FUNDAMENTALS OF MACHINE LEARNING IN DATA SCIENCE

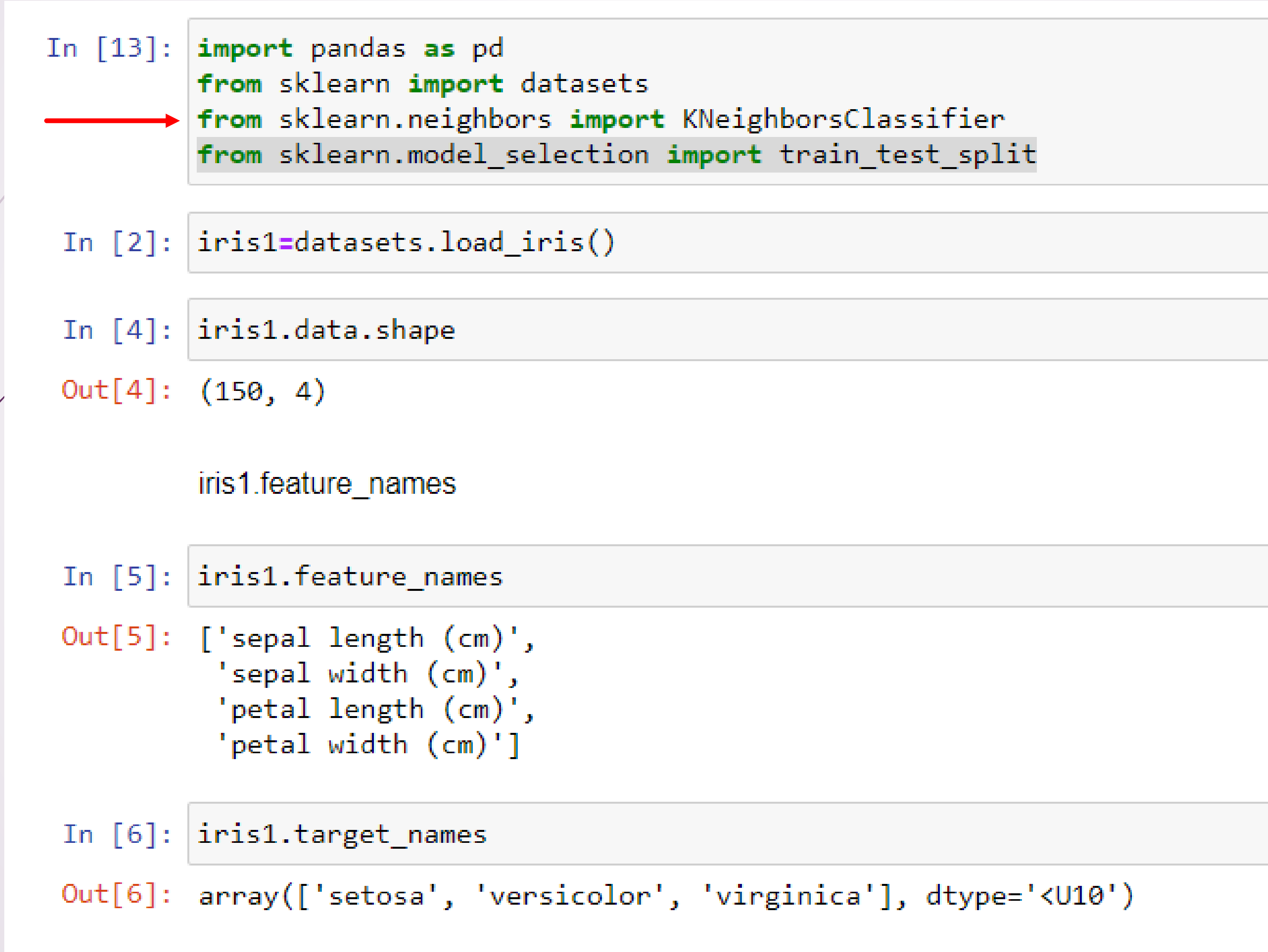
CSIS 3290

SUPERVISED LEARNING 2 (SVM, KNN, NAÏVE BAYES)

IN SKLEARN

FATEMEH AHMADI

KNN



```
In [13]: import pandas as pd
         from sklearn import datasets
         from sklearn.neighbors import KNeighborsClassifier
         from sklearn.model_selection import train_test_split
```

```
In [2]: iris1=datasets.load_iris()
```

```
In [4]: iris1.data.shape

Out[4]: (150, 4)
```

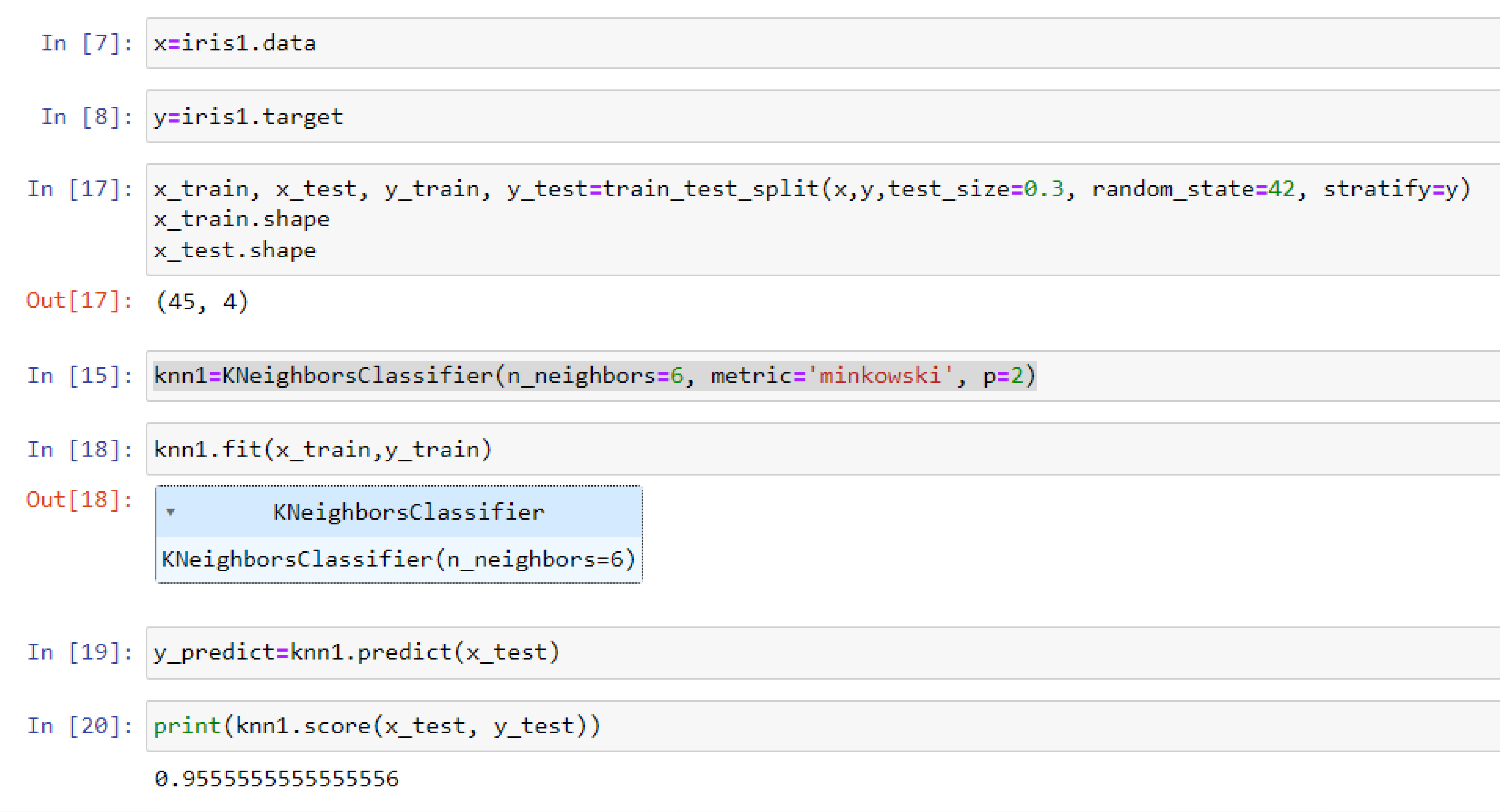
iris1.feature_names

```
In [5]: iris1.feature_names

Out[5]: ['sepal length (cm)',
         'sepal width (cm)',
         'petal length (cm)',
         'petal width (cm)']
```

```
In [6]: iris1.target_names

Out[6]: array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

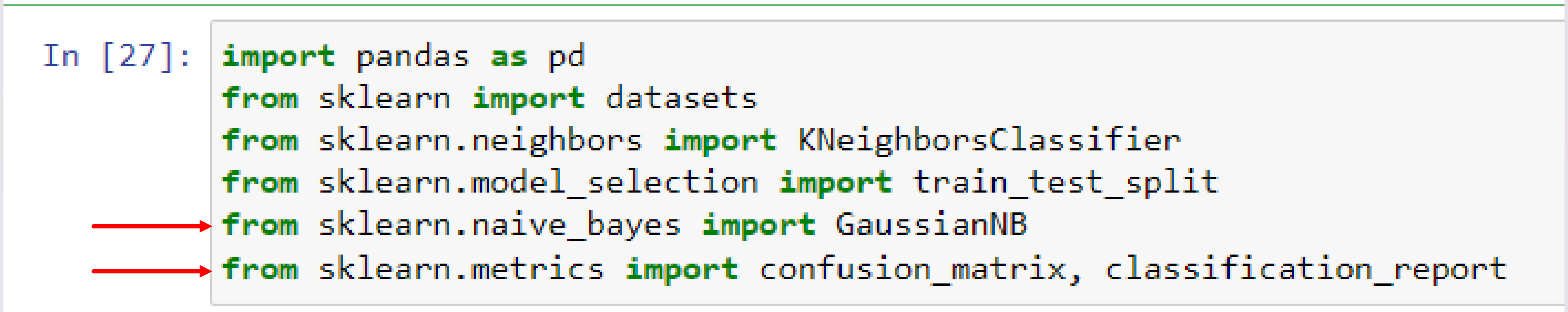
KNN

The parameter p is used to specify the **power parameter** for the Minkowski metric. When p is set to 1, this is equivalent to using manhattan_distance (l1). When we set p=2, which is its default value, the Minkowski metric works as the euclidean distance metric.



```
In [7]:  x=iris1.data

In [8]:  y=iris1.target

In [17]: x_train, x_test, y_train, y_test=train_test_split(x,y,test_size=0.3, random_state=42, stratify=y)
         x_train.shape
         x_test.shape

Out[17]: (45, 4)

In [15]: knn1=KNeighborsClassifier(n_neighbors=6, metric='minkowski', p=2)

In [18]: knn1.fit(x_train,y_train)

Out[18]: ▾        KNeighborsClassifier
         KNeighborsClassifier(n_neighbors=6)

In [19]: y_predict=knn1.predict(x_test)

In [20]: print(knn1.score(x_test, y_test))

         0.9555555555555556
```

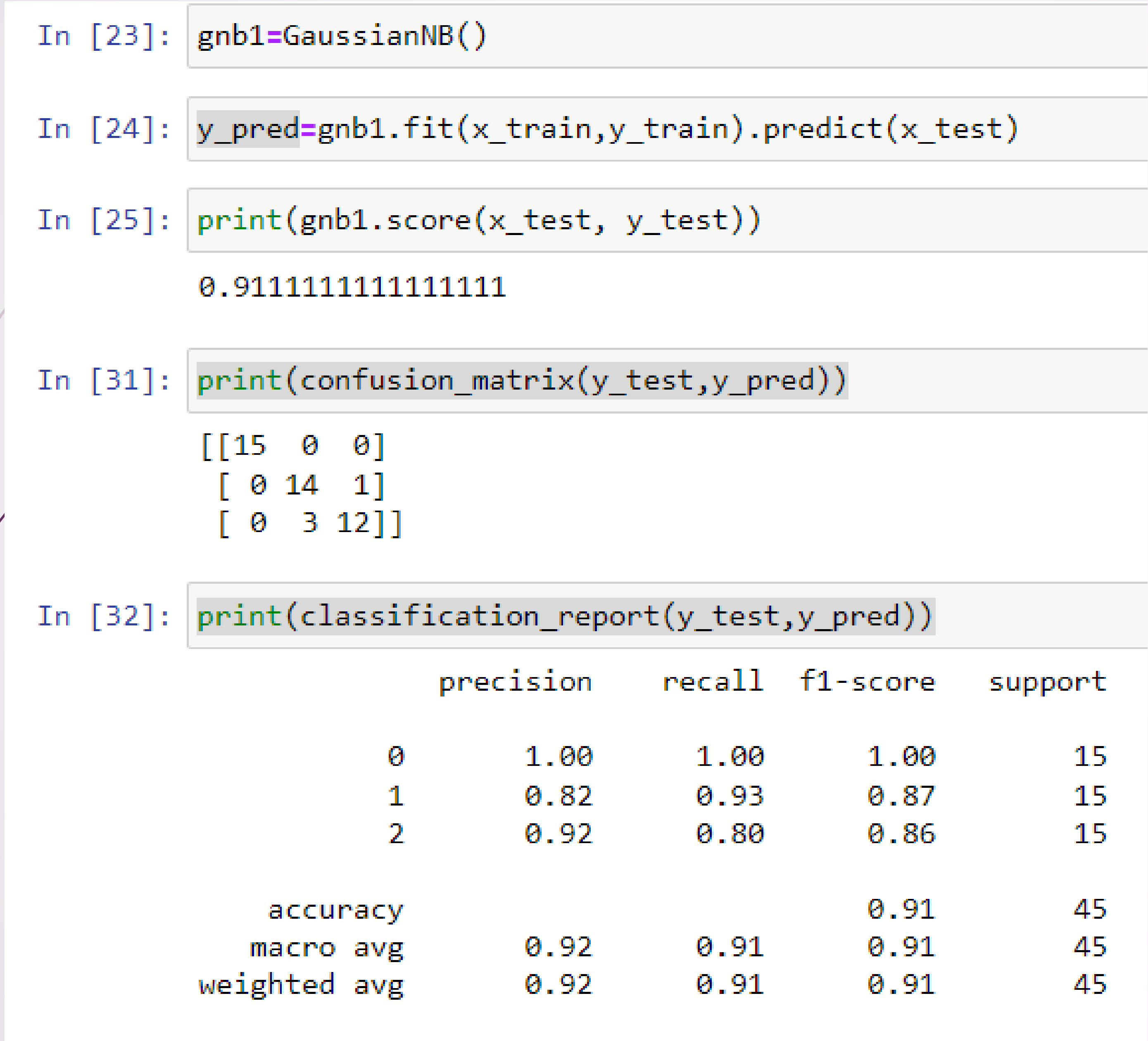
Naïve Bayes



In [27]:

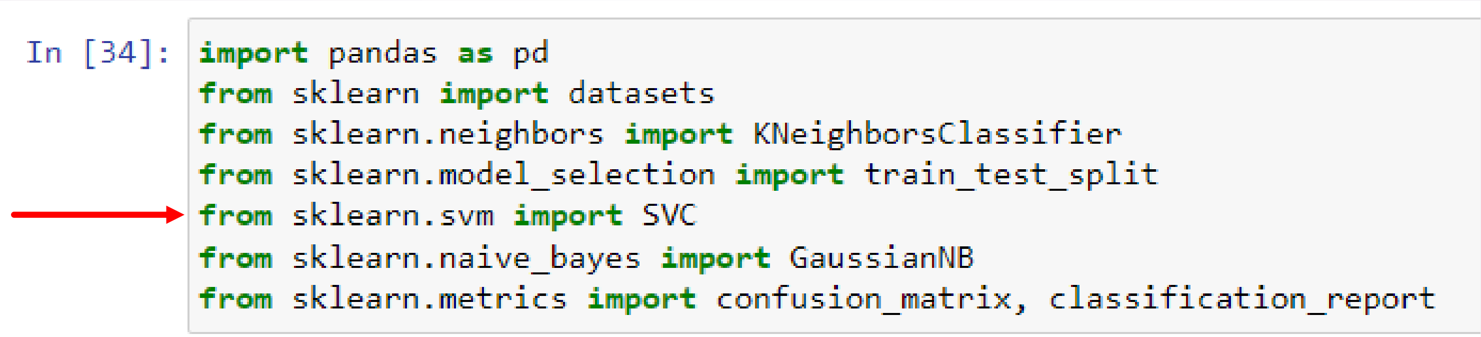
```
import pandas as pd
from sklearn import datasets
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, classification_report
```

Naïve Bayes

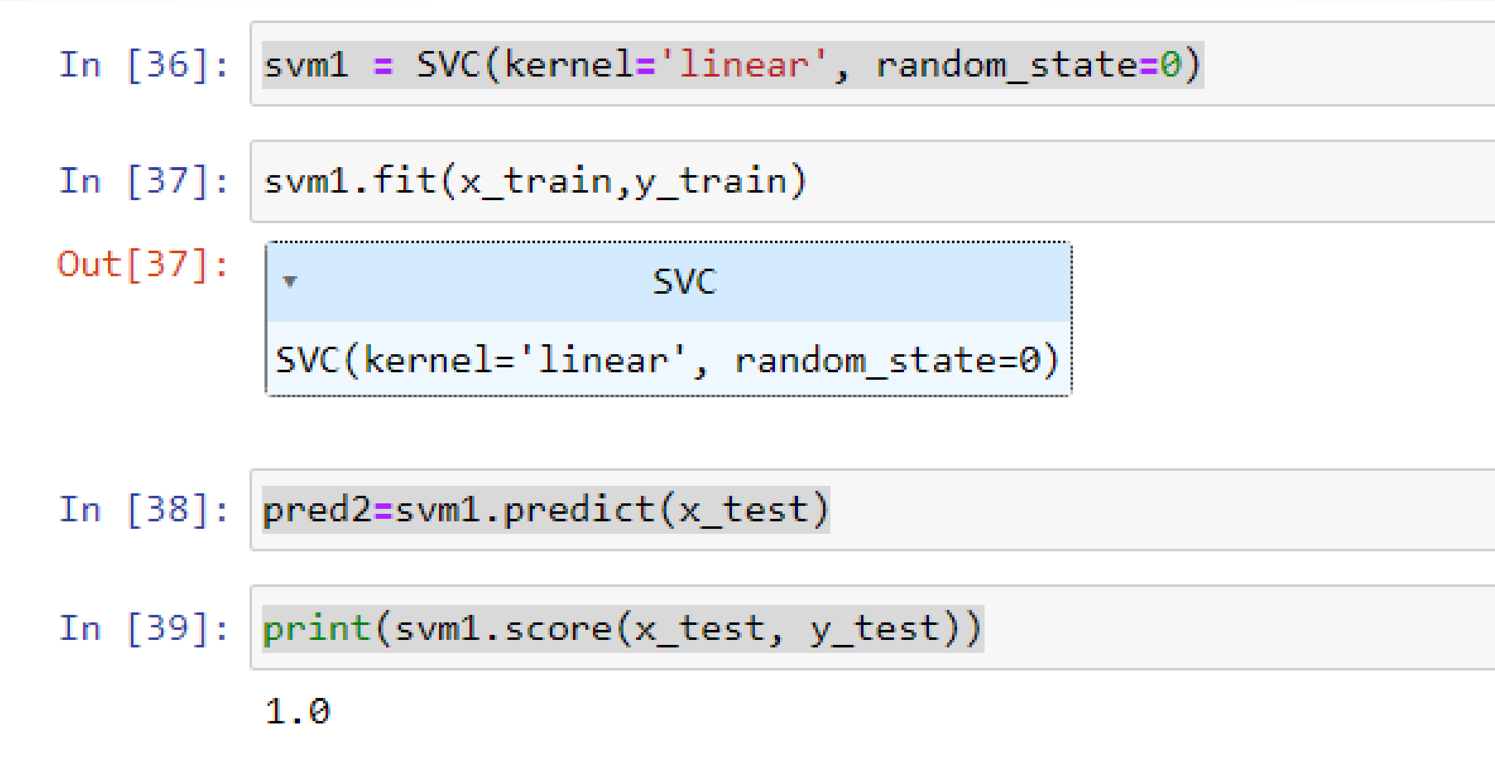


```
In [23]: gnb1=GaussianNB()

In [24]: y_pred=gnb1.fit(x_train,y_train).predict(x_test)

In [25]: print(gnb1.score(x_test, y_test))

         0.9111111111111111

In [31]: print(confusion_matrix(y_test,y_pred))

         [[15  0  0]
          [ 0 14  1]
          [ 0  3 12]]

In [32]: print(classification_report(y_test,y_pred))

                       precision    recall  f1-score   support

                    0       1.00      1.00      1.00        15
                    1       0.82      0.93      0.87        15
                    2       0.92      0.80      0.86        15

             accuracy                           0.91        45
            macro avg       0.92      0.91      0.91        45
         weighted avg       0.92      0.91      0.91        45
```

SVM (with Linear Kernel)



```
In [34]: import pandas as pd
         from sklearn import datasets
         from sklearn.neighbors import KNeighborsClassifier
         from sklearn.model_selection import train_test_split
         from sklearn.svm import SVC
         from sklearn.naive_bayes import GaussianNB
         from sklearn.metrics import confusion_matrix, classification_report
```



```
In [36]: svm1 = SVC(kernel='linear', random_state=0)

In [37]: svm1.fit(x_train,y_train)

Out[37]: ▾ SVC
         SVC(kernel='linear', random_state=0)

In [38]: pred2=svm1.predict(x_test)

In [39]: print(svm1.score(x_test, y_test))

         1.0
```