

```
# -*- coding: utf-8 -*-  
"""Irisdataset.ipynb
```

Automatically generated by Colaboratory.

Original file is located at
<https://colab.research.google.com/drive/1pUP-rHH8qeRmxmQH5tj50uTLWK77Pb-I>
"""

```
cd /content/drive/My Drive/lp1/DA/assignment1
```

```
# Commented out IPython magic to ensure Python compatibility.
```

```
# %matplotlib inline
```

```
import matplotlib.pyplot as plt
```

```
import pandas as pd
```

```
import numpy as np
```

```
names=['sepal-length','sepal-width','petal-length','petal-width','class'];
```

```
data=pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-  
databases/iris/iris.data",names=names);#skilrows=1 beacause we want to skip header row  
print(data[0:10]) #print first 10rows
```

```
data.describe()
```

```
data.shape
```

```
list(data.columns)
```

```
data.dtypes
```

```
print(data['class'].describe()); #we can specify column name only also to fins statistical values
```

```
print("-----Class Distribution-----")
```

```
print(data.groupby('class').size())
```

```
#histograms
```

```
print("Histrogram")
```

```
plt.hist(data['petal-length'],bins=30,color=['red']) #bins represent no. of bars
```

```
plt.ylabel('No of times')
```

```
plt.xlabel("PETAL_LENGTH");
```

```
plt.title("PETAL_LENGTH HISTOGRAM IRIS")
```

```
plt.show()
```

```
print("histogram for petal-width")
```

```
plt.hist(data['petal-width'],bins=20,color=['orange'])
```

```
plt.xlabel("petal Width");
```

```
plt.ylabel('Frequency')
```

```
plt.show();
```

```
#box and whisker plot #Method 1
```

```
print("Displaying Box Plot");
```

```
data.plot(kind='box', subplots=True, layout=(2,2), sharex=False, sharey=False, figsize =  
(8,8),notch=True)
```

```
plt.show()
```

```
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
```

```
train_data,test_data=train_test_split(data,test_size=0.30, random_state=2)
train_data.shape, test_data.shape
```

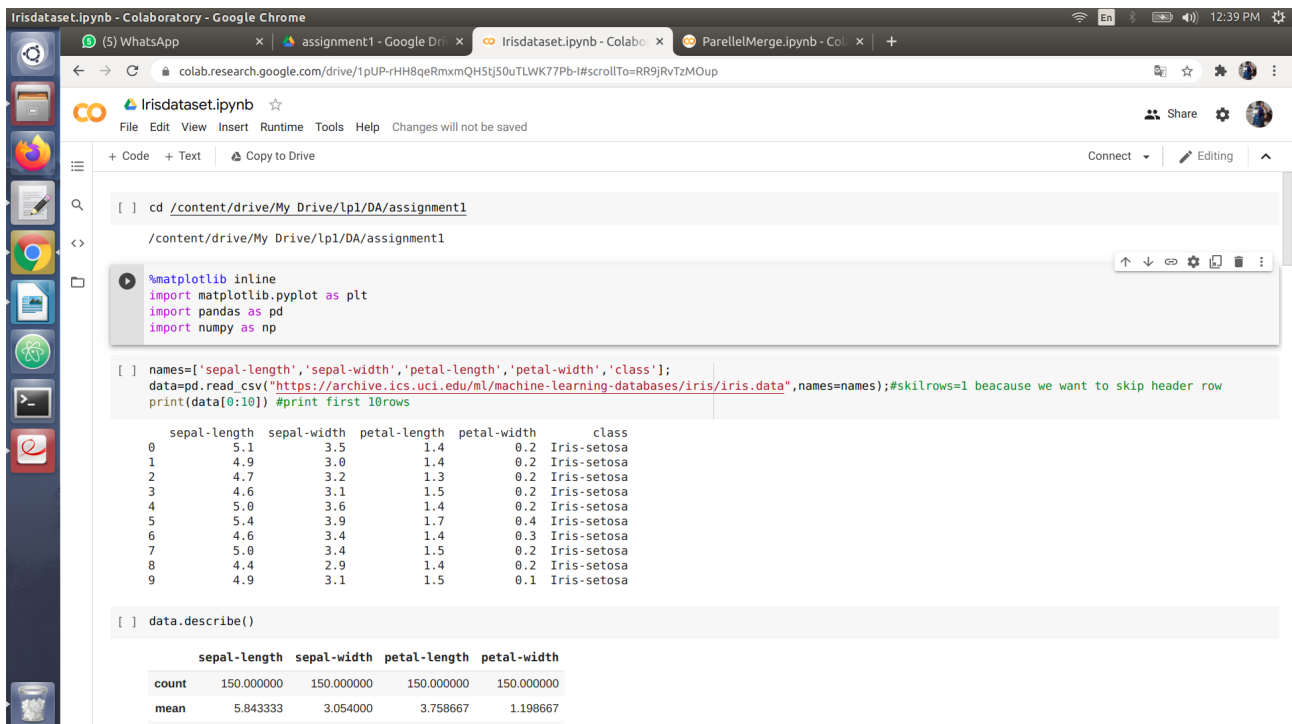
```
train_labels=train_data["class"].copy();
train_data=train_data.drop("class",axis=1);
test_labels=test_data["class"].copy();
test_data=test_data.drop("class",axis=1);
```

```
classifier=GaussianNB();
classifier.fit(train_data,train_labels);
predict=classifier.predict(test_data);
```

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

```
accuracy=accuracy_score(test_labels,predict)
accuracy
```

OUTPUT:::



The screenshot shows a Google Colaboratory notebook titled 'Irisdataset.ipynb'. The code executed includes:

```
[ ] cd /content/drive/My Drive/lp1/DA/assignment1
/content/drive/My Drive/lp1/DA/assignment1

%matplotlib inline
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

[ ] names=['sepal-length','sepal-width','petal-length','petal-width','class'];
data=pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data",names=names);#skilrows=1 beacuse we want to skip header row
print(data[0:10]) #print first 10rows
```

The output shows the first 10 rows of the Iris dataset:

| | 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 |
| 5 | 5.4 | 3.9 | 1.7 | 0.4 | Iris-setosa |
| 6 | 4.6 | 3.4 | 1.4 | 0.3 | Iris-setosa |
| 7 | 5.0 | 3.4 | 1.5 | 0.2 | Iris-setosa |
| 8 | 4.4 | 2.9 | 1.4 | 0.2 | Iris-setosa |
| 9 | 4.9 | 3.1 | 1.5 | 0.1 | Iris-setosa |

The next code cell shows the result of `data.describe()`:

```
[ ] data.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 | 0.435264 | 1.761875 | 0.762238 |
| min | 4.300000 | 2.000000 | 1.000000 | 0.100000 |
| max | 7.900000 | 4.400000 | 6.900000 | 4.700000 |

