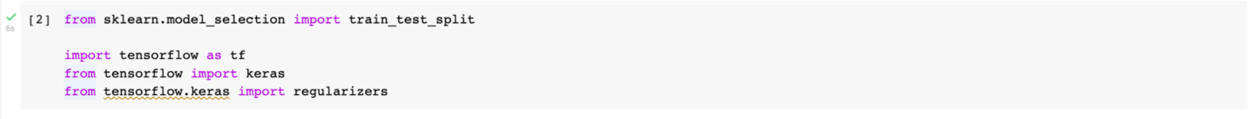
Machine Learning
Final Project

Prince David Sanam
WSU ID: M339J323

Creating the Deep learning models that make up a committee and obtain the accuracies of each model and the accuracy of the committee which is the average of the three deep learning models.

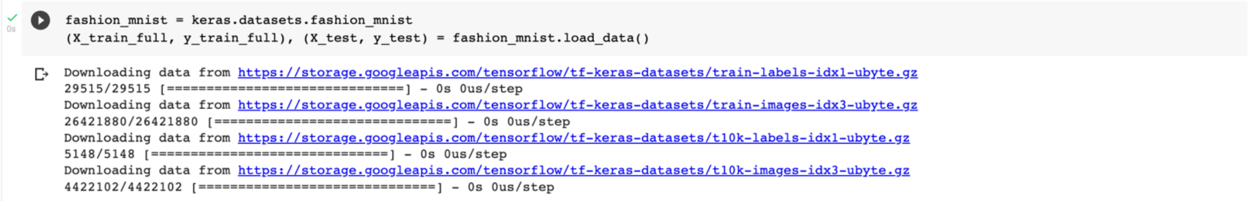
Step 1:
Import the necessary libraries which are to be used.



```
[2] from sklearn.model_selection import train_test_split

    import tensorflow as tf
    from tensorflow import keras
    from tensorflow.keras import regularizers
```

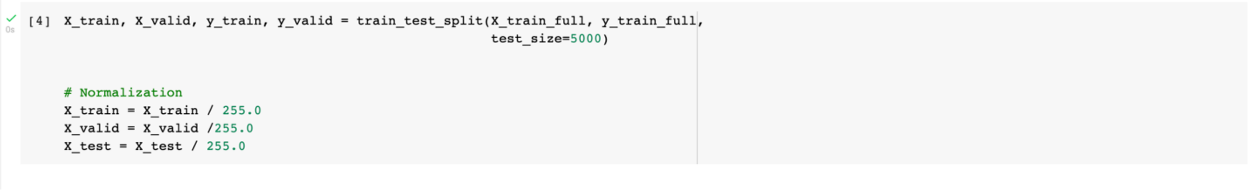
Step 2:
Importing the dataset,



```
fashion_mnist = keras.datasets.fashion_mnist
(X_train_full, y_train_full), (X_test, y_test) = fashion_mnist.load_data()

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
29515/29515 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 [==============================] - 0s 0us/step
```

Step 3:
Split the data set into train and validation sets,

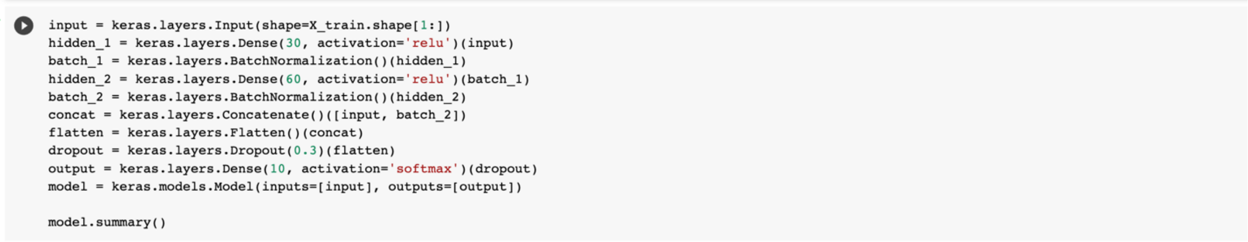


```
[4] X_train, X_valid, y_train, y_valid = train_test_split(X_train_full, y_train_full,
                                                          test_size=5000)


    # Normalization
    X_train = X_train / 255.0
    X_valid = X_valid /255.0
    X_test = X_test / 255.0
```

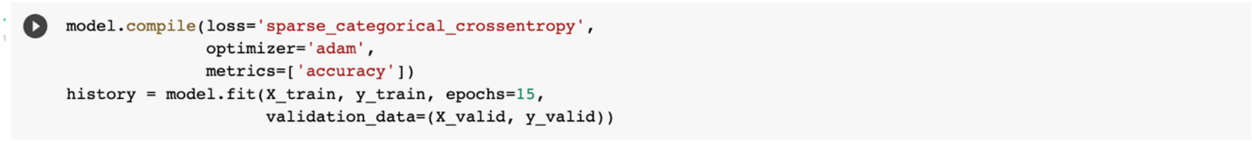
Now as asked, we must create 3 individual models which makes a committee,

Step 4:
Creating model_1,



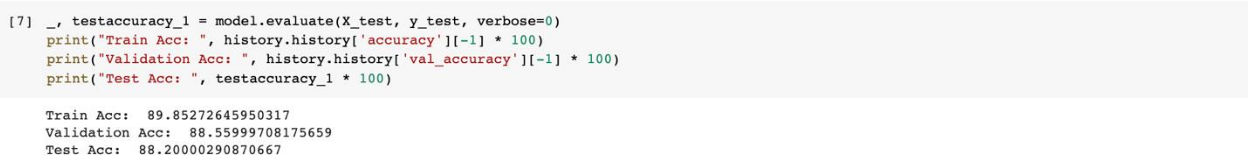
```
input = keras.layers.Input(shape=X_train.shape[1:])
hidden_1 = keras.layers.Dense(30, activation='relu')(input)
batch_1 = keras.layers.BatchNormalization()(hidden_1)
hidden_2 = keras.layers.Dense(60, activation='relu')(batch_1)
batch_2 = keras.layers.BatchNormalization()(hidden_2)
concat = keras.layers.Concatenate()([input, batch_2])
flatten = keras.layers.Flatten()(concat)
dropout = keras.layers.Dropout(0.3)(flatten)
output = keras.layers.Dense(10, activation='softmax')(dropout)
model = keras.models.Model(inputs=[input], outputs=[output])

model.summary()
```



```
model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
history = model.fit(X_train, y_train, epochs=15,
                    validation_data=(X_valid, y_valid))
```

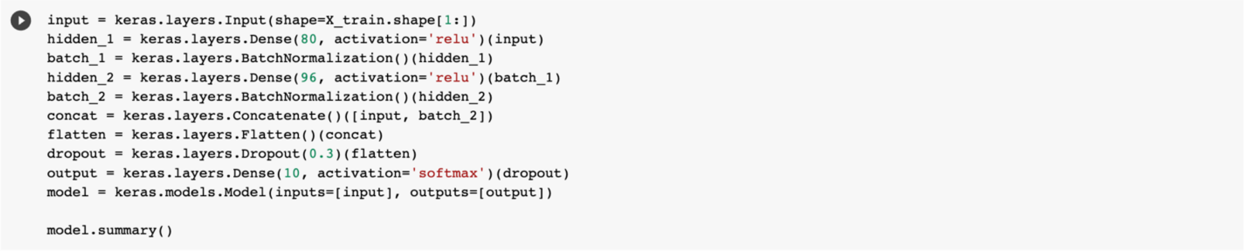
Obtain the accuracy of the model_1,



```
[7] _, testaccuracy_1 = model.evaluate(X_test, y_test, verbose=0)
    print("Train Acc: ", history.history['accuracy'][-1] * 100)
    print("Validation Acc: ", history.history['val_accuracy'][-1] * 100)
    print("Test Acc: ", testaccuracy_1 * 100)
```

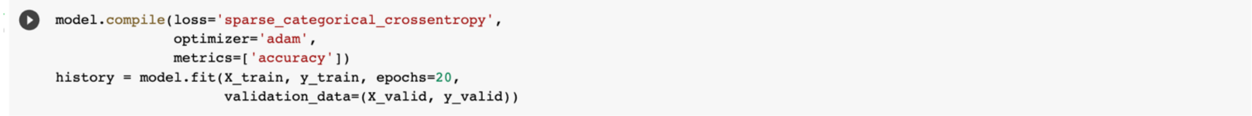
```
    Train Acc:  89.85272645950317
    Validation Acc:  88.55999708175659
    Test Acc:  88.20000290870667
```

Step 5:
Creating model_2:



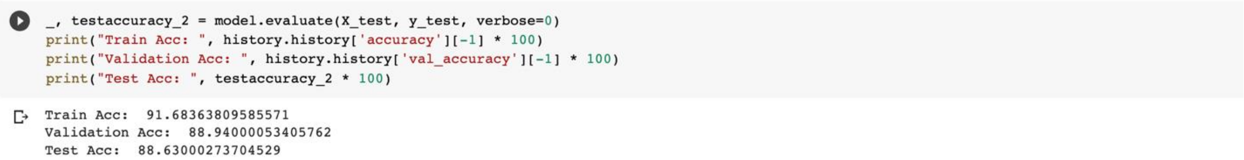
```
input = keras.layers.Input(shape=X_train.shape[1:])
hidden_1 = keras.layers.Dense(80, activation='relu')(input)
batch_1 = keras.layers.BatchNormalization()(hidden_1)
hidden_2 = keras.layers.Dense(96, activation='relu')(batch_1)
batch_2 = keras.layers.BatchNormalization()(hidden_2)
concat = keras.layers.Concatenate()([input, batch_2])
flatten = keras.layers.Flatten()(concat)
dropout = keras.layers.Dropout(0.3)(flatten)
output = keras.layers.Dense(10, activation='softmax')(dropout)
model = keras.models.Model(inputs=[input], outputs=[output])

model.summary()
```



```
model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
history = model.fit(X_train, y_train, epochs=20,
                    validation_data=(X_valid, y_valid))
```

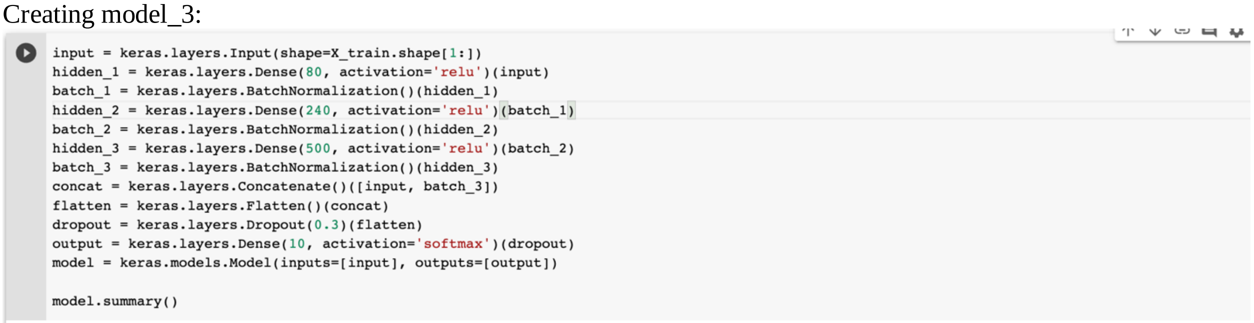
Obtain the accuracy of the model,



```
_, testaccuracy_2 = model.evaluate(X_test, y_test, verbose=0)
print("Train Acc: ", history.history['accuracy'][-1] * 100)
print("Validation Acc: ", history.history['val_accuracy'][-1] * 100)
print("Test Acc: ", testaccuracy_2 * 100)
```

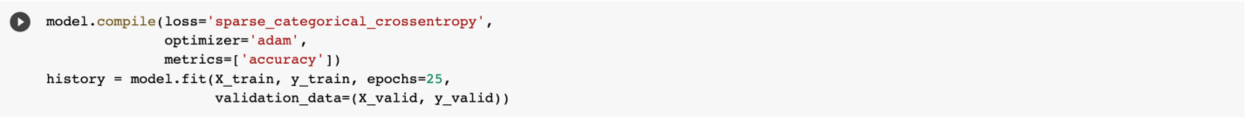
```
Train Acc:  91.68363809585571
Validation Acc:  88.94000053405762
Test Acc:  88.63000273704529
```

Step 6:
Creating model_3:



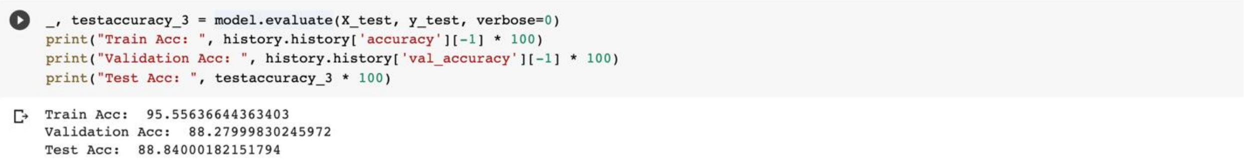
```
input = keras.layers.Input(shape=X_train.shape[1:])
hidden_1 = keras.layers.Dense(80, activation='relu')(input)
batch_1 = keras.layers.BatchNormalization()(hidden_1)
hidden_2 = keras.layers.Dense(240, activation='relu')(batch_1)
batch_2 = keras.layers.BatchNormalization()(hidden_2)
hidden_3 = keras.layers.Dense(500, activation='relu')(batch_2)
batch_3 = keras.layers.BatchNormalization()(hidden_3)
concat = keras.layers.Concatenate()([input, batch_3])
flatten = keras.layers.Flatten()(concat)
dropout = keras.layers.Dropout(0.3)(flatten)
output = keras.layers.Dense(10, activation='softmax')(dropout)
model = keras.models.Model(inputs=[input], outputs=[output])

model.summary()
```



```
model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
history = model.fit(X_train, y_train, epochs=25,
                    validation_data=(X_valid, y_valid))
```

Obtain the accuracy of the model,

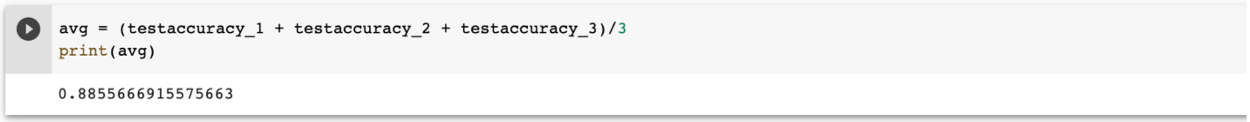


```
_, testaccuracy_3 = model.evaluate(X_test, y_test, verbose=0)
print("Train Acc: ", history.history['accuracy'][-1] * 100)
print("Validation Acc: ", history.history['val_accuracy'][-1] * 100)
print("Test Acc: ", testaccuracy_3 * 100)
```

```
Train Acc:  95.55636644363403
Validation Acc:  88.27999830245972
Test Acc:  88.84000182151794
```

Now we have created 3 models and obtained the accuracies of each model. We must get the accuracy of the committee and interpret the values.

Step 7:
Obtain the accuracy of the committee (Average accuracy),



```
avg = (testaccuracy_1 + testaccuracy_2 + testaccuracy_3)/3
print(avg)
```

```
0.8855666915575663
```

Analysis:

We have created a committee of three deep learning models using the Fashion MNIST dataset. We have obtained the accuracies of each deep learning models names model1, model2 and model3 respectively. The train accuracy, validation accuracy and test accuracy for each deep learning model is as follows,

Model1:
Train Acc: 89.85272645950317
Validation Acc: 88.55999708175659
Test Acc: 88.20000290870667

Model2:
Train Acc: 91.68363809585571
Validation Acc: 88.94000053405762
Test Acc: 88.63000273704529

Model3:
Train Acc: 95.55636644363403
Validation Acc: 88.27999830245972
Test Acc: 88.84000182151794

These three models make up the committee, so the total accuracy of the committee will be the average of the individual accuracies of the deep learning models. Now, the accuracy of the committee is as follows,

Accuracy = (testaccuracy_1 + testaccuracy_2 + testaccuracy_3)/3
= 0.8855666915575663

From the above values we can understand that the committee is 88.5% accurate