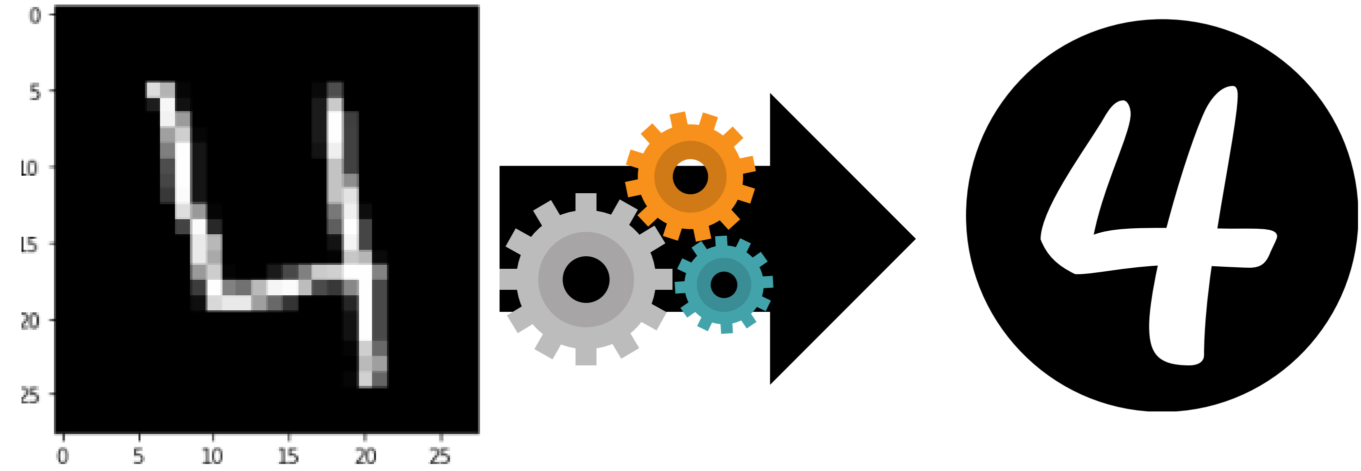
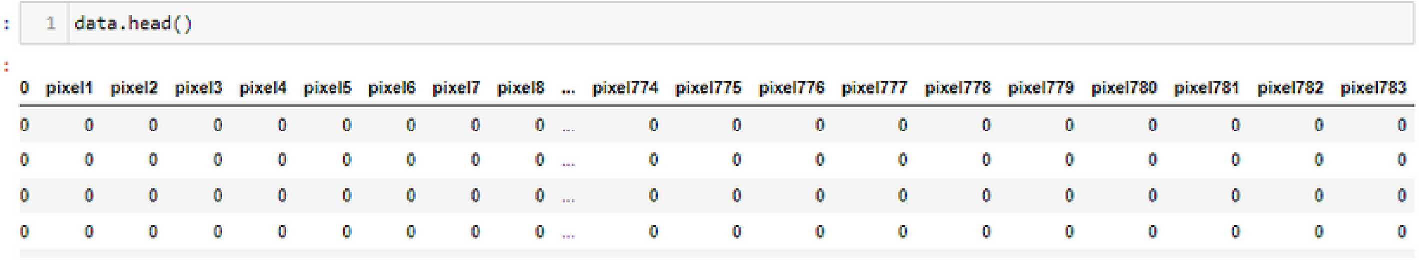
Project No #6

digit recognition using Machine Learning

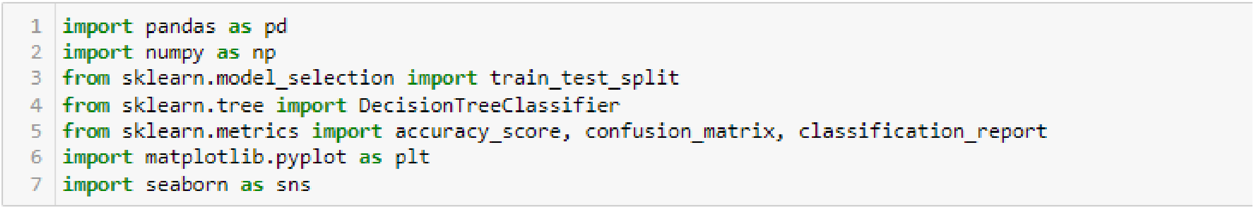
Using Supervised Algorithms



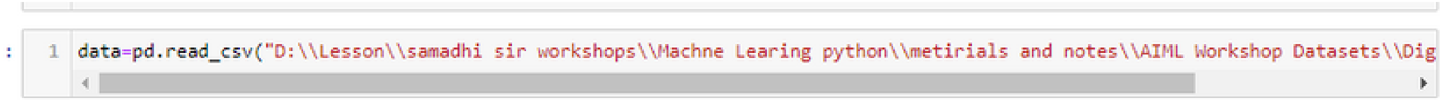


```
data.head()
```

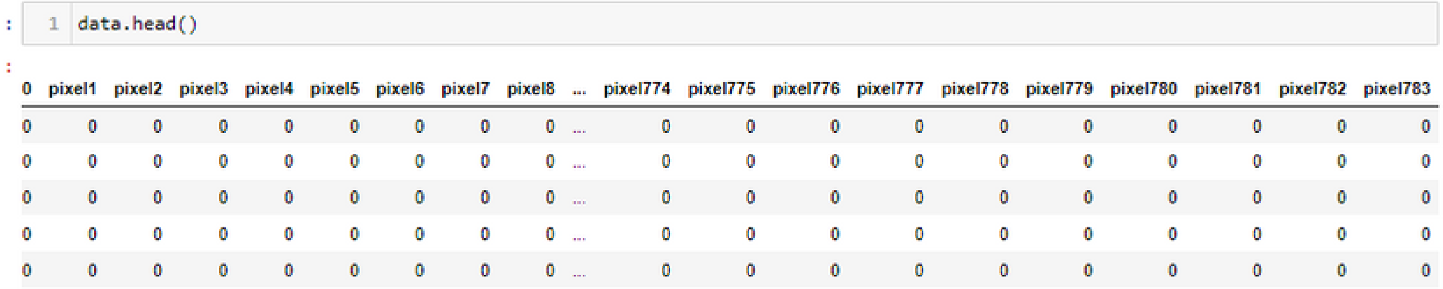
0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	...	pixel774	pixel775	pixel776	pixel777	pixel778	pixel779	pixel780	pixel781	pixel782	pixel783
0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0



```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import matplotlib.pyplot as plt
import seaborn as sns
```



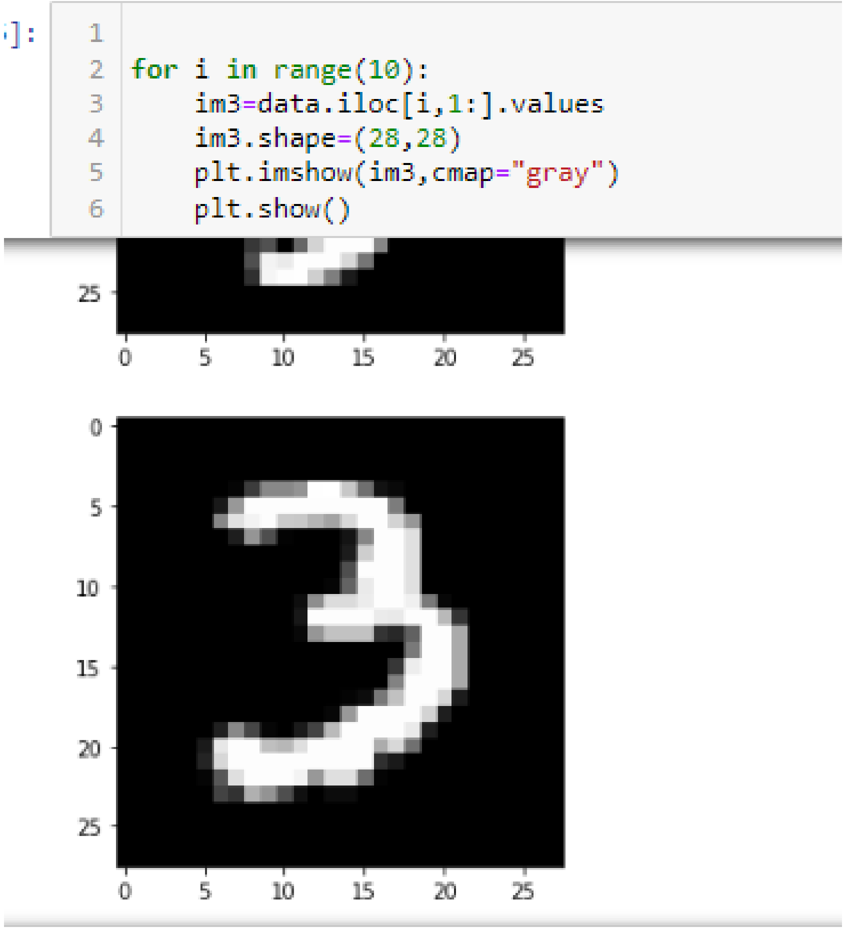
```
data=pd.read_csv("D:\\Lesson\\samadhi sir workshops\\Machne Learing python\\metirials and notes\\AIML Workshop Datasets\\Dig
```



```
data.head()
```

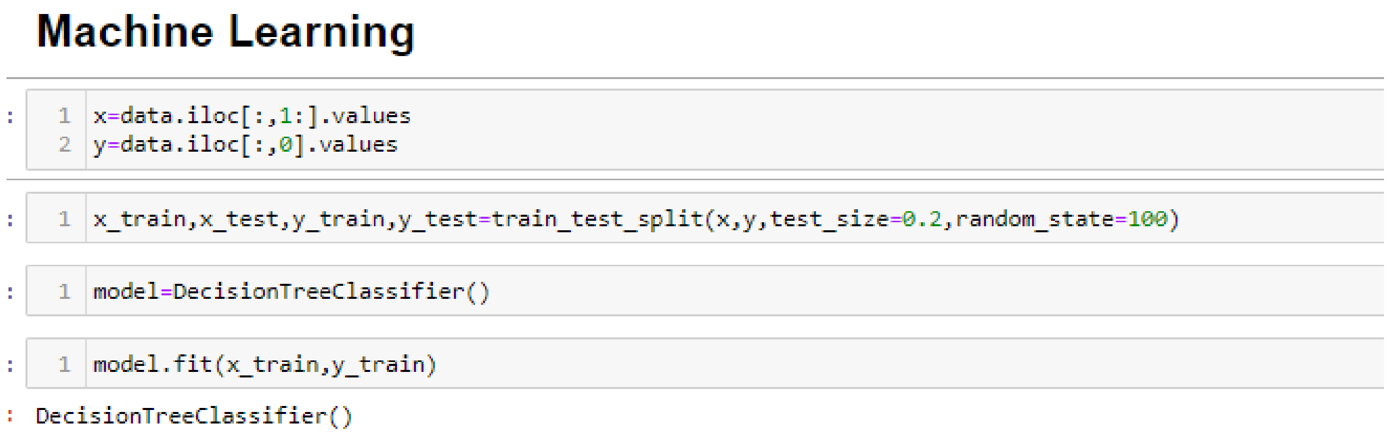
0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	...	pixel774	pixel775	pixel776	pixel777	pixel778	pixel779	pixel780	pixel781	pixel782	pixel783
0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

Visualizing letters



```

for i in range(10):
    im3=data.iloc[i,1:].values
    im3.shape=(28,28)
    plt.imshow(im3,cmap="gray")
    plt.show()
```



Machine Learning

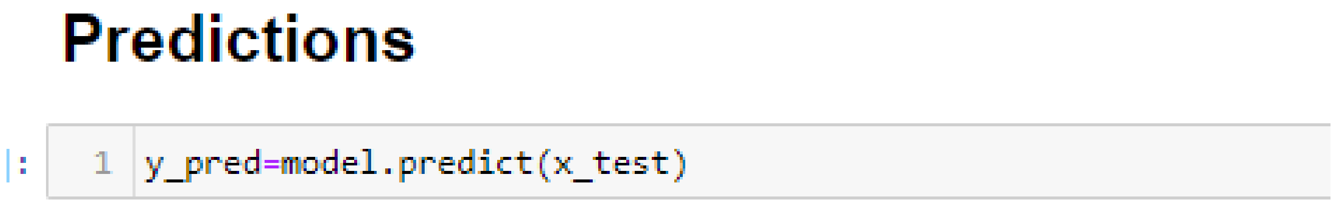
```
x=data.iloc[:,1:].values
y=data.iloc[:,0].values
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=100)
```

```
model=DecisionTreeClassifier()
```

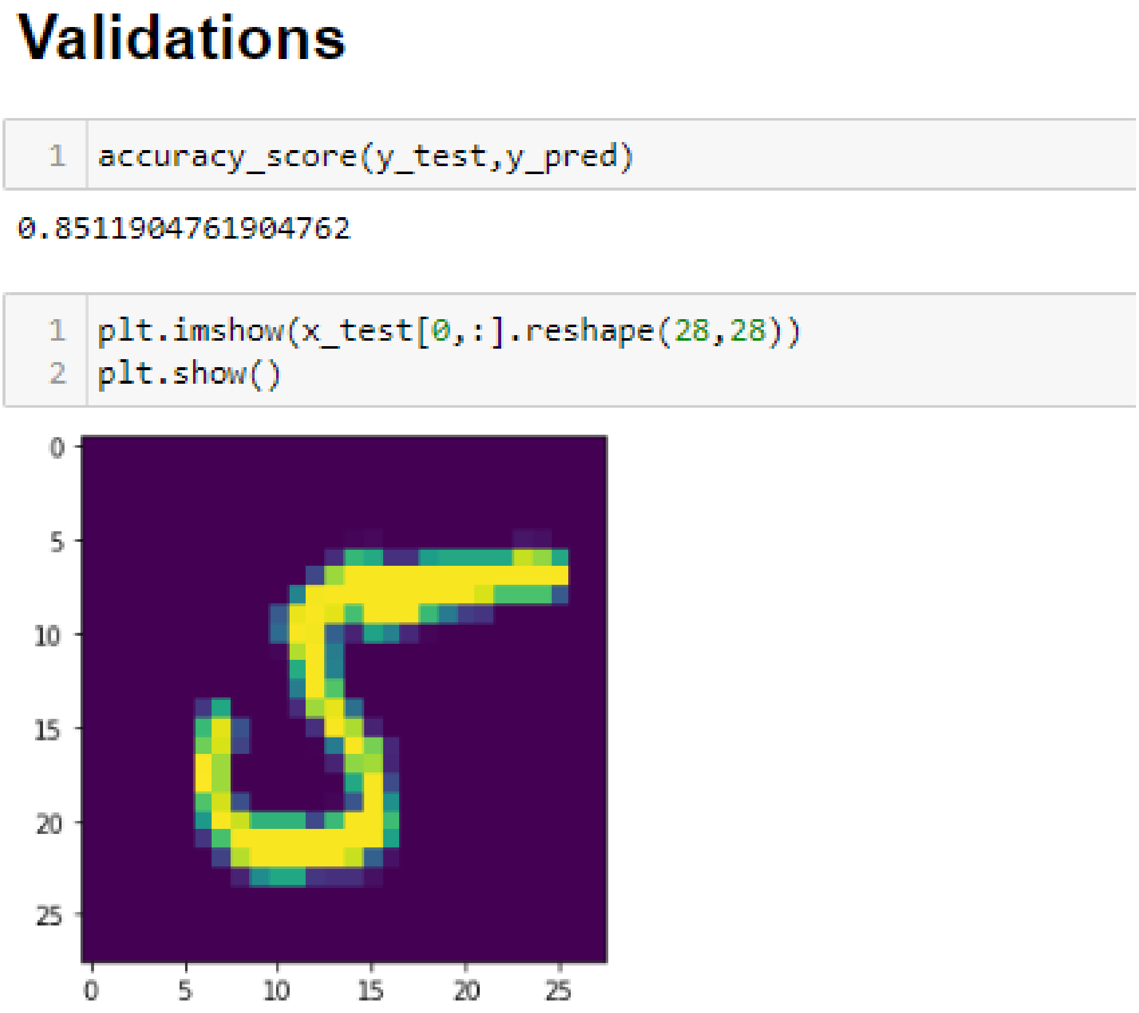
```
model.fit(x_train,y_train)
```

```
DecisionTreeClassifier()
```



Predictions

```
y_pred=model.predict(x_test)
```



Validations

```
accuracy_score(y_test,y_pred)
```

```
0.8511904761904762
```

```
plt.imshow(x_test[0,:].reshape(28,28))
plt.show()
```

```
y_test[0]
```

```
5
```

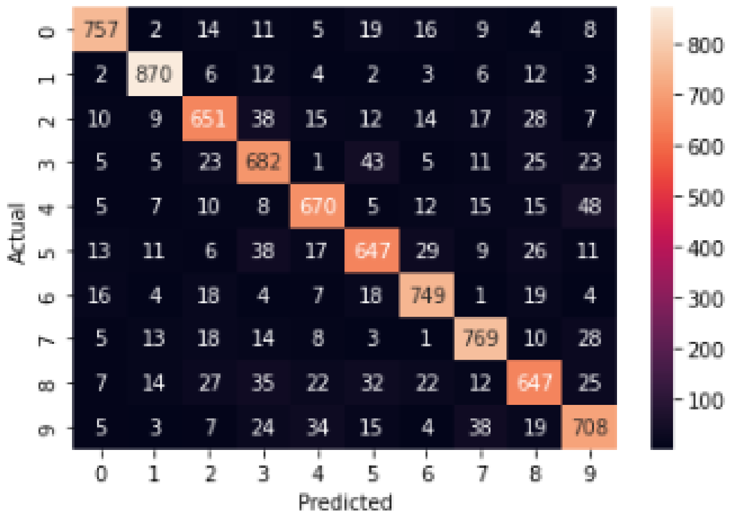
```
y_pred[0]
```

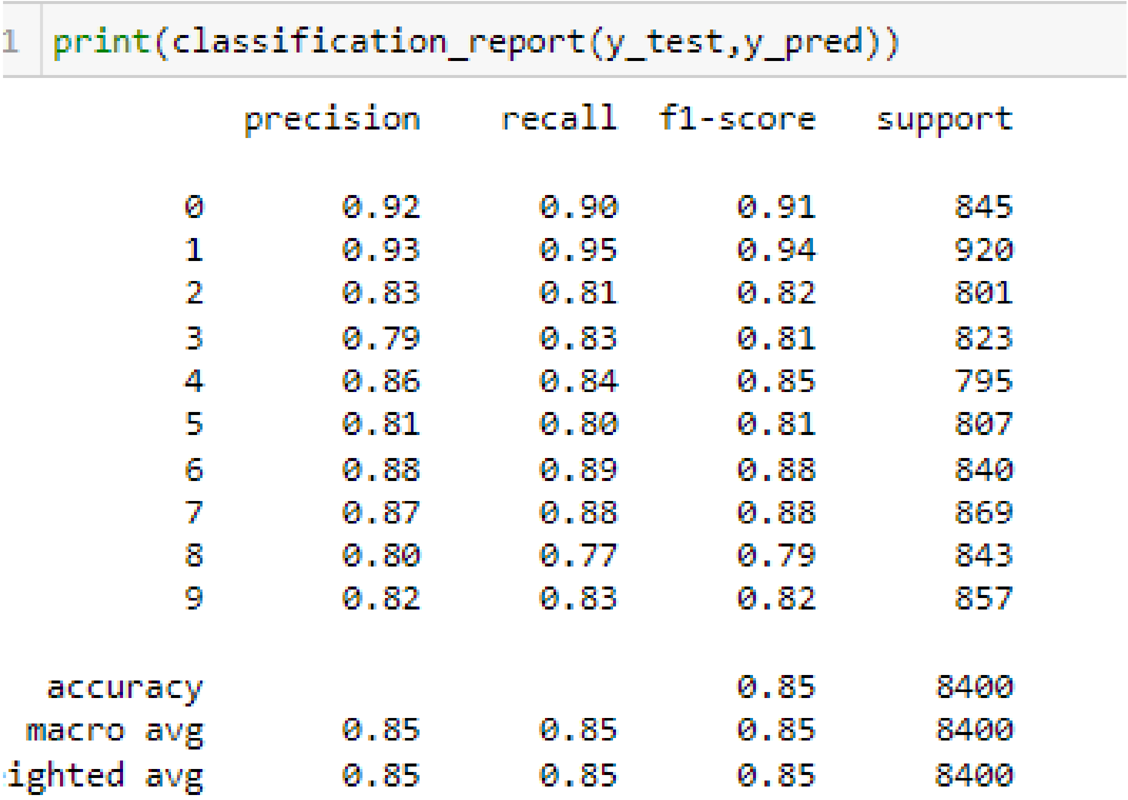
```
5
```

```
confusion_matrix(y_test,y_pred)
```

```
array([[757,   2,  14,  11,   5,  19,  16,   9,   4,   8],
       [  2, 870,   6,  12,   4,   2,   3,   6,  12,   3],
       [ 10,   9, 651,  38,  15,  12,  14,  17,  28,   7],
       [  5,   5,  23, 682,   1,  43,   5,  11,  25,  23],
       [  5,   7,  10,   8, 670,   5,  12,  15,  15,  48],
       [ 13,  11,   6,  38,  17, 647,  29,   9,  26,  11],
       [ 16,   4,  18,   4,   7,  18, 749,   1,  19,   4],
       [  5,  13,  18,  14,   8,   3,   1, 769,  10,  28],
       [  7,  14,  27,  35,  22,  32,  22,  12, 647,  25],
       [  5,   3,   7,  24,  34,  15,   4,  38,  19, 708]], dtype=int64)
```

```
sns.heatmap(confusion_matrix(y_test,y_pred),annot=True,fmt="g")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```



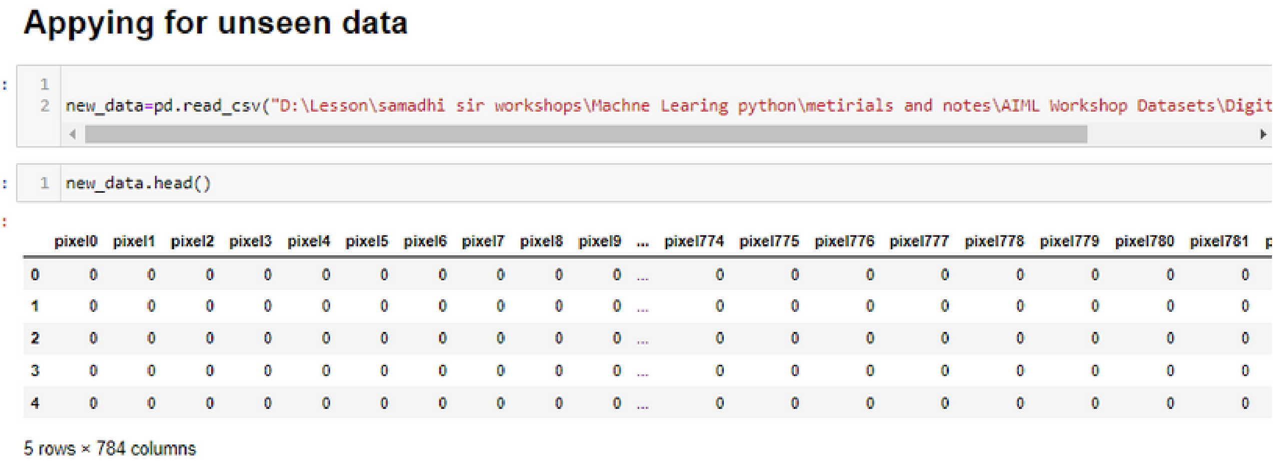


```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.92      0.90      0.91       845
           1       0.93      0.95      0.94       920
           2       0.83      0.81      0.82       801
           3       0.79      0.83      0.81       823
           4       0.86      0.84      0.85       795
           5       0.81      0.80      0.81       807
           6       0.88      0.89      0.88       840
           7       0.87      0.88      0.88       869
           8       0.80      0.77      0.79       843
           9       0.82      0.83      0.82       857

    accuracy                           0.85      8400
   macro avg       0.85      0.85      0.85      8400
ighted avg       0.85      0.85      0.85      8400
```



Appying for unseen data

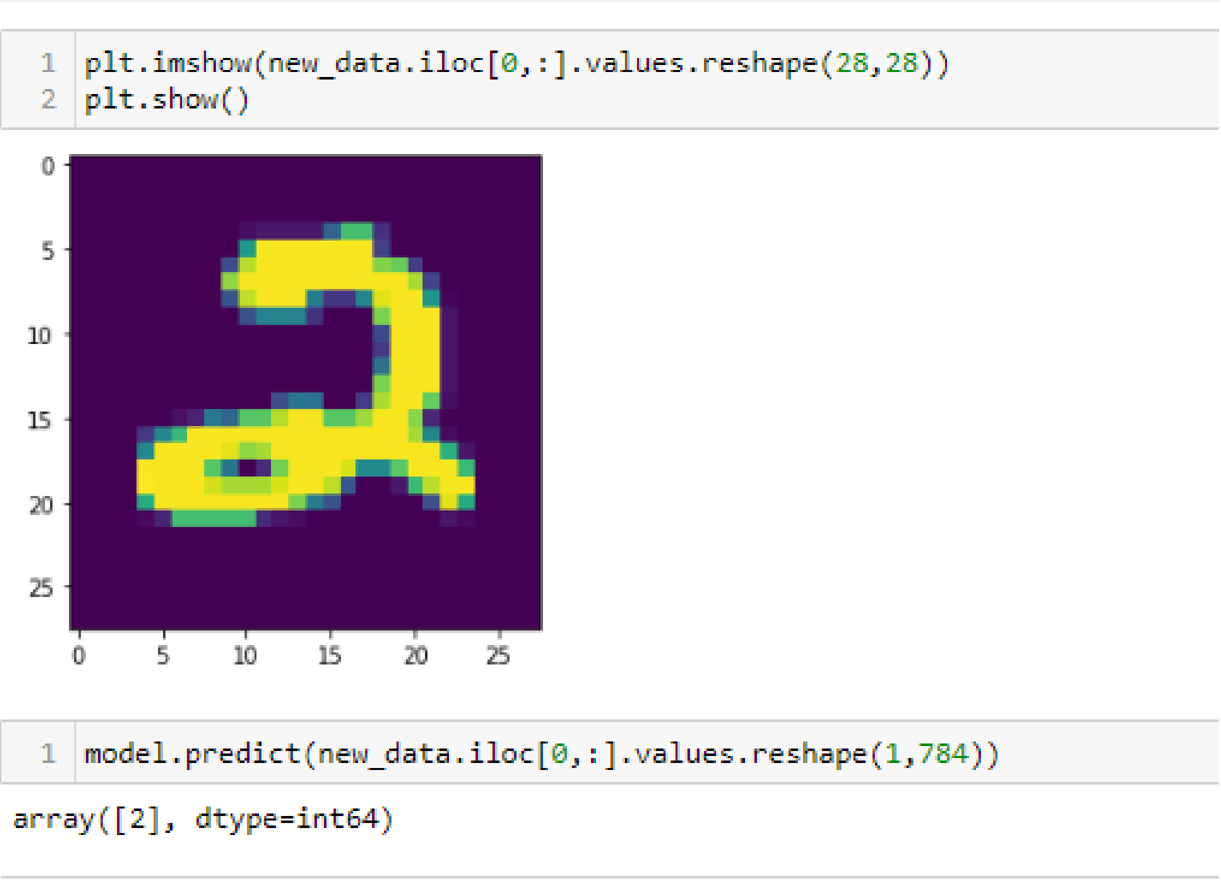
```

new_data=pd.read_csv("D:\Lesson\samadhi sir workshops\Machne Learing python\metirials and notes\AIML Workshop Datasets\Digit
```

```
new_data.head()
```

	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel774	pixel775	pixel776	pixel777	pixel778	pixel779	pixel780	pixel781	p
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	

5 rows × 784 columns



```
plt.imshow(new_data.iloc[0,:].values.reshape(28,28))
plt.show()
```

```
model.predict(new_data.iloc[0,:].values.reshape(1,784))
```

```
array([2], dtype=int64)
```