**Deep Learning & Neural Networks**

**Description: -**

I worked on Boston housing dataset. The problem was to predict the selling label. The dataset consisted of 13 features and 1 output. I had to implement three algorithms of which are: -

* Stochastic gradient descent
* Batch Gradient Descent
* Mini-batch Gradient Descent

To achieve accurate output, we had to follow few steps which are to do forward propagation which is the dot product. This was further divided into two parts which is from input to hidden layer and hidden to output layer respectively. After taking dot product of each neuron we had to activate it by an activation function. We used Relu function to activate our function. I calculated errors for respective algorithms. In stochastic you have to calculate error per instance and then take average at the end of epoch. However, in Batch you have to calculate forward pass for all instance first and then calculate error of the last instance meaning 1 error per epoch. Lastly, in mini-batch you calculate error per batch. In the end I did backpropagation to improve our weights. Lastly, I plotted graph of epochs against loss function.

**Rubric for grading:**

1. Correct implementation of the basic neural net.

**[10 points]**

Answer: - Check Code

1. Modularization of the code. It means that you must create functions to calculate for instance **[10 points]**

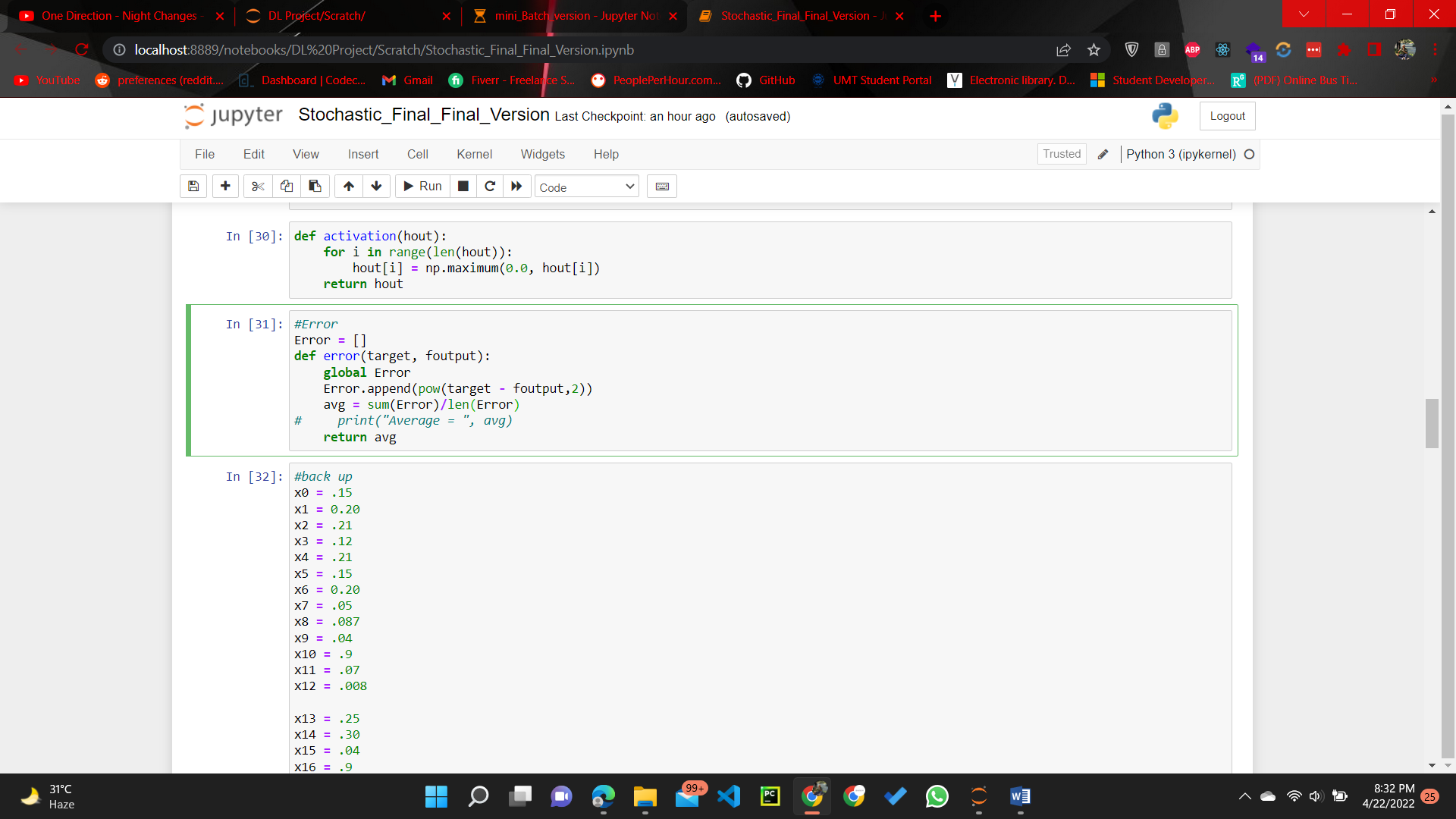
* Error()
* dotProduct()
* sigmoidActivation()
* sigmoidDerivative() etc.

**Note: Do not use the built-in functions.**

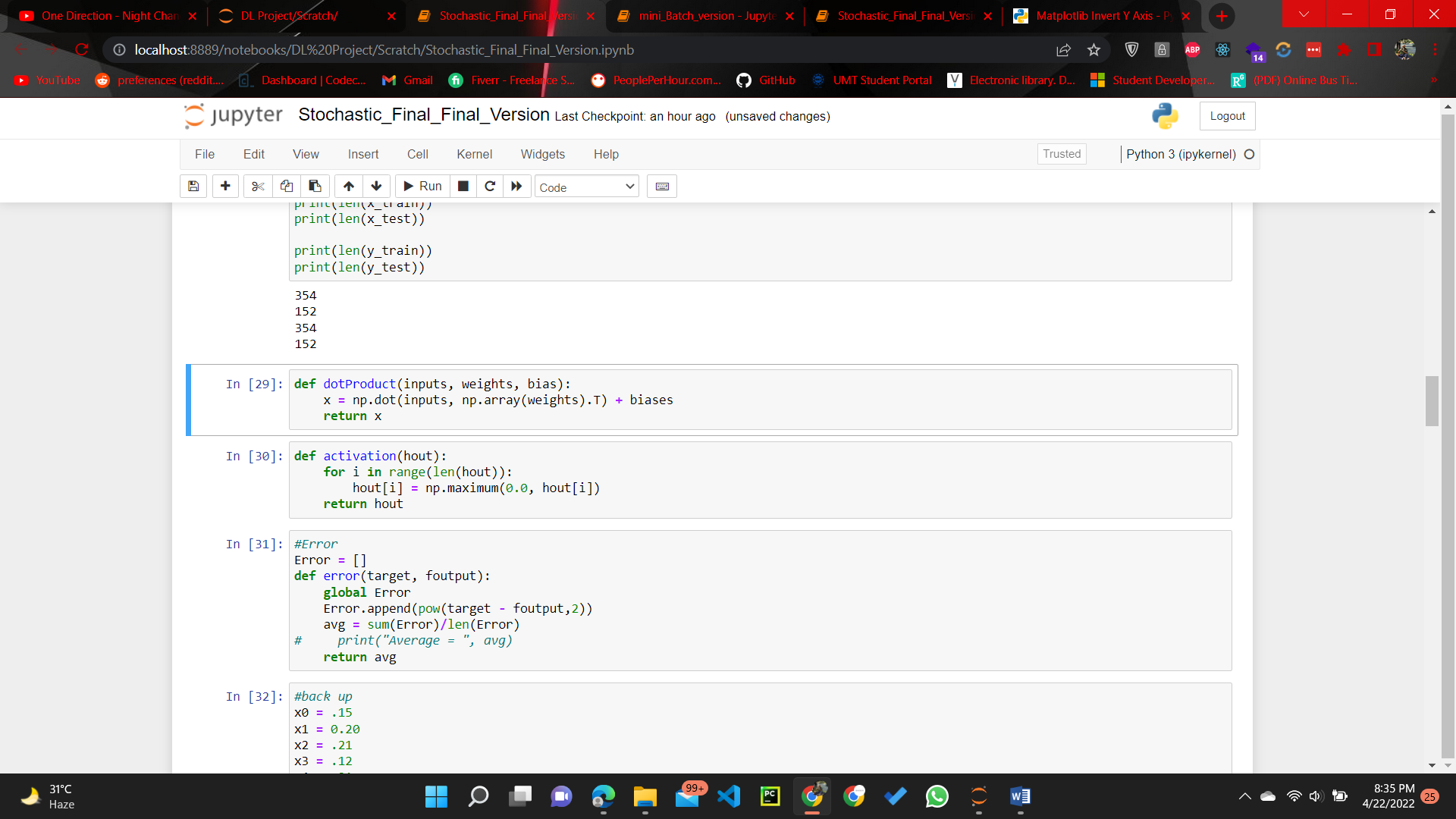
**Answer: -**

Error function, dot product and activation functions are same for all algorithms.

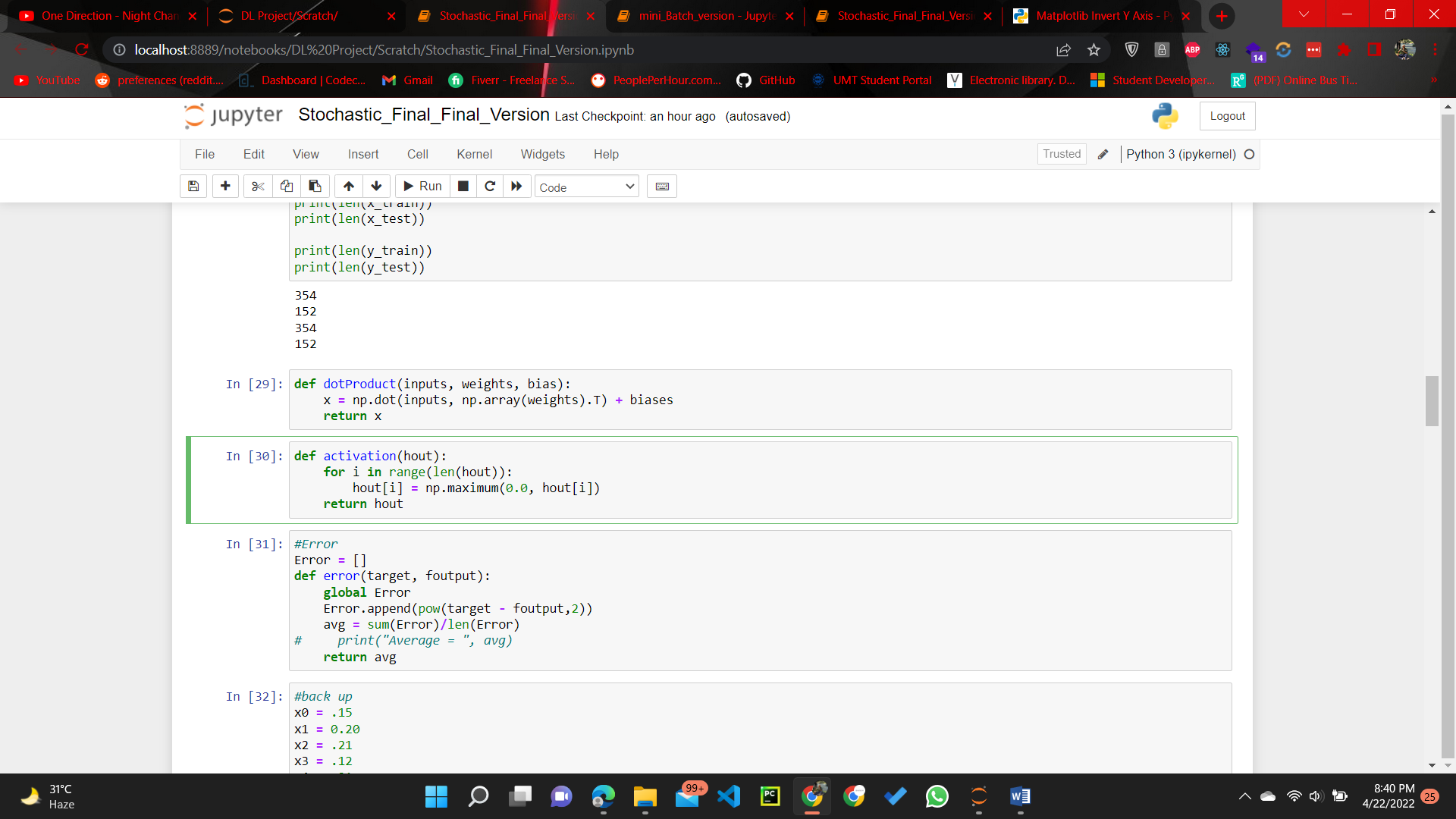
* Error()



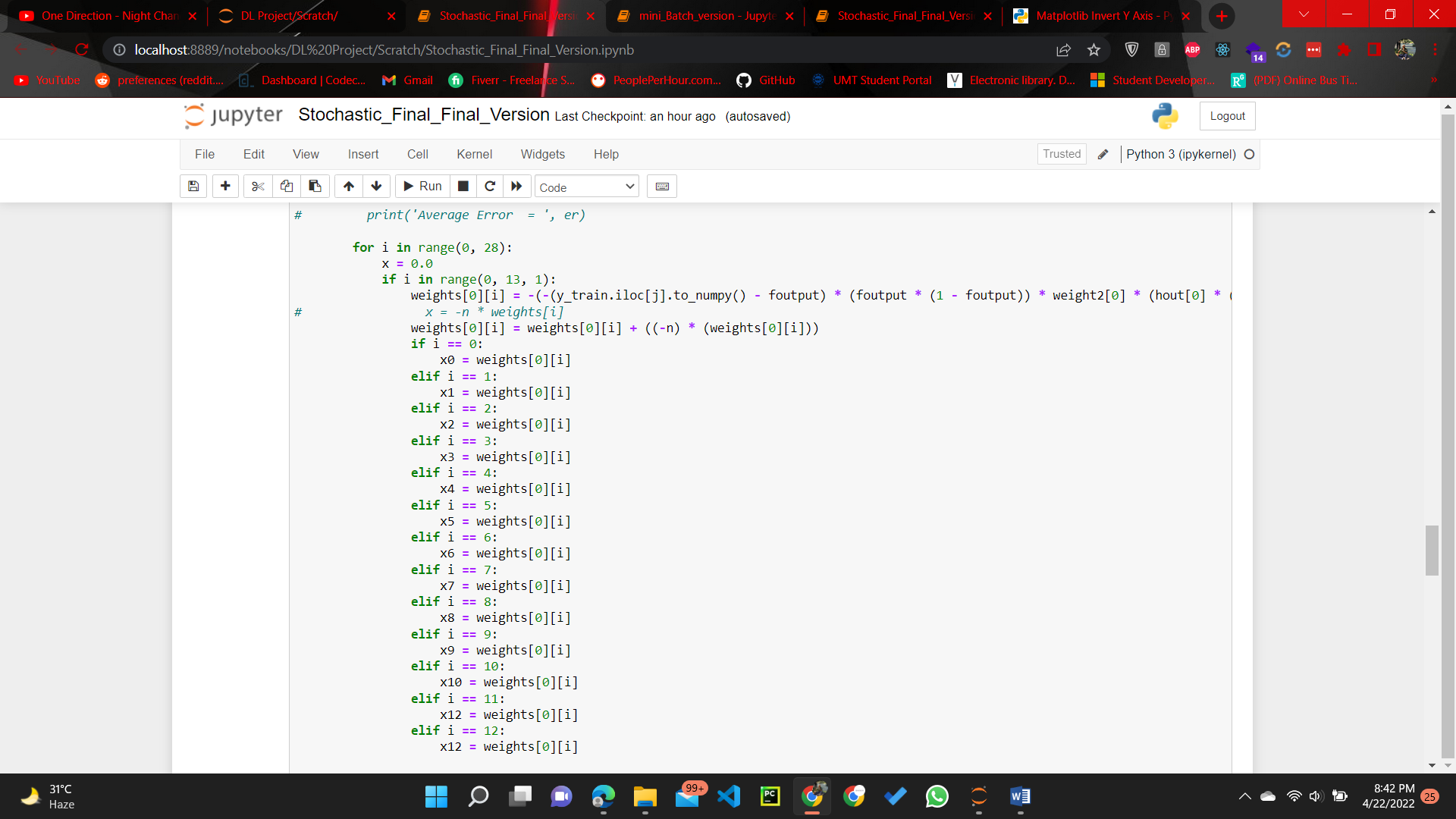
* Dot Product

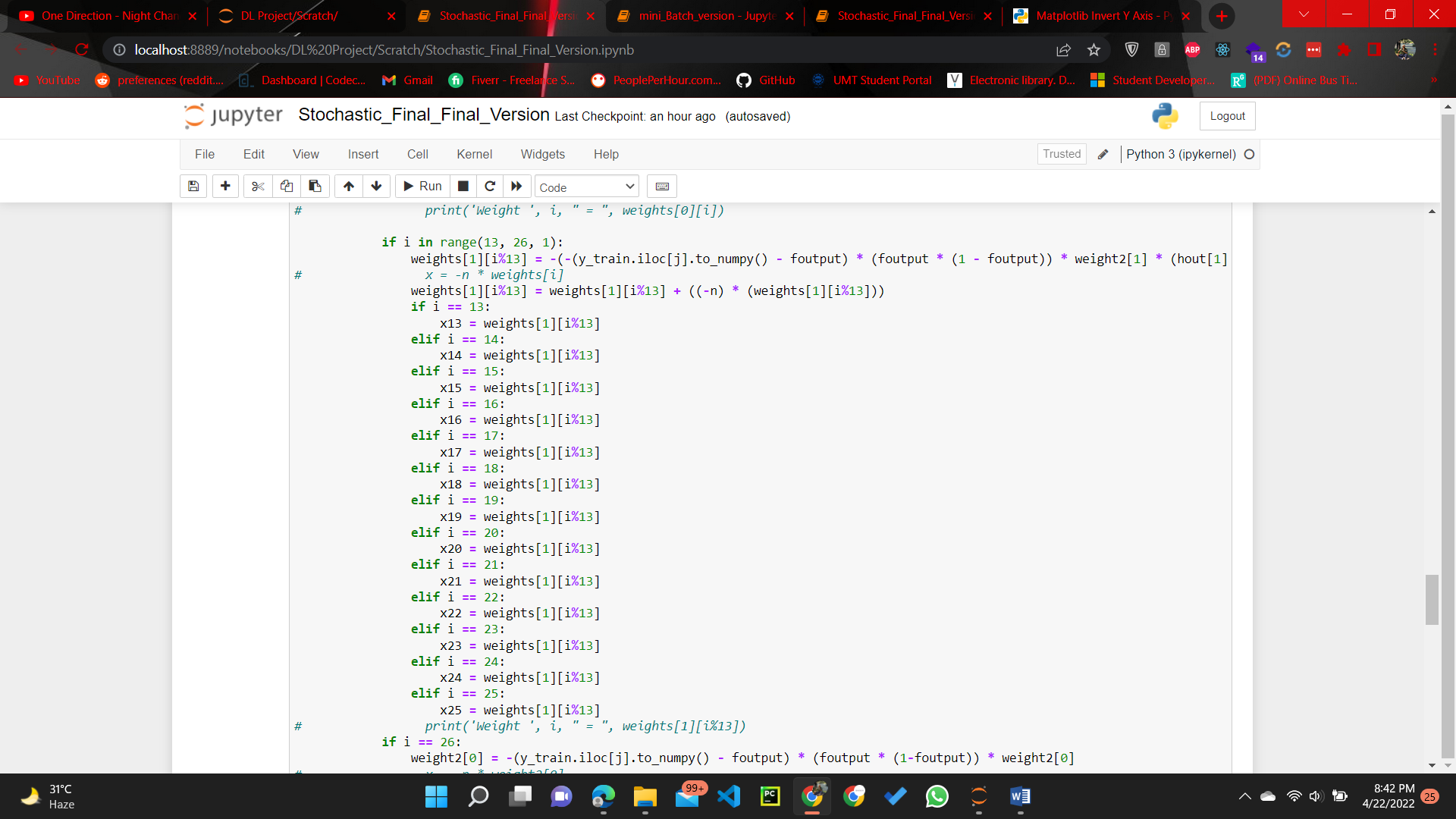


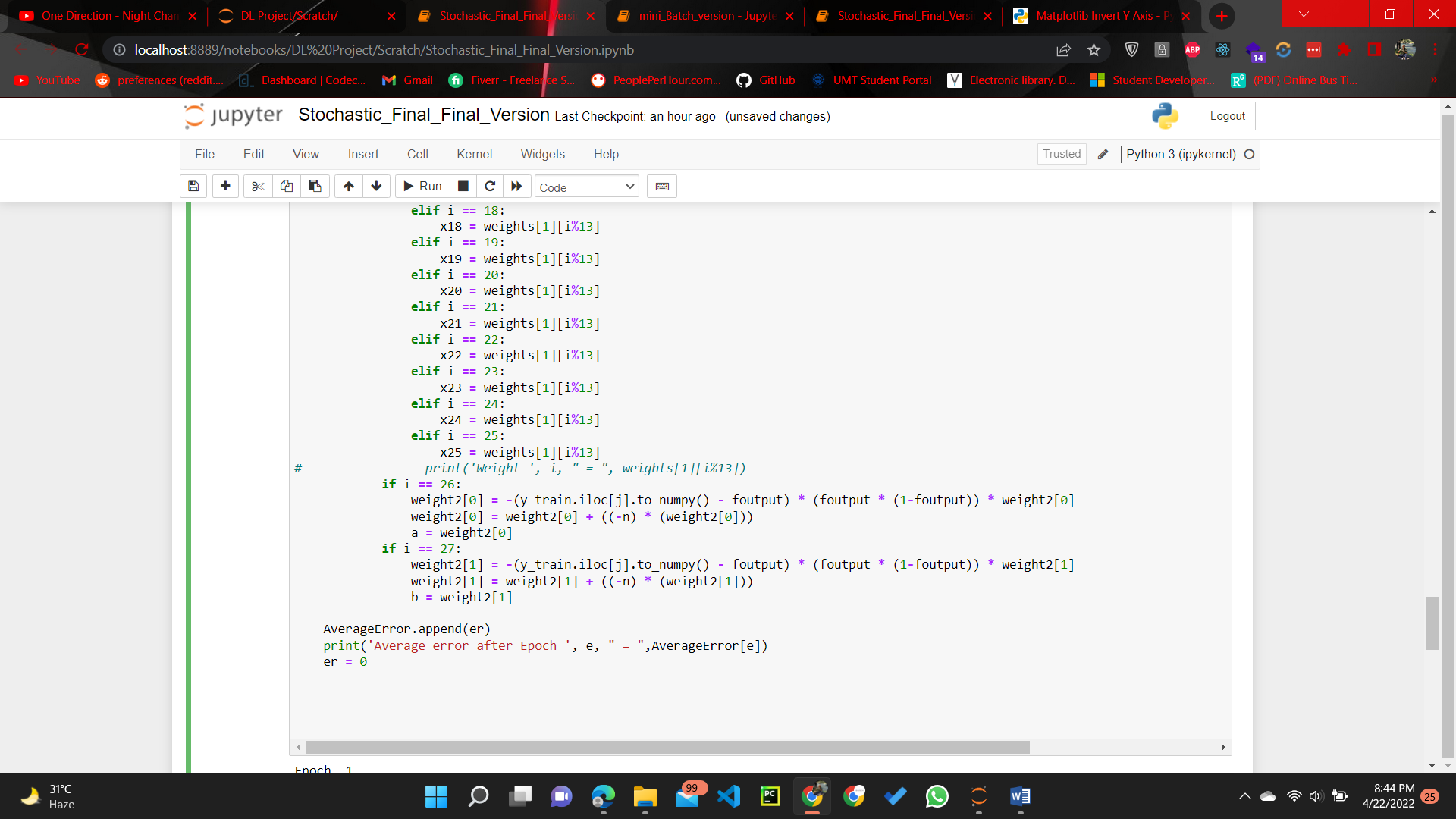
* Activation Function



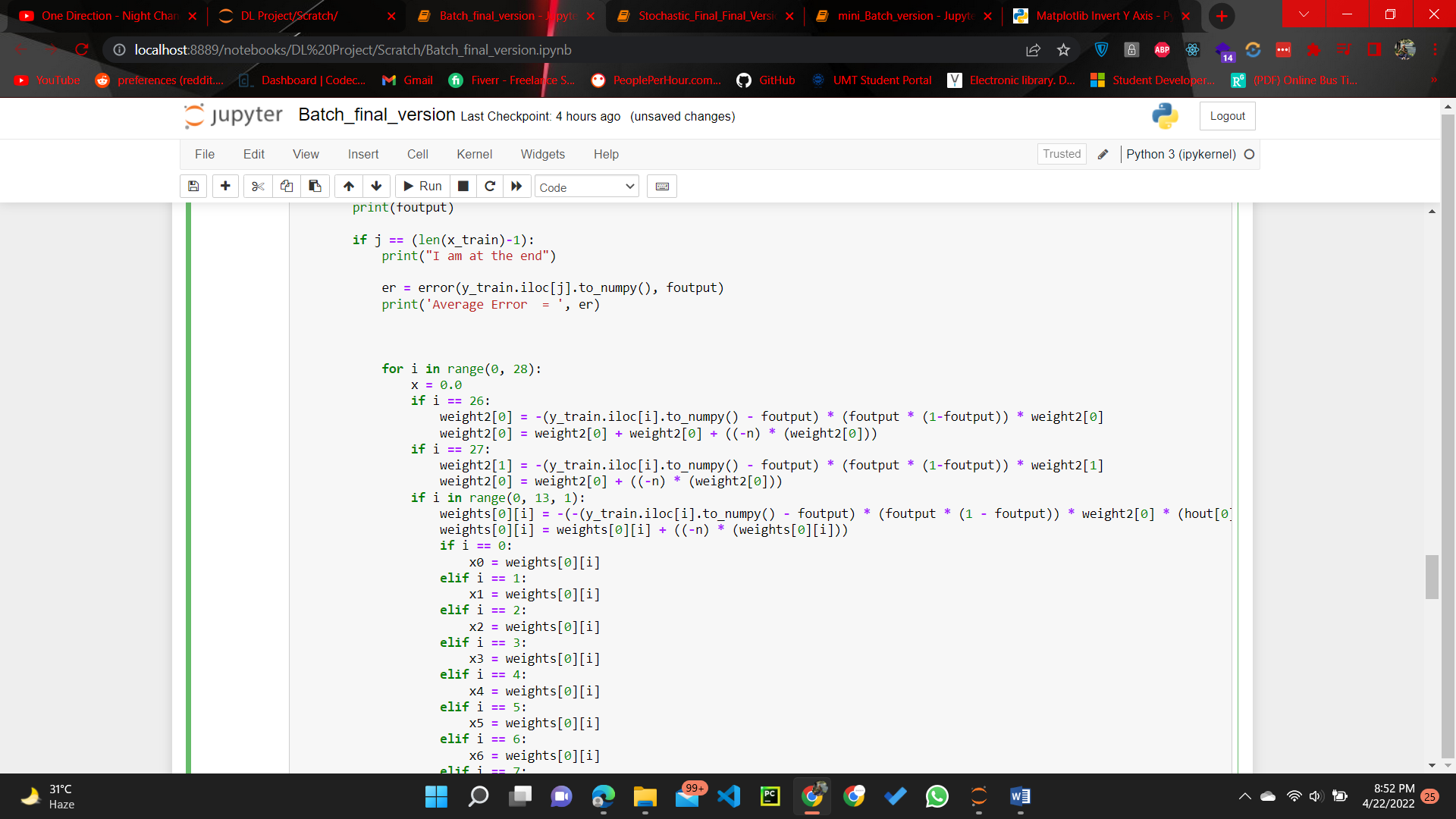
**Derivation for Gradient Descent**

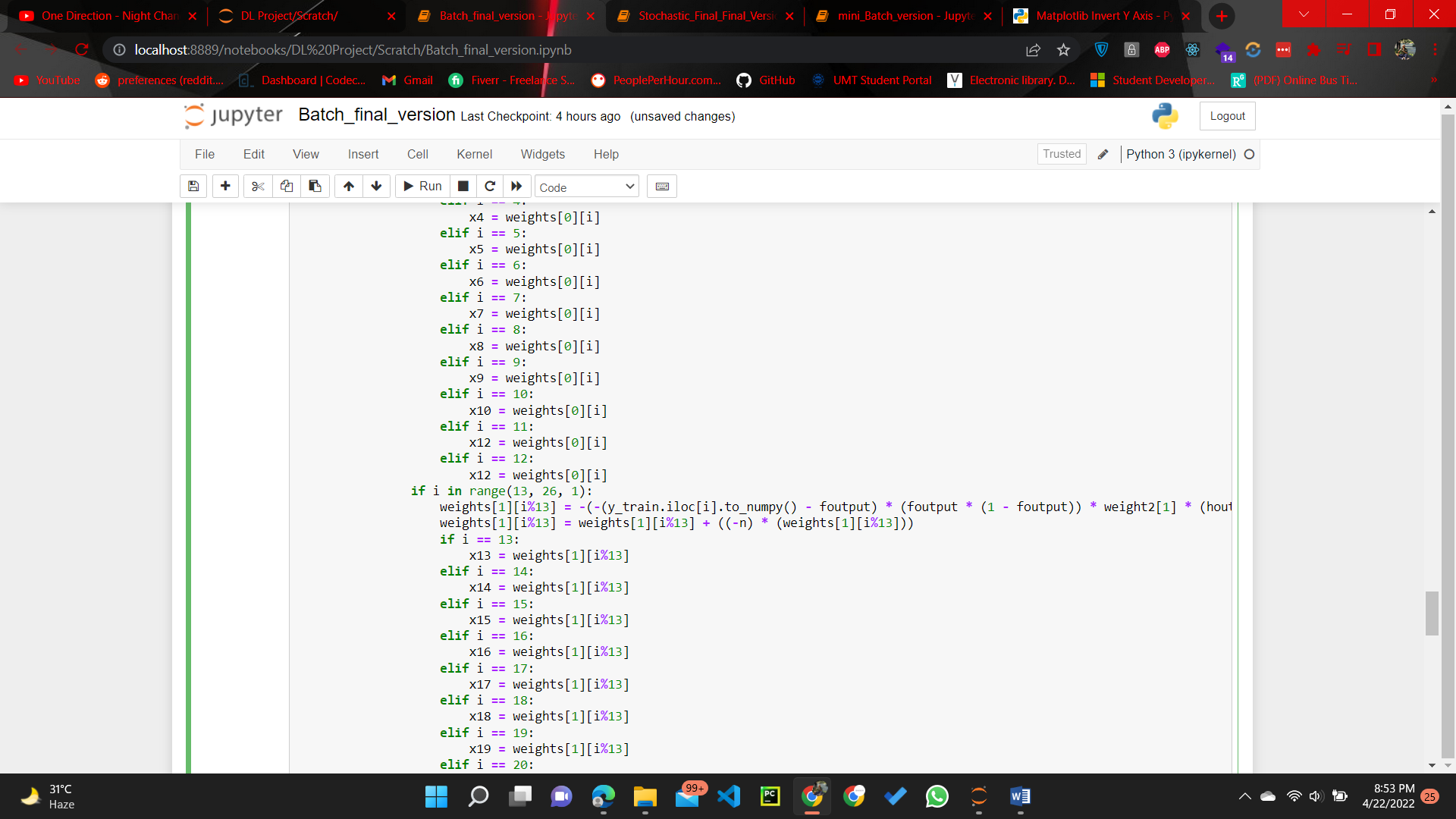


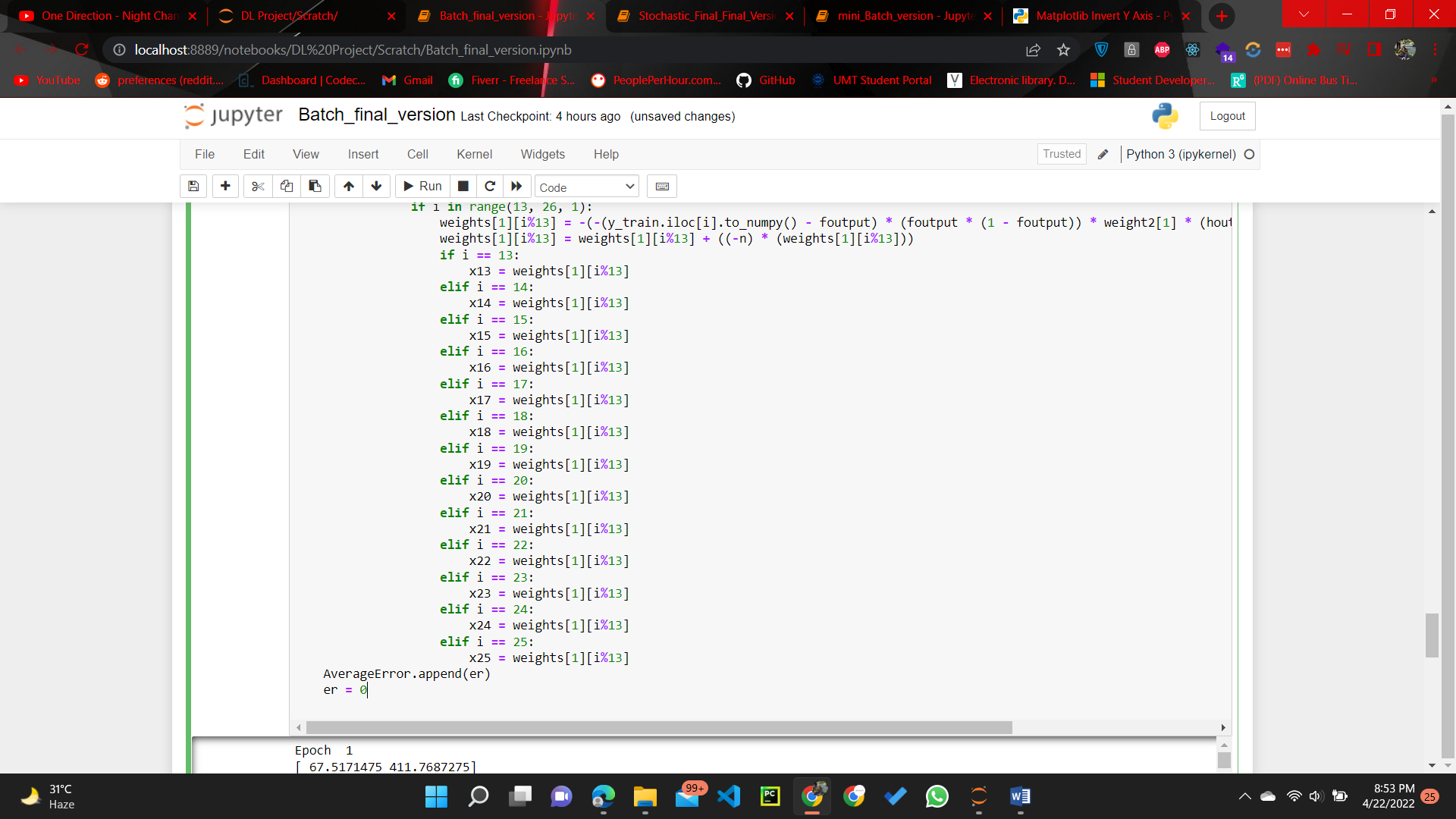




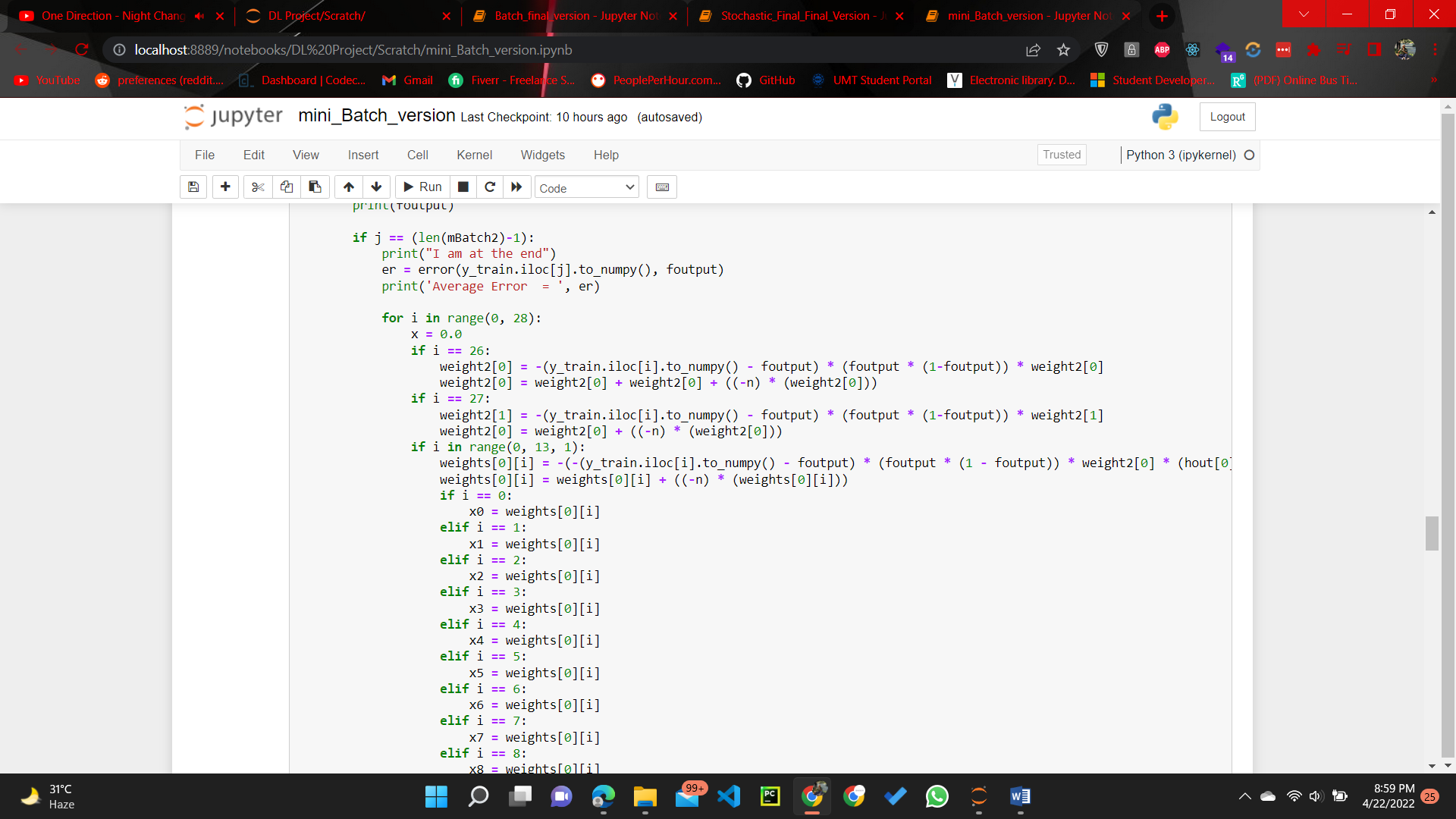
**Derivation for Batch Gradient**

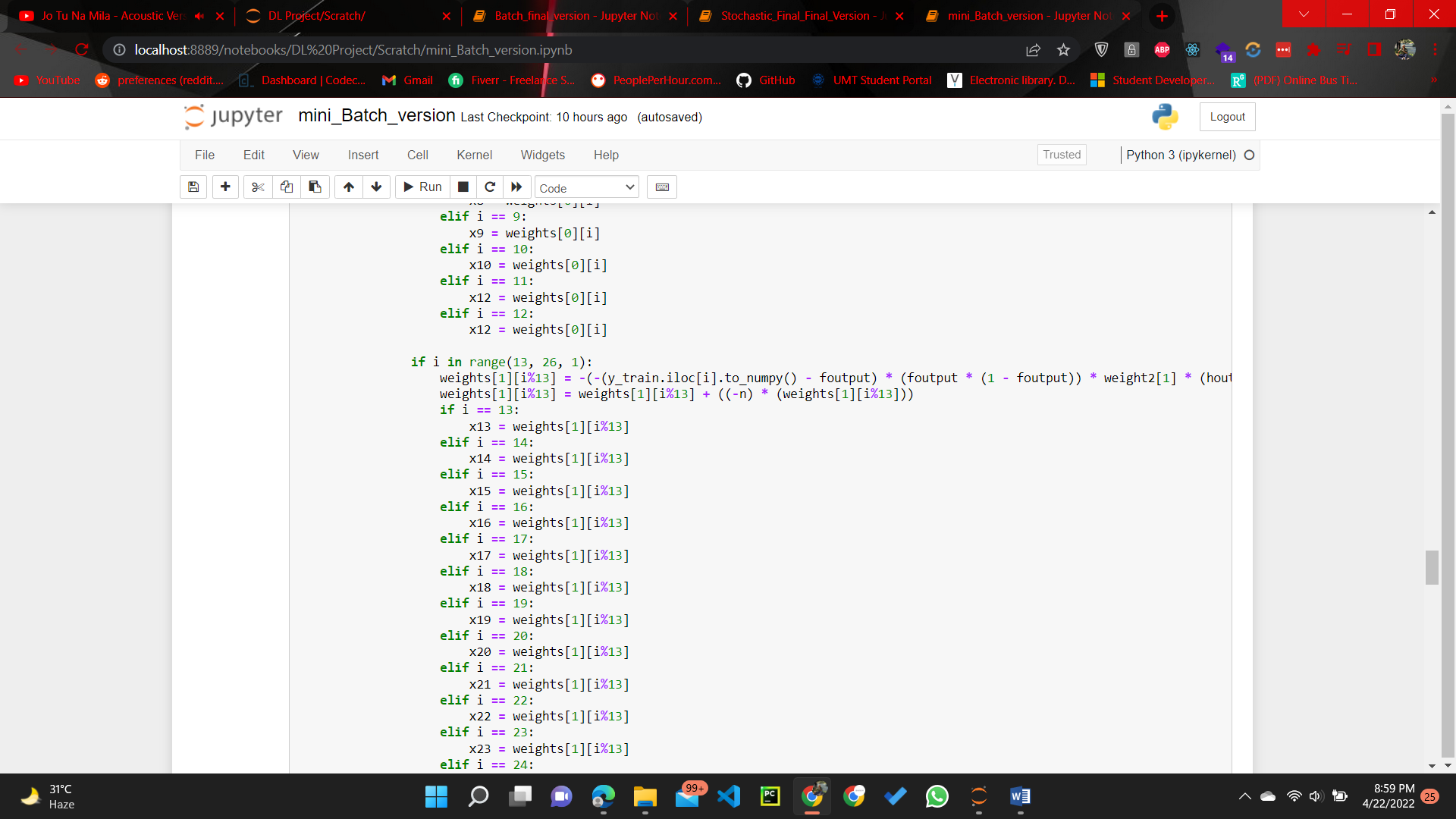


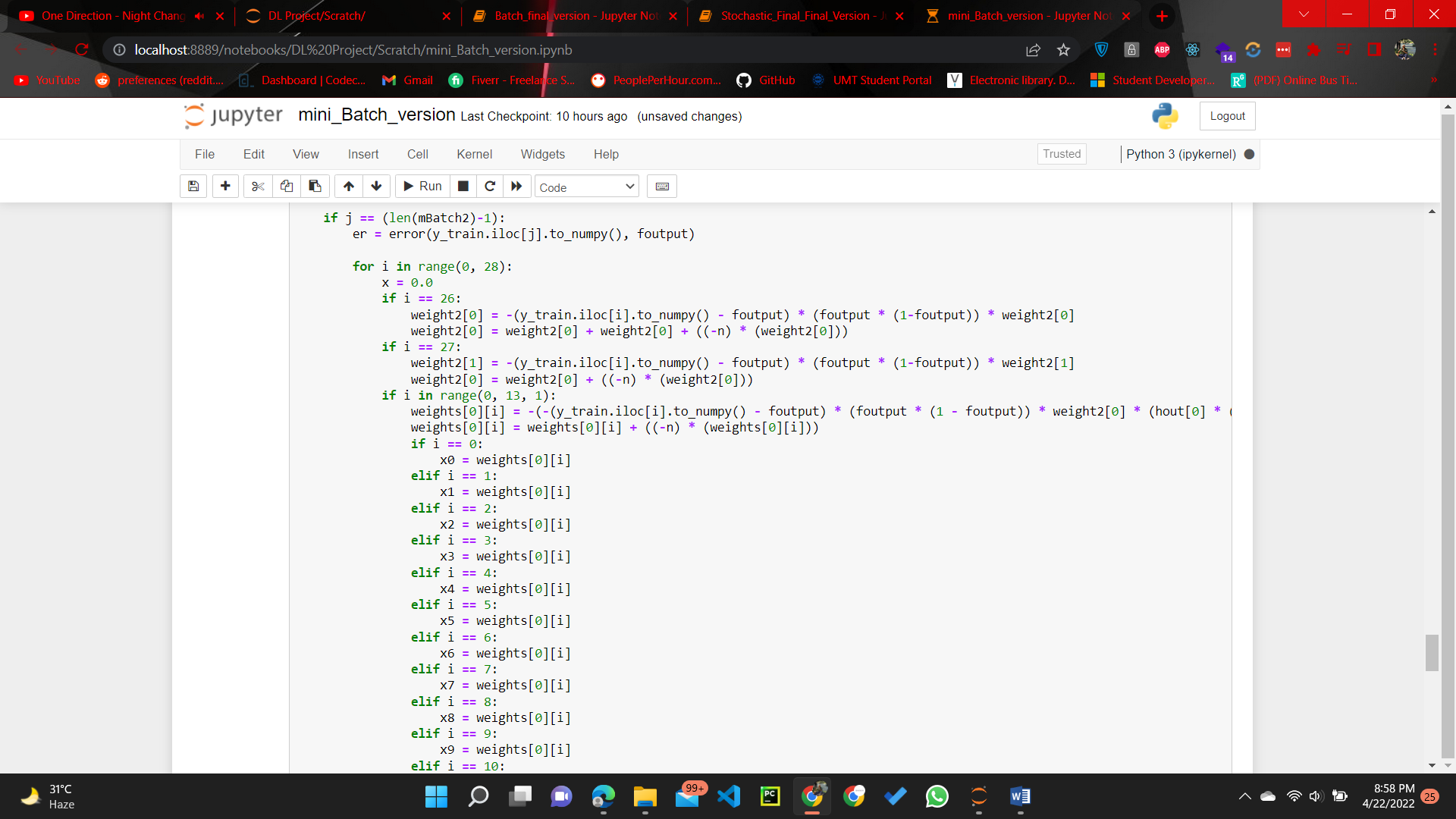


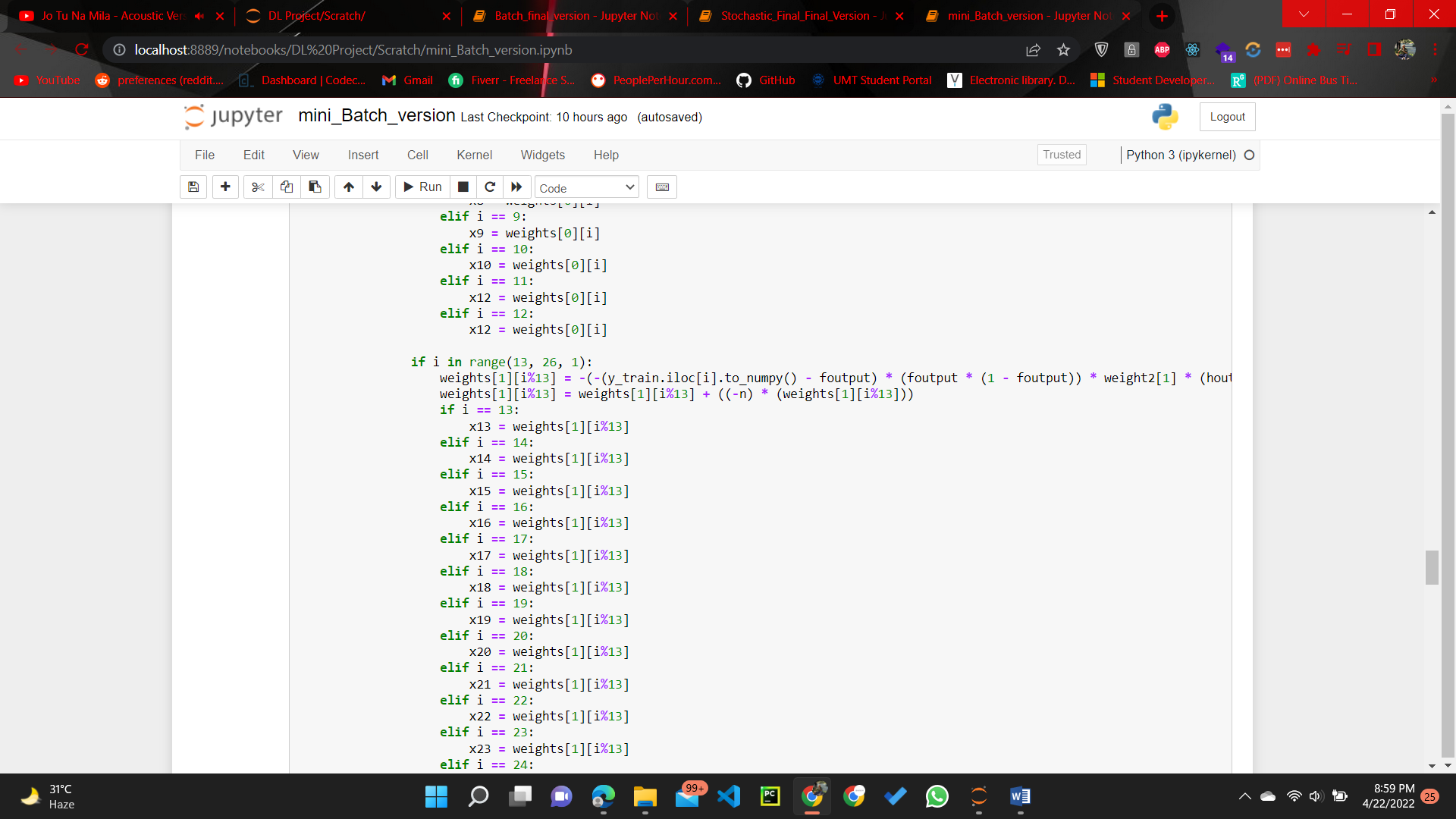


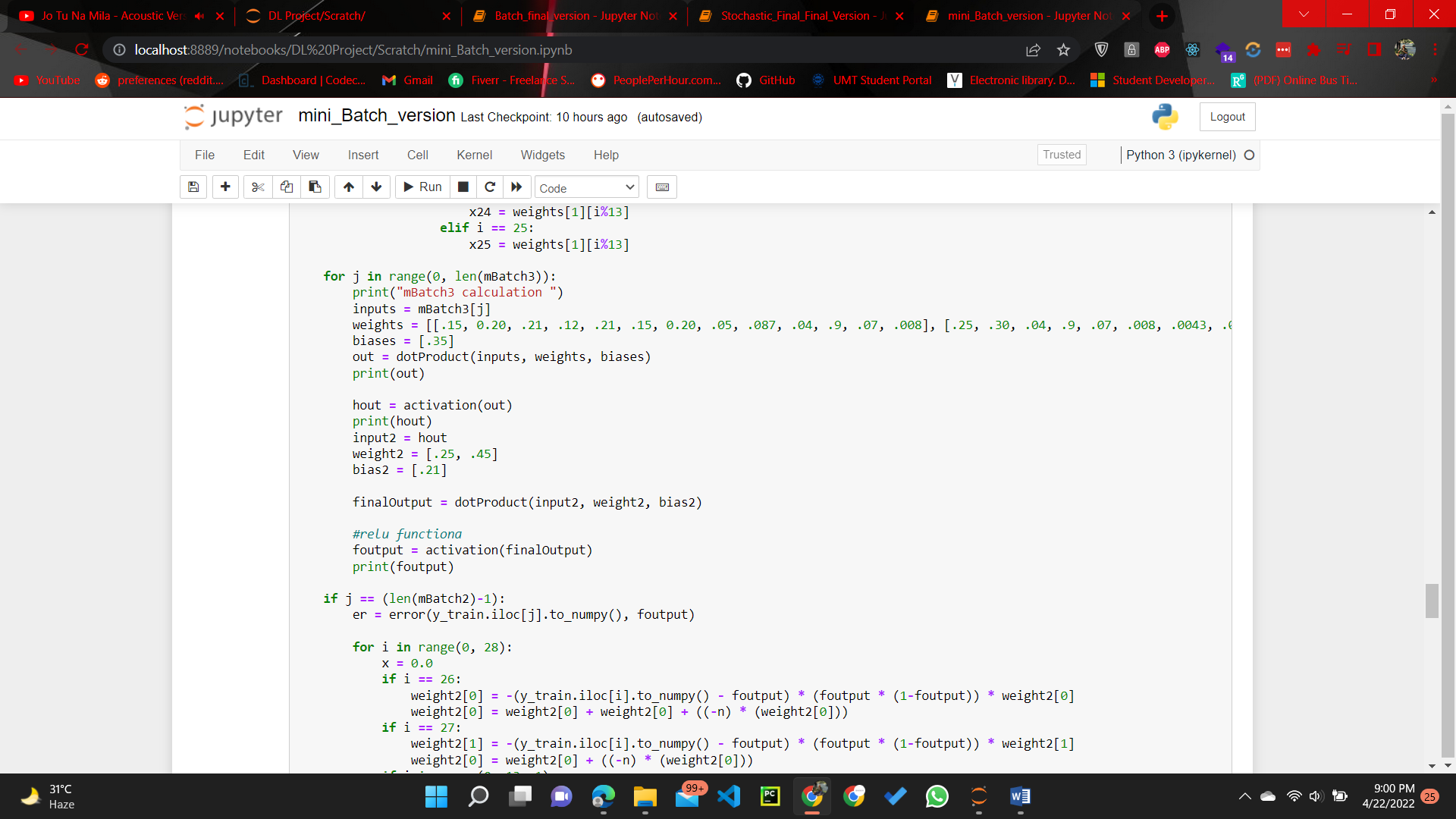
**Derivation for Mini-batch**

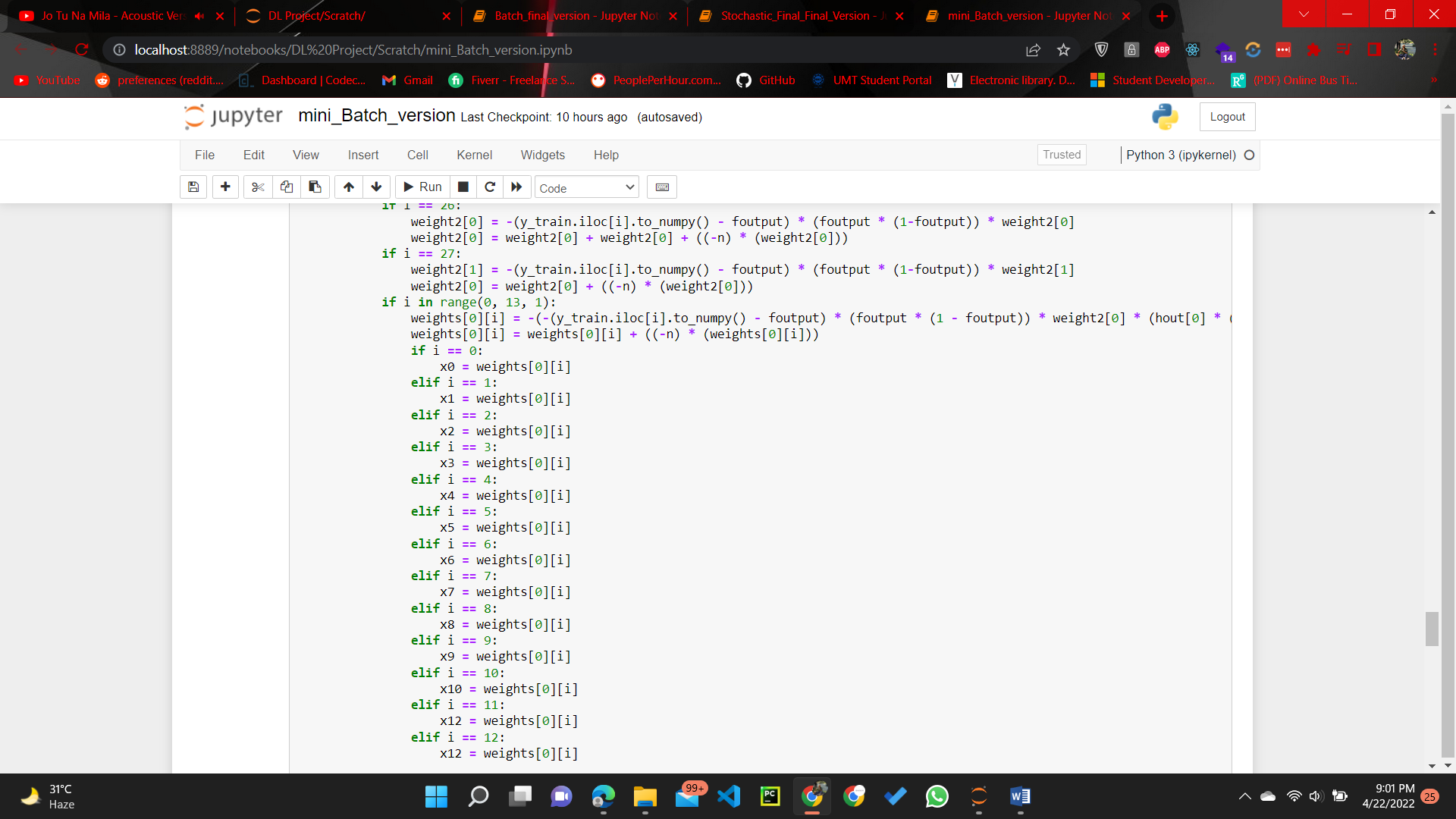


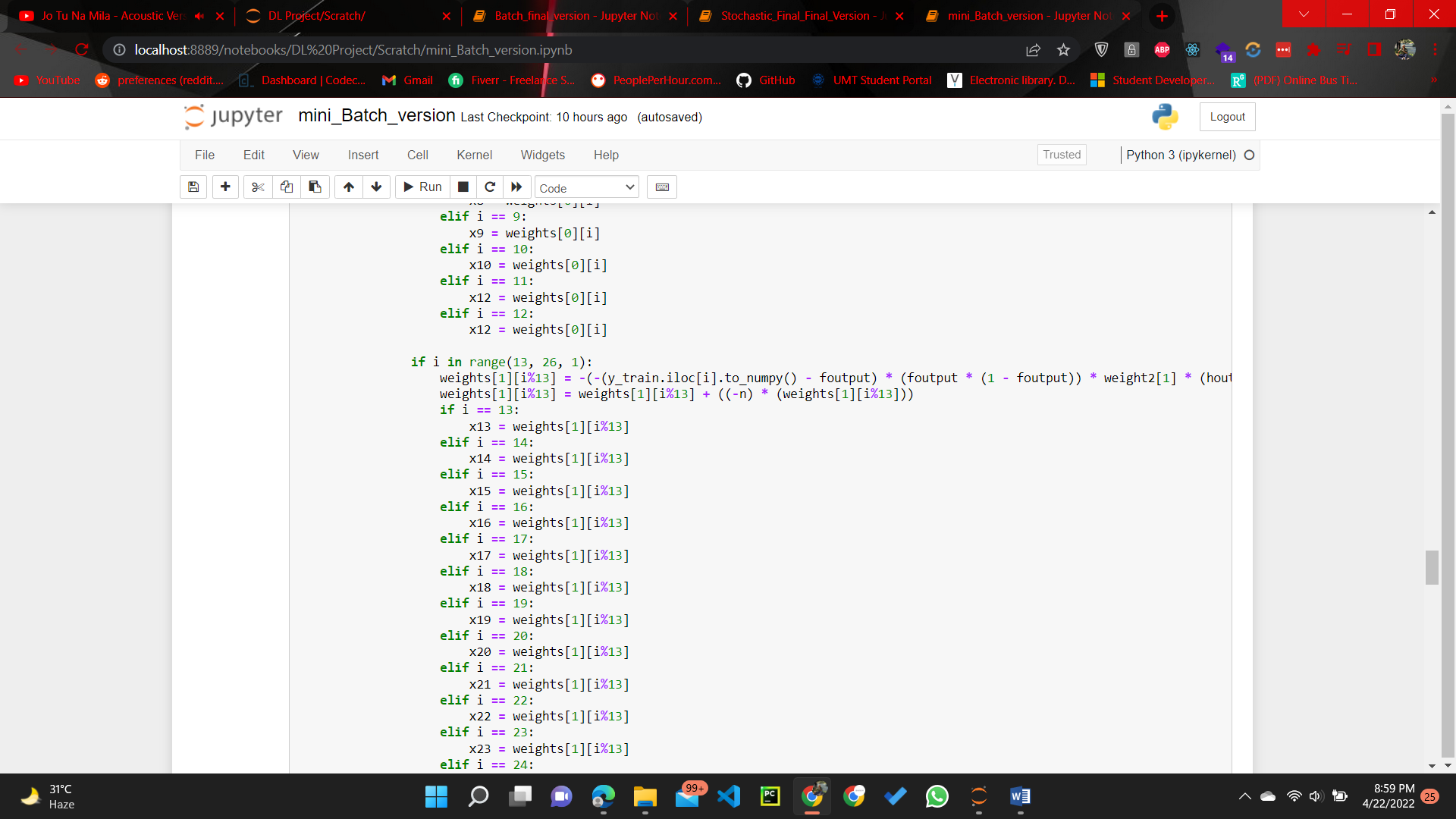






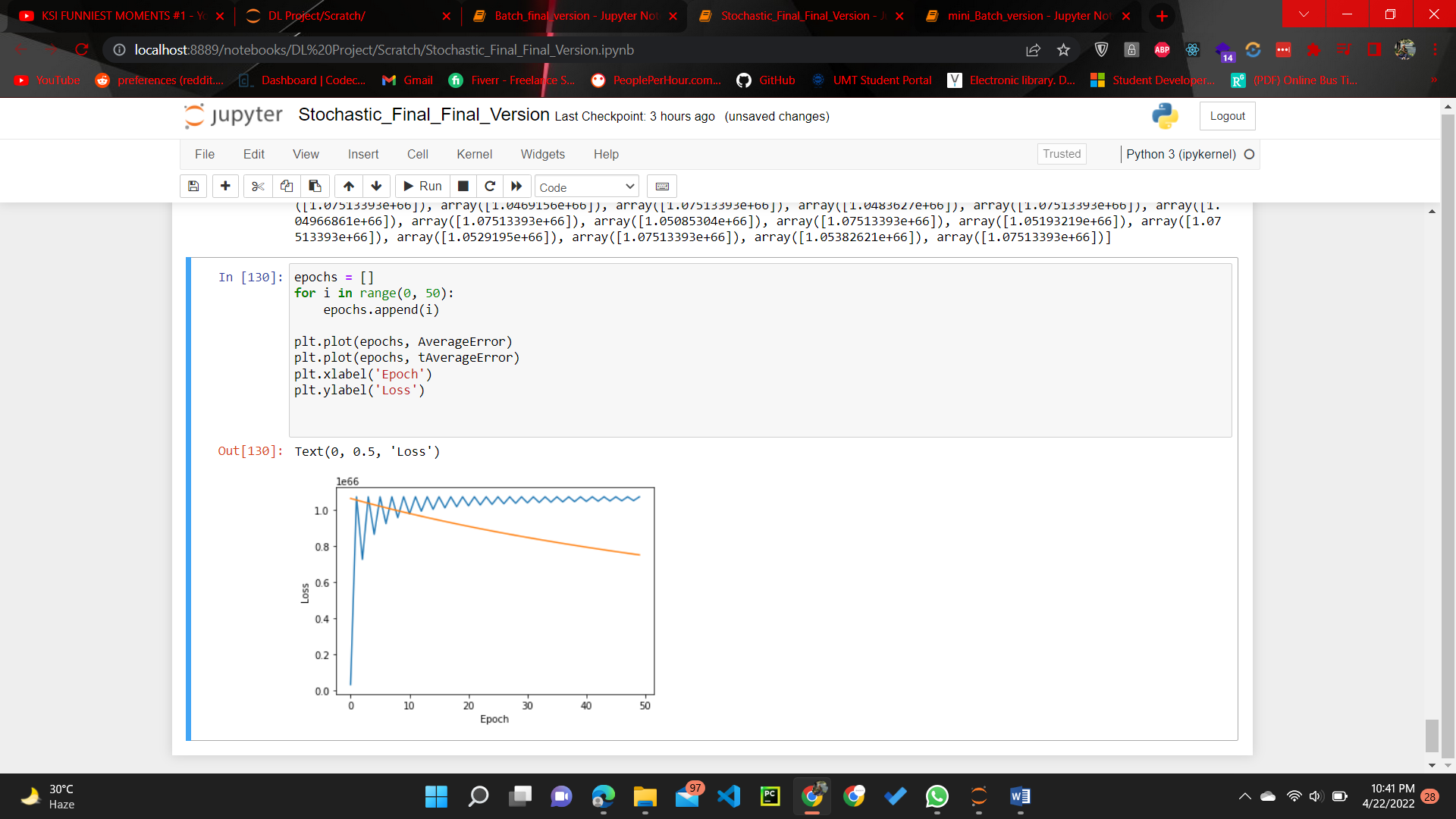




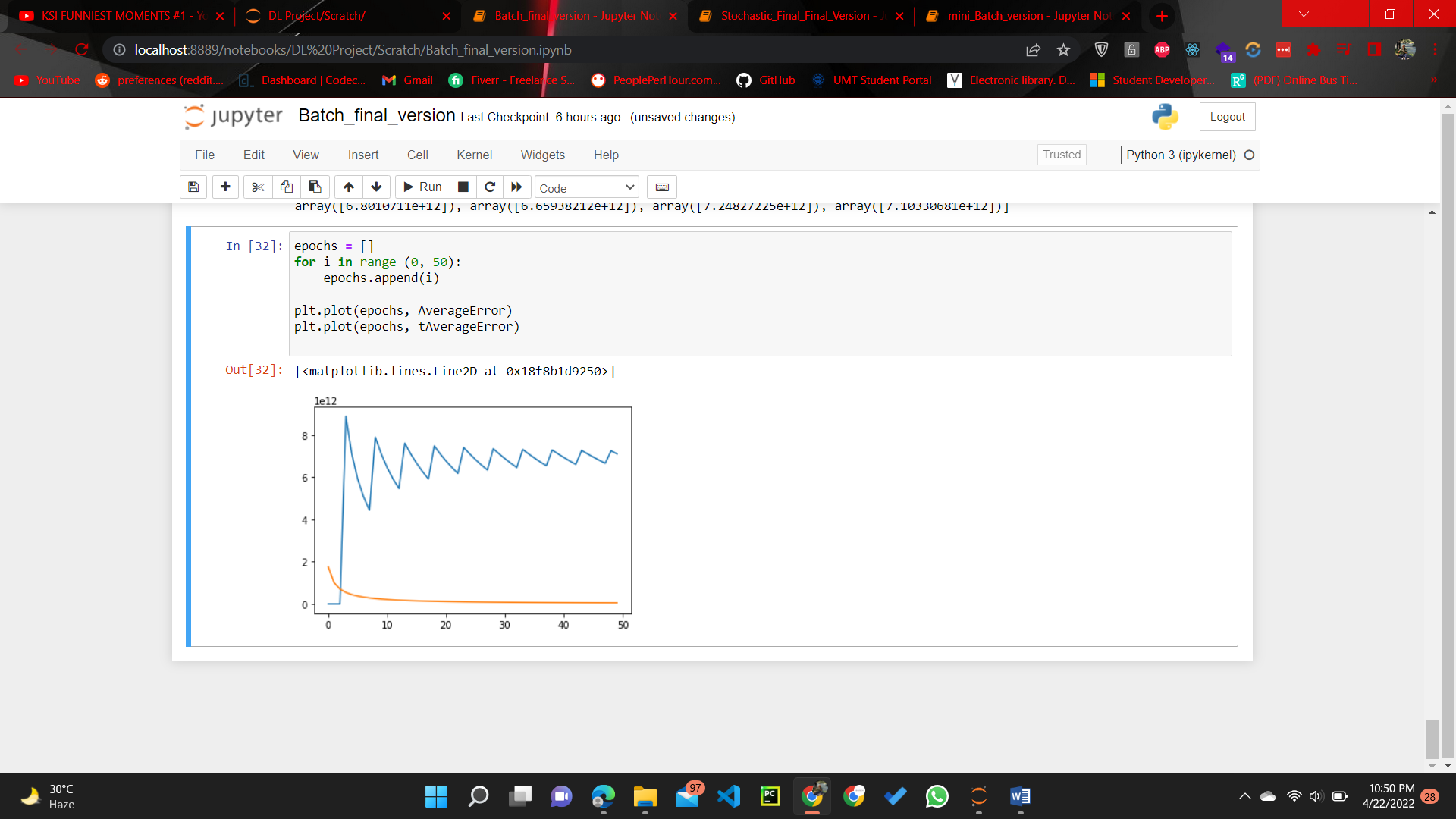


1. **Learning curve [10+10+10]**

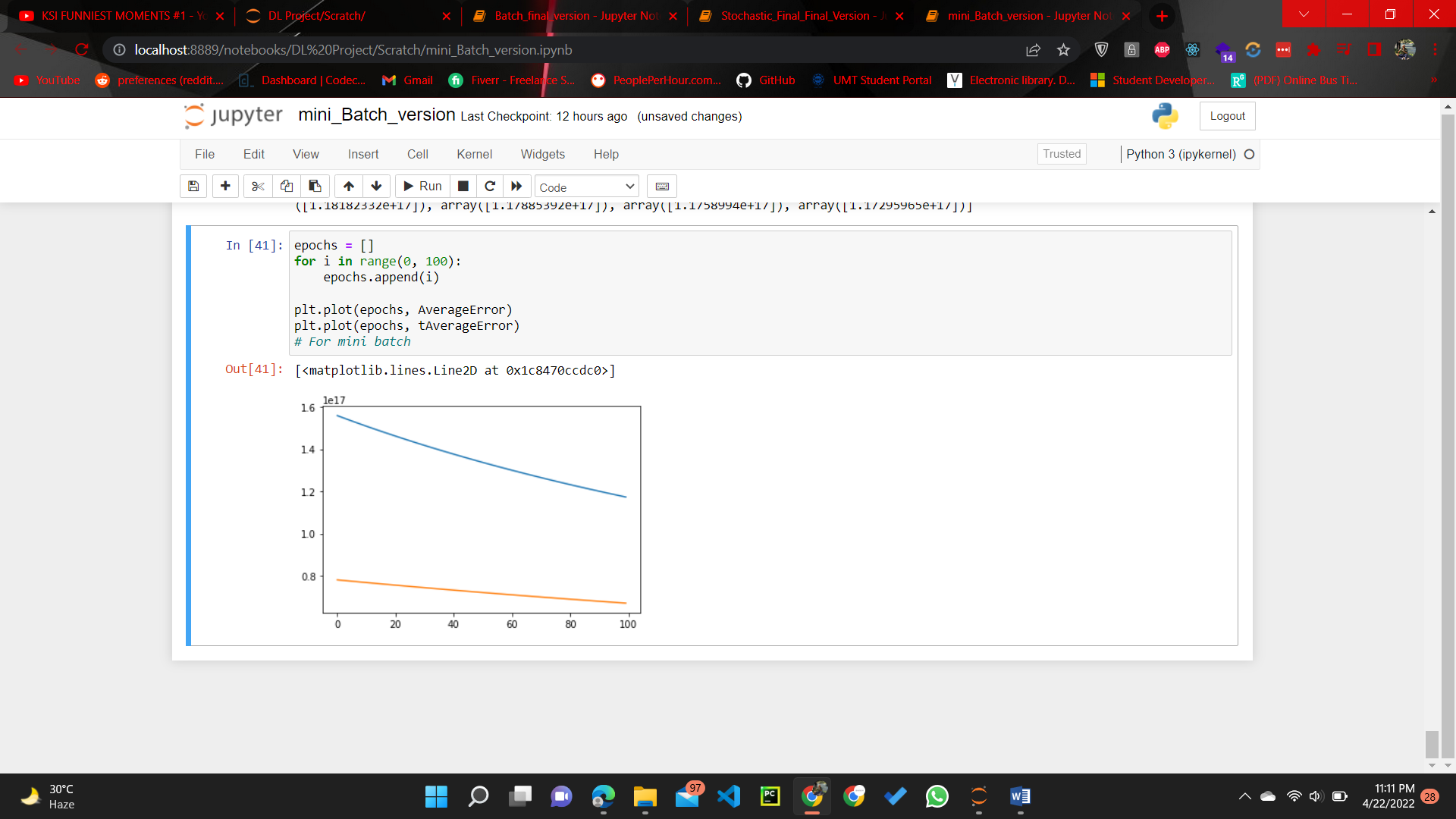
**Stochastic**



**Batch**

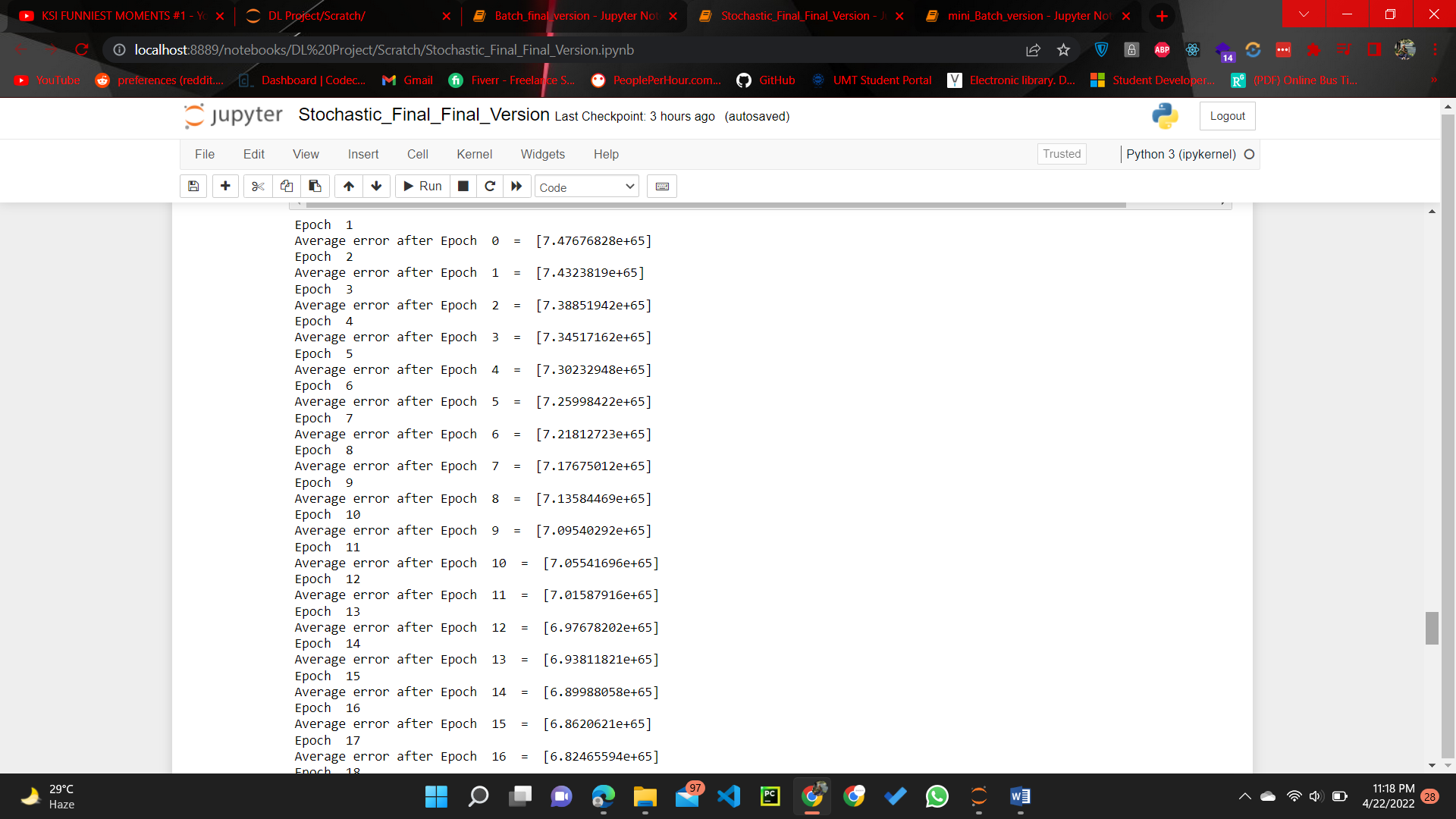


**Mini-Batch**

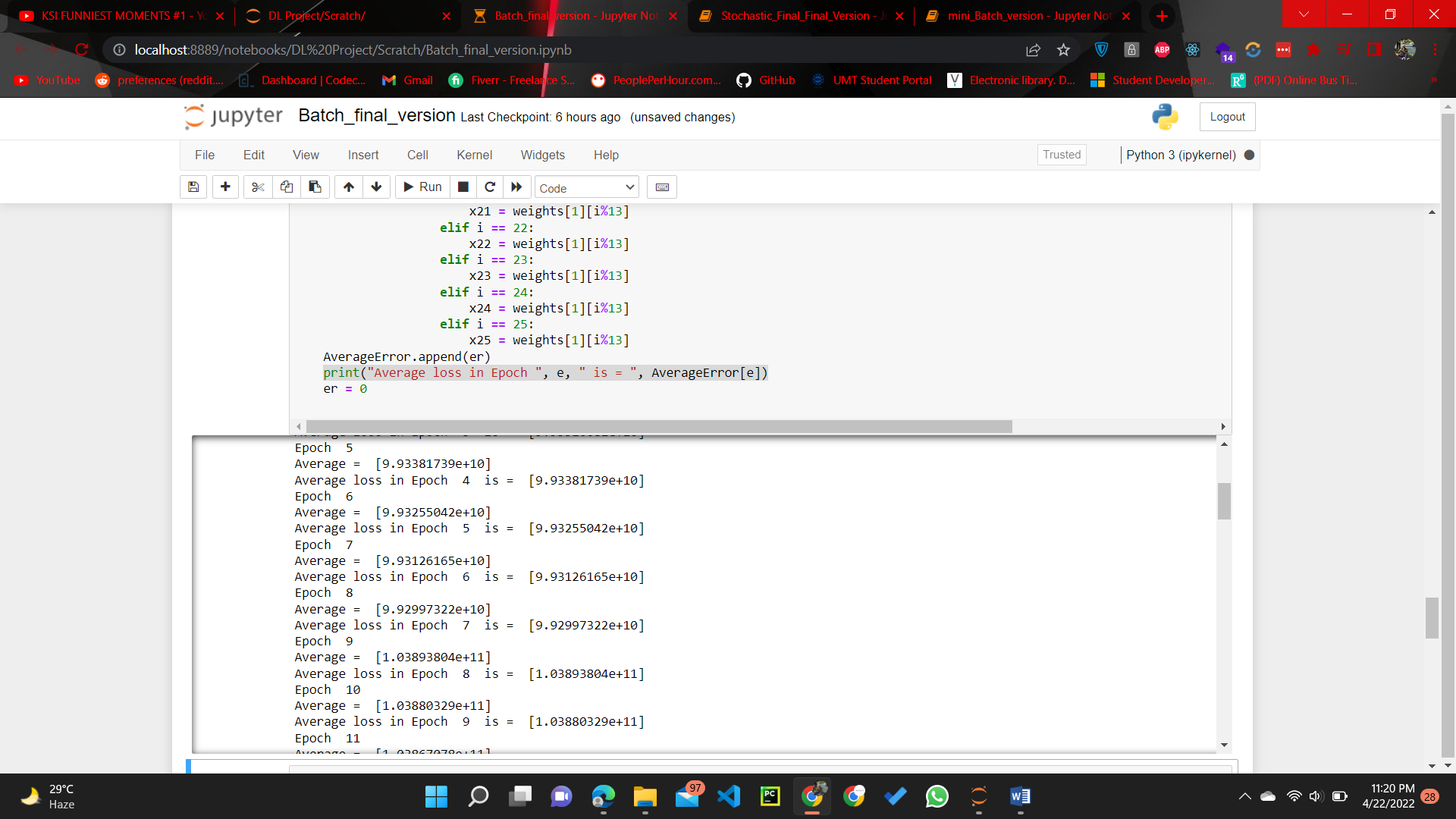


**Textual output:**

Stochastic: -



Batch: -



Mini-Batch: -

