

```
In [41]: import tensorflow as tf
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tensorflow import keras
```

```
In [42]: fashion_mnist = keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
32768/29515 [=====] - 0s 8us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26427392/26421880 [=====] - 46s 2us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
8192/5148 [=====] - 0s 0s/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4423680/4422102 [=====] - 11s 3us/step
```

```
In [45]: train_images.shape, test_images.shape
```

```
Out[45]: ((60000, 28, 28), (10000, 28, 28))
```

```
In [50]: train_images[0,23,23]
```

```
Out[50]: 194
```

```
In [53]: train_images[:3,:5,:7]
```

```
Out[53]: array([[ [ 0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0]],

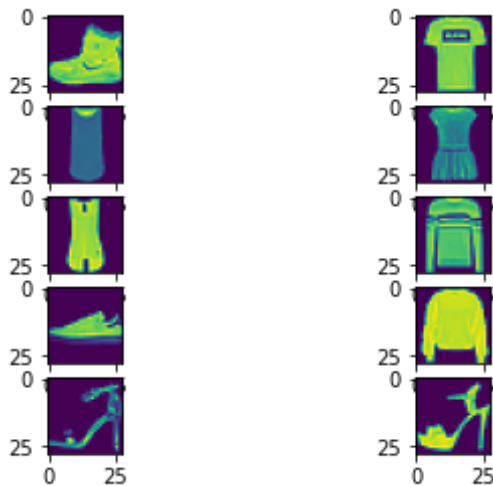
 [[ [ 0,  0,  0,  0,  0,  1,  0],
 [ 0,  0,  0,  1,  0,  0,  0],
 [ 0,  0,  0,  0,  0, 14, 176],
 [ 0,  0,  0,  0,  0,188, 219],
 [ 0,  0,  0,  0, 51, 219, 199]])
```

```
[[ 0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0]], dtype=uint8)
```

```
In [54]: train_labels[:10]
```

```
Out[54]: array([9, 0, 0, 3, 0, 2, 7, 2, 5, 5], dtype=uint8)
```

```
In [97]: for i in range(10):
          plt.subplot(5,2,1+i)
          plt.imshow(train_images[i])
```



```
In [71]: train_images = train_images/255
          test_images = test_images /255
          train_images[:,3,:5,:7]
```

```
Out[71]: array([[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
                  0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
                 [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
                  0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
                 [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
                  0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
                 [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
                  0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
                 [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
                  0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
```

```
[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00]],

[[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 1.53787005e-05, 0.00000000e+00],
[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 1.53787005e-05,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 2.15301807e-04, 2.70665129e-03],
[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 2.89119569e-03, 3.36793541e-03],
[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
7.84313725e-04, 3.36793541e-03, 3.06036140e-03]],

[[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00],
[0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00]]])
```

```
In [99]: class_names = ["Sandel", "T-shirt", "Trouser", "Dress", "Pullover", "Shirt", "Bag", "Coat", "Sneaker", "Ankle Boot"]
```

```
In [80]: model = keras.Sequential([
keras.layers.Flatten(input_shape=(28,28)),
keras.layers.Dense(128,activation = "relu"),
keras.layers.Dense(10,activation="softmax")
])
```

```
In [82]: model.compile(optimizer="adam",
loss="sparse_categorical_crossentropy",
metrics=["accuracy"])
```

```
In [84]: model.fit(train_images,train_labels,epochs=10)
```

```
Epoch 1/10
1875/1875 [=====] - 13s 7ms/step - loss: 0.5234 - accuracy: 0.8145
Epoch 2/10
1875/1875 [=====] - 10s 5ms/step - loss: 0.4943 - accuracy: 0.8247
```

```

Epoch 3/10
1875/1875 [=====] - 7s 4ms/step - loss: 0.4731 - accuracy: 0.8332
Epoch 4/10
1875/1875 [=====] - 11s 6ms/step - loss: 0.4579 - accuracy: 0.8378
Epoch 5/10
1875/1875 [=====] - 12s 7ms/step - loss: 0.4453 - accuracy: 0.8427
Epoch 6/10
1875/1875 [=====] - 18s 10ms/step - loss: 0.4354 - accuracy: 0.8463
Epoch 7/10
1875/1875 [=====] - 15s 8ms/step - loss: 0.4259 - accuracy: 0.8501 0s - loss: 0.4254 - accur
acy:
Epoch 8/10
1875/1875 [=====] - 17s 9ms/step - loss: 0.4191 - accuracy: 0.8519
Epoch 9/10
1875/1875 [=====] - 17s 9ms/step - loss: 0.4117 - accuracy: 0.8547
Epoch 10/10
1875/1875 [=====] - 17s 9ms/step - loss: 0.4059 - accuracy: 0.8570

```

Out[84]: <tensorflow.python.keras.callbacks.History at 0x216604be880>

```
In [87]: test_loss, test_acc = model.evaluate(test_images, test_labels, verbose=1)
        print(test_acc)
```

```

313/313 [=====] - 2s 4ms/step - loss: 0.4367 - accuracy: 0.8451
0.8450999855995178

```

```
In [89]: pred_images = model.predict(test_images)
```

```
In [90]: pred_images[0]
```

```
Out[90]: array([1.6935498e-07, 4.0761531e-08, 2.7129456e-06, 2.8052634e-06,
                6.9736393e-06, 1.6168678e-01, 8.2573642e-06, 2.5487080e-01,
                4.5638327e-03, 5.7885754e-01], dtype=float32)
```

```
In [91]: np.argmax(pred_images[0])
```

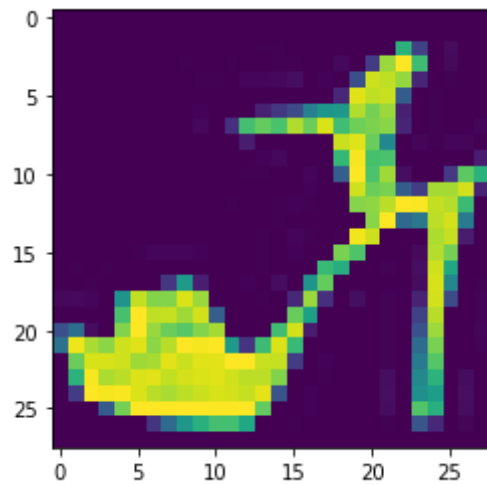
Out[91]: 9

```
In [100]: class_names[np.argmax(pred_images[0])]
```

Out[100]: 'Ankle Boot'

```
In [98]: plt.imshow(train_images[9])
```

```
Out[98]: <matplotlib.image.AxesImage at 0x21662e78730>
```



```
In [ ]:
```