**REPORT FOR ARTIFICIAL INTELLEGENCE**

**NO 1.**

**Step 1:**

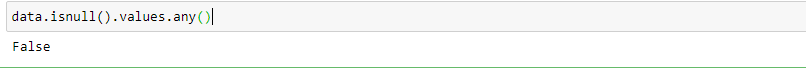
Libraries used:



Dataset used was from:

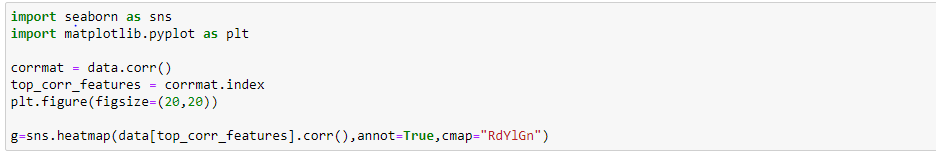
C:\Users\Lau_Lecture\Pictures\2.PNG

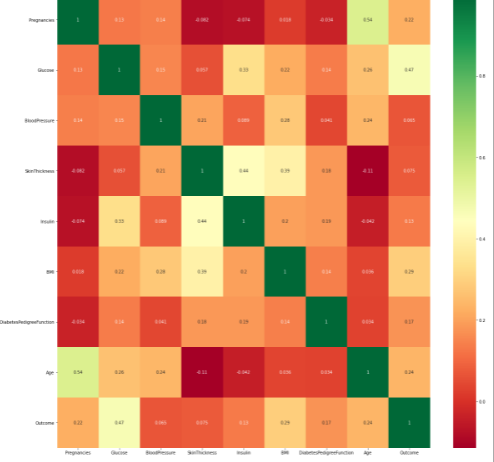
Checking whether dataset is empty:



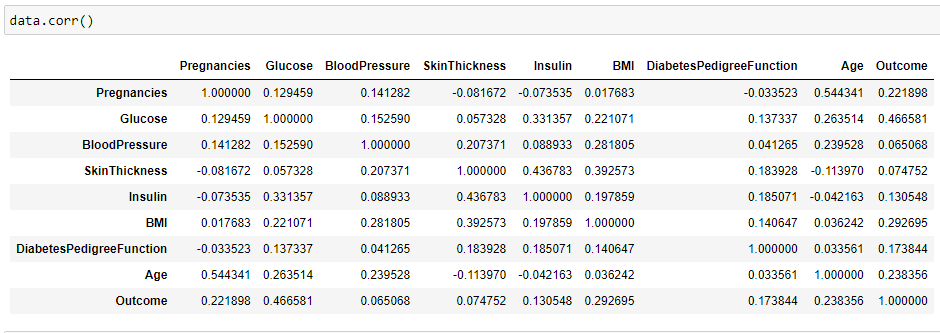
The above diagram shows that the dataset is available.

We imported these libraries for getting correlation features of dataset, and then plotted them by heap map.



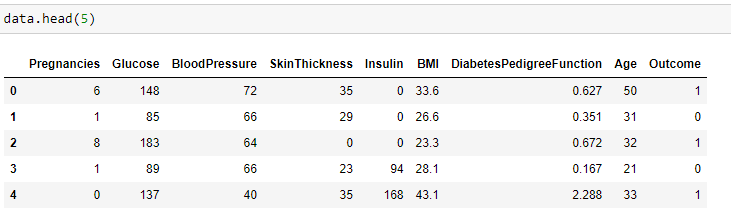


The figure below shows correlations of dataset features.

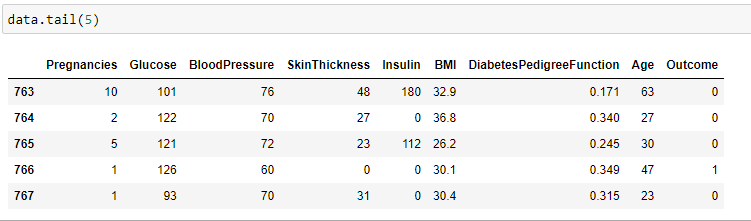


In this step, there are outputs of records from the head and tail with two figures:

**From head:**



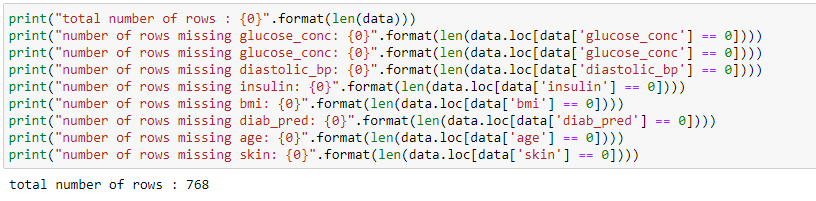
**From the tail:**



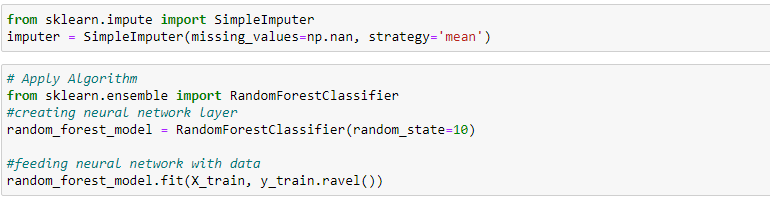
The below diagram allows user to enter number of N records.



The below diagram shows the number of rows available:



We created neural network that we build with our dataset. Figure below shows that:



**NO 2.**

**Step 1:**

The libraries that were imported are tensorflow, numpy and matplotlib.



The dataset used was from fashion mnist which contains 70 000 grayscale images with 10 categories and with low resolution (28 by 28 pixels). Here, 60 000 images are used to train the network and 10 000 images to evaluate how accurately the network learned to classify clothes. There were 10 classes of clothes and 10 labels used, this where numpy used where it stores images in the form of array and then maps the labels with corresponded classes.

**Step 2:**

Explore the data whereby the format of dataset was explored before training the model. The dataset explored contains 60 000 images used to train the network. Each image was represented in 28 by 28 pixels. In addition, 10 000 images for model accuracy was explored and represented each image in 28 by 28 pixels.

**Step 3:**

**Preprocess the data**

The data was preprocessed before training the model.

**Step 4:**

**Build the model**

We built neural network, configure the layers which were input layer, hidden layer and output layer, and compile the model. The first layer in the neural network, transform the format of images from two-dimensional array into one-dimensional array (28 x 28 = 784 pixels). This layer reformats the data

After the pixels are flattened, the network consists two layers.

**Train the model**

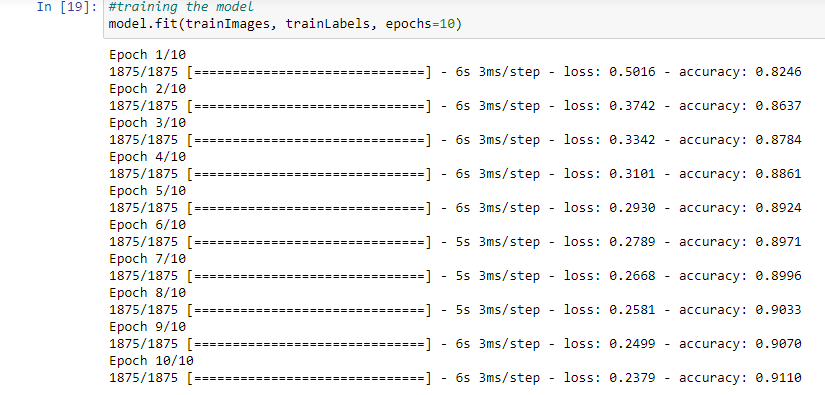
Training the neural network model requires the following steps:

1. Feed the training data to the model. In this example, the training data is in the trainImages and trainLabels arrays.
2. The model learns to associate images and labels.
3. The model makes predictions about a test set-in this example, the testImages array.
4. Verify that the predictions match the labels from the testLabels array.

**Feed the model**

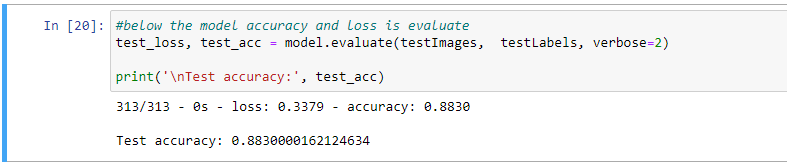
To start training, we called the **model.fit** method.

As the model trains, the loss and accuracy metrics are displayed. This model reaches an accuracy of about 0.91 (or 91%) on the training data.

****

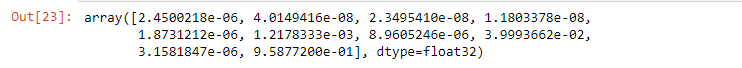
**Evaluate accuracy**

Here we compared how the model performs on the test dataset. Which gives 88% test accuracy.



**Make predictions**

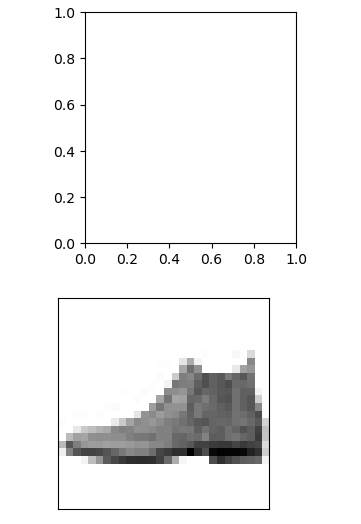
With the model trained, we were able to use it to make predictions about some images. The model's linear outputs, logist. Attach a softmax layer to convert the logits to probabilities, which are easier to interpret.

****

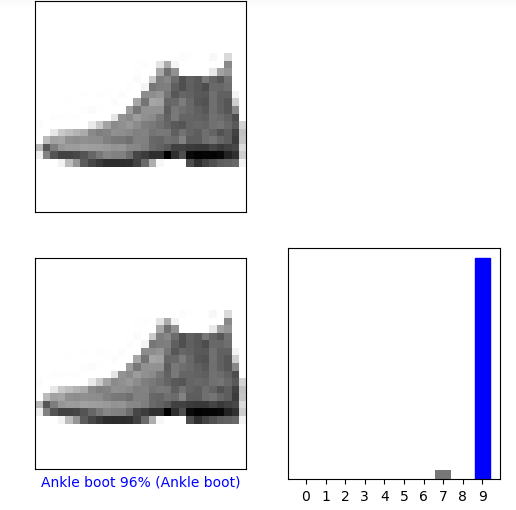
A prediction is an array of 10 numbers. They represent the model's confidence that the image corresponds to each of the 10 different articles of clothing. Where we were able to see which label has the highest confidence value.

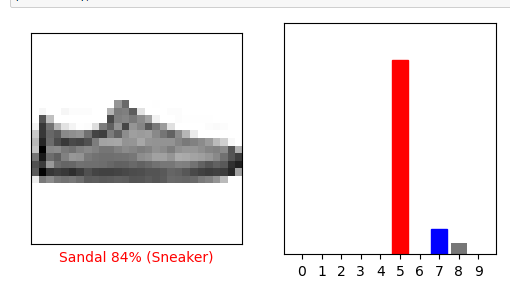
**C:\Users\Lau_Lecture\Pictures\Capture5.PNG**

The label was able to be predicted as **9** and graphed with the help of matplotlib library.

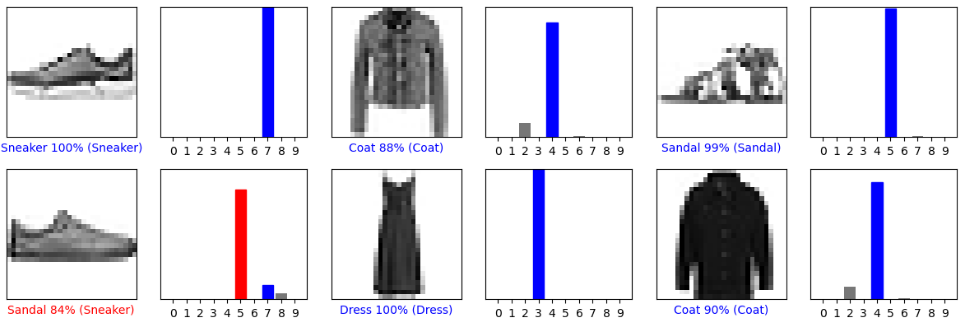
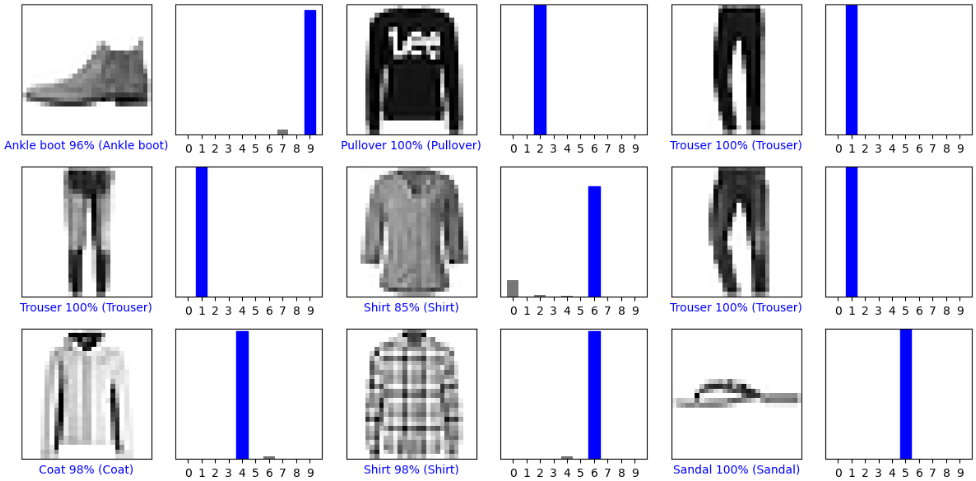
****

**Verify predictions**

With the model trained, we were able to use it to make predictions about some images. ****

****

Below are predicted labels on several images with their accuracy percentage.

****