Report Assignment 3 ML

1) *Find the trainign loss and accruacy, validation loss and accuracy and test loss and accuracy*

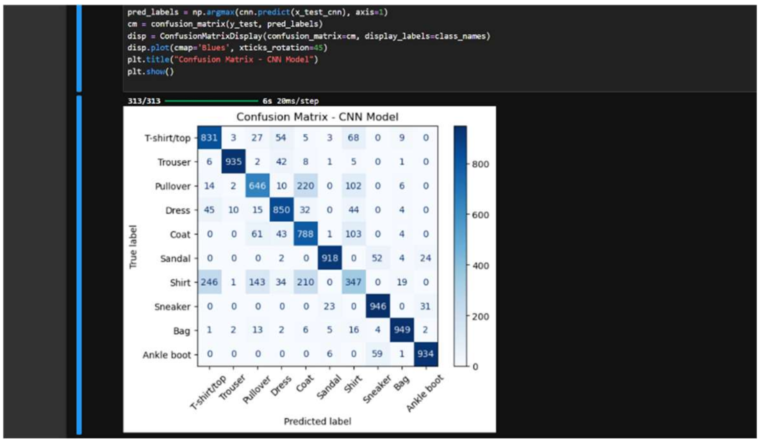
From the CNN training results, the recorded values were:

- **Training Accuracy:** 0.9198
- **Training Loss:** 0.2141
- **Validation Accuracy:** 0.9178
- **Validation Loss:** 0.2353
- **Test Accuracy:** 0.910
- **Test Loss:** 0.230

These results show that the model trained well and generalized effectively.
The validation and test accuracy are close to the training accuracy, which means the model did not overfit and performed consistently on unseen data.

2) *Plot the confusion matrix:*

The confusion matrix shows how well the model classified each clothing item from the Fashion MNIST dataset.
Most predictions are correct (visible as darker blue diagonal values). The model performed very well on categories such as T-shirt/top**,** Trouser**,** Sneaker, and Bag**.**
However, it sometimes confused visually similar items like Shirt and Coat, or Sandal and Sneaker, which is normal since these items can look alike in grayscale images.



```
pred_labels = np.argmax(cnn.predict(x_test_cnn), axis=1)
cm = confusion_matrix(y_test, pred_labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=class_names)
disp.plot(cmap='Blues', xticks_rotation=45)
plt.title("Confusion Matrix - CNN Model")
plt.show()
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 6s 20ms/step
```

Confusion Matrix - CNN Model

True label	T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
T-shirt/top	831	3	27	54	5	3	68	0	9	0
Trouser	6	935	2	42	8	1	5	0	1	0
Pullover	14	2	646	10	220	0	102	0	6	0
Dress	45	10	15	850	32	0	44	0	4	0
Coat	0	0	61	43	788	1	103	0	4	0
Sandal	0	0	0	2	0	918	0	52	4	24
Shirt	246	1	143	34	210	0	347	0	19	0
Sneaker	0	0	0	0	0	23	0	946	0	31
Bag	1	2	13	2	6	5	16	4	949	2
Ankle boot	0	0	0	0	0	6	0	59	1	934

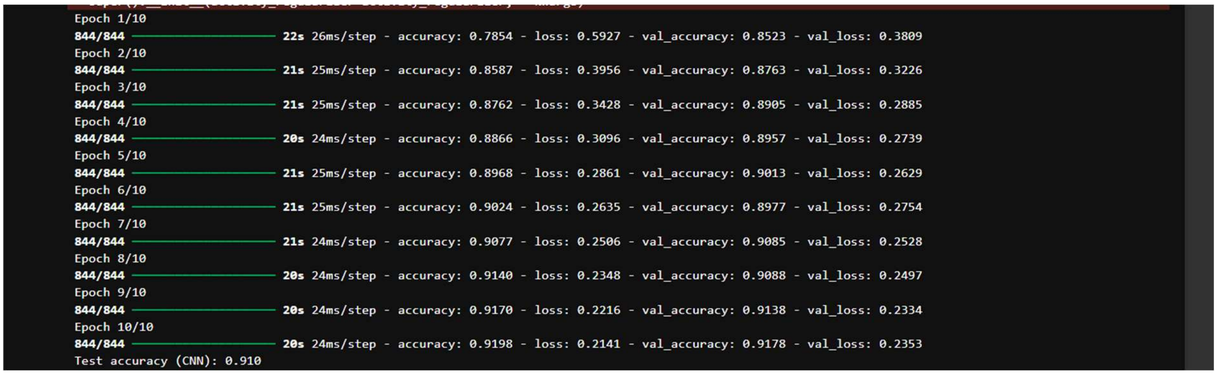
3) Hyperparameters Used and why:

For this task, I built a Convolutional Neural Network (CNN) model to classify clothing items from the Fashion MNIST dataset.

The CNN had two convolution layers, each followed by max pooling, then one dense layer, and finally an output layer.

- The first convolution layer used 32 filters, and the second used 64 filters, both with a 3×3 kernel.
  This helped the model detect small and detailed features in the images.
- Each convolution layer was followed by 2×2 max pooling to reduce the image size and prevent overfitting.
- After flattening the data, I added a dense layer with 128 neurons to combine learned features.
- A dropout layer (0.5) was included to prevent overfitting by randomly turning off neurons during training.
- The ReLU activation function was used in the hidden layers for faster and more efficient learning, while Softmax was used in the output layer to generate class probabilities for the 10 clothing categories.
- The model was trained using the Adam optimizer (for adaptive learning rates) and categorical crossentropy as the loss function, which is standard for multi-class classification.
- It was trained for 10 epochs with a batch size of 64 and a validation split of 0.1.

These hyperparameters worked well, resulting in a stable and high accuracy of around 91% on the test data.



```
Epoch 1/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 22s 26ms/step - accuracy: 0.7854 - loss: 0.5927 - val_accuracy: 0.8523 - val_loss: 0.3809
Epoch 2/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 21s 25ms/step - accuracy: 0.8587 - loss: 0.3956 - val_accuracy: 0.8763 - val_loss: 0.3226
Epoch 3/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 21s 25ms/step - accuracy: 0.8762 - loss: 0.3428 - val_accuracy: 0.8905 - val_loss: 0.2885
Epoch 4/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 20s 24ms/step - accuracy: 0.8866 - loss: 0.3096 - val_accuracy: 0.8957 - val_loss: 0.2739
Epoch 5/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 21s 25ms/step - accuracy: 0.8968 - loss: 0.2861 - val_accuracy: 0.9013 - val_loss: 0.2629
Epoch 6/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 21s 25ms/step - accuracy: 0.9024 - loss: 0.2635 - val_accuracy: 0.8977 - val_loss: 0.2754
Epoch 7/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 21s 24ms/step - accuracy: 0.9077 - loss: 0.2506 - val_accuracy: 0.9085 - val_loss: 0.2528
Epoch 8/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 20s 24ms/step - accuracy: 0.9140 - loss: 0.2348 - val_accuracy: 0.9088 - val_loss: 0.2497
Epoch 9/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 20s 24ms/step - accuracy: 0.9170 - loss: 0.2216 - val_accuracy: 0.9138 - val_loss: 0.2334
Epoch 10/10
844/844 ━━━━━━━━━━━━━━━━━━━━ 20s 24ms/step - accuracy: 0.9198 - loss: 0.2141 - val_accuracy: 0.9178 - val_loss: 0.2353
Test accuracy (CNN): 0.910
```

Collaboration and contribution:

This assignment was completed individually.
No group collaboration was involved.

The work was divided into the following steps:

- Researching the Fashion MNIST dataset — understanding what it contains and where it comes from.
- Downloading the dataset using TensorFlow and displaying a sample image to confirm it loaded correctly.
- Building and training the neural network using a CNN model.
- Visualizing and analyzing the results using plots and the confusion matrix.