Aadarsh Goyal

MIS: 111915001

--

Importing the dataset

In [1]:

```
import pandas as pd

# Location of dataset
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"

# Assign colum names to the dataset
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']

# Read dataset to pandas dataframe
df = pd.read_csv(url, names=names)
```

In [2]:

```
df.head()
```

Out[2]:

	sepal-length	sepal-width	petal-length	petal-width	Class
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

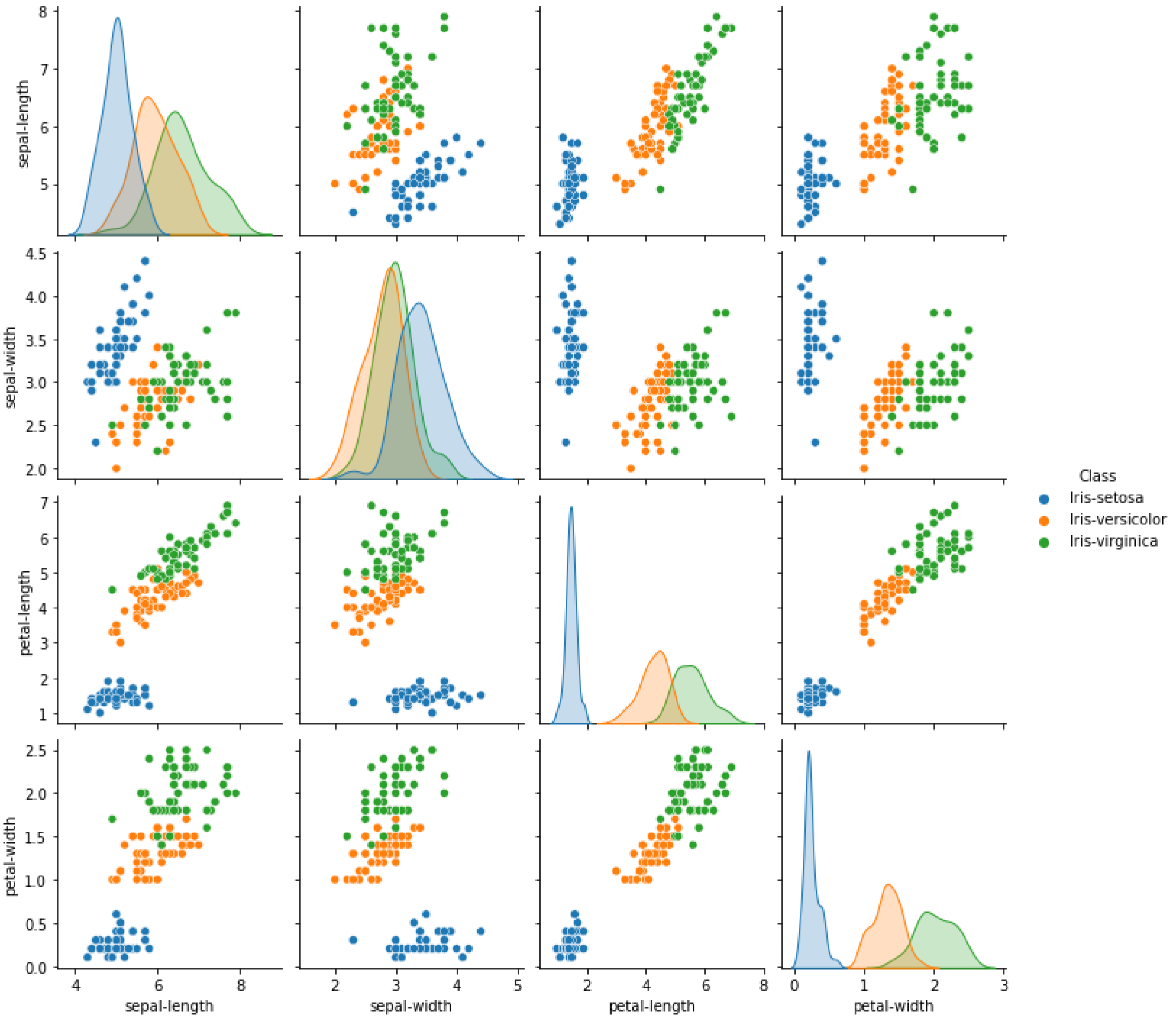
In [3]:

```
import seaborn as sns

sns.pairplot(data = df, hue = 'Class')
```

Out[3]:

```
<seaborn.axisgrid.PairGrid at 0x1b6e673e948>
```



In [4]:

```
X = df.iloc[:,0:4]
y = df.iloc[:,4]
```

In [5]:

```
y.head()
```

Out[5]:

```
0    Iris-setosa
1    Iris-setosa
2    Iris-setosa
3    Iris-setosa
4    Iris-setosa
Name: Class, dtype: object
```

In [6]:

```
y.unique()
```

Out[6]:

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

In [7]:

```
from sklearn import preprocessing

encoder = preprocessing.LabelEncoder()
y = encoder.fit_transform(y)
```

Test/Train Split

In [8]:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state =
```

Scaling the data

In [9]:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)

X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

Build the Neural Network model

In [10]:

```
from sklearn.neural_network import MLPClassifier

mlp = MLPClassifier(hidden_layer_sizes=(100, 100, 100), max_iter=1000)
mlp.fit(X_train, y_train)
```

Out[10]:

```
MLPClassifier(hidden_layer_sizes=(100, 100, 100), max_iter=1000)
```

In [11]:

```
predictions = mlp.predict(X_test)
```

In [12]:

```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,plot_conf
```

In [13]:

```
confusion_matrix(y_test,predictions)
```

Out[13]:

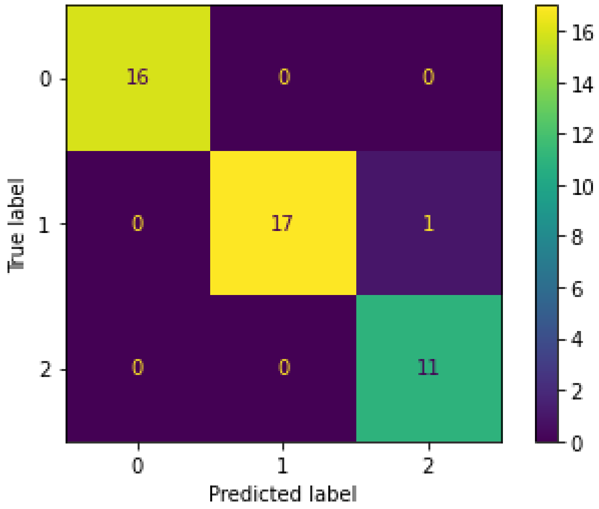
```
array([[16,  0,  0],
       [ 0, 17,  1],
       [ 0,  0, 11]], dtype=int64)
```

In [14]:

```
plot_confusion_matrix(mlp,X_test,y_test)
```

Out[14]:

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1b6eb020
d48>
```

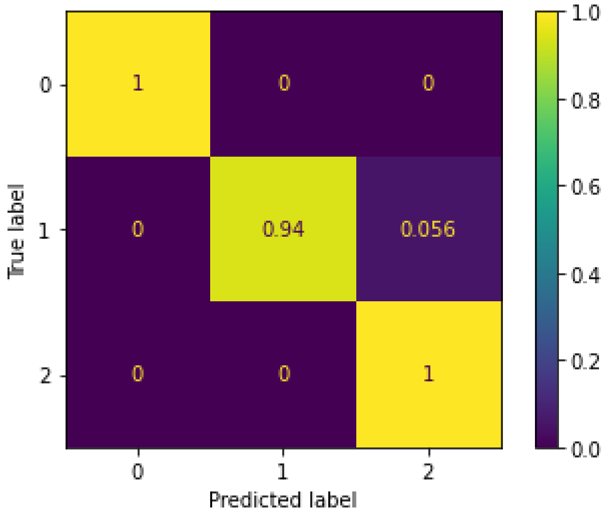


In [15]:

```
plot_confusion_matrix(mlp,X_test,y_test,normalize='true')
```

Out[15]:

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1b6eb020
c48>
```



In [16]:

```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        16
           1       1.00      0.94      0.97        18
           2       0.92      1.00      0.96        11

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

In [ ]: