.3, 3. , 1.1, 0.1],
       [6. , 2.2, 5. , 1.5],
       [7.2, 3.2, 6. , 1.8],
       [4.6, 3.1, 1.5, 0.2],
       [5.1, 3.5, 1.4, 0.3],
       [4.4, 3. , 1.3, 0.2],
       [6.3, 2.5, 4.9, 1.5],
       [6.3, 3.4, 5.6, 2.4],
       [4.6, 3.4, 1.4, 0.3],
       [6.8, 3. , 5.5, 2.1],
       [6.3, 3.3, 6. , 2.5],
       [4.7, 3.2, 1.3, 0.2],
       [6.1, 2.9, 4.7, 1.4],
       [6.5, 2.8, 4.6, 1.5],
       [6.2, 2.8, 4.8, 1.8],
       [7. , 3.2, 4.7, 1.4],
       [6.4, 3.2, 5.3, 2.3],
       [5.1, 3.8, 1.6, 0.2],
       [6.9, 3.1, 5.4, 2.1],
       [5.9, 3. , 4.2, 1.5],
       [6.5, 3. , 5.2, 2. ],
       [5.7, 2.6, 3.5, 1. ],
       [5.2, 2.7, 3.9, 1.4],
       [6.1, 3. , 4.6, 1.4],
       [4.5, 2.3, 1.3, 0.3],
       [6.6, 2.9, 4.6, 1.3],
       [5.5, 2.6, 4.4, 1.2],
       [5.3, 3.7, 1.5, 0.2],
       [5.6, 3. , 4.1, 1.3],
       [7.3, 2.9, 6.3, 1.8],
       [6.7, 3.3, 5.7, 2.1],
       [5.1, 3.7, 1.5, 0.4],
       [4.9, 2.4, 3.3, 1. ],
       [6.7, 3.3, 5.7, 2.5],
       [7.2, 3. , 5.8, 1.6],
       [4.9, 3.1, 1.5, 0.1],
```

```
       [6.7, 3.1, 5.6, 2.4],
       [4.9, 3. , 1.4, 0.2],
       [6.9, 3.1, 4.9, 1.5],
       [7.4, 2.8, 6.1, 1.9],
       [6.3, 2.9, 5.6, 1.8],
       [5.7, 2.8, 4.1, 1.3],
       [6.5, 3. , 5.5, 1.8],
       [6.3, 2.3, 4.4, 1.3],
       [6.4, 2.9, 4.3, 1.3],
       [5.6, 2.8, 4.9, 2. ],
       [5.9, 3. , 5.1, 1.8],
       [5.4, 3.4, 1.7, 0.2],
       [6.1, 2.8, 4. , 1.3],
       [4.9, 2.5, 4.5, 1.7],
       [5.8, 4. , 1.2, 0.2],
       [5.8, 2.6, 4. , 1.2],
       [7.1, 3. , 5.9, 2.1]])
```

_

```
x_test
```

```
array([[6.1, 2.8, 4.7, 1.2],
       [5.7, 3.8, 1.7, 0.3],
       [7.7, 2.6, 6.9, 2.3],
       [6. , 2.9, 4.5, 1.5],
       [6.8, 2.8, 4.8, 1.4],
       [5.4, 3.4, 1.5, 0.4],
       [5.6, 2.9, 3.6, 1.3],
       [6.9, 3.1, 5.1, 2.3],
       [6.2, 2.2, 4.5, 1.5],
       [5.8, 2.7, 3.9, 1.2],
       [6.5, 3.2, 5.1, 2. ],
       [4.8, 3. , 1.4, 0.1],
       [5.5, 3.5, 1.3, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.1, 3.8, 1.5, 0.3],
       [6.3, 3.3, 4.7, 1.6],
       [6.5, 3. , 5.8, 2.2],
       [5.6, 2.5, 3.9, 1.1],
       [5.7, 2.8, 4.5, 1.3],
       [6.4, 2.8, 5.6, 2.2],
       [4.7, 3.2, 1.6, 0.2],
       [6.1, 3. , 4.9, 1.8],
       [5. , 3.4, 1.6, 0.4],
       [6.4, 2.8, 5.6, 2.1],
       [7.9, 3.8, 6.4, 2. ],
       [6.7, 3. , 5.2, 2.3],
       [6.7, 2.5, 5.8, 1.8],
```

```
       [6.8, 3.2, 5.9, 2.3],
       [4.8, 3. , 1.4, 0.3],
       [4.8, 3.1, 1.6, 0.2],
       [4.6, 3.6, 1. , 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [6.7, 3.1, 4.4, 1.4],
       [4.8, 3.4, 1.6, 0.2],
       [4.4, 3.2, 1.3, 0.2],
       [6.3, 2.5, 5. , 1.9],
       [6.4, 3.2, 4.5, 1.5],
       [5.2, 3.5, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.2, 4.1, 1.5, 0.1],
       [5.8, 2.7, 5.1, 1.9],
       [6. , 3.4, 4.5, 1.6],
       [6.7, 3.1, 4.7, 1.5],
       [5.4, 3.9, 1.3, 0.4],
       [5.4, 3.7, 1.5, 0.2]])
```

```
y_train
```

```
array(['Iris-versicolor', 'Iris-virginica', 'Iris-virginica',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-
setosa',
       'Iris-setosa', 'Iris-versicolor', 'Iris-virginica', 'Iris-
setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa',
       'Iris-virginica', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-setosa', 'Iris-versicolor', 'Iris-virginica', 'Iris-
setosa',
       'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor', 'Iris-
setosa',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-
versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-virginica', 'Iris-virginica', 'Iris-versicolor',
       'Iris-setosa', 'Iris-setosa', 'Iris-virginica', 'Iris-
virginica',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-virginica', 'Iris-setosa', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
```

```
       'Iris-virginica', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-virginica',
       'Iris-setosa', 'Iris-versicolor', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-virginica', 'Iris-
setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-virginica',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-virginica',
       'Iris-setosa', 'Iris-versicolor', 'Iris-virginica', 'Iris-
setosa',
       'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

_

```
y_test
```

```
array(['Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa', 'Iris-
setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-
virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica',
       'Iris-setosa', 'Iris-virginica', 'Iris-setosa', 'Iris-
virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-
setosa',
       'Iris-setosa', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa'], dtype=object)
```

SCALING/NORMALISATION

Normalization in machine learning is the process of translating data into the range [0, 1] (or any other range).

```
scaler=MinMaxScaler()
scaler.fit(x_train)
x_train=scaler.transform(x_train)
x_test=scaler.transform(x_test)
```

_

```python
#Normalized training data
x_train
```

```
array([[0.35294118, 0.18181818, 0.46428571, 0.375     ],
       [0.58823529, 0.36363636, 0.71428571, 0.58333333],
       [0.61764706, 0.5       , 0.78571429, 0.70833333],
       [0.67647059, 0.45454545, 0.58928571, 0.54166667],
       [0.85294118, 0.72727273, 0.89285714, 1.        ],
       [0.41176471, 0.40909091, 0.55357143, 0.5       ],
       [0.97058824, 0.45454545, 0.98214286, 0.83333333],
       [0.38235294, 0.45454545, 0.60714286, 0.58333333],
       [0.23529412, 0.68181818, 0.05357143, 0.04166667],
       [1.        , 0.36363636, 1.        , 0.79166667],
       [0.44117647, 0.31818182, 0.53571429, 0.375     ],
       [0.26470588, 0.63636364, 0.05357143, 0.04166667],
       [0.20588235, 0.68181818, 0.03571429, 0.08333333],
       [0.23529412, 0.81818182, 0.14285714, 0.125     ],
       [0.20588235, 0.        , 0.42857143, 0.375     ],
       [0.58823529, 0.31818182, 0.67857143, 0.70833333],
       [0.14705882, 0.63636364, 0.14285714, 0.04166667],
       [0.20588235, 0.45454545, 0.08928571, 0.04166667],
       [0.23529412, 0.59090909, 0.10714286, 0.16666667],
       [0.38235294, 0.31818182, 0.55357143, 0.5       ],
       [0.23529412, 0.63636364, 0.07142857, 0.04166667],
       [0.41176471, 0.45454545, 0.55357143, 0.45833333],
       [1.        , 0.81818182, 1.        , 0.875     ],
       [0.08823529, 0.54545455, 0.05357143, 0.04166667],
       [0.55882353, 0.40909091, 0.57142857, 0.5       ],
       [0.41176471, 0.22727273, 0.69642857, 0.79166667],
       [0.35294118, 1.        , 0.05357143, 0.04166667],
       [0.5       , 0.45454545, 0.66071429, 0.70833333],
       [0.44117647, 0.31818182, 0.71428571, 0.75      ],
       [0.5       , 0.09090909, 0.51785714, 0.375     ],
       [0.32352941, 0.45454545, 0.60714286, 0.58333333],
       [0.55882353, 0.63636364, 0.76785714, 0.91666667],
       [0.35294118, 0.13636364, 0.51785714, 0.5       ],
       [0.32352941, 0.86363636, 0.10714286, 0.125     ],
       [0.20588235, 0.13636364, 0.39285714, 0.375     ],
       [0.61764706, 0.31818182, 0.75      , 0.75      ],
       [0.20588235, 0.59090909, 0.05357143, 0.04166667],
       [0.20588235, 0.54545455, 0.01785714, 0.04166667],
       [0.35294118, 0.18181818, 0.48214286, 0.41666667],
       [0.70588235, 0.45454545, 0.69642857, 0.66666667],
       [0.17647059, 0.5       , 0.07142857, 0.        ],
       [0.44117647, 0.36363636, 0.71428571, 0.95833333],
       [0.20588235, 0.63636364, 0.07142857, 0.04166667],
       [0.20588235, 0.68181818, 0.08928571, 0.20833333],
       [0.47058824, 0.54545455, 0.66071429, 0.70833333],
       [0.23529412, 0.22727273, 0.33928571, 0.41666667],
       [0.76470588, 0.54545455, 0.82142857, 0.91666667],
```

```
       [0.5       , 0.31818182, 0.71428571, 0.625     ],
       [0.52941176, 0.27272727, 0.80357143, 0.54166667],
       [1.        , 0.45454545, 0.89285714, 0.91666667],
       [0.35294118, 0.22727273, 0.51785714, 0.5       ],
       [0.02941176, 0.40909091, 0.05357143, 0.04166667],
       [0.        , 0.45454545, 0.        , 0.        ],
       [0.5       , 0.09090909, 0.69642857, 0.58333333],
       [0.85294118, 0.54545455, 0.875     , 0.70833333],
       [0.08823529, 0.5       , 0.07142857, 0.04166667],
       [0.23529412, 0.68181818, 0.05357143, 0.08333333],
       [0.02941176, 0.45454545, 0.03571429, 0.04166667],
       [0.58823529, 0.22727273, 0.67857143, 0.58333333],
       [0.58823529, 0.63636364, 0.80357143, 0.95833333],
       [0.08823529, 0.63636364, 0.05357143, 0.08333333],
       [0.73529412, 0.45454545, 0.78571429, 0.83333333],
       [0.58823529, 0.59090909, 0.875     , 1.        ],
       [0.11764706, 0.54545455, 0.03571429, 0.04166667],
       [0.52941176, 0.40909091, 0.64285714, 0.54166667],
       [0.64705882, 0.36363636, 0.625     , 0.58333333],
       [0.55882353, 0.36363636, 0.66071429, 0.70833333],
       [0.79411765, 0.54545455, 0.64285714, 0.54166667],
       [0.61764706, 0.54545455, 0.75      , 0.91666667],
       [0.23529412, 0.81818182, 0.08928571, 0.04166667],
       [0.76470588, 0.5       , 0.76785714, 0.83333333],
       [0.47058824, 0.45454545, 0.55357143, 0.58333333],
       [0.64705882, 0.45454545, 0.73214286, 0.79166667],
       [0.41176471, 0.27272727, 0.42857143, 0.375     ],
       [0.26470588, 0.31818182, 0.5       , 0.54166667],
       [0.52941176, 0.45454545, 0.625     , 0.54166667],
       [0.05882353, 0.13636364, 0.03571429, 0.08333333],
       [0.67647059, 0.40909091, 0.625     , 0.5       ],
       [0.35294118, 0.27272727, 0.58928571, 0.45833333],
       [0.29411765, 0.77272727, 0.07142857, 0.04166667],
       [0.38235294, 0.45454545, 0.53571429, 0.5       ],
       [0.88235294, 0.40909091, 0.92857143, 0.70833333],
       [0.70588235, 0.59090909, 0.82142857, 0.83333333],
       [0.23529412, 0.77272727, 0.07142857, 0.125     ],
       [0.17647059, 0.18181818, 0.39285714, 0.375     ],
       [0.70588235, 0.59090909, 0.82142857, 1.        ],
       [0.85294118, 0.45454545, 0.83928571, 0.625     ],
       [0.17647059, 0.5       , 0.07142857, 0.        ],
       [0.70588235, 0.5       , 0.80357143, 0.95833333],
       [0.17647059, 0.45454545, 0.05357143, 0.04166667],
       [0.76470588, 0.5       , 0.67857143, 0.58333333],
       [0.91176471, 0.36363636, 0.89285714, 0.75      ],
       [0.58823529, 0.40909091, 0.80357143, 0.70833333],
       [0.41176471, 0.36363636, 0.53571429, 0.5       ],
       [0.64705882, 0.45454545, 0.78571429, 0.70833333],
       [0.58823529, 0.13636364, 0.58928571, 0.5       ],
```

```
       [0.61764706, 0.40909091, 0.57142857, 0.5       ],
       [0.38235294, 0.36363636, 0.67857143, 0.79166667],
       [0.47058824, 0.45454545, 0.71428571, 0.70833333],
       [0.32352941, 0.63636364, 0.10714286, 0.04166667],
       [0.52941176, 0.36363636, 0.51785714, 0.5       ],
       [0.17647059, 0.22727273, 0.60714286, 0.66666667],
       [0.44117647, 0.90909091, 0.01785714, 0.04166667],
       [0.44117647, 0.27272727, 0.51785714, 0.45833333],
       [0.82352941, 0.45454545, 0.85714286, 0.83333333]])
```

_

```
#Normalized testing data
x_test
```

```
array([[ 0.52941176,  0.36363636,  0.64285714,  0.45833333],
       [ 0.41176471,  0.81818182,  0.10714286,  0.08333333],
       [ 1.        ,  0.27272727,  1.03571429,  0.91666667],
       [ 0.5       ,  0.40909091,  0.60714286,  0.58333333],
       [ 0.73529412,  0.36363636,  0.66071429,  0.54166667],
       [ 0.32352941,  0.63636364,  0.07142857,  0.125     ],
       [ 0.38235294,  0.40909091,  0.44642857,  0.5       ],
       [ 0.76470588,  0.5       ,  0.71428571,  0.91666667],
       [ 0.55882353,  0.09090909,  0.60714286,  0.58333333],
       [ 0.44117647,  0.31818182,  0.5       ,  0.45833333],
       [ 0.64705882,  0.54545455,  0.71428571,  0.79166667],
       [ 0.14705882,  0.45454545,  0.05357143,  0.        ],
       [ 0.35294118,  0.68181818,  0.03571429,  0.04166667],
       [ 0.17647059,  0.5       ,  0.07142857,  0.        ],
       [ 0.23529412,  0.81818182,  0.07142857,  0.08333333],
       [ 0.58823529,  0.59090909,  0.64285714,  0.625     ],
       [ 0.64705882,  0.45454545,  0.83928571,  0.875     ],
       [ 0.38235294,  0.22727273,  0.5       ,  0.41666667],
       [ 0.41176471,  0.36363636,  0.60714286,  0.5       ],
       [ 0.61764706,  0.36363636,  0.80357143,  0.875     ],
       [ 0.11764706,  0.54545455,  0.08928571,  0.04166667],
       [ 0.52941176,  0.45454545,  0.67857143,  0.70833333],
       [ 0.20588235,  0.63636364,  0.08928571,  0.125     ],
       [ 0.61764706,  0.36363636,  0.80357143,  0.83333333],
       [ 1.05882353,  0.81818182,  0.94642857,  0.79166667],
       [ 0.70588235,  0.45454545,  0.73214286,  0.91666667],
       [ 0.70588235,  0.22727273,  0.83928571,  0.70833333],
       [ 0.73529412,  0.54545455,  0.85714286,  0.91666667],
       [ 0.14705882,  0.45454545,  0.05357143,  0.08333333],
       [ 0.14705882,  0.5       ,  0.08928571,  0.04166667],
       [ 0.08823529,  0.72727273, -0.01785714,  0.04166667],
       [ 0.41176471,  1.09090909,  0.07142857,  0.125     ],
       [ 0.70588235,  0.5       ,  0.58928571,  0.54166667],
       [ 0.14705882,  0.63636364,  0.08928571,  0.04166667],
```

```
       [ 0.02941176,  0.54545455,  0.03571429,  0.04166667],
       [ 0.58823529,  0.22727273,  0.69642857,  0.75      ],
       [ 0.61764706,  0.54545455,  0.60714286,  0.58333333],
       [ 0.26470588,  0.68181818,  0.07142857,  0.04166667],
       [ 0.20588235,  0.72727273,  0.05357143,  0.04166667],
       [ 0.26470588,  0.95454545,  0.07142857,  0.        ],
       [ 0.44117647,  0.31818182,  0.71428571,  0.75      ],
       [ 0.5       ,  0.63636364,  0.60714286,  0.625     ],
       [ 0.70588235,  0.5       ,  0.64285714,  0.58333333],
       [ 0.32352941,  0.86363636,  0.03571429,  0.125     ],
       [ 0.32352941,  0.77272727,  0.07142857,  0.04166667]])
```

MODEL CREATION

CLASSIFICATION ALGORITHAMS

1) K-Nearest Neighbors algorithm(KNN)::

```
knn=KNeighborsClassifier(n_neighbors=7)
knn.fit(x_train,y_train)

#Predicting using test data

y_pred_knn=knn.predict(x_test)
y_pred_knn

array(['Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa', 'Iris-
setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-
virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica',
       'Iris-setosa', 'Iris-virginica', 'Iris-setosa', 'Iris-
virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-
setosa',
       'Iris-setosa', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa'], dtype=object)
```

_

```
y_test

array(['Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
```

```
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa', 'Iris-
setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-
virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica',
       'Iris-setosa', 'Iris-virginica', 'Iris-setosa', 'Iris-
virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-
setosa',
       'Iris-setosa', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa'], dtype=object)
```

2) Naive Bayes algorithm

```
naiv=GaussianNB()
naiv.fit(x_train,y_train)

y_pred_naiv=naiv.predict(x_test)
y_pred_naiv
```

```
array(['Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa', 'Iris-
setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-virginica', 'Iris-
virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica',
       'Iris-setosa', 'Iris-virginica', 'Iris-setosa', 'Iris-
virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-
setosa',
       'Iris-setosa', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa'], dtype='<U15')
```

3) Support Vector Machine algorithm(SVM)

```
sv=SVC()
sv.fit(x_train,y_train)

y_pred_sv=sv.predict(x_test)
y_pred_sv
```

```
array(['Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa', 'Iris-
setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-
virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica',
       'Iris-setosa', 'Iris-virginica', 'Iris-setosa', 'Iris-
virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-
setosa',
       'Iris-setosa', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa'], dtype=object)
```

PERFORMANCE EVALUATION

```
#KNN
report=classification_report(y_pred_knn,y_test)
print(report)
```

```
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        19
Iris-versicolor       1.00      1.00      1.00        13
 Iris-virginica       1.00      1.00      1.00        13

       accuracy                           1.00        45
      macro avg       1.00      1.00      1.00        45
   weighted avg       1.00      1.00      1.00        45
```

–

```
score=accuracy_score(y_test,y_pred_knn)
score
```

```
1.0
```

–

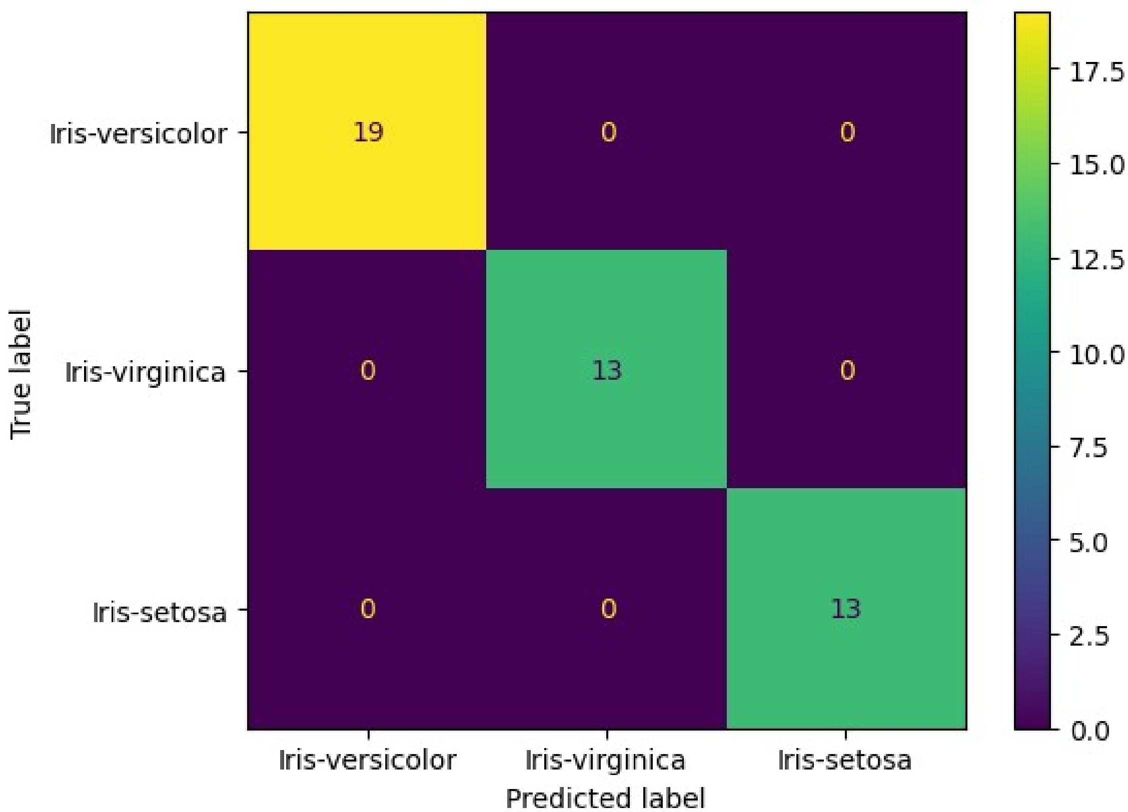
```
matx=confusion_matrix(y_test,y_pred_knn)
print(matx)
```

```
[[19  0  0]
 [ 0 13  0]
 [ 0  0 13]]
```

_

```
labels=['Iris-versicolor','Iris-virginica','Iris-setosa']
cmd=ConfusionMatrixDisplay(matx,display_labels=labels)
cmd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x7f9a96180ac0>
```



_

```
#Naive Bayes
report=classification_report(y_pred_naiv,y_test)
print(report)
```

```
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        19
Iris-versicolor       0.92      1.00      0.96        12
 Iris-virginica       1.00      0.93      0.96        14
```

```
    accuracy                           0.98        45
   macro avg       0.97      0.98      0.97        45
weighted avg       0.98      0.98      0.98        45
```

```python
score1=accuracy_score(y_pred_naiv,y_test)
print(score1)
```

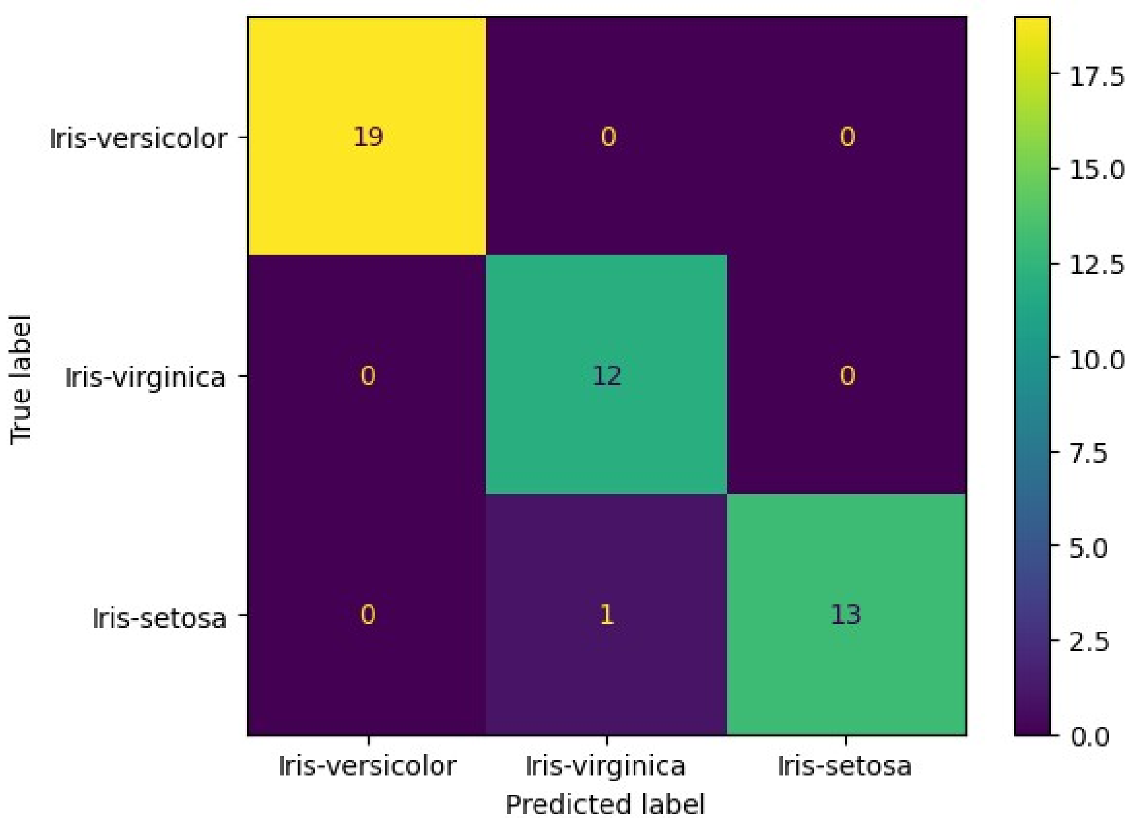
```
0.9777777777777777
```

```python
matx1=confusion_matrix(y_pred_naiv,y_test)
matx1
```

```
array([[19,  0,  0],
       [ 0, 12,  0],
       [ 0,  1, 13]])
```

```python
labels=['Iris-versicolor','Iris-virginica','Iris-setosa']
cmd1=ConfusionMatrixDisplay(matx1,display_labels=labels)
cmd1.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x7f9a96821fa0>
```



```
#svm
report=classification_report(y_pred_sv,y_test)
print(report)
```

```
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        19
Iris-versicolor       1.00      1.00      1.00        13
 Iris-virginica       1.00      1.00      1.00        13

       accuracy                           1.00        45
      macro avg       1.00      1.00      1.00        45
   weighted avg       1.00      1.00      1.00        45
```

```
score=accuracy_score(y_pred_sv,y_test)
print(score)
```

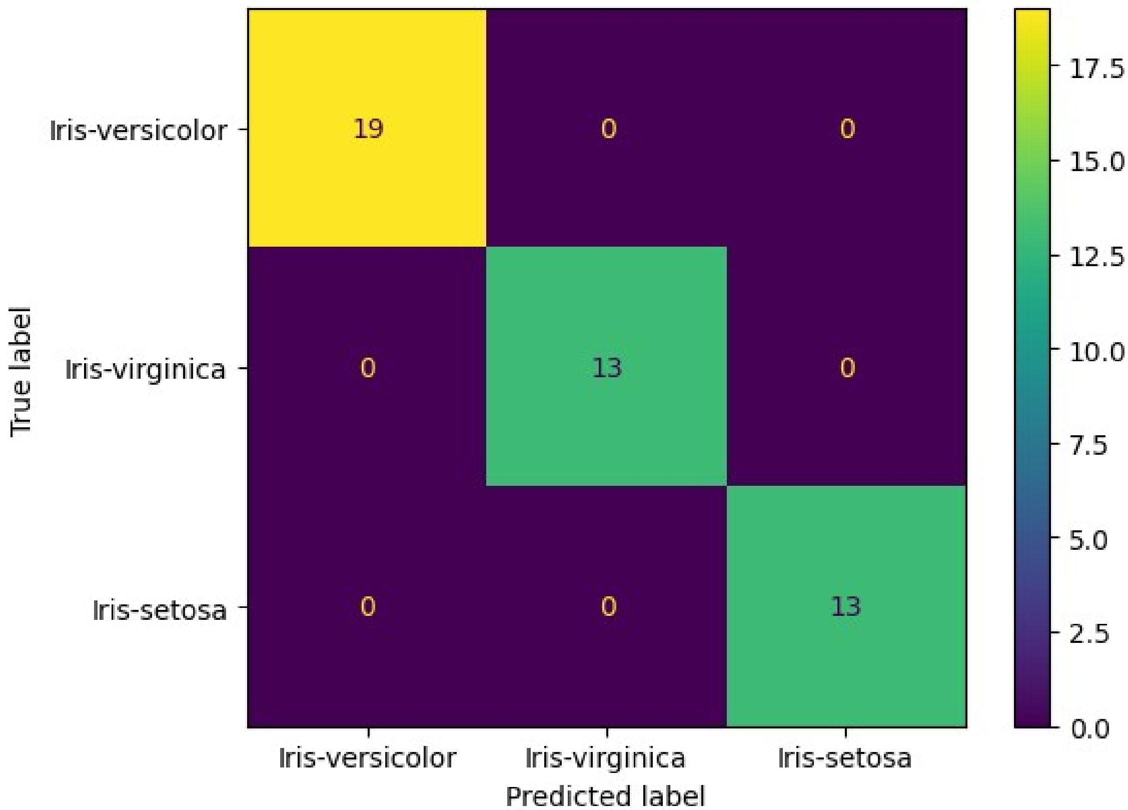
```
1.0
```

```
matx2=confusion_matrix(y_pred_sv,y_test)
matx2
```

```
array([[19,  0,  0],
       [ 0, 13,  0],
       [ 0,  0, 13]])
```

```
labels=['Iris-versicolor','Iris-virginica','Iris-setosa']
cmd=ConfusionMatrixDisplay(matx2,display_labels=labels)
cmd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x7f9a9664e7c0>
```




```
print(knn.predict([[4.5,5.2,1.4,0.3]]))

['Iris-virginica']

print(naiv.predict([[4.5,5.2,1.4,0.3]]))

['Iris-virginica']

print(sv.predict([[4.5,5.2,1.4,0.3]]))

['Iris-virginica']
```

Analyzing different classification techniques and considering their respective accuracies, it can be inferred that all the models demonstrated accuracies within the range of 98% to 100%.The highest accuracy is given by knn and svm.