In [ ]:
```
import tensorflow as tf
```

In [ ]:
```
from tensorflow import keras
```

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import random
%matplotlib inline
```

In [ ]:
```
mnist=tf.keras.datasets.mnist
(x_train,y_train),(x_test,y_test)=mnist.load_data()
```

In [ ]:
```
plt.matshow(x_train[0])
```

In [ ]:
```
x_train=x_train/255
x_test=x_test/255
```

In [ ]:
```
x_train[0]
```

In [ ]:
```
model=keras.Sequential([
    keras.layers.Flatten(input_shape=(28,28)),
    keras.layers.Dense(128,activation='relu'),
    keras.layers.Dense(10,activation='softmax')
])
```

In [ ]:
```
model.summary()
```

In [ ]:
```
model.compile(optimizer='sgd',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
```

In [ ]:
```
history=model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=10)
```

In [ ]:
```
test_loss,test_acc=model.evaluate(x_test,y_test)
print("Loss=%.3f" %test_loss)
print("Accuracy=%.3f" %test_acc)
```

In [ ]:
```
n=random.randint(0,9999)
plt.imshow(x_test[n])
plt.show
```

In [ ]:
```
test_predict=model.predict(x_test)
test_predict_labels=np.argmax(test_predict,axis=1)
confusion_matrix=tf.math.confusion_matrix(labels=y_test,predictions=test_predict_labels)
print('confusion matrix of the test set :\n', confusion_matrix)
```

In [ ]:
```

```