# Assignnment 1

## Importing the libraries

In [1]:

```
import numpy as np
import pandas as pd
import tensorflow as tf
```

In [2]:

```
tf.__version__
```

Out[2]:

```
'2.18.0'
```

# Part 1 - Data Preprocessing

## Importing the dataset

In [3]:

```
ds = pd.read_csv('boston_housing.csv')
ds
```

Out[3]:

| | CRIM | ZN | INDUS | CHAS | NOX | RM | AGE | DIS | RAD | TAX | PTRATIO | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.00632 | 18.0 | 2.31 | 0 | 0.538 | 6.575 | 65.2 | 4.0900 | 1 | 296 | 15.3 | 396 |
| **1** | 0.02731 | 0.0 | 7.07 | 0 | 0.469 | 6.421 | 78.9 | 4.9671 | 2 | 242 | 17.8 | 396 |
| **2** | 0.02729 | 0.0 | 7.07 | 0 | 0.469 | 7.185 | 61.1 | 4.9671 | 2 | 242 | 17.8 | 392 |
| **3** | 0.03237 | 0.0 | 2.18 | 0 | 0.458 | 6.998 | 45.8 | 6.0622 | 3 | 222 | 18.7 | 394 |
| **4** | 0.06905 | 0.0 | 2.18 | 0 | 0.458 | 7.147 | 54.2 | 6.0622 | 3 | 222 | 18.7 | 396 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **501** | 0.06263 | 0.0 | 11.93 | 0 | 0.573 | 6.593 | 69.1 | 2.4786 | 1 | 273 | 21.0 | 391 |
| **502** | 0.04527 | 0.0 | 11.93 | 0 | 0.573 | 6.120 | 76.7 | 2.2875 | 1 | 273 | 21.0 | 396 |
| **503** | 0.06076 | 0.0 | 11.93 | 0 | 0.573 | 6.976 | 91.0 | 2.1675 | 1 | 273 | 21.0 | 396 |
| **504** | 0.10959 | 0.0 | 11.93 | 0 | 0.573 | 6.794 | 89.3 | 2.3889 | 1 | 273 | 21.0 | 393 |
| **505** | 0.04741 | 0.0 | 11.93 | 0 | 0.573 | 6.030 | 80.8 | 2.5050 | 1 | 273 | 21.0 | 396 |

506 rows × 14 columns

In [4]:

```
X = ds.iloc[:,:-1].values
y = ds.iloc[:, -1].values
```

## Splitting the dataset into the Training set and Test set

In [5]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, randc
```

### Feature Scaling

In [6]:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

## Part 2 - Building the ANN

### Initializing the ANN

In [7]:

```
ann = tf.keras.models.Sequential()
```

### Adding the input layer and the first hidden layer

In [8]:

```
ann.add(tf.keras.layers.Dense(units=150, activation='relu'))
```

### Adding the second hidden layer

In [9]:

```
ann.add(tf.keras.layers.Dense(units=100, activation='relu'))
```

In [10]:

```
ann.add(tf.keras.layers.Dense(units=50, activation='relu'))
```

### Adding the output layer

In [11]:

```
ann.add(tf.keras.layers.Dense(units=1, activation='relu'))
```

## Part 3 - Training the ANN

### Compiling the ANN

In [12]:

```
ann.compile(optimizer = 'adam', loss = 'mse', metrics = ['mae'])
```

### Training the ANN on the Training set

In [13]:

```
ann.fit(X_train, y_train, batch_size = 32, epochs = 100)
```

```
Epoch 1/100
```

```
13/13 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - loss: 563.7682 - mae: 21.9091
Epoch 2/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 487.1405 - mae: 20.1474
Epoch 3/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 341.0410 - mae: 16.0792
Epoch 4/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 111.9793 - mae: 8.5422
Epoch 5/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 56.9797 - mae: 5.7580
Epoch 6/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 31.1616 - mae: 4.0791
Epoch 7/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 22.3824 - mae: 3.4530
Epoch 8/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 20.5991 - mae: 3.2987
Epoch 9/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 19.8079 - mae: 3.2045
Epoch 10/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 23.2158 - mae: 3.2876
Epoch 11/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 14.2159 - mae: 2.6943
Epoch 12/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 13.4323 - mae: 2.6931
Epoch 13/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 11.0760 - mae: 2.4843
Epoch 14/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 14.3434 - mae: 2.5998
Epoch 15/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 11.7372 - mae: 2.5316
Epoch 16/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 10.7616 - mae: 2.3284
Epoch 17/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 10.6445 - mae: 2.3264
Epoch 18/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 9.4197 - mae: 2.2221
Epoch 19/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 10.1069 - mae: 2.4366
Epoch 20/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 10.1548 - mae: 2.3020
Epoch 21/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 8.2259 - mae: 2.1241
Epoch 22/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 9.3914 - mae: 2.2269
Epoch 23/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 9.7292 - mae: 2.1913
Epoch 24/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 8.8796 - mae: 2.1650
Epoch 25/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 8.1743 - mae: 2.0516
Epoch 26/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 7.7929 - mae: 2.0550
Epoch 27/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 8.4374 - mae: 2.0709
Epoch 28/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 7.1634 - mae: 1.9542
Epoch 29/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 6.8790 - mae: 1.9721
Epoch 30/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 6.9440 - mae: 1.9513
Epoch 31/100
```

```
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 6.5877 - mae: 1.9434
Epoch 32/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 7.2339 - mae: 2.0440
Epoch 33/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 7.5452 - mae: 1.9965
Epoch 34/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 6.3850 - mae: 1.8628
Epoch 35/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 6.4143 - mae: 1.9188
Epoch 36/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 6.4348 - mae: 1.9004
Epoch 37/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 7.0753 - mae: 1.9538
Epoch 38/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 6.8706 - mae: 1.9252
Epoch 39/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 7.5215 - mae: 2.0026
Epoch 40/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 6.2592 - mae: 1.8697
Epoch 41/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 6.3253 - mae: 1.8757
Epoch 42/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.8362 - mae: 1.7908
Epoch 43/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 6.2725 - mae: 1.8872
Epoch 44/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.8477 - mae: 1.8701
Epoch 45/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.5362 - mae: 1.7230
Epoch 46/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 6.2573 - mae: 1.8399
Epoch 47/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.8049 - mae: 1.7763
Epoch 48/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.7743 - mae: 1.7914
Epoch 49/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.0485 - mae: 1.7178
Epoch 50/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.5995 - mae: 1.7030
Epoch 51/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.8756 - mae: 1.7738
Epoch 52/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 4.8289 - mae: 1.6777
Epoch 53/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.8117 - mae: 1.6290
Epoch 54/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.2025 - mae: 1.7635
Epoch 55/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.4743 - mae: 1.7728
Epoch 56/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.3838 - mae: 1.7171
Epoch 57/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.8033 - mae: 1.6608
Epoch 58/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.3749 - mae: 1.6105
Epoch 59/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.6322 - mae: 1.6517
Epoch 60/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.5509 - mae: 1.5971
Epoch 61/100
```

```
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.6773 - mae: 1.6010
Epoch 62/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.3954 - mae: 1.5994
Epoch 63/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.4361 - mae: 1.7930
Epoch 64/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.0977 - mae: 1.7404
Epoch 65/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.5783 - mae: 1.5902
Epoch 66/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.8521 - mae: 1.7237
Epoch 67/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.8779 - mae: 1.7235
Epoch 68/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.5348 - mae: 1.6140
Epoch 69/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 4.7998 - mae: 1.6934
Epoch 70/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.0253 - mae: 1.6833
Epoch 71/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.7800 - mae: 1.5010
Epoch 72/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.2671 - mae: 1.5452
Epoch 73/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.8306 - mae: 1.4788
Epoch 74/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.0283 - mae: 1.5410
Epoch 75/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.6946 - mae: 1.4680
Epoch 76/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.4699 - mae: 1.4643
Epoch 77/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.1583 - mae: 1.5345
Epoch 78/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.2960 - mae: 1.4199
Epoch 79/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.8190 - mae: 1.4887
Epoch 80/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.9262 - mae: 1.4995
Epoch 81/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.6127 - mae: 1.4592
Epoch 82/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.7006 - mae: 1.4414
Epoch 83/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.3318 - mae: 1.3775
Epoch 84/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 3.5521 - mae: 1.4353
Epoch 85/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.3651 - mae: 1.4300
Epoch 86/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 3.6556 - mae: 1.4559
Epoch 87/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.2425 - mae: 1.4052
Epoch 88/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.8450 - mae: 1.5218
Epoch 89/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.7119 - mae: 1.4949
Epoch 90/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.8289 - mae: 1.3033
Epoch 91/100
```

```
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.5812 - mae: 1.2433
Epoch 92/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.2835 - mae: 1.3468
Epoch 93/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.5620 - mae: 1.4550
Epoch 94/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.8241 - mae: 1.2703
Epoch 95/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.9526 - mae: 1.3182
Epoch 96/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.6215 - mae: 1.2318
Epoch 97/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.9691 - mae: 1.5211
Epoch 98/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.9109 - mae: 1.2978
Epoch 99/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.7802 - mae: 1.2513
Epoch 100/100
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.6324 - mae: 1.2657
```

```
Out[13]: <keras.src.callbacks.history.History at 0x16cb334da10>
```

# Part 4 - Making the predictions and evaluating the model

## Predicting the Test set results

```
In [14]: y_pred=ann.predict(X_test)
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step
```

```
In [15]: from sklearn.metrics import r2_score
         r2_score(y_test,y_pred)
```

```
Out[15]: 0.7992892662530684
```

```
In [ ]:
```

# Assignment 2

In [108…

```
from keras.datasets import imdb
from keras.preprocessing.sequence import pad_sequences
from keras.layers import Dense
from keras.models import Sequential
import numpy as np
```

In [109…

```
(x_train,y_train),(x_test,y_test)=imdb.load_data(num_words=1000)
```

In [110…

```
x_train, y_train = np.array(x_train), np.array(y_train)
```

In [111…

```
max_len=250
```

In [112…

```
x_train=pad_sequences(x_train,maxlen=max_len)
x_test=pad_sequences(x_test,maxlen=max_len)
```

In [113…

```
model=Sequential()
```

In [114…

```
model.add((Dense(units=100,activation='relu')))
```

In [115…

```
model.add((Dense(units=50,activation='relu')))
```

In [116…

```
# model.add((Dense(units=50,activation='relu')))
```

In [117…

```
model.add(Dense(units=1,activation='sigmoid'))
```

In [118…

```
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
```

In [119…

```
model.fit(x_train,y_train,epochs=30,batch_size=64)
```

```
Epoch 1/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.5034 - loss: 18.6443
Epoch 2/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.5186 - loss: 1.1738
Epoch 3/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.5218 - loss: 0.7723
Epoch 4/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.5329 - loss: 0.7365
Epoch 5/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.5301 - loss: 0.7327
Epoch 6/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.5347 - loss: 0.7282
Epoch 7/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.5359 - loss: 0.7298
Epoch 8/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.5483 - loss: 0.6916
Epoch 9/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.5565 - loss: 0.7049
Epoch 10/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.5803 - loss: 0.6753
Epoch 11/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.5865 - loss: 0.6689
Epoch 12/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.6154 - loss: 0.6448
Epoch 13/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.6274 - loss: 0.6279
Epoch 14/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.6563 - loss: 0.5957
Epoch 15/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.6797 - loss: 0.5732
Epoch 16/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.7063 - loss: 0.5389
Epoch 17/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.7208 - loss: 0.5240
Epoch 18/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.7506 - loss: 0.4859
Epoch 19/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.7649 - loss: 0.4642
Epoch 20/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.7861 - loss: 0.4308
Epoch 21/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.7950 - loss: 0.4158
Epoch 22/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8183 - loss: 0.3790
Epoch 23/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8315 - loss: 0.3568
Epoch 24/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8469 - loss: 0.3323
Epoch 25/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8623 - loss: 0.3051
Epoch 26/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8697 - loss: 0.2922
Epoch 27/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8810 - loss: 0.2718
Epoch 28/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8907 - loss: 0.2521
Epoch 29/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8972 - loss: 0.2422
Epoch 30/30
391/391 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.9033 - loss: 0.2225
```

```
Out[119…  <keras.src.callbacks.history.History at 0x27b9f214e10>
```

```
In [120…  loss,acc=model.evaluate(x_test,y_test,batch_size=128)
```

```
196/196 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.5088 - loss: 2.1961
```

```
In [ ]:
```

# Assignment 3

In [38]:
```
from tensorflow.keras.datasets import fashion_mnist
import tensorflow as tf
from tensorflow.keras.utils import to_categorical
```

In [2]:
```
(x_train,y_train),(x_test,y_test)=fashion_mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-dataset
s/train-labels-idx1-ubyte.gz
29515/29515 ━━━━━━━━━━━━━━━━━━━━ 0s 3us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-dataset
s/train-images-idx3-ubyte.gz
26421880/26421880 ━━━━━━━━━━━━━━━━━━━━ 13s 1us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-dataset
s/t10k-labels-idx1-ubyte.gz
5148/5148 ━━━━━━━━━━━━━━━━━━━━ 0s 1us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-dataset
s/t10k-images-idx3-ubyte.gz
4422102/4422102 ━━━━━━━━━━━━━━━━━━━━ 2s 0us/step
```

In [5]:
```
x_train=x_train/255.0
x_test=x_test/255.0
```

In [ ]:
```
x_train=x_train.reshape(-1,28,28,1)
x_test=x_test.reshape(-1,28,28,1)
```

In [39]:
```
# One-hot encode the labels
y_train = to_categorical(y_train,10)
y_test = to_categorical(y_test,10)
```

In [41]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropou

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
    MaxPooling2D(2, 2),

    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),

    Flatten(),
    Dropout(0.5),
    Dense(128, activation='relu'),
    Dense(10, activation='softmax')
])
```

```
C:\Users\acer\AppData\Roaming\Python\Python311\site-packages\keras\src\layers\con
volutional\base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim
` argument to a layer. When using Sequential models, prefer using an `Input(shap
e)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

In [43]:
```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accur
# Train
model.fit(x_train, y_train, epochs=10,batch_size=32,validation_data=(x_test,y_te
```

```
Epoch 1/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 19s 9ms/step - accuracy: 0.8910 - loss: 0.2891 - v
al_accuracy: 0.8981 - val_loss: 0.2781
Epoch 2/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 18s 9ms/step - accuracy: 0.9017 - loss: 0.2636 - v
al_accuracy: 0.8910 - val_loss: 0.2799
Epoch 3/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 17s 9ms/step - accuracy: 0.9058 - loss: 0.2491 - v
al_accuracy: 0.9002 - val_loss: 0.2591
Epoch 4/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 20s 10ms/step - accuracy: 0.9126 - loss: 0.2321 -
val_accuracy: 0.9128 - val_loss: 0.2381
Epoch 5/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 17s 9ms/step - accuracy: 0.9163 - loss: 0.2226 - v
al_accuracy: 0.9124 - val_loss: 0.2404
Epoch 6/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 17s 9ms/step - accuracy: 0.9201 - loss: 0.2106 - v
al_accuracy: 0.9099 - val_loss: 0.2392
Epoch 7/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 17s 9ms/step - accuracy: 0.9213 - loss: 0.2040 - v
al_accuracy: 0.9140 - val_loss: 0.2421
Epoch 8/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 17s 9ms/step - accuracy: 0.9233 - loss: 0.1984 - v
al_accuracy: 0.9119 - val_loss: 0.2370
Epoch 9/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 17s 9ms/step - accuracy: 0.9268 - loss: 0.1912 - v
al_accuracy: 0.9122 - val_loss: 0.2408
Epoch 10/10
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 18s 10ms/step - accuracy: 0.9308 - loss: 0.1795 -
val_accuracy: 0.9134 - val_loss: 0.2407
```

```
Out[43]: <keras.src.callbacks.history.History at 0x21c28b8ec90>
```

```
In [44]: test_loss, test_accuracy = model.evaluate(x_test, y_test)
         print(f"Test Accuracy: {test_accuracy:.4f}")
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - accuracy: 0.9144 - loss: 0.2440
Test Accuracy: 0.9134
```

```
In [ ]:
```