

INTRODUCTION TO MACHINE LEARNING

LAB ASSIGNMENT – 6

NAME : PRATHAPANI SATWIKA

REG.NO. :20BCD7160

Implement SVM on the data set provided.

```
import pandas as pd
from sklearn import svm
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, classification_report
from sklearn.model_selection import train_test_split
```

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

```
[ ] train_data=pd.read_csv('/content/drive/My Drive/lab4/mnist_train.csv')
    print(train_data)
```

```
0      5  0  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  ...  0.608  0.609  0.610  \
1      4  0  0  0  0  0  0  0  0  0  0  ...  0  0  0
2      1  0  0  0  0  0  0  0  0  0  0  ...  0  0  0
3      9  0  0  0  0  0  0  0  0  0  0  ...  0  0  0
4      2  0  0  0  0  0  0  0  0  0  0  ...  0  0  0
...    ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
59994  8  0  0  0  0  0  0  0  0  0  0  ...  0  0  0
59995  3  0  0  0  0  0  0  0  0  0  0  ...  0  0  0
59996  5  0  0  0  0  0  0  0  0  0  0  ...  0  0  0
59997  6  0  0  0  0  0  0  0  0  0  0  ...  0  0  0
59998  8  0  0  0  0  0  0  0  0  0  0  ...  0  0  0

      0.611  0.612  0.613  0.614  0.615  0.616  0.617
0          0          0          0          0          0          0
1          0          0          0          0          0          0
2          0          0          0          0          0          0
3          0          0          0          0          0          0
4          0          0          0          0          0          0
...    ...    ...    ...    ...    ...    ...
59994      0          0          0          0          0          0
59995      0          0          0          0          0          0
59996      0          0          0          0          0          0
59997      0          0          0          0          0          0
59998      0          0          0          0          0          0
```

[59999 rows x 785 columns]

```
test_data=pd.read_csv('/content/drive/My Drive/Colab Notebooks/mnist_test.csv')
print(test_data)
```

```

0      7  0  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  ...  0.658  0.659  0.660  \
0      2  0    0    0    0    0    0    0    0    0  ...    0    0    0
1      1  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
3      4  0    0    0    0    0    0    0    0    0  ...    0    0    0
4      1  0    0    0    0    0    0    0    0    0  ...    0    0    0
...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
9994   2  0    0    0    0    0    0    0    0    0  ...    0    0    0
9995   3  0    0    0    0    0    0    0    0    0  ...    0    0    0
9996   4  0    0    0    0    0    0    0    0    0  ...    0    0    0
9997   5  0    0    0    0    0    0    0    0    0  ...    0    0    0
9998   6  0    0    0    0    0    0    0    0    0  ...    0    0    0

      0.661  0.662  0.663  0.664  0.665  0.666  0.667
0          0          0          0          0          0          0
1          0          0          0          0          0          0
2          0          0          0          0          0          0
3          0          0          0          0          0          0
4          0          0          0          0          0          0
...  ...  ...  ...  ...  ...  ...
9994   0          0          0          0          0          0
9995   0          0          0          0          0          0
9996   0          0          0          0          0          0
9997   0          0          0          0          0          0
9998   0          0          0          0          0          0

```

[9999 rows x 785 columns]

```

x=train_data.iloc[:,1:785]
y=train_data.iloc[:,0]
print("X.shape :",x.shape)
print("Y.shape :",y.shape)

```

X.shape : (59999, 784)
Y.shape : (59999,)

```

[ ] x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=10)
    model=svm.SVC()
    print(model.fit(x_train,y_train))

```

SVC()

```
[ ] y_train_predict=model.predict(x_train)
y_test_predict=model.predict(x_test)
print("Accuracy Score:",accuracy_score(y_train,y_train_predict))
print("\n CONFUSION MATRIX\n")
print(confusion_matrix(y_test,y_test_predict))
```

Accuracy Score: 0.9889759279982857

CONFUSION MATRIX

```
[[1812    0    4    0    0    2    6    0    1    2]
 [   0 2016    8    4    3    0    1    2    1    3]
 [    7    4 1717    2    9    2    4   15   10    1]
 [    0    2   12 1705    1   22    1    8   16   10]
 [    1    3    0    0 1711    0    3    2    1   19]
 [    5    1    5    7    4 1602   10    1    4    3]
 [    7    2    1    0    2    8 1731    0    1    0]
 [    1    3   11    2   12    0    0 1860    2   11]
 [    3    9    8    8    5   11    3    1 1674    6]
 [    5    1    1   10   22    6    0   19    8 1751]]
```



```
print("\n CLASSIFICATION REPORT \n")
print(classification_report(y_test,y_test_predict))
```

CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.98	0.99	0.99	1827
1	0.99	0.99	0.99	2038
2	0.97	0.97	0.97	1771
3	0.98	0.96	0.97	1777
4	0.97	0.98	0.98	1740
5	0.97	0.98	0.97	1642
6	0.98	0.99	0.99	1752
7	0.97	0.98	0.98	1902
8	0.97	0.97	0.97	1728
9	0.97	0.96	0.97	1823
accuracy			0.98	18000
macro avg	0.98	0.98	0.98	18000
weighted avg	0.98	0.98	0.98	18000