In [ ]:
```
#import depensencies
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 [5]:
```
#using pandas to read the database stored in the same folder
data=pd.read_csv('mnist_test.csv')
```

In [6]:
```
#viewing column heads
data.head()
```

Out[6]:

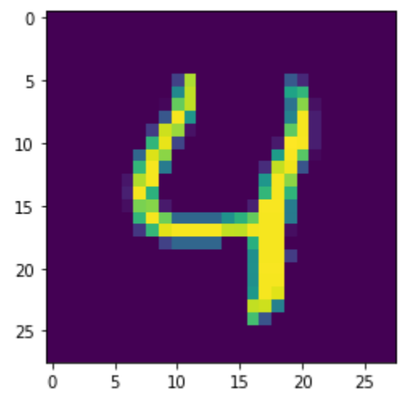
	7	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	...	0.658	0.659	0.660	0.661	0.662	0.663	0.664	0.665	0.666	0.667
0	2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 785 columns

In [7]:
```
#extracting data from the dataset and viewing them up close
a=data.iloc[3,1:].values
```

In [10]:
```
#reshaping the extracted data into a reasonable size
a=a.reshape(28,28).astype('uint8')
plt.imshow(a)
```

Out[10]:
```
<matplotlib.image.AxesImage at 0x21aa087388>
```



In [11]:
```
#preparing the data
#seperting lables and data values
df_x=data.iloc[:,1:]
df_y=data.iloc[:,0]
```

In [12]:
```
#creating test and train sizes/batches
x_train,x_test,y_train,y_test=train_test_split(df_x,df_y,test_size=0.2,random_state=4)
```

In [13]:
```
#check data
x_train.head()
```

Out[13]:

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	...	0.658	0.659	0.660	0.661	0.662	0.663	0.664	0.665	0.666	0.667
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 [14]:
```
y_train.head()
```

Out[14]:
```
4983    1
6789    3
2221    3
6043    1
1564    0
Name: 7, dtype: int64
```

In [15]:
```
#call rf classifier
rf=RandomForestClassifier(n_estimators=100)
```

In [16]:
```
#fit the model
rf.fit(x_train,y_train)
```

Out[16]:
```
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 on the test data
pred=rf.predict(x_test)
```

In [18]:
```
pred
```

Out[18]:
```
array([2, 2, 2, ..., 0, 9, 2], dtype=int64)
```

In [20]:
```
#check prediction accuracy
s=y_test.values

#calculatenumber of correctly predicted values
count=0
for i in range(len(pred)):
    if pred[i]==[i]:
        count=count+1
```

In [21]:
```
count
```

Out[21]:
```
2
```

In [22]:
```
#total values that the prediction code was run on
len(pred)
```

Out[22]:
```
2000
```

In [23]:
```
#accuracy value
2/2000
```

Out[23]:
```
0.001
```

In [ ]:
```

```