

use MNIST fashion dataset and create a classifier to classify fashion clothing into categories

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

class_name = ['T-shirt/top', 'Trouser', 'Pollover', 'Dress', 'Coat', 'Sandel',
'Shirt', 'Sneakers', 'Bag', 'Ankleboot']

df1 = pd.read_csv("fashion-mnist_train.csv")
df2 = pd.read_csv("fashion-mnist_test.csv")

x_train = df1.drop("label", axis = 1).values
y_train = df1["label"].values
x_train.shape

x_test = df2.drop("label", axis = 1).values
y_test = df2["label"].values

x_train = x_train.reshape(60000, 28, 28)
x_test = x_test.reshape(10000, 28, 28)

plt.imshow(x_train[0],cmap='gray')

x_train = x_train/255
x_test = x_test/255

x_train = x_train.reshape(60000, 28, 28,1)
x_test = x_test.reshape(10000, 28, 28,1)

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, MaxPooling2D, Flatten

model = Sequential()
model.add(Conv2D(64, (3,3), activation= 'relu', input_shape = (28,28,1)))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(64, (3,3), activation= 'relu'))
model.add(MaxPooling2D((2,2)))
model.add(Flatten())
model.add(Dense(128, activation= 'relu'))
model.add(Dense(10, activation= 'softmax'))
model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy',
metrics = ['accuracy'])
model.summary()

model.fit(x_train, y_train, epochs= 5, validation_data= (x_test, y_test))

predictions = model.predict(x_test)

import numpy as np
index = 10
print(predictions[index])
final_value = np.argmax(predictions[index])
```

```
print("Actual: ",y_test[index])
print("Predicted: ", final_value)
print("Class Label: ", class_name[final_value])

plt.imshow(x_test[index])

loss, accuracy = model.evaluate(x_test,y_test)
print("Loss : ", loss)
print("Accuracy: ", accuracy*100)
```