EXPERIMENT – 8

PREDICTING MODEL -KNN

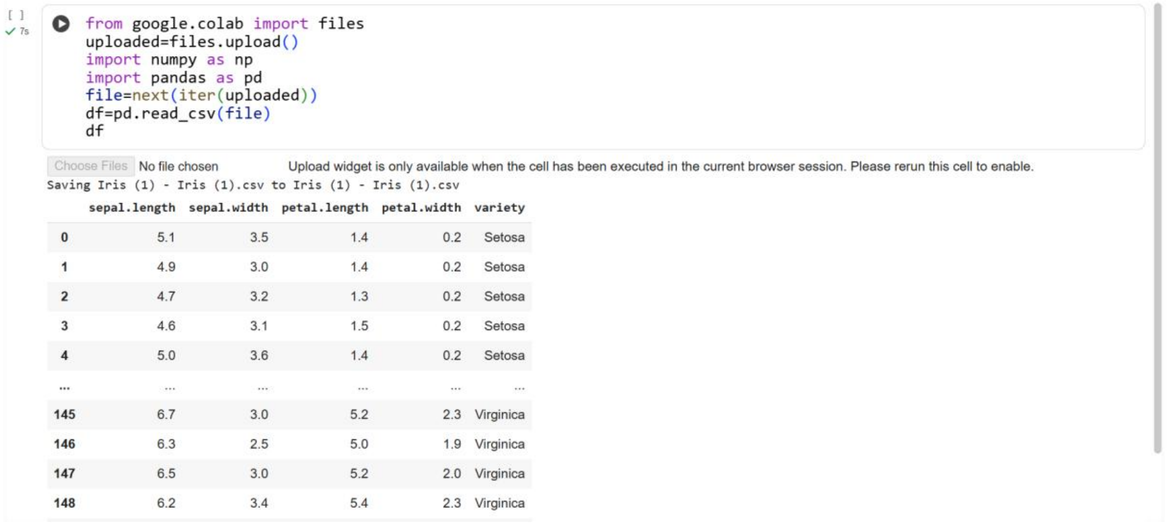
Aim:

To perform model classification using K-nearest neighbours

Procedure:

- Upload a given dataset
- Import all necessities
- Read and make it as dataframe
- Through sklearn train the model
- Test the model

Program:

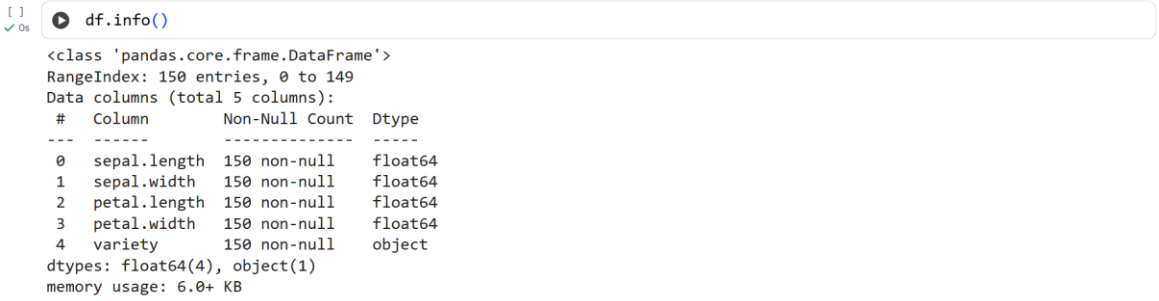


```
from google.colab import files
uploaded=files.upload()
import numpy as np
import pandas as pd
file=next(iter(uploaded))
df=pd.read_csv(file)
df
```

No file chosen   Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

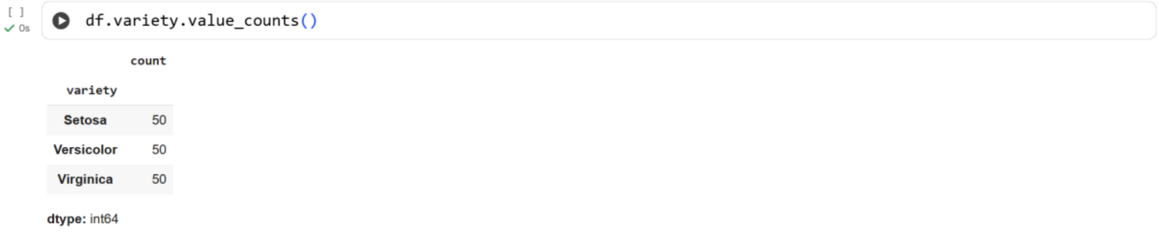
```
Saving Iris (1) - Iris (1).csv to Iris (1) - Iris (1).csv
```

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica



```
df.info()
```

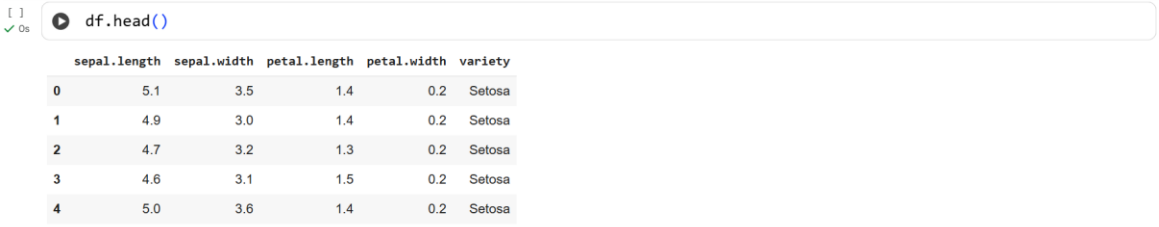
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal.length  150 non-null    float64
 1   sepal.width   150 non-null    float64
 2   petal.length  150 non-null    float64
 3   petal.width   150 non-null    float64
 4   variety       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```



```
df.variety.value_counts()
```

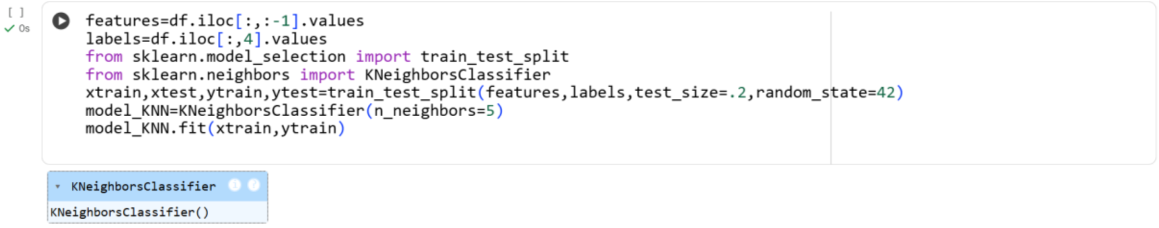
	count
variety	
Setosa	50
Versicolor	50
Virginica	50

dtype: int64



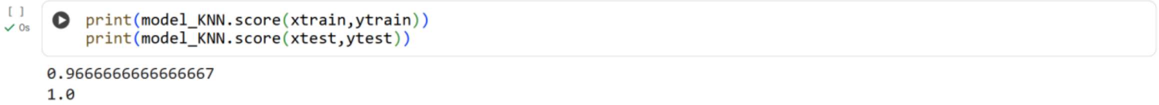
```
df.head()
```

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa



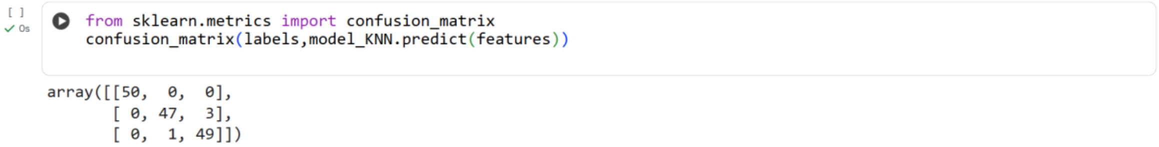
```
features=df.iloc[:,:-1].values
labels=df.iloc[:,4].values
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
xtrain,xtest,ytrain,ytest=train_test_split(features,labels,test_size=.2,random_state=42)
model_KNN=KNeighborsClassifier(n_neighbors=5)
model_KNN.fit(xtrain,ytrain)
```

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```



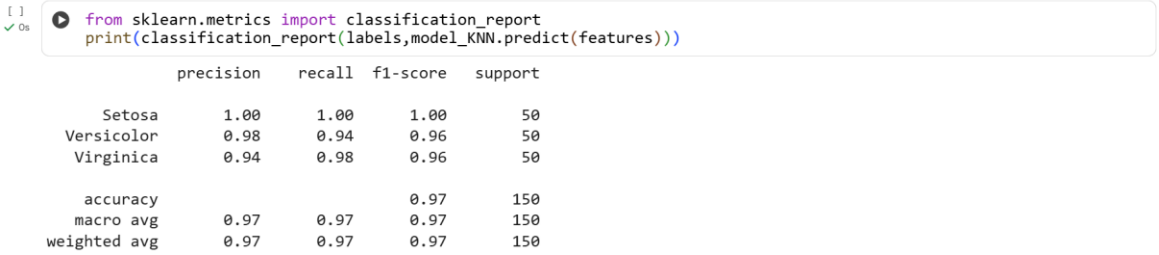
```
print(model_KNN.score(xtrain,ytrain))
print(model_KNN.score(xtest,ytest))
```

```
0.9666666666666667
1.0
```



```
from sklearn.metrics import confusion_matrix
confusion_matrix(labels,model_KNN.predict(features))
```

```
array([[50,  0,  0],
       [ 0, 47,  3],
       [ 0,  1, 49]])
```



```
from sklearn.metrics import classification_report
print(classification_report(labels,model_KNN.predict(features)))
```

```
              precision    recall  f1-score   support

      Setosa       1.00      1.00      1.00        50
  Versicolor       0.98      0.94      0.96        50
   Virginica       0.94      0.98      0.96        50

    accuracy                           0.97       150
   macro avg       0.97      0.97      0.97       150
weighted avg       0.97      0.97      0.97       150
```

Result:

Thus the python program to perform model classification using KNN is executed and verified