In [11]:

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

In [12]:

```
data=pd.read_csv('mnist.csv')#reading the data
```

In [13]:

```
data.head()
```

Out[13]:

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

5 rows × 785 columns

In [14]:

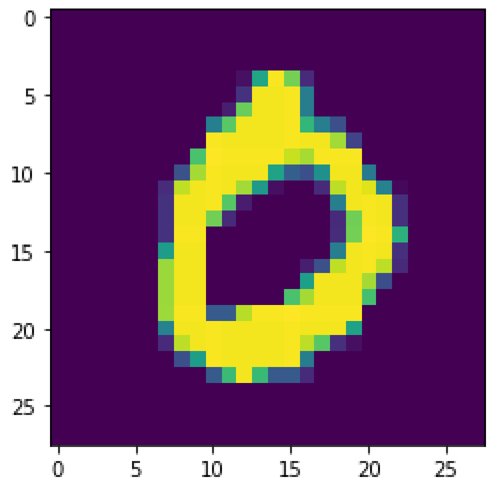
```
a = data.iloc[3,1:].values
```

In [15]:

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

Out[15]:

```
<matplotlib.image.AxesImage at 0x1c89e8bb788>
```



In [16]:

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

In [17]:

```
x_train, x_test, y_train, y_test = train_test_split(df_x, df_y, test_siz
e = 0.2, random_state=4)
```

In [20]:

```
x_train.head()
```

Out[20]:

	1x1	1x2	1x3	1x4	1x5	1x6	1x7	1x8	1x9	1x10	...	28x19	28x20	28x21	28x22
4983	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
2221	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
1564	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0

5 rows × 784 columns

In [19]:

```
y_train.head()
```

Out[19]:

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

In [21]:

```
rf = RandomForestClassifier(n_estimators=100)
```

In [22]:

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

Out[22]:

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='a
uto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=Non
e,
                       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 [23]:

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

In [24]:

```
pred
```

Out[24]:

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

In [25]:

```
s = y_test.values

count = 0
for i in range(len(pred)):
    if pred[i] == s[i]:
        count = count+1
```

In [26]:

```
count
```

Out[26]:

```
1895
```

In [27]:

```
len(pred)
```

Out[27]:

```
2000
```

In [28]:

```
1895/200
```

Out[28]:

```
9.475
```