Gradient descent for predicting Exports

By

Saara

1. **Introduction**

Gradient descent is a machine learning optimization process used to discover the best parameters for a model. It works by initially guessing the parameters and then iteratively modifying them to minimize a cost function that quantifies how well the model fits the data. This step is repeated until the procedure converges to a cost function minimum. Gradient descent is a popular method for doing problems including regression, classification, and neural networks.

The beginning point is simply an arbitrary point from which we can assess performance. We'll find the derivative (or slope) from that starting point, and then use a tangent line to see how steep the slope is.

The beginning point is simply an arbitrary point from which we can assess performance. Let us find the derivative (or slope) from that starting point, and then use a tangent line to see how steep the slope is. The slope will affect parameter updates. The slope will be steeper at the start, but as new parameters are generated, it should steadily decrease

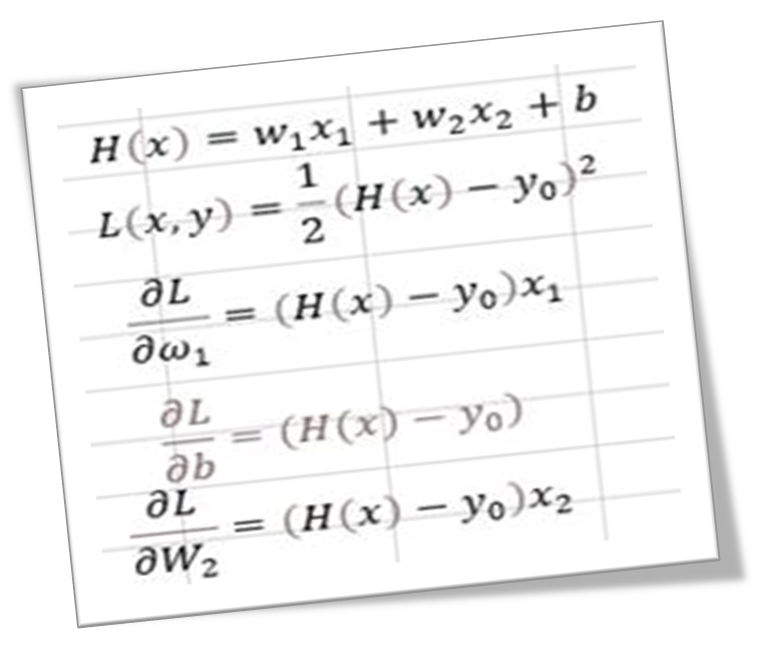
1. **Data Set**

The data set I generated had two variables: export and import. It includes 30 rows. It's a time series.

1. **Steps**

* **Set up Initial parameters:** Setting up of parameters like learning rate, the no. of iterations and the initial values of slope(i.e., s1,s2,sy)
* **X1 and X2:** Using the initial parameters we got x1, x2 and y0.
* **Finding H(x) and L(x,y):** Using initial values give W1 and W2,
* **Calculate the error:** Then calculate the error between the predicted values and the actual values of the variable.
* **Calculate the partial derivatives:** Calculate the error's partial derivatives in regard to the slope and intercept.
* **Update the parameters**
* **Calculate Total Loss**
* **Plotting of graph (**Dependent variable Exports**)**

**Fig 1 : Formula**

****

1. **Graphs**

**Fig 2**

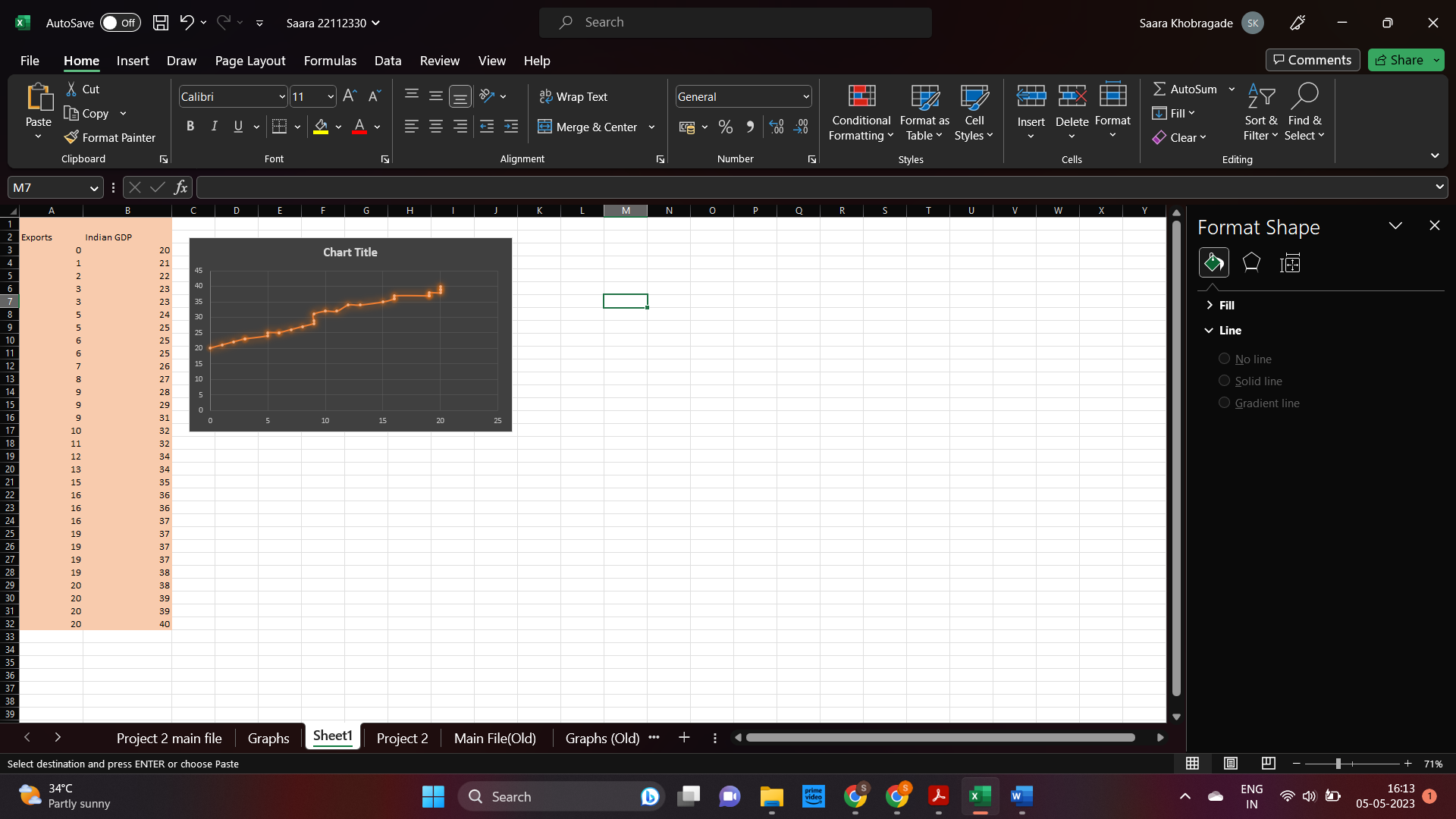
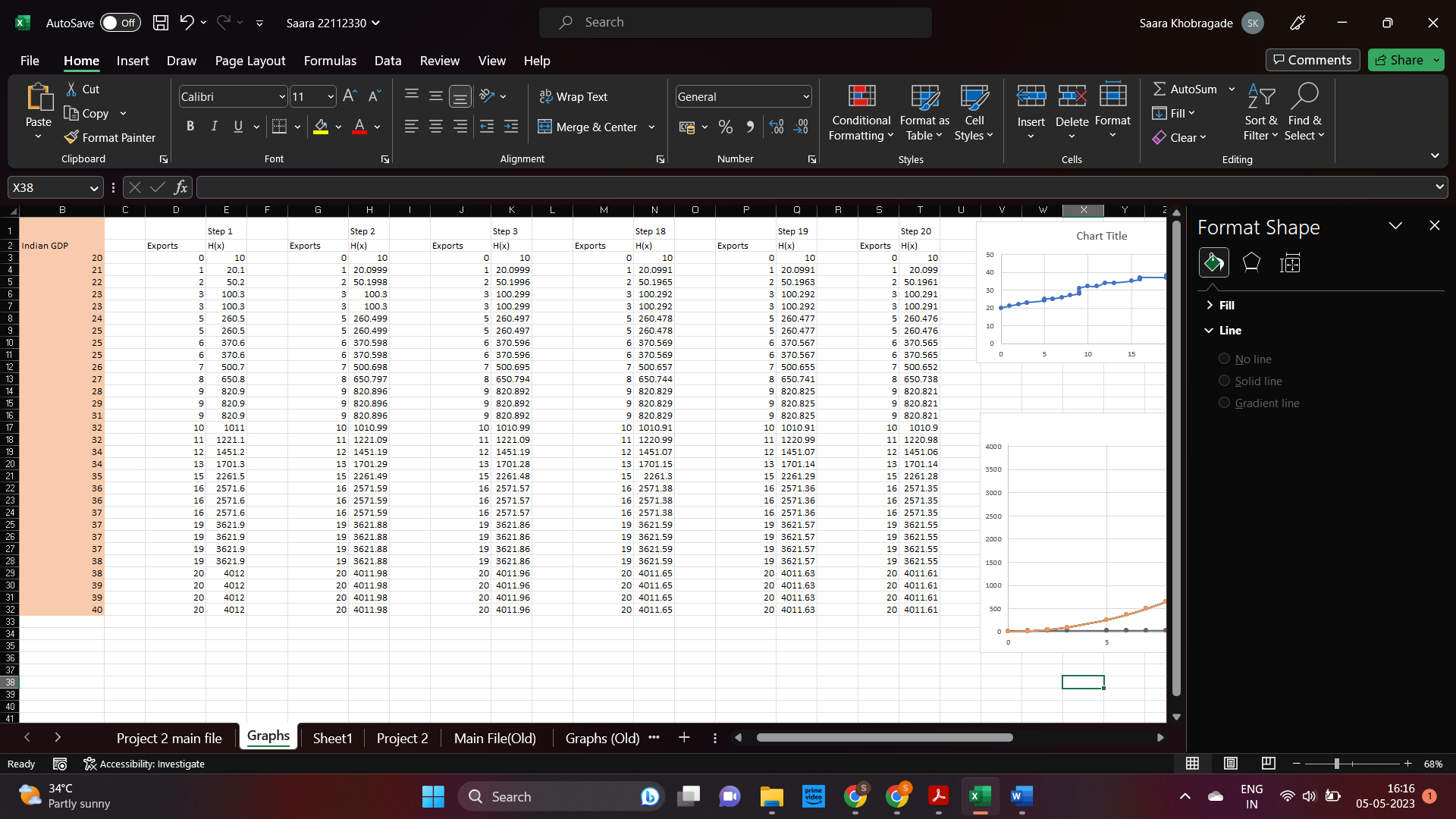
****

Figure 2 shows the initial dataset the variables exports and imports.

**Figure 3&4**



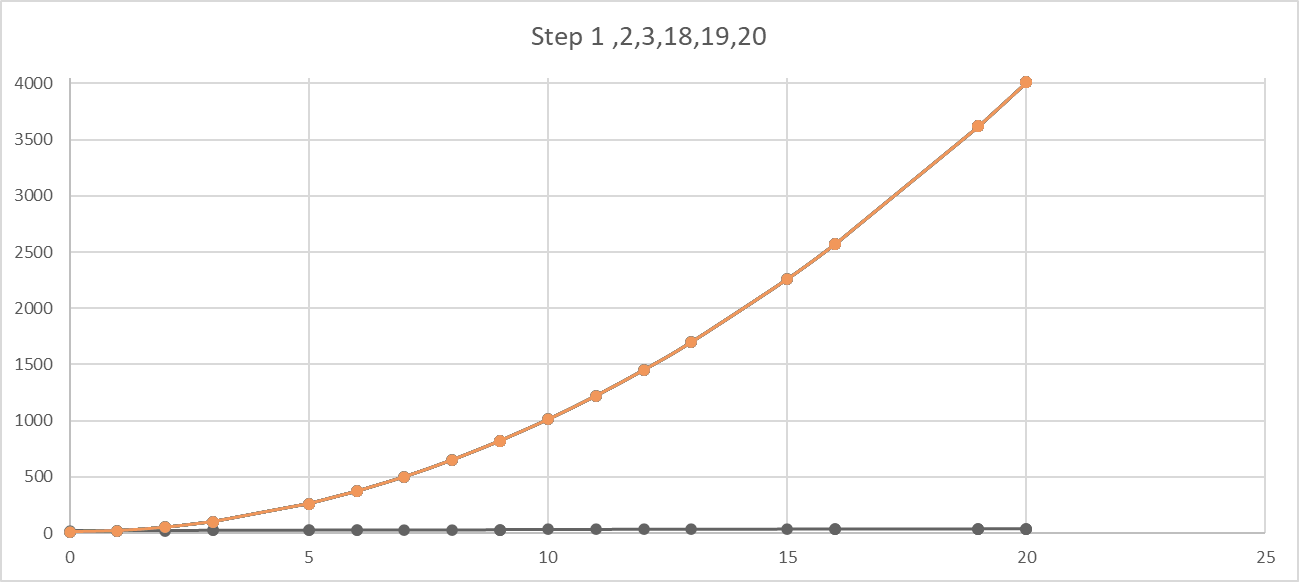


Figure 3&4 shows the data we derived by performing a number of steps. The table shows the initial dataset and result of step 1,2,3,18,19,20. And the following graph is the comparison between the given steps.

**Figure 5**

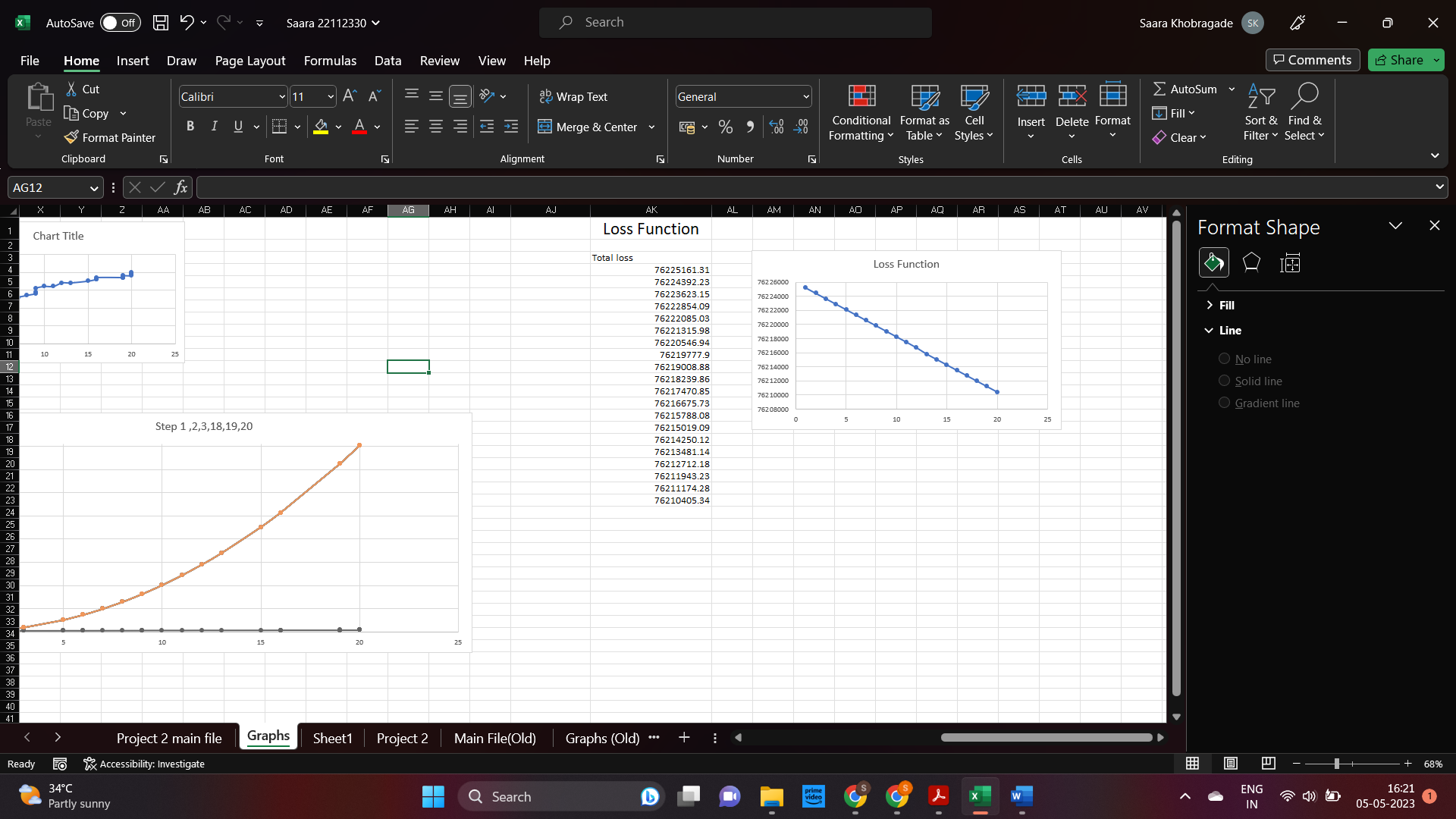


Figure 5 shows the total loss from all the 20 steps done. There is not a noticeable decrease in the loss. There were some issues with the model, Therefore I couldn't Minimize the loss and error.

**Figure 6**

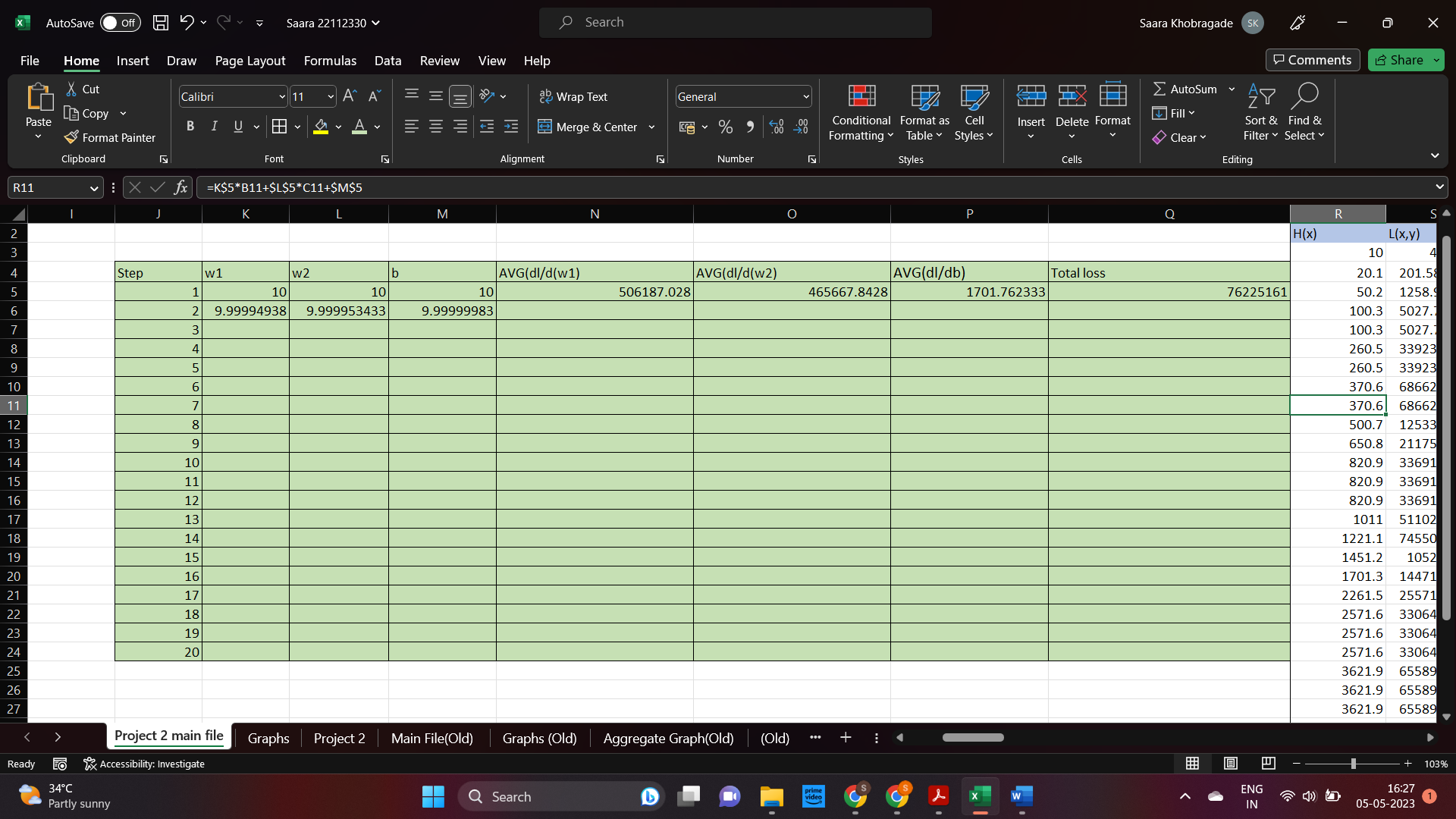


Figure 6 shows the average of (W,1, W2, b)and changes while going from step 1 to 2.

To find the average of the given,

W1 it is [H(x)-y]\*x1

W2 it is [H(x)-y]\*x2

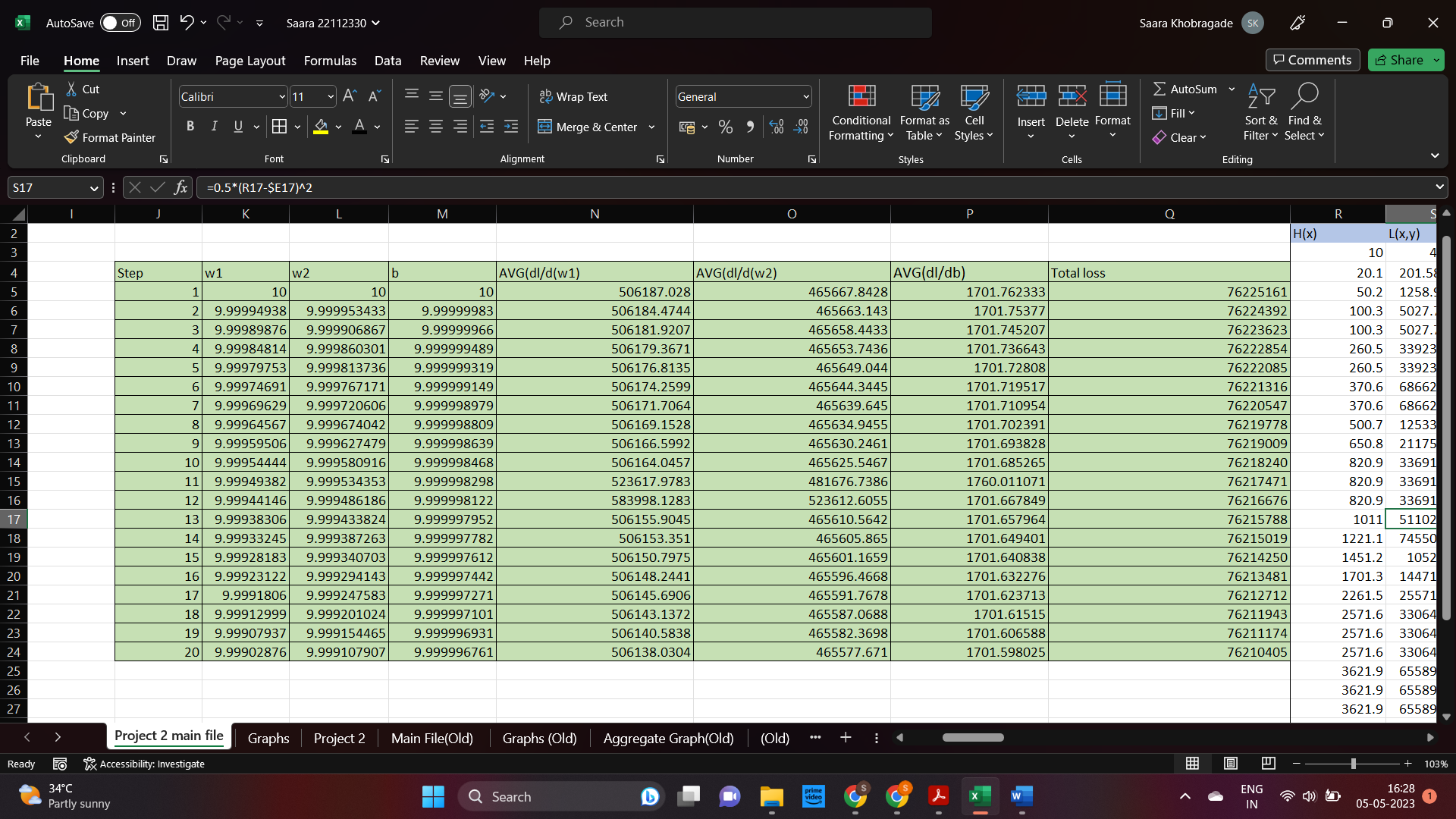
Avg of dl/db is H(X)- Y0

Then we derive the w1, w2 and b for the next step.

[(w1/w2/b)-a]\* avg of w1,w2,b respectively. From there we go to the next step.

Finally we find the total loss of each step.

**Figure 7**



It is repeated till the total loss is minimum. (Although in the project it did not decrease by a considerable level )

1. **Result**

The initial loss value was 76225161 and at the end of step 20 the loss measured was 14754 less than the initial loss (i.e., 76210407). Predicting the future value for export didn’t work out because of some errors.

1. **Conclusion**

Users can take advantage of Excel's many features, including conditional formatting, data visualization, and data analysis tools, to perform gradient descent and obtain insights into the behavior of the model and the optimization procedure.The built-in functions in Excel offer a simple and effective method for structuring and resolving optimisation issues, including gradient descent.

|  |
| --- |