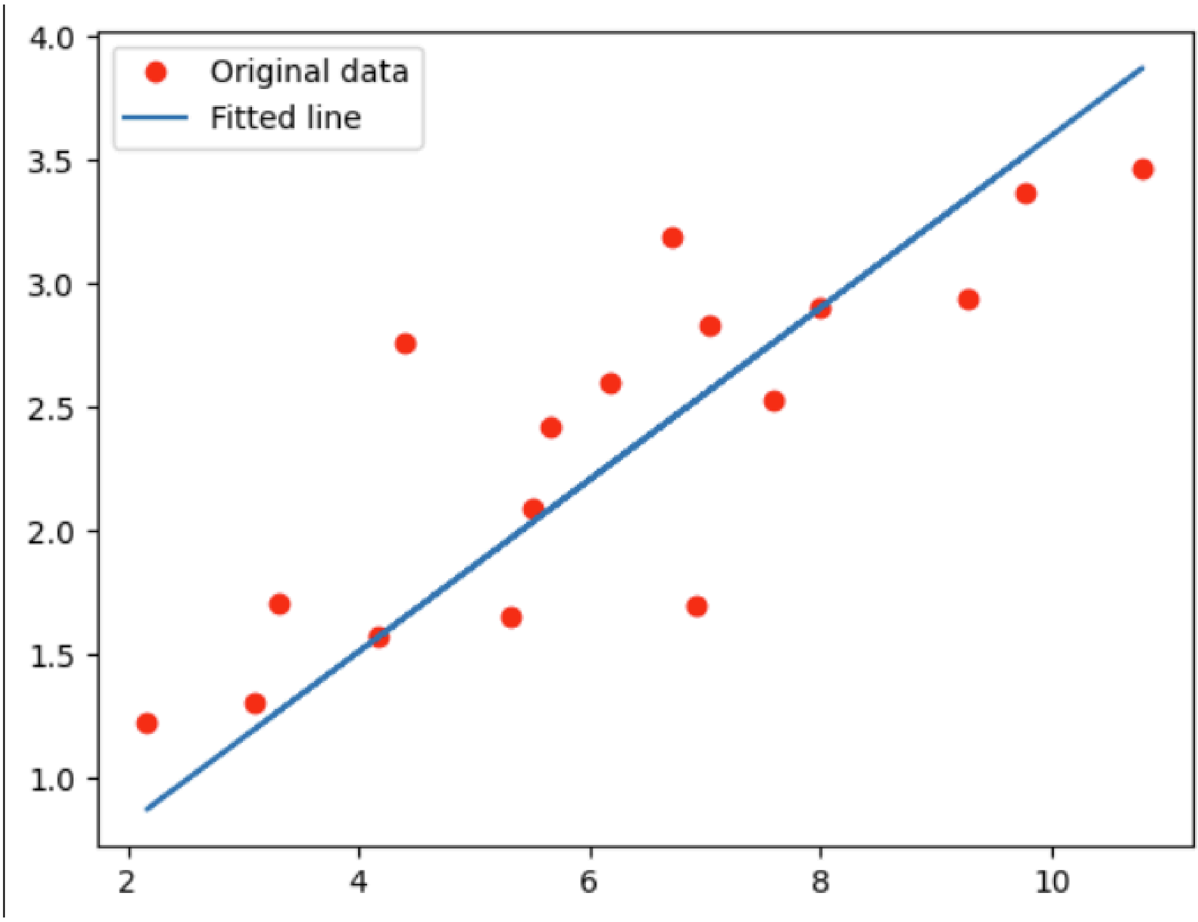
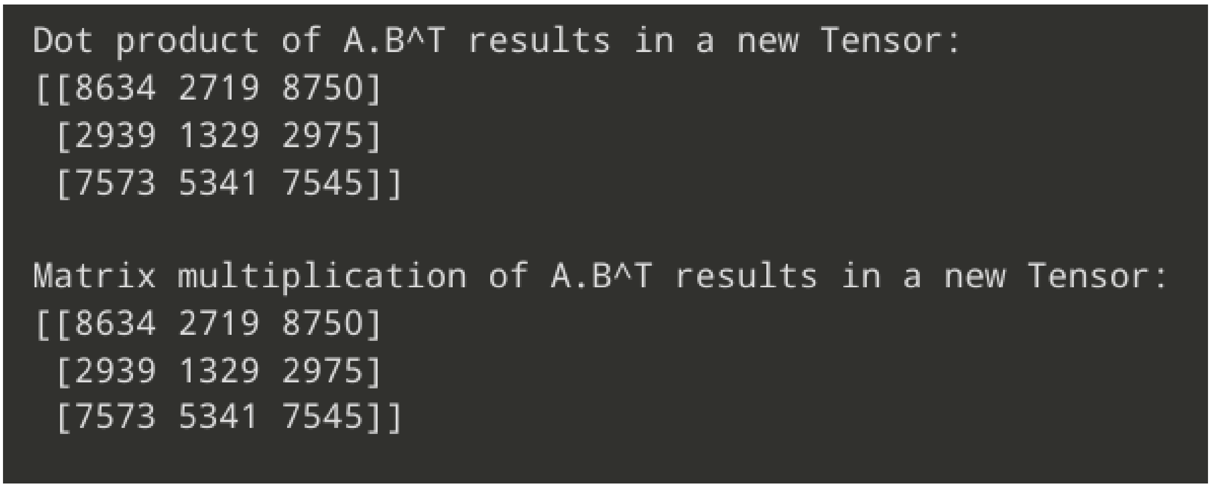
LAB logbook

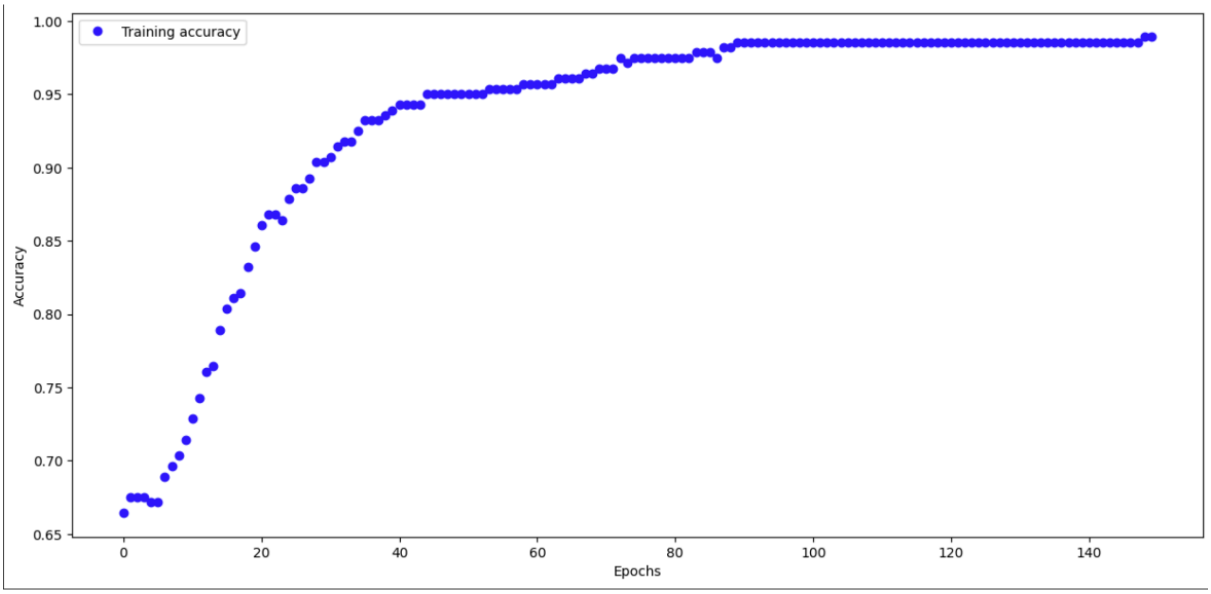
Week - 1

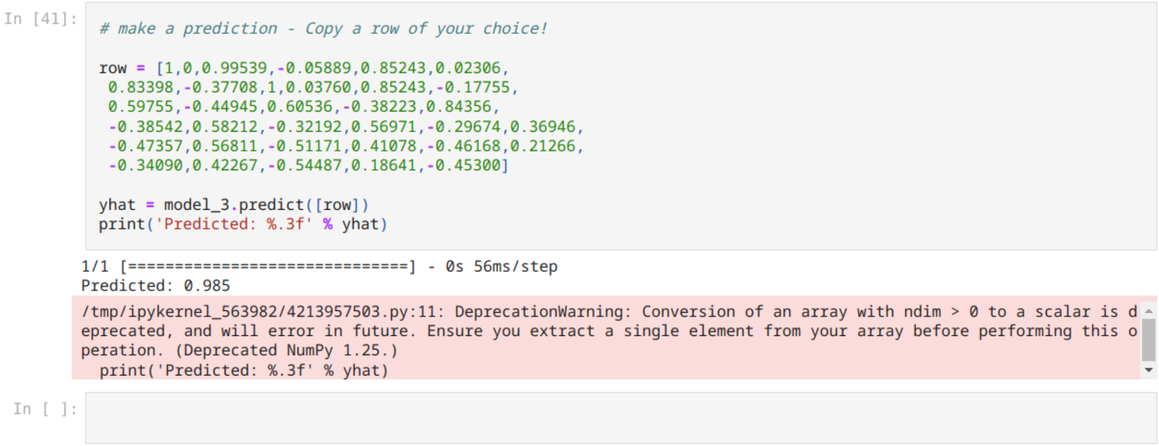




```
Dot product of A.B^T results in a new Tensor:
[[8634 2719 8750]
 [2939 1329 2975]
 [7573 5341 7545]]

Matrix multiplication of A.B^T results in a new Tensor:
[[8634 2719 8750]
 [2939 1329 2975]
 [7573 5341 7545]]
```

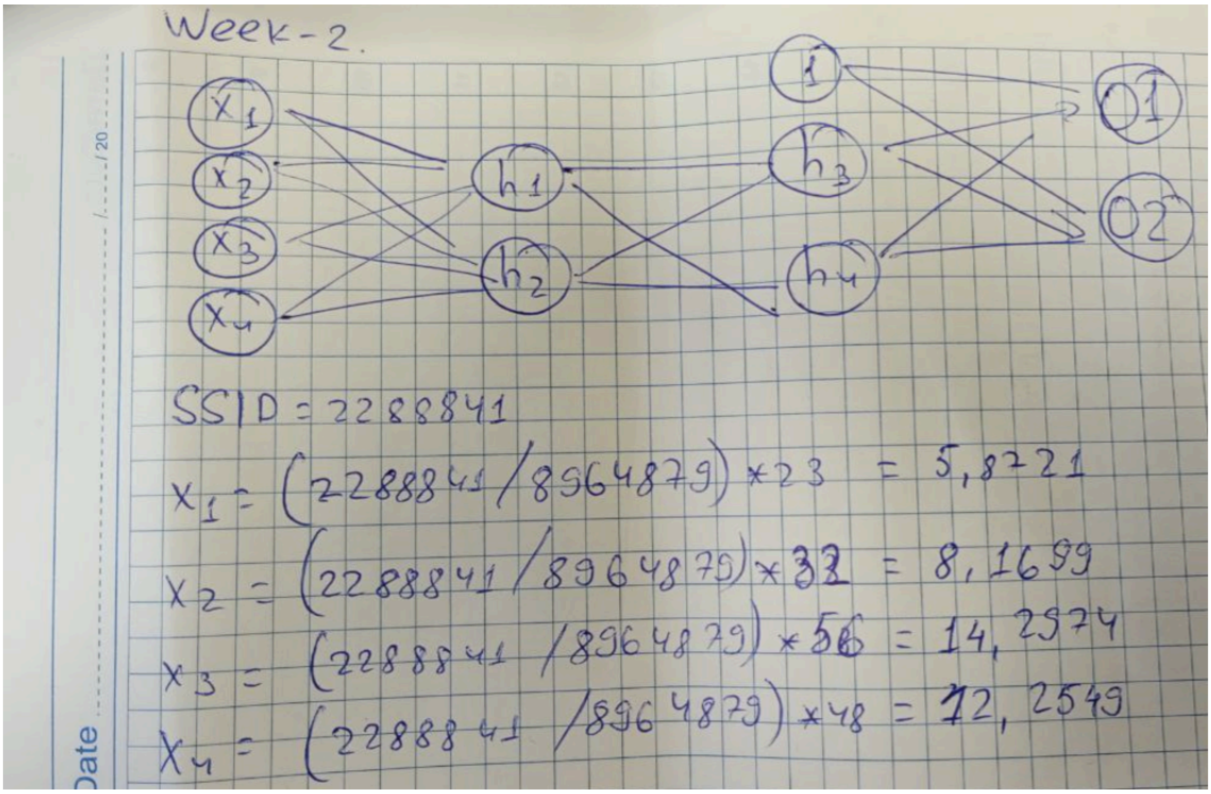
Week-2





```
In [41]:  # make a prediction - Copy a row of your choice!

          row = [1,0,0.99539,-0.05889,0.85243,0.02306,
           0.83398,-0.37708,1,0.03760,0.85243,-0.17755,
           0.59755,-0.44945,0.60536,-0.38223,0.84356,
           -0.38542,0.58212,-0.32192,0.56971,-0.29674,0.36946,
           -0.47357,0.56811,-0.51171,0.41078,-0.46168,0.21266,
           -0.34090,0.42267,-0.54487,0.18641,-0.45300]

          yhat = model_3.predict([row])
          print('Predicted: %.3f' % yhat)

         1/1 [==============================] - 0s 56ms/step
         Predicted: 0.985
         /tmp/ipykernel_563982/4213957503.py:11: DeprecationWarning: Conversion of an array with ndim > 0 to a scalar is d
         eprecated, and will error in future. Ensure you extract a single element from your array before performing this o
         peration. (Deprecated NumPy 1.25.)
           print('Predicted: %.3f' % yhat)

In [ ]:
```

Task 1: accuracy = 0.998


Date
/ 20
x_1
x_2
x_3
x_4
h_1
h_2
1
h_3
h_4
O1
O2

Week-2.

SSID = 2288841

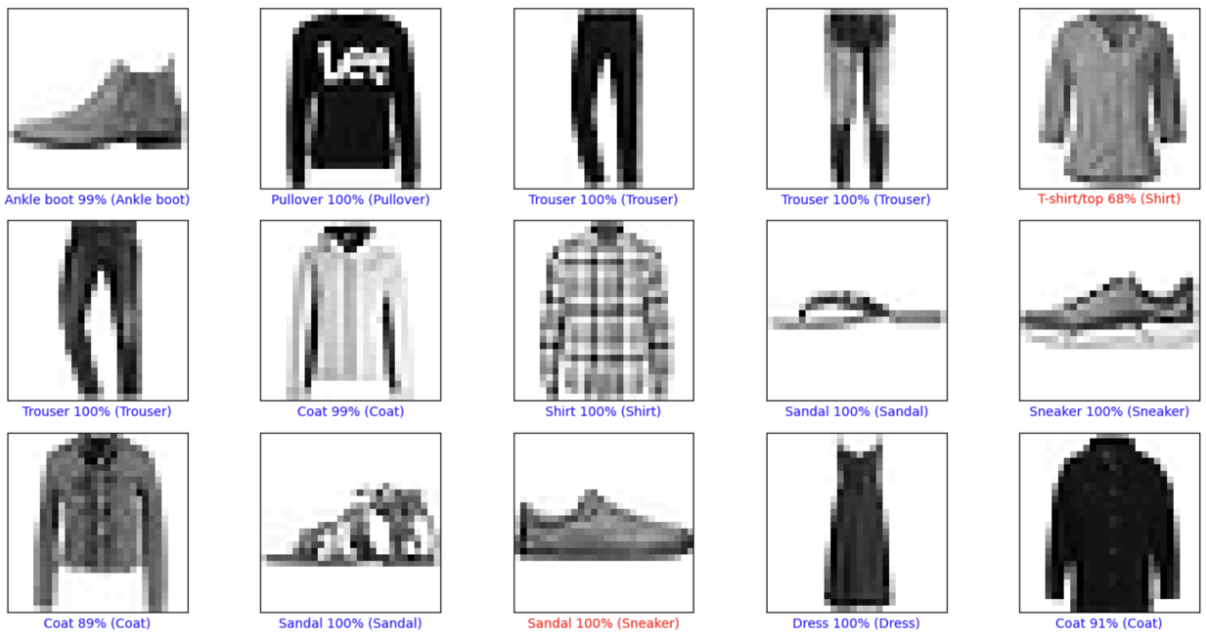
$x_1 = (2288841 / 8964879) * 23 = 5,8721$

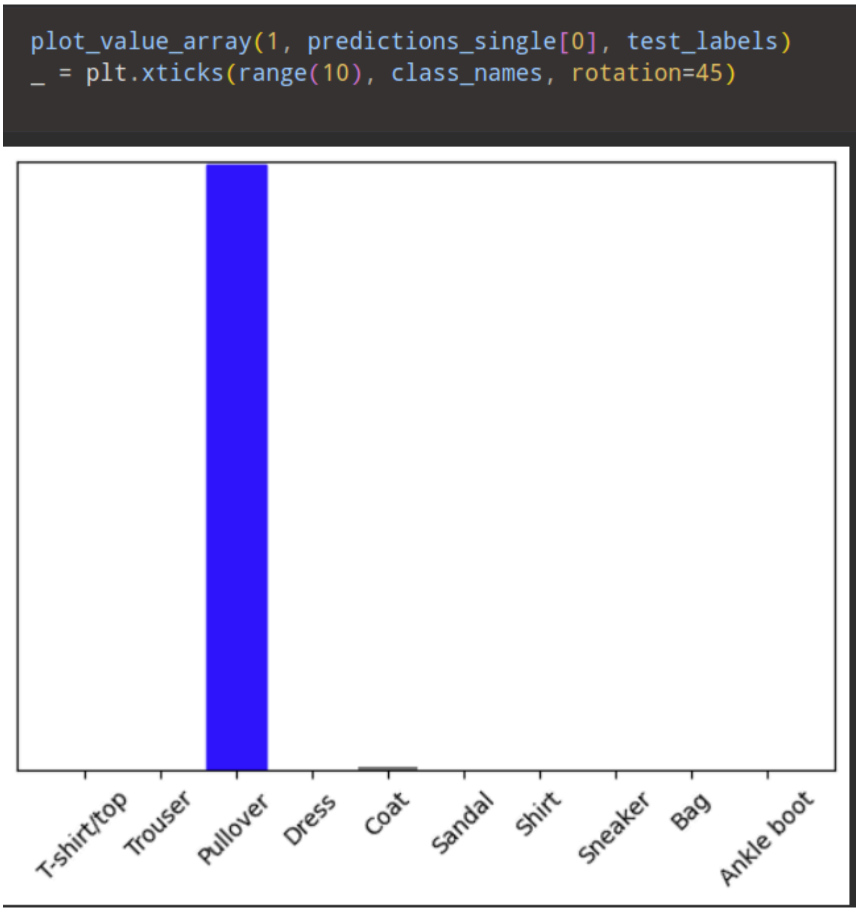
$x_2 = (2288841 / 8964879) * 32 = 8,1699$

$x_3 = (2288841 / 8964879) * 56 = 14,2974$

$x_4 = (2288841 / 8964879) * 48 = 12,2549$

week-3

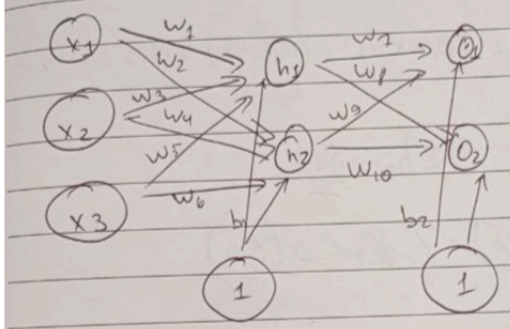




```
plot_value_array(1, predictions_single[0], test_labels)
_ = plt.xticks(range(10), class_names, rotation=45)
```

week-3

Backpropogation by hand



$w_1 = 0,1$ $w_3 = 0,3$ $w_5 = 0,5$ $w_7 = 0,7$

$w_2 = 0,2$ $w_4 = 0,4$ $w_6 = 0,6$ $w_8 = 0,8$

$w_9 = 0,5$ $w_{10} = 0,1$ $b_1 = 0,5$ $b_2 = 0,5$

$x_1 = 1,$ $x_2 = 4,$ $x_3 = 5$ $t_1 = 0.1$

$t_2 = 0.05$

Forward Propogate

Memo No.
Date / /

I → Hidden Layer

$(0,1 * 1) + (0,3 \cdot 4) + (0,5 \cdot 5) + 0,5 = zh_1$

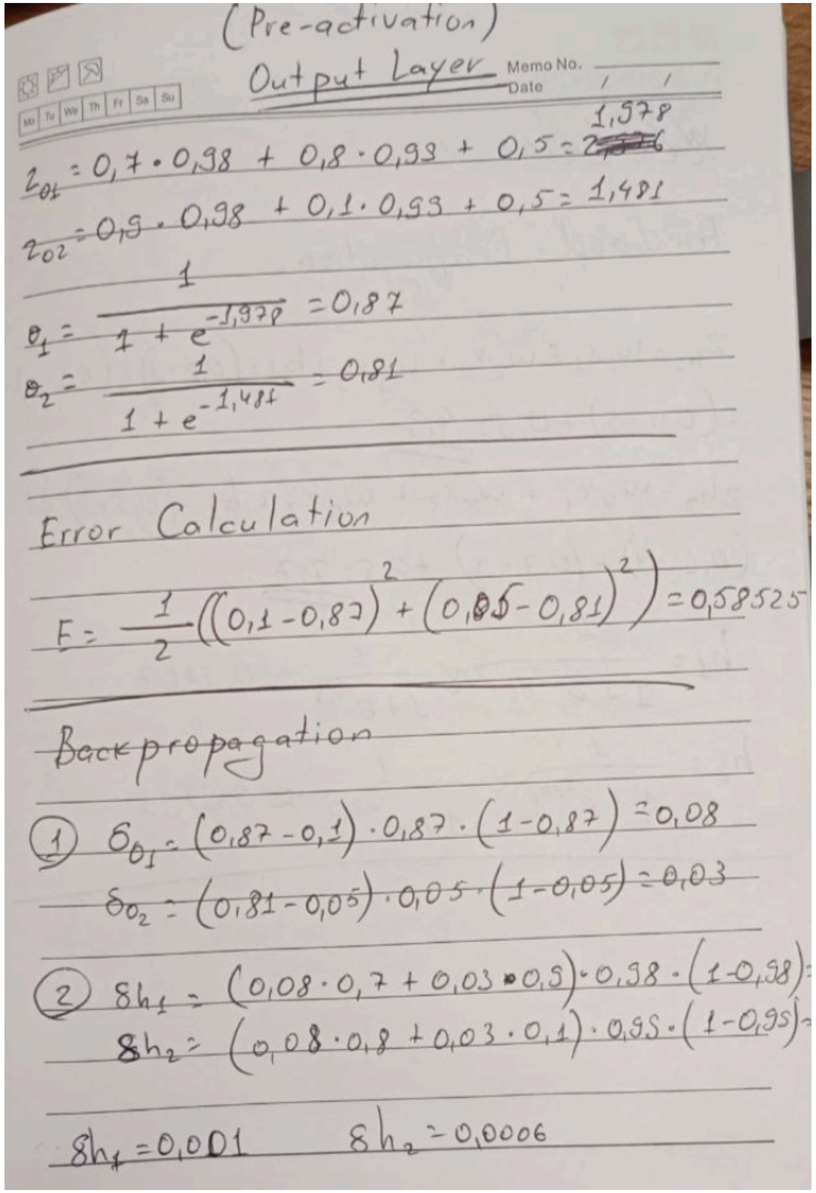
$(0,2 * 1) + (0,4 * 4) + (0,6 \cdot 5) + 0,5 = zh_2$

$\underline{\underline{z_{h_1} = 4,3}}$ $\underline{\underline{zh_2 = 5,3}}$

~~$h_1 = \sigma(4,3)$~~ ~~$h_2 = \sigma(5,3)$~~

$h_1 = \dfrac{1}{1 + e^{-zh_1}} = 0,98$

$h_2 = \dfrac{1}{1 + e^{-zh_2}} = 0,99$



(Pre-activation)

Mo Tu We Th Fr Sa Su

Output Layer Memo No. ______ Date / /

$z_{o1} = 0,7 \cdot 0,98 + 0,8 \cdot 0,99 + 0,5 =$ ~~2,[illegible]6~~ 1,978

$z_{o2} = 0,9 \cdot 0,98 + 0,1 \cdot 0,99 + 0,5 = 1,481$

$$o_1 = \frac{1}{1 + e^{-1,978}} = 0,87$$

$$o_2 = \frac{1}{1 + e^{-1,481}} = 0,81$$

Error Calculation

$$E = \frac{1}{2}\left((0,1 - 0,87)^2 + (0,05 - 0,81)^2\right) = 0,58525$$

Backpropagation

① $\delta_{o_1} = (0,87 - 0,1) \cdot 0,87 \cdot (1 - 0,87) = 0,08$

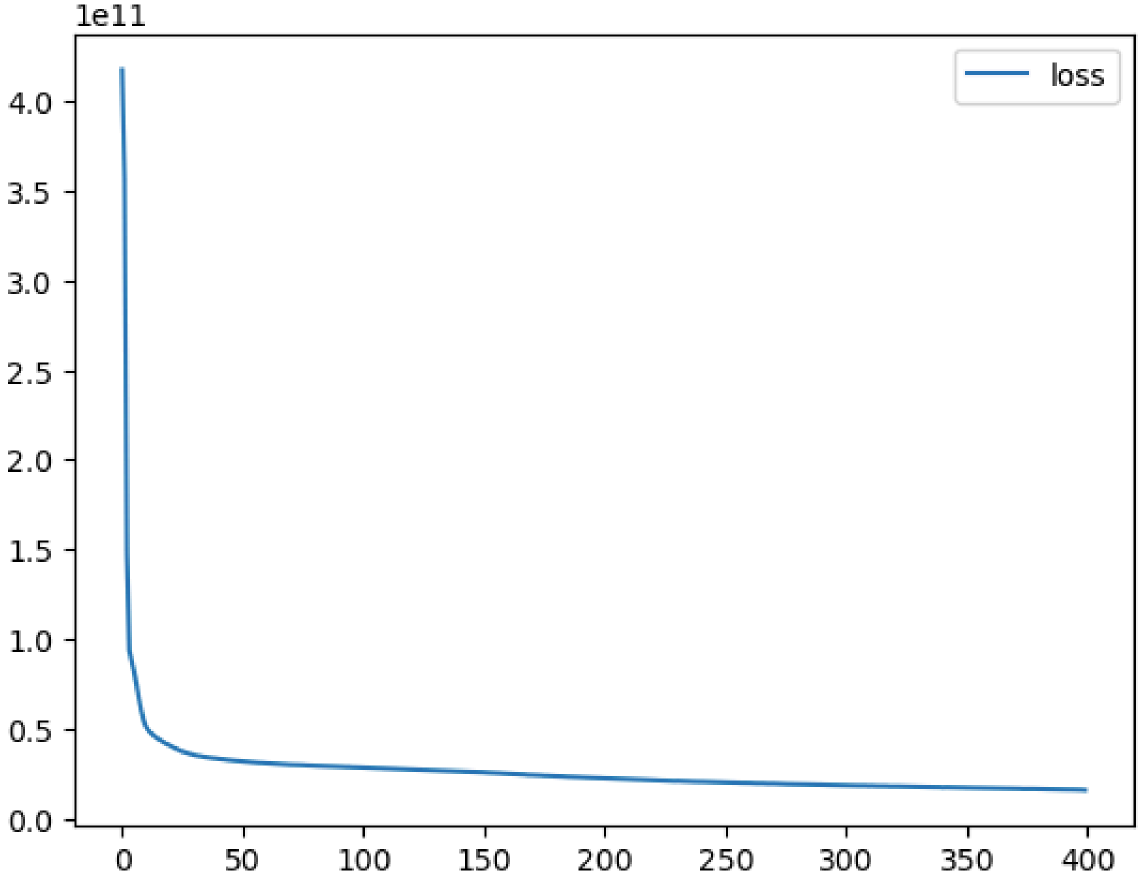
$\delta_{o_2} = (0,81 - 0,05) \cdot 0,05 \cdot (1 - 0,05) = 0,03$

② $\delta h_1 = (0,08 \cdot 0,7 + 0,03 \cdot 0,5) \cdot 0,98 \cdot (1 - 0,98) =$

$\delta h_2 = (0,08 \cdot 0,8 + 0,03 \cdot 0,1) \cdot 0,95 \cdot (1 - 0,95) =$

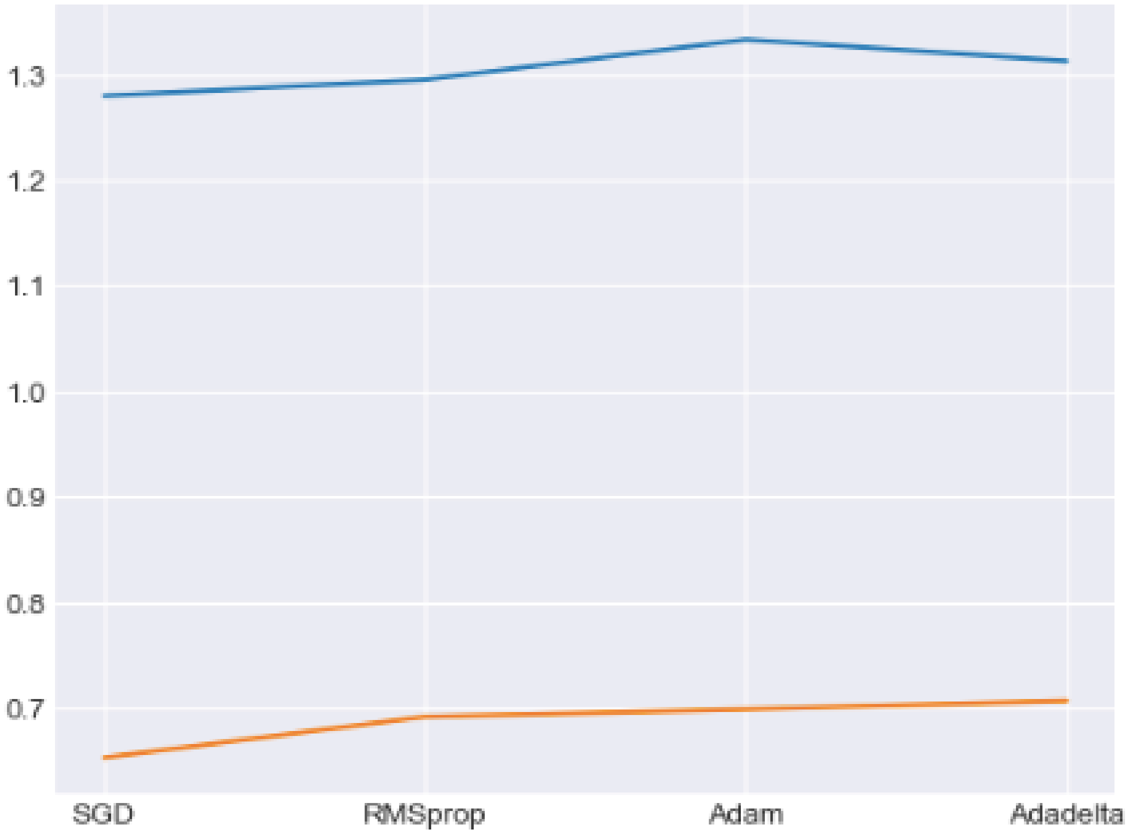
$\delta h_1 = 0,001$ $\quad \delta h_2 = 0,0006$

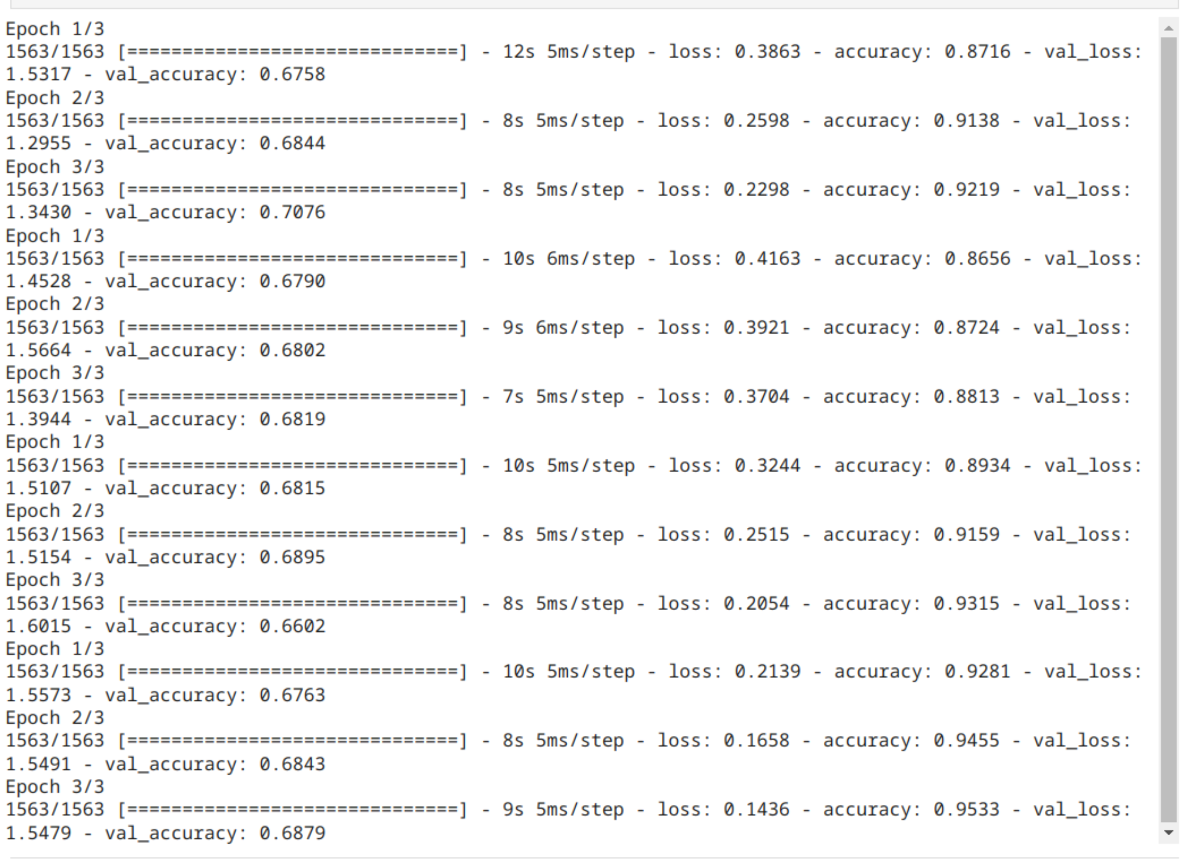
Week-4


1e11
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
250
300
350
400
loss

```
Model: "sequential"
_________________________________________________________________
 Layer (type) Output Shape Param #
=====================================================  ==============
 dense (Dense) (None, 128) 2560

 dense_1 (Dense) (None, 64) 8256

 dense_2 (Dense) (None, 32) 2080

 dense_3 (Dense) (None, 16) 528

 dense_4 (Dense) (None, 1) 17

=====================================================  ==============
Total params: 13441 (52.50 KB)
Trainable params: 13441 (52.50 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

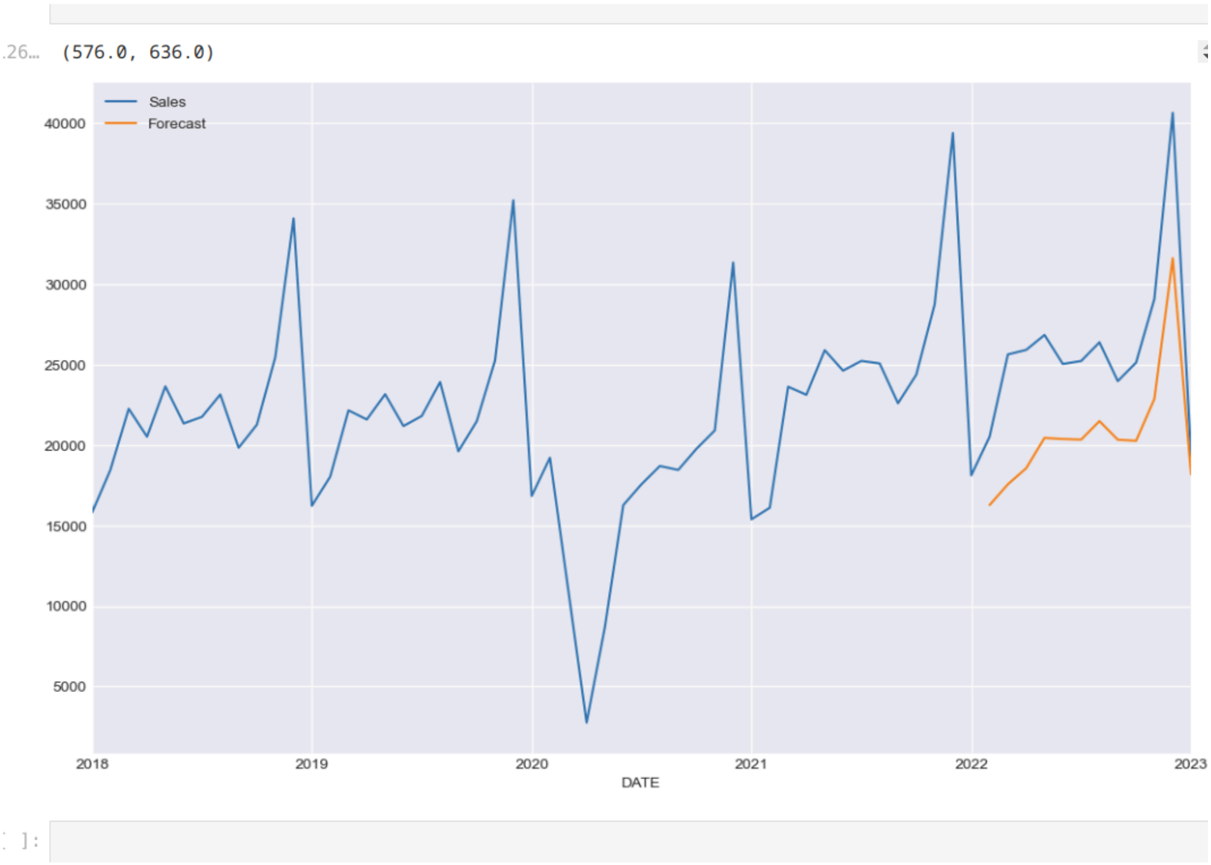
Week - 5





```
Epoch 1/3
1563/1563 [==============================] - 12s 5ms/step - loss: 0.3863 - accuracy: 0.8716 - val_loss:
1.5317 - val_accuracy: 0.6758
Epoch 2/3
1563/1563 [==============================] - 8s 5ms/step - loss: 0.2598 - accuracy: 0.9138 - val_loss:
1.2955 - val_accuracy: 0.6844
Epoch 3/3
1563/1563 [==============================] - 8s 5ms/step - loss: 0.2298 - accuracy: 0.9219 - val_loss:
1.3430 - val_accuracy: 0.7076
Epoch 1/3
1563/1563 [==============================] - 10s 6ms/step - loss: 0.4163 - accuracy: 0.8656 - val_loss:
1.4528 - val_accuracy: 0.6790
Epoch 2/3
1563/1563 [==============================] - 9s 6ms/step - loss: 0.3921 - accuracy: 0.8724 - val_loss:
1.5664 - val_accuracy: 0.6802
Epoch 3/3
1563/1563 [==============================] - 7s 5ms/step - loss: 0.3704 - accuracy: 0.8813 - val_loss:
1.3944 - val_accuracy: 0.6819
Epoch 1/3
1563/1563 [==============================] - 10s 5ms/step - loss: 0.3244 - accuracy: 0.8934 - val_loss:
1.5107 - val_accuracy: 0.6815
Epoch 2/3
1563/1563 [==============================] - 8s 5ms/step - loss: 0.2515 - accuracy: 0.9159 - val_loss:
1.5154 - val_accuracy: 0.6895
Epoch 3/3
1563/1563 [==============================] - 8s 5ms/step - loss: 0.2054 - accuracy: 0.9315 - val_loss:
1.6015 - val_accuracy: 0.6602
Epoch 1/3
1563/1563 [==============================] - 10s 5ms/step - loss: 0.2139 - accuracy: 0.9281 - val_loss:
1.5573 - val_accuracy: 0.6763
Epoch 2/3
1563/1563 [==============================] - 8s 5ms/step - loss: 0.1658 - accuracy: 0.9455 - val_loss:
1.5491 - val_accuracy: 0.6843
Epoch 3/3
1563/1563 [==============================] - 9s 5ms/step - loss: 0.1436 - accuracy: 0.9533 - val_loss:
1.5479 - val_accuracy: 0.6879
```

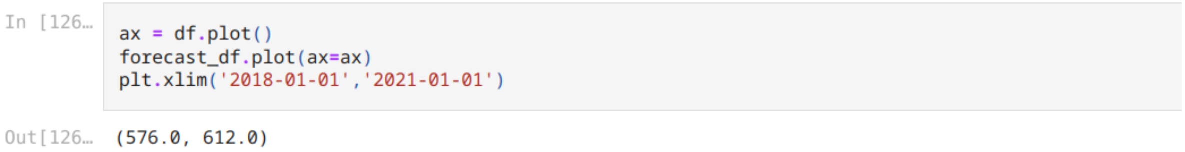
Week - 6



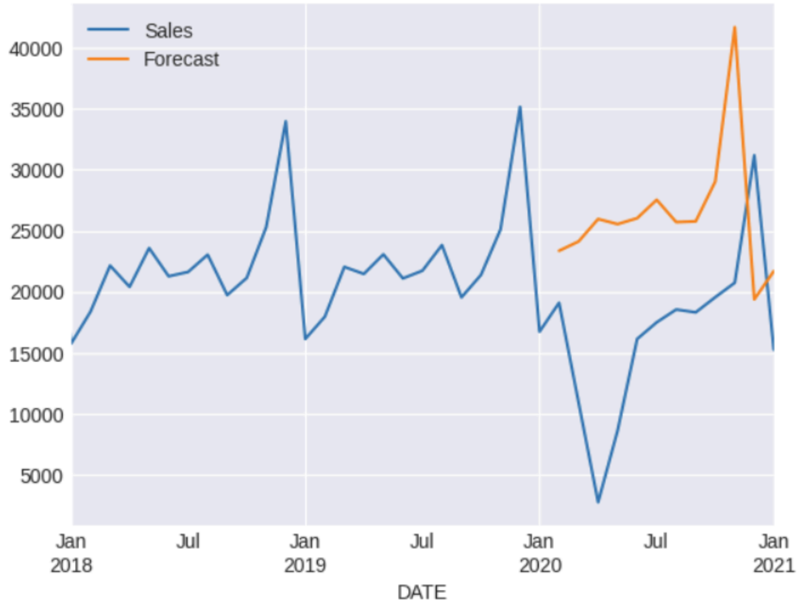
```
.26…  (576.0, 636.0)
```

```
[ ]:
```

DATE



```
In [126…  ax = df.plot()
          forecast_df.plot(ax=ax)
          plt.xlim('2018-01-01','2021-01-01')

Out[126…  (576.0, 612.0)
```

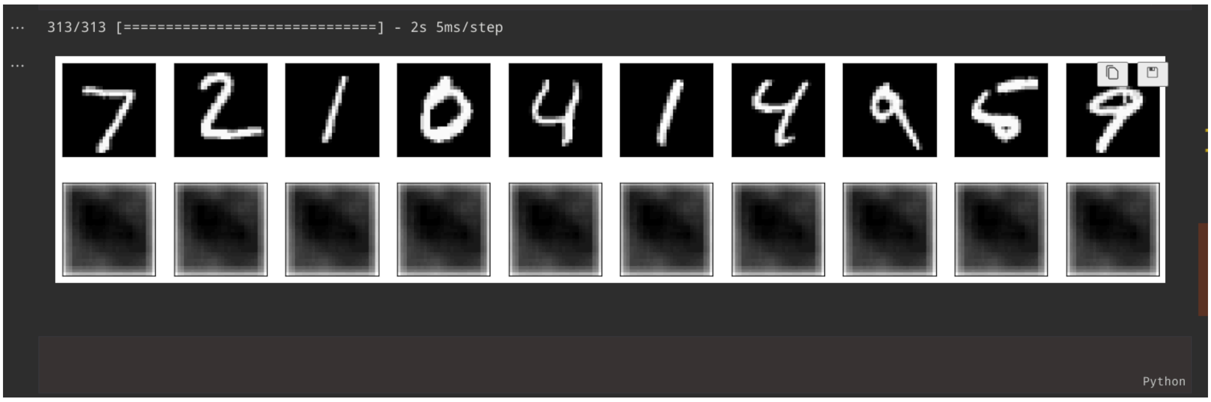


Week - 7

task-1

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type) Output Shape Param #
========================================================= ===============
 sequential (Sequential) (None, 25) 420625

 sequential_1 (Sequential) (None, 28, 28) 421384

========================================================= ===============
Total params: 842009 (3.21 MB)
Trainable params: 842009 (3.21 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

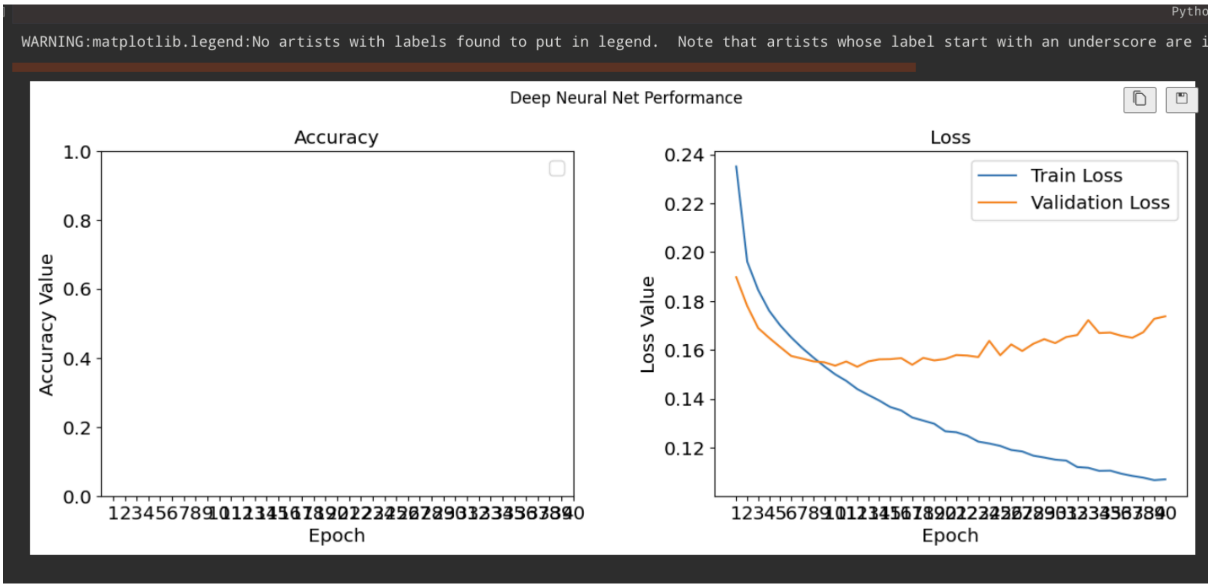
task - 2



```
313/313 [==============================] - 2s 5ms/step
```

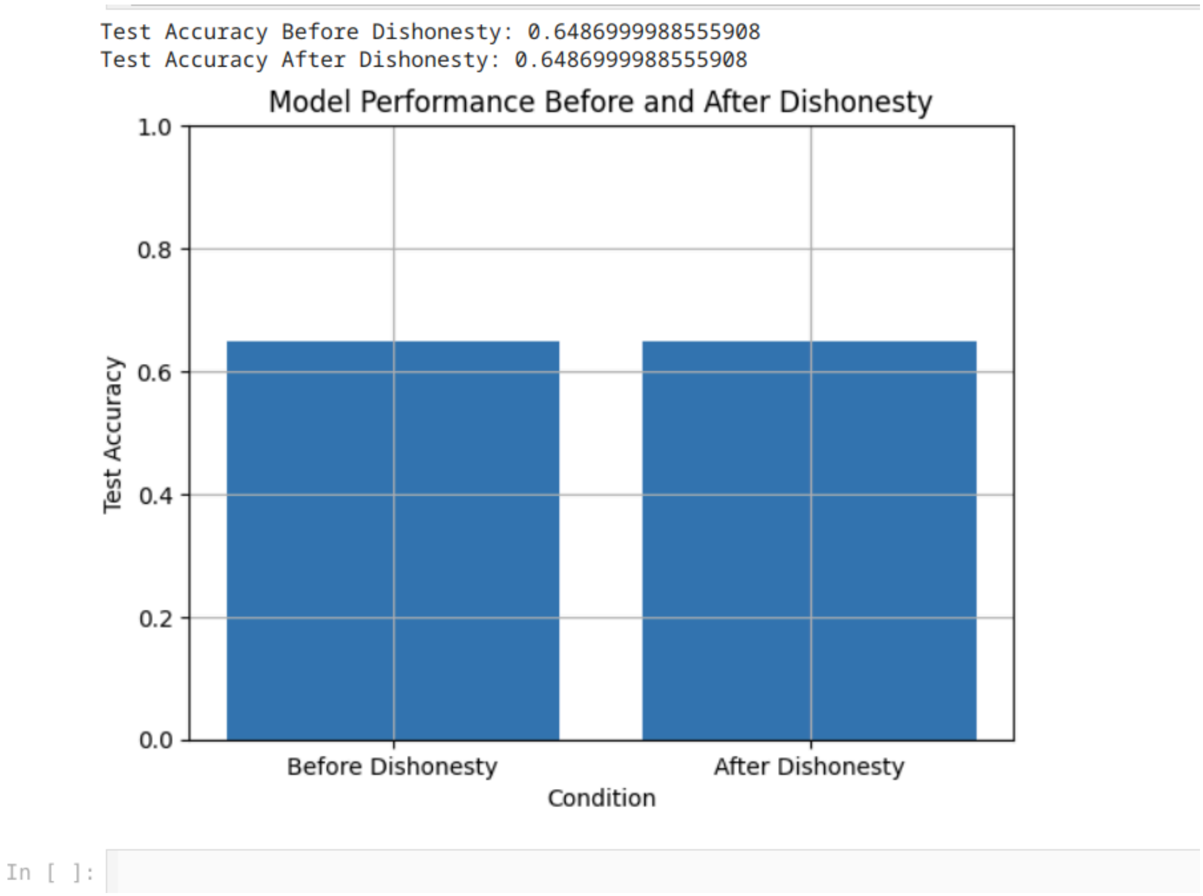
Week - 8

Week - 9



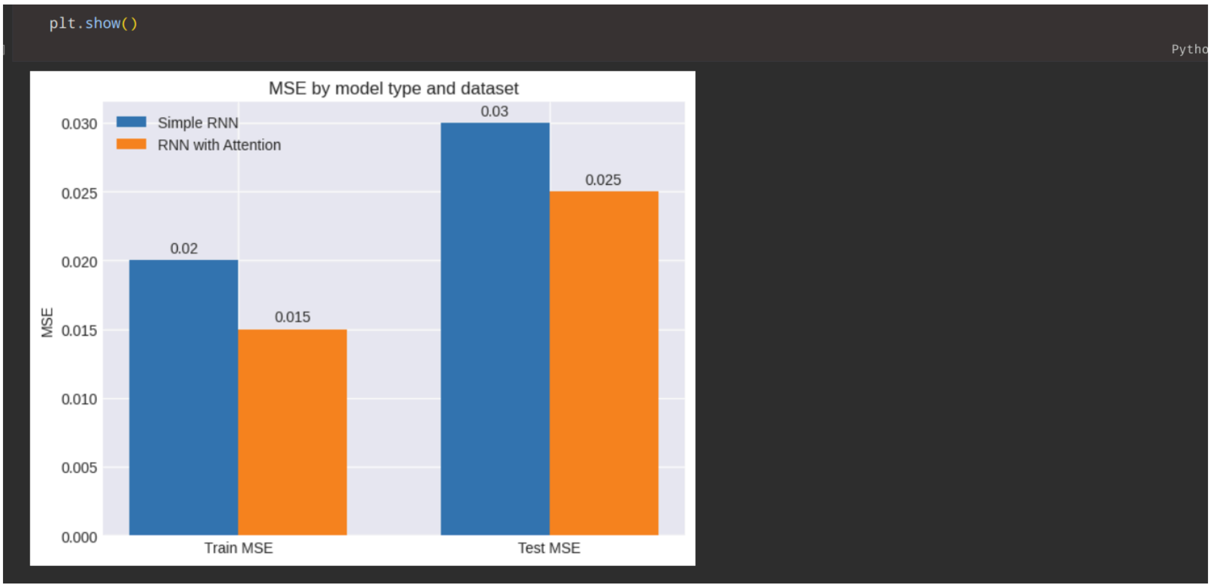
```
WARNING:matplotlib.legend:No artists with labels found to put in legend.  Note that artists whose label start with an underscore are i
```

Week - 10



```
Test Accuracy Before Dishonesty: 0.6486999988555908
Test Accuracy After Dishonesty: 0.6486999988555908
```

Week - 11



```
plt.show()
```