

In [1]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
%matplotlib inline
```

In [2]:

```
data=pd.read_csv("mnist_test.csv.zip")
```

In [7]:

```
data.head()
```

Out[7]:

	label	1x1	1x2	1x3	1x4	1x5	1x6	1x7	1x8	1x9	...	28x19	28x20	28x21	28x22	28x23	28x24	28x25	28x26	28x27	28x28
0	7	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	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	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	0	0
4	4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 785 columns

In [4]:

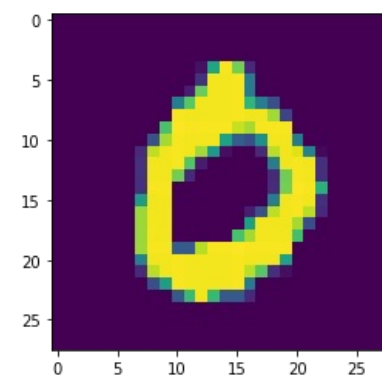
```
a=data.iloc[3,1:].values
```

In [5]:

```
#reshaping
a = a.reshape(28,28).astype('uint8')
plt.imshow(a)
```

Out[5]:

<matplotlib.image.AxesImage at 0x1963314f648>



In [9]:

```
df_x= data.iloc[:,1:]
df_y=data.iloc[:,0]
```

In [11]:

```
##creating test and train model
x_train,x_test,y_train,y_test = train_test_split(df_x,df_y,test_size=0.2,random_state=4)
```

In [12]:

```
#check_data
x_train.head()
```

Out[12]:

	1x1	1x2	1x3	1x4	1x5	1x6	1x7	1x8	1x9	1x10	...	28x19	28x20	28x21	28x22	28x23	28x24	28x25	28x26	28x27	28x28
4983	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
6789	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2221	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
6043	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1564	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 784 columns

In [13]:

```
y_train.head()
```

Out[13]:

```
4983    3
6789    1
2221    1
6043    5
1564    7
Name: label, dtype: int64
```

In [14]:

```
# call rf classifier
rf=RandomForestClassifier(n_estimators=100)
```

In [15]:

```
# fit the model
rf.fit(x_train,y_train)
```

Out[15]:

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                        criterion='gini', max_depth=None, max_features='auto',
                        max_leaf_nodes=None, max_samples=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, n_estimators=100,
                        n_jobs=None, oob_score=False, random_state=None,
                        verbose=0, warm_start=False)
```

In [17]:

```
#prediction
pred=rf.predict(x_test)
```

In [18]:

```
pred
```

Out[18]:

```
array([1, 3, 7, ..., 0, 1, 0], dtype=int64)
```

In [19]:

```
#check prediction accuracy
s=y_test.values
# calculate no. of correctly predicted values
count=0
for i in range(len(pred)):
    if pred[i]==s[i]:
        count=count+1
```

In [20]:

```
count
```

Out[20]:

```
1893
```

In [21]:

```
#total values that from prediction code was run on  
len(pred)
```

Out[21]:

```
2000
```

In [22]:

```
#accuracy  
1893/2000
```

Out[22]:

```
0.9465
```

In [ ]: