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

```
%matplotlib inline
```

```
fashion_train_df= pd.read_csv('fashion-mnist_train.csv')
```

```
fashion_test_df = pd.read_csv('fashion-mnist_test.csv')
```

```
fashion_train_df.head()
```

	label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	.
0	2	0	0	0	0	0	0	0	0	0	
1	9	0	0	0	0	0	0	0	0	0	
2	6	0	0	0	0	0	0	0	5	0	
3	0	0	0	0	1	2	0	0	0	0	
4	3	0	0	0	0	0	0	0	0	0	

5 rows × 785 columns

```
fashion_train_df.tail()
```

	label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9
3306	3	0	0	0	0	0	0	0	1	1
3307	5	0	0	0	0	0	0	0	0	0
3308	0	0	0	0	0	0	0	0	0	0
3309	3	0	0	0	0	0	0	0	0	0
3310	8	0	0	0	0	0	0	0	0	0

5 rows × 785 columns

```
fashion_train_df.shape
```

```
(3311, 785)
```

```
fashion_test_df.shape
```

```
(4249, 785)
```

```
training = np.array(fashion_train_df,dtype='float32')
testing = np.array(fashion_test_df,dtype='float32')
```

```
training.shape
```

```
(3311, 785)
```

```
import random
```

```
W_grid = 7
L_grid = 7

fig,axes = plt.subplots(L_grid,W_grid,figsize =(17,17))

axes = axes.ravel()
n_training = len(training)



for i in np.arange(0,W_grid*L_grid):
        index = np.random.randint(0,n_training)
        axes[i].imshow(training[index,1:].reshape((28,28)))
        axes[i].set_title(training[index,0],fontsize = 8)
```

```
        axes[i].axis('off')

plt.subplots_adjust(hspace=0.4)
```


```
X_train = training[:,1:]/255
y_train = training[:,0]
X_test = testing[:,1:]/255
y_test = testing[:,0]


from sklearn.model_selection import train_test_split
X_train, X_validate, y_train, y_validate = train_test_split(X_train, y_train,test_size = 0.2,random_state = 12345)


X_train = X_train.reshape(X_train.shape[0],*(28,28,1))
X_test = X_test.reshape(X_test.shape[0],*(28,28,1))
X_validate = X_validate.reshape(X_validate.shape[0],*(28,28,1))


X_train.shape
```

```
     (2648, 28, 28, 1)
```

```
X_test.shape
```

```
     (4249, 28, 28, 1)
```

```
X_validate.shape
```

```
(663, 28, 28, 1)
```

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dense,Flatten,Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import TensorBoard


cnn_model = Sequential()
cnn_model.add(Conv2D(32,3,3,input_shape = (28,28,1),activation = 'relu'))
cnn_model.add(MaxPooling2D(pool_size= (2,2)))
cnn_model.add(Flatten())
cnn_model.add(Dense(32,activation = 'relu'))
cnn_model.add(Dense(10,activation = 'sigmoid'))
cnn_model.compile(loss ='sparse_categorical_crossentropy',optimizer = Adam(learning_rate=0.001),metrics= ['accuracy'])



epochs = 200



cnn_model.fit(X_train,y_train,batch_size =512,epochs = epochs,verbose = 1,validation_data = (X_validate,y_validate) )
```

```
Epoch 1/200
6/6 [==============================] - 2s 72ms/step - loss: nan - accuracy: 0.0974 - val_loss: nan - val_accuracy: 0.1071
Epoch 2/200
6/6 [==============================] - 0s 27ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 3/200
6/6 [==============================] - 0s 28ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 4/200
6/6 [==============================] - 0s 30ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 5/200
6/6 [==============================] - 0s 29ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 6/200
6/6 [==============================] - 0s 30ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 7/200
6/6 [==============================] - 0s 30ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 8/200
6/6 [==============================] - 0s 30ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 9/200
6/6 [==============================] - 0s 27ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 10/200
6/6 [==============================] - 0s 27ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 11/200
6/6 [==============================] - 0s 30ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 12/200
6/6 [==============================] - 0s 27ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 13/200
6/6 [==============================] - 0s 29ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 14/200
6/6 [==============================] - 0s 27ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 15/200
6/6 [==============================] - 0s 26ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 16/200
6/6 [==============================] - 0s 28ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 17/200
6/6 [==============================] - 0s 33ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 18/200
6/6 [==============================] - 0s 44ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 19/200
6/6 [==============================] - 0s 45ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 20/200
6/6 [==============================] - 0s 47ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 21/200
6/6 [==============================] - 0s 53ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 22/200
6/6 [==============================] - 0s 45ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 23/200
6/6 [==============================] - 0s 53ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 24/200
6/6 [==============================] - 0s 54ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 25/200
6/6 [==============================] - 0s 45ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 26/200
6/6 [==============================] - 0s 54ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 27/200
6/6 [==============================] - 0s 27ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 28/200
6/6 [==============================] - 0s 32ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
Epoch 29/200
6/6 [==============================] - 0s 27ms/step - loss: nan - accuracy: 0.1042 - val_loss: nan - val_accuracy: 0.1071
```

```
evaluation = cnn_model.evaluate(X_test,y_test)
print('Test Accuracy : {:.3f}'.format(evaluation[1]))
```

```
133/133 [==============================] - 0s 2ms/step - loss: nan - accuracy: 0.0986
Test Accuracy : 0.099
```

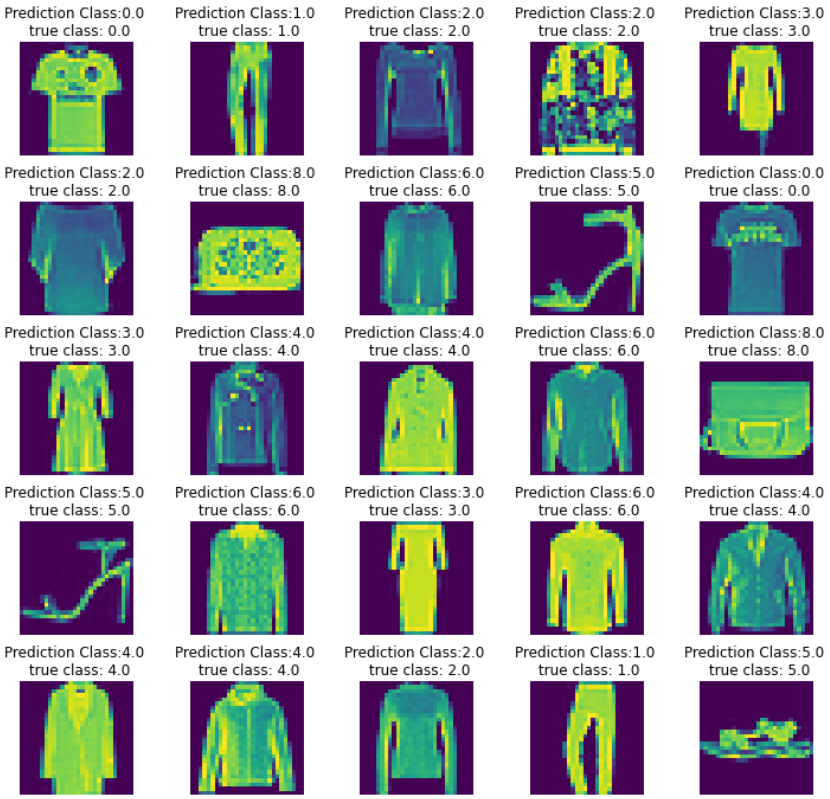
```
predicted_classes = np.argmax(cnn_model.predict(X_test),axis=-1)

predicted_classes
```

```
133/133 [==============================] - 1s 4ms/step
array([0, 0, 0, ..., 0, 0, 0])
```

```
L = 5
W = 5

fig,axes = plt.subplots(L,W,figsize = (12,12))
axes = axes.ravel()
for i in np.arange(0,L*W):
    axes[i].imshow(X_test[i].reshape(28,28))
    axes[i].set_title('Prediction Class:{1} \n true class: {1}'.format(predicted_classes[i],y_test[i]))
    axes[i].axis('off')
plt.subplots_adjust(wspace = 0.5)
```




```
from sklearn.metrics import classification_report

classes = 10
targets = ["Class {}".format(i) for i in range(classes)]
print(classification_report(y_test, predicted_classes, target_names = targets))
```

```
                precision    recall  f1-score   support

     Class 0       0.10      1.00      0.18       419
     Class 1       0.00      0.00      0.00       409
     Class 2       0.00      0.00      0.00       406
     Class 3       0.00      0.00      0.00       442
     Class 4       0.00      0.00      0.00       421
     Class 5       0.00      0.00      0.00       431
     Class 6       0.00      0.00      0.00       450
     Class 7       0.00      0.00      0.00       403
     Class 8       0.00      0.00      0.00       437
     Class 9       0.00      0.00      0.00       431

    accuracy                           0.10      4249
```

```
   macro avg       0.01      0.10      0.02      4249
weighted avg       0.01      0.10      0.02      4249

/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are i
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are i
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are i
  _warn_prf(average, modifier, msg_start, len(result))
```