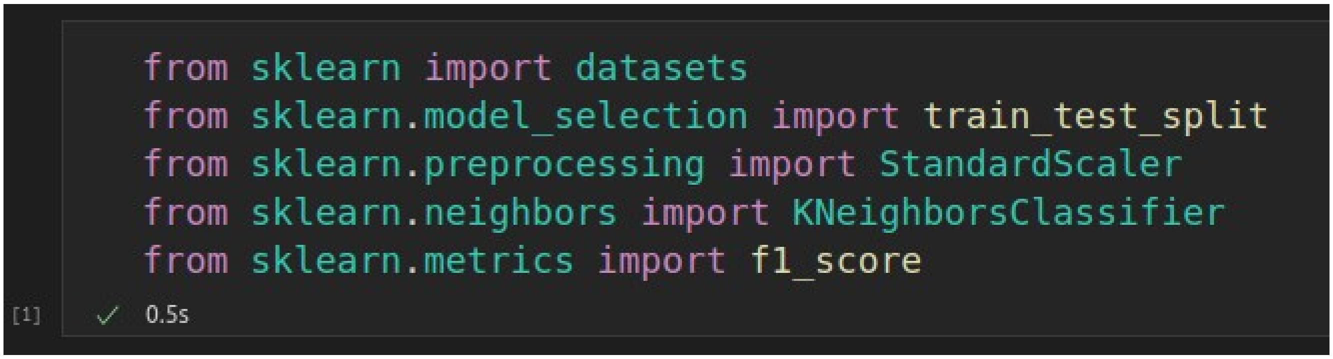
Peer Assignment
Sheikh Muhammad Sabih (2303.KHI.DEG.010)
M Humza Moeen (2303.KHI.DEG.019)

Assignment (3.4)

Implement a single classification model of your choice and try to achieve at least an 80% F1 score on the wine dataset provided by Sklearn.

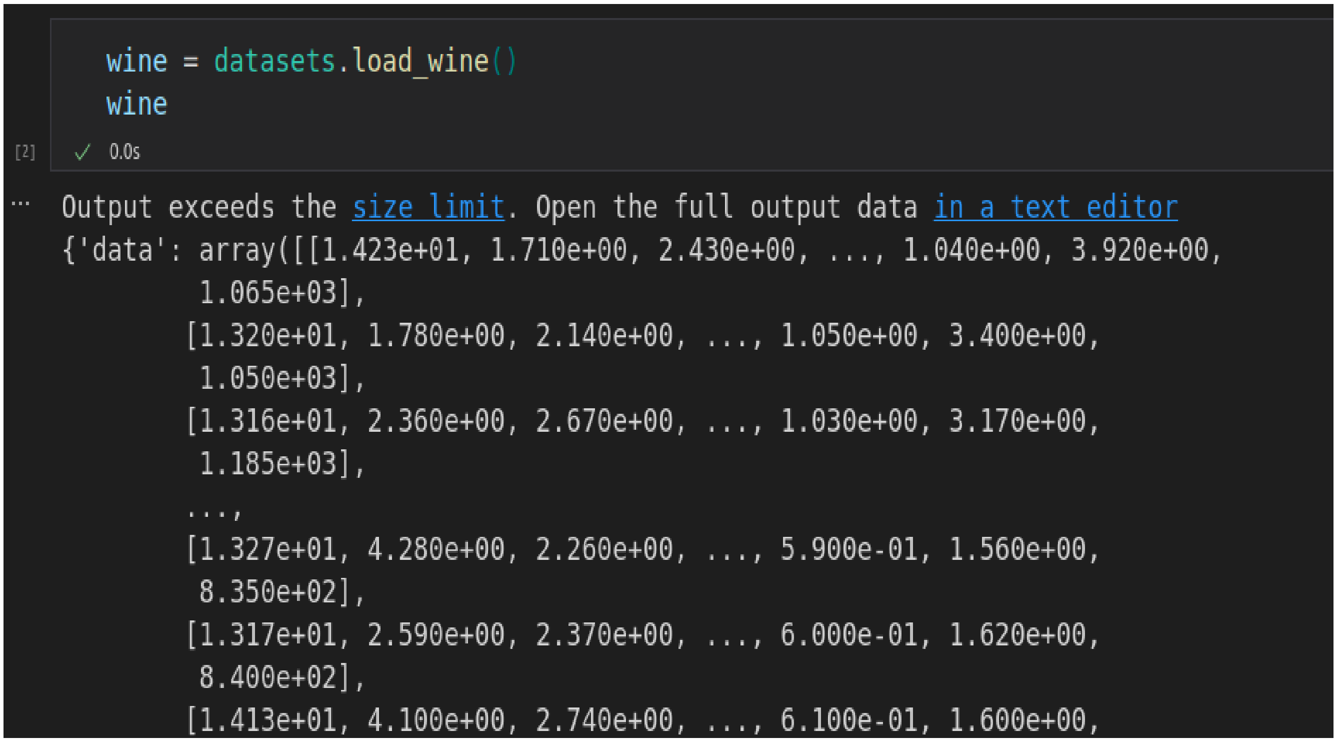
Solution

1. First we import all required **libraries**.



```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import f1_score
```

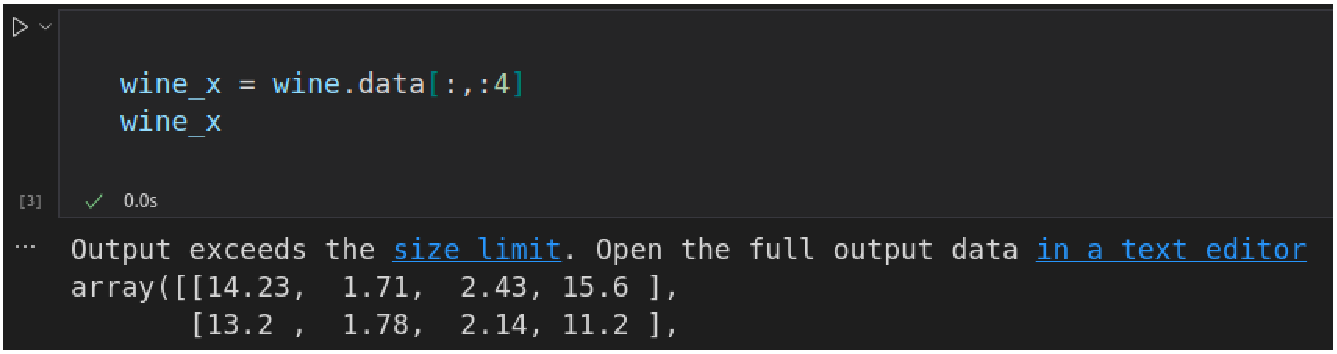
2. Then load the **wine dataset** from **sklearn**.



```
wine = datasets.load_wine()
wine
```

```
Output exceeds the size limit. Open the full output data in a text editor
{'data': array([[1.423e+01, 1.710e+00, 2.430e+00, ..., 1.040e+00, 3.920e+00,
         1.065e+03],
        [1.320e+01, 1.780e+00, 2.140e+00, ..., 1.050e+00, 3.400e+00,
         1.050e+03],
        [1.316e+01, 2.360e+00, 2.670e+00, ..., 1.030e+00, 3.170e+00,
         1.185e+03],
        ...,
        [1.327e+01, 4.280e+00, 2.260e+00, ..., 5.900e-01, 1.560e+00,
         8.350e+02],
        [1.317e+01, 2.590e+00, 2.370e+00, ..., 6.000e-01, 1.620e+00,
         8.400e+02],
        [1.413e+01, 4.100e+00, 2.740e+00, ..., 6.100e-01, 1.600e+00,
```

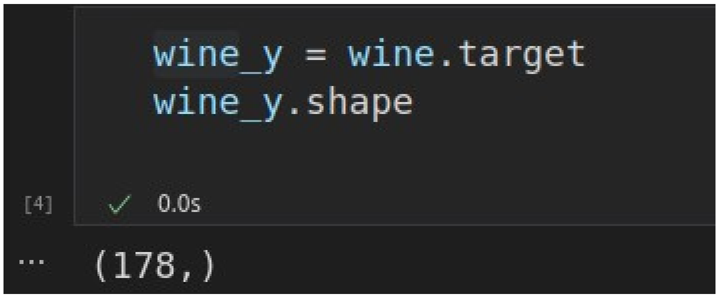
3. Now, select last **4 columns of data** as **wine_x**.



```
wine_x = wine.data[:,:4]
wine_x
```

```
Output exceeds the size limit. Open the full output data in a text editor
array([[14.23,   1.71,   2.43, 15.6 ],
       [13.2 ,   1.78,   2.14, 11.2 ],
```

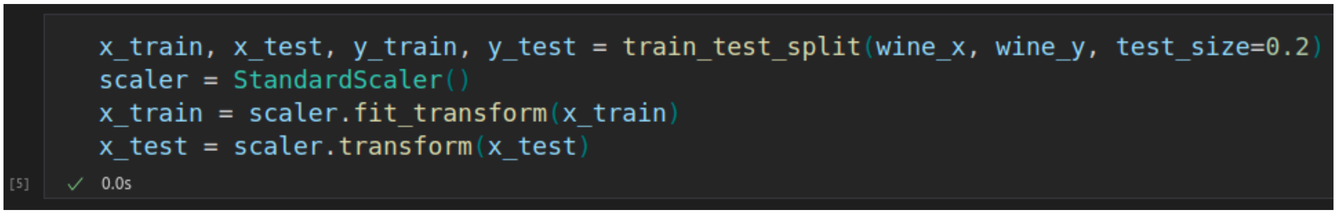
4. And select **target data as wine_y**.



```
wine_y = wine.target
wine_y.shape
```

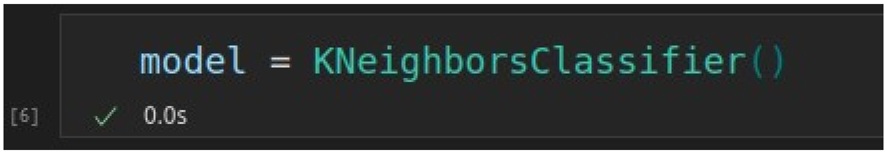
```
(178,)
```

5. Then split data into train and test, transform and fit transform.



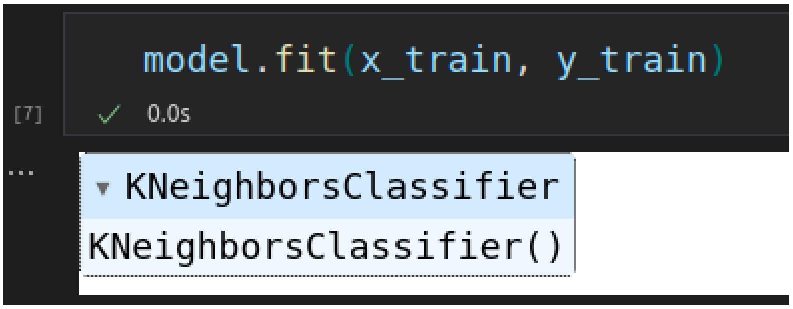
```
x_train, x_test, y_train, y_test = train_test_split(wine_x, wine_y, test_size=0.2)
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
```

6. Now use **KneighborsClassifier as model**.



```
model = KNeighborsClassifier()
```

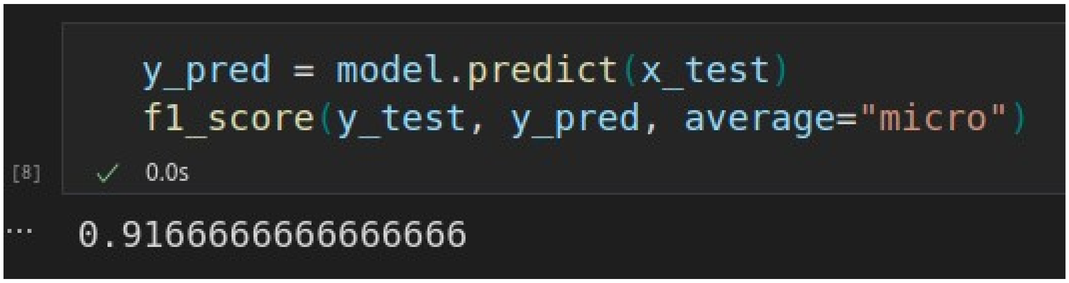
7. Now, use **model.fit**.



```
model.fit(x_train, y_train)
```

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

8. Now, **predict the model** and show the **f1 score**.



```
y_pred = model.predict(x_test)
f1_score(y_test, y_pred, average="micro")
```

```
0.9166666666666666
```

In the end, we are getting **91%** accuracy.