```
import tensorflow as tf    ass1
from tensorflow import keras
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import random
%matplotlib inline

mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
len(x_train)
len(x_test)
len(y_test)
len(y_train)
x_train.shape
x_train[0]
import matplotlib.pyplot as plt
plt.matshow(x_train[0])
plt.imshow(-x_train[0], cmap="gray")
x_train = x_train / 255
x_test = x_test / 255
x_train = x_train / 255
x_test = x_test / 255
x_train[0]
model = keras.Sequential([
keras.layers.Flatten(input_shape=(28, 28)),
keras.layers.Dense(128, activation="relu"),
keras.layers.Dense(10, activation="softmax")
])
model.summary()

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

history=model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=10)

test_loss,test_acc=model.evaluate(x_test,y_test)
print("Loss=%.3f" %test_loss)
print("Accuracy=%.3f" %test_acc)

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

predicted_value=model.predict(x_test)
print("Handwritten number in the image is= %d" %np.argmax(predicted_value[n]))

history.history??
history.history.keys()

plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```