Import the libraries

In [1]:

```
#importing the libraries
import tensorflow as tf
from tensorflow import keras
import numpy as np
import pandas as pd
```

```
WARNING:tensorflow:From C:\Users\Teo Boon Kean\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\s
rc\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses
.sparse_softmax_cross_entropy instead.
```

Load the data and data pre-processing

In [2]:

```
#load the datasets
baseline_df = pd.read_excel('extracted_features_baseline.xlsx')
toolwear_df = pd.read_excel('extracted_features_toolwear.xlsx')
```

In [3]:

```
#labelling the datasets. 0 for baseline, 1 for toolwear. This will be the variable the model tries to predict
baseline_df["state"] = 0
toolwear_df["state"] = 1
```

In [4]:

```
#concantanate the datasets
combined_df = pd.concat([baseline_df, toolwear_df], axis=0)
print(combined_df.shape)
```

```
(840, 67)
```

In [5]:

```
#getting the y label
state = combined_df["state"].values
print(state.shape)
```

```
(840,)
```

In [6]:

```
#getting the features to train the model
features = combined_df.drop('state', axis=1).values
print(features.shape)
```

```
(840, 66)
```

In [7]:

```
#train test split
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(features, state, test_size=0.2, random_state=50)
```

In [8]:

```
#data scalling
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Constructing the stacked autoencoder and training it

In [9]:

```
#specify the number of condensed features for the 3 encoders. This will be the number of neurons in the hidden
condensed_e1 = 50
condensed_e2 = 30
condensed_e3 = 10
```

In [10]:

```
#constructing the model

#input layer which number of neurons equals the number of original features
l_in = keras.Input(features.shape[1])

#hidden layer of encoder 1
l_e1 = keras.layers.Dense(condensed_e1)(l_in)

#hidden layer of encoder 2
l_e2 = keras.layers.Dense(condensed_e2)(l_e1)

#hidden layer of encoder 3
l_e3 = keras.layers.Dense(condensed_e3)(l_e2)

#hidden layer of decoder 2
l_d2 = keras.layers.Dense(condensed_e2)(l_e3)

#hidden layer of decoder 1
l_d1 = keras.layers.Dense(condensed_e1)(l_d2)
```

```
#output layer which is the same as the input
l_out = keras.layers.Dense(features.shape[1])(l_d1)
```

```
WARNING:tensorflow:From C:\Users\Teo Boon Kean\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\s
rc\backend.py:1398: The name tf.executing_eagerly_outside_functions is deprecated. Please use tf.compat.v1.execu
ting_eagerly_outside_functions instead.
```

In [11]:

```
#defining the autoencode
autoencoder = keras.Model(l_in, l_out)
```

In [12]:

```
autoencoder.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 input_1 (InputLayer)        [(None, 66)]              0         
                                                                 
 dense (Dense)               (None, 50)                3350      
                                                                 
 dense_1 (Dense)             (None, 30)                1530      
                                                                 
 dense_2 (Dense)             (None, 10)                310       
                                                                 
 dense_3 (Dense)             (None, 30)                330       
                                                                 
 dense_4 (Dense)             (None, 50)                1550      
                                                                 
 dense_5 (Dense)             (None, 66)                3366      
                                                                 
=================================================================
Total params: 10436 (40.77 KB)
Trainable params: 10436 (40.77 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [13]:

```
#compile the model
autoencoder.compile(optimizer='adam', loss='mse')
#train the model
autoencoder.fit(X_train, X_train, epochs = 50, batch_size = 8, validation_split = 0.1)
```

```
WARNING:tensorflow:From C:\Users\Teo Boon Kean\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\s
rc\optimizers\__init__.py:309: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimize
r instead.

Epoch 1/50
WARNING:tensorflow:From C:\Users\Teo Boon Kean\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\s
rc\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated. Please use tf.compat.v1.ragged.Rag
gedTensorValue instead.

76/76 [==============================] - 6s 14ms/step - loss: 0.7300 - val_loss: 0.3674
Epoch 2/50
76/76 [==============================] - 0s 5ms/step - loss: 0.3338 - val_loss: 0.2611
Epoch 3/50
76/76 [==============================] - 0s 5ms/step - loss: 0.2582 - val_loss: 0.2160
Epoch 4/50
76/76 [==============================] - 0s 5ms/step - loss: 0.2133 - val_loss: 0.1840
Epoch 5/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1864 - val_loss: 0.1638
Epoch 6/50
76/76 [==============================] - 0s 6ms/step - loss: 0.1709 - val_loss: 0.1560
Epoch 7/50
76/76 [==============================] - 0s 6ms/step - loss: 0.1620 - val_loss: 0.1505
Epoch 8/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1559 - val_loss: 0.1461
Epoch 9/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1530 - val_loss: 0.1422
Epoch 10/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1512 - val_loss: 0.1431
Epoch 11/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1501 - val_loss: 0.1401
Epoch 12/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1493 - val_loss: 0.1420
Epoch 13/50
76/76 [==============================] - 0s 6ms/step - loss: 0.1480 - val_loss: 0.1392
Epoch 14/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1475 - val_loss: 0.1395
Epoch 15/50
76/76 [==============================] - 0s 6ms/step - loss: 0.1470 - val_loss: 0.1406
Epoch 16/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1475 - val_loss: 0.1387
Epoch 17/50
```

```
76/76 [==============================] - 0s 6ms/step - loss: 0.1470 - val_loss: 0.1401
Epoch 18/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1474 - val_loss: 0.1389
Epoch 19/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1465 - val_loss: 0.1393
Epoch 20/50
76/76 [==============================] - 0s 6ms/step - loss: 0.1461 - val_loss: 0.1363
Epoch 21/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1461 - val_loss: 0.1368
Epoch 22/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1461 - val_loss: 0.1391
Epoch 23/50
76/76 [==============================] - 0s 6ms/step - loss: 0.1465 - val_loss: 0.1379
Epoch 24/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1463 - val_loss: 0.1366
Epoch 25/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1460 - val_loss: 0.1389
Epoch 26/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1474 - val_loss: 0.1382
Epoch 27/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1461 - val_loss: 0.1381
Epoch 28/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1461 - val_loss: 0.1386
Epoch 29/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1469 - val_loss: 0.1384
Epoch 30/50
76/76 [==============================] - 0s 6ms/step - loss: 0.1466 - val_loss: 0.1372
Epoch 31/50
76/76 [==============================] - 0s 6ms/step - loss: 0.1457 - val_loss: 0.1367
Epoch 32/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1454 - val_loss: 0.1371
Epoch 33/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1459 - val_loss: 0.1385
Epoch 34/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1455 - val_loss: 0.1372
Epoch 35/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1452 - val_loss: 0.1371
Epoch 36/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1453 - val_loss: 0.1365
Epoch 37/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1453 - val_loss: 0.1366
Epoch 38/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1455 - val_loss: 0.1359
Epoch 39/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1454 - val_loss: 0.1387
Epoch 40/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1457 - val_loss: 0.1362
Epoch 41/50
76/76 [==============================] - 0s 6ms/step - loss: 0.1461 - val_loss: 0.1371
Epoch 42/50
76/76 [==============================] - 0s 6ms/step - loss: 0.1463 - val_loss: 0.1376
Epoch 43/50
76/76 [==============================] - 0s 6ms/step - loss: 0.1461 - val_loss: 0.1380
Epoch 44/50
76/76 [==============================] - 0s 6ms/step - loss: 0.1456 - val_loss: 0.1375
Epoch 45/50
76/76 [==============================] - 0s 6ms/step - loss: 0.1457 - val_loss: 0.1392
Epoch 46/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1475 - val_loss: 0.1381
Epoch 47/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1459 - val_loss: 0.1363
Epoch 48/50
76/76 [==============================] - 0s 6ms/step - loss: 0.1458 - val_loss: 0.1374
Epoch 49/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1453 - val_loss: 0.1377
Epoch 50/50
76/76 [==============================] - 0s 5ms/step - loss: 0.1452 - val_loss: 0.1389
```

Out[13]:

```
<keras.src.callbacks.History at 0x20630cb3ed0>
```

Model Evaluation

In [14]:

```python
from sklearn.metrics import mean_absolute_error

#Mean square error of the model
pred = autoencoder.predict(X_test)
print(mean_absolute_error(X_test,pred))
```

```
6/6 [==============================] - 0s 4ms/step
0.27588827367051055
```