IB Maths HL AA IA

Exploration of Linear Regression and Machine Learning and its Application in Prediction

Page Count: 20

Introduction

Machine learning is a topic in which a computer program can learn from data to execute certain processes. As I further studied this topic through an online course called Machine Learning taught by Andrew Ng from Stanford University, I was intrigued by the integration of mathematics and computer science. As I studied the mathematics behind machine learning, I realized that linear regression was a major part of machine learning. Linear regression is a method of representing a linear relationship between a dependent variable and one or more independent variables through observing a dataset of selected variables. In the course, Professor Ng used house price as an example to show how one can predict the house price based on multiple independent variables such as the size of the house and the number of room it has. Although nothing in the world is dependent only on a single variable, I found the need to understand machine learning from a simplistic viewpoint because it helps me learn the underlying foundation of linear regression and machine learning. Thus, I was drawn to univariate linear regression, a method of representing a linear relationship between a dependent variable and a single independent variable.

As I further studied linear regression and its incorporation in machine learning, I learned that the main objective of linear regression is to determine the optimal values of the coefficients of the regression line, a linear model that best fits a dataset. In other words, I need to find the gradient and the y-intercept of the regression line. I discovered that a very simple way of doing this is to find the coefficients that minimizes the average error between the datapoint of a dataset and the regression line. An error function, $J(\theta_0, \theta_1)$ where θ_0 is the y-intercept and θ_1 is the gradient, is a multivariable function that represents the average error between the datapoint and the regression line. There are many well known error functions such as Mean Squared Error (MSE) and Mean Absolute Error (MAE) that I will be taking it into further details. To minimize the error function, I can use partial derivatives, derivatives of a multivariable function with respect to each variable. Therefore, I will have two derivatives that are with respect to different variable. I can use the gaussian elimination technique to figure out the optimal values of the coefficients. While I can use system of equations to determine the values of θ_0 and θ_1, vectors and matrices are preferred in machine learning. Therefore, I'll be using the gaussian elimination technique.

I was greatly fascinated by this simple yet powerful concept. Therefore, I started to wonder whether it could be applied to my own projects. Therefore, I decided to use the Mean Squared Error and partial derivatives to determine the regression line of my dataset that maps the relationship between my semester exam grades and my course grades. I will then create my own error function to determine the regression line of the dataset. Using these two regression lines, I will predict my IB Math HL semester exam grade based on my current course grade in IB Math HL.

Partial Derivatives

Partial derivatives allow us to derive a multivariable function with respect to a single variable, which is exactly what I need for my calculations. I realized that the definition of derivatives using limits helps showed that this was possible. For example, if I have a multivariable function $f(x, y) = 2xy$, the partial derivative of $f(x, y)$ with respect to x can be represented as

$$\frac{\partial f}{\partial x} = \lim_{h \to 0} \frac{f(x+h, y) - f(x, y)}{h}.$$

$$\frac{\partial f}{\partial x} = \lim_{h \to 0} \frac{2(x+h)y - 2xy}{h} = \lim_{h \to 0} \frac{2xy + 2yh - 2xy}{h} = 2y$$

I notice that the partial derivative of $f(x, y)$ with respect x equals to taking the derivative of $f(x, y)$ while treating y as a constant[1]. With this same mathematical logic, the partial derivative of $f(x, y)$ with respect to y would treat x as a constant and give a partial derivative of $2x$. When addressing the partial derivative of $f(x, y)$ with respect to a variable, it is also acceptable to represent the partial derivative with a subscript under the regression line[2]. For example, $f_x(x, y)$ is equal to $\frac{\partial f}{\partial x}$. For

[1] "Introduction to Partial Derivatives (Article)."

[2] "Second Partial Derivatives (Article)."

second partial derivative with respect x, I can simply append another subscript under the regression line. For example, $f_{xx}(x, y)$ equals to the partial derivative of $\frac{\partial f}{\partial x}$ with respect to x.

As our error function is a multivariable function that is represented in terms of θ_0 and θ_1, partial derivatives will be used to find the derivates with respect to both variables.

<u>Error Functions</u>

As I mentioned before, error function is a representation of the average error between the data points and the regression line. One common way to represent this error is by using the average vertical distance between the data points and the regression line. My regression line can be represented as $f(x) = \theta_0 + \theta_1 x$ where x is the independent variable. The y-component of my dataset that corresponds to the input value of x can be represented as y_x. The vertical distance between a datapoint and the regression line at a particular x value would be equal to $f(x) - y_x$. Thus, I figured that the average vertical distance between the regression line and the datapoint would be $\frac{\sum_{i=1}^{n} f(x_i) - y_i}{n}$ where x_i is the x value at the i^{th} index of the dataset, y_i is the y value at the i^{th} index of the dataset, and n is the total number of datapoint in a dataset. However, I realized that when some distances are negative while some others are positive, it poses a significant inaccuracy to my process of linear regression. For example, if there are two data points and the vertical distance from each points to the regression line was -1 and 1, the formula above gives an average distance of 0 which is in fact a misleading representation. Hence, I realized that these errors values need to be manipulated so that the values are either all positive or all negative. Hence, I came across Mean Absolute Error which uses the absolute value of the distance: $\frac{\sum_{i=1}^{n} |f(x_i) - y_i|}{n}$. To find the values of θ_1 and θ_0, I must take the partial derivatives of it with respect to both θ_1 and θ_0. However, I realized that by taking the absolute value of the vertical distance, the error function will not be differentiable at certain points. Therefore, for the error function to be differentiable for any point, I decided to use the square of the vertical distance between the datapoint and the regression line. This gives me the Mean Squared Error, as mentioned before. The equation is as following: $\frac{\sum_{i=1}^{n}(f(x_i) - y_i)^2}{n}$. In my case, the only unknowns are the two coefficients from the regression line. Thus, I can have a function called $J(\theta_0, \theta_1)$ that represents the error function in terms of these two coefficients. Therefore, finding the partial derivatives with respect to these coefficients will give me the optimal values of θ_0 and θ_1 that minimizes the error function. The local minimum is not always the global minimum. However, to solve for θ_0 and θ_1, I need two equations in terms of these two variables. Partial derivatives with respect to each variable is a simple method that allows me to obtain two equations to solve for these two coefficients.

Second Partial Derivative Test

Even if I had found the values of the coefficients of the regression line using partial derivatives, it does not guarantee that at those values, the error function will be at its local minimum. In fact, it can also be at a local maximum or a saddle point, which is a critical point that is neither a local minimum nor a local maximum[3]. I came across the second partial derivative test that determines if a stable point is at its local minimum, maximum, or a saddle point. I will not derive nor prove this test as it is not relevant to my aim. If I have an equation $H = f_{xx}(a, b)f_{yy}(a, b) - f_{xy}(a, b)$ where (a, b) is a coordinate on function $f(x, y)$, when $H > 0$ and $f_{xx}(a, b) > 0$ or $f_{yy}(a, b) > 0$, there is a local minimum point at (a, b)[4]. There are other conditions

---

[3] "Maxima, Minima, and Saddle Points (Article)."

[4] "Second Partial Derivative Test (Article)."

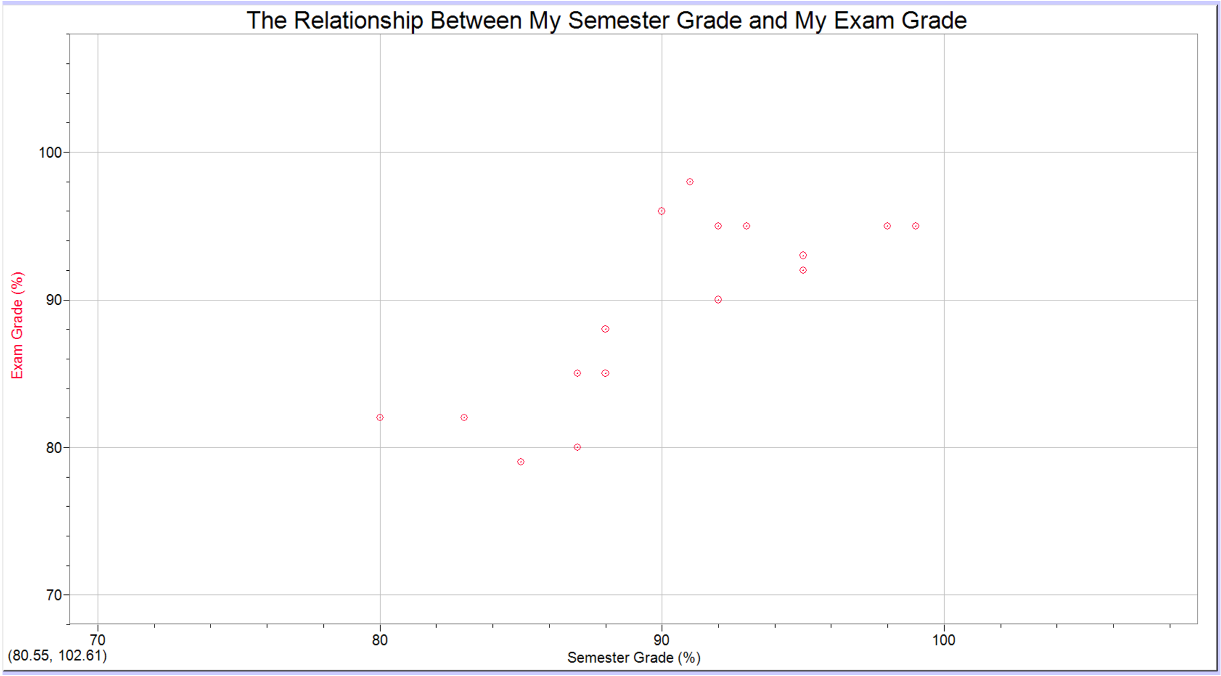
that check for the local maximum and the saddle point. However, I will not go through those as they are not relevant to my aim.

Dataset

The Relationship Between My Semester Grade to My Exam Grade

Semester Grade (%)	Exam Grade (%)
95	92
80	82
88	85
93	95
91	98
98	95
99	95
85	79
83	82
90	96
92	90
98	95
88	88
87	80
87	85
95	93
92	95

Scatter Plot of the Semester and Exam Grades Dataset With Estimated Line of Best Fit



I'll be using partial derivatives and mean squared error to create a regression line. After, I'll create my own error function and determine a new regression line. Using these, I'll predict my IB Math HL semester exam grade using my current IB Math HL course grade as my input.

Variables and Equations

$f(x_i) = \theta_0 + \theta_1 x_i$ is the regression line of the dataset.

x_i is the semester grade in percentage grade at the i^{th} index in the dataset.

y_i is the exam grade in percentage grade at the i^{th} index in the dataset.

$f(x_i)$ is a function of x that represent the predicted exam grade in percentage.

θ_0 and θ_1 are coefficients from the regression line that I need to find.

$J(\theta_0, \theta_1)$ is the error function, a function of both θ_0 and θ_1, that represents the average error between the regression line and the data points.

Calculation of the Minimum Value of MSE

As I have graphed the dataset, I now have to find the line of best-fit for this particular data, using the MSE. For better visualization, I will expand $J(\theta_0, \theta_1)$.

$$J(\theta_0, \theta_1) = \frac{\sum_{i=1}^{17} (f(x_i) - y_i)^2}{17}$$

Expanded Form:

$$\begin{aligned} J(\theta_0, \theta_1) &= \frac{1}{17}\{(f(x_1) - y_1)^2 + (f(x_2) - y_2)^2 + \ldots + (f(x_{17}) - y_{17})^2\} \\ &= \frac{1}{17}\{(\theta_0 + \theta_1 x_1 - y_1)^2 + (\theta_0 + \theta_1 x_2 - y_2)^2 + \ldots + (\theta_0 + \theta_1 x_{17} - y_{17})^2\} \\ &= \frac{1}{17}\{(\theta_0 + 95\theta_1 - 92)^2 + (\theta_0 + 80\theta_1 - 82)^2 + \ldots + (\theta_0 + 92\theta_1 - 95)^2\} \end{aligned}$$

For $J(\theta_0, \theta_1)$ to be minimized, the optimal coefficients of the regression line would be when the partial derivatives of $J(\theta_0, \theta_1)$ are equal to 0. $\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0} = 0$ and $\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1} = 0$.

$$\begin{aligned} \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0} &= \frac{1}{17}\{2(\theta_0 + 95\theta_1 - 92) + 2(\theta_0 + 80\theta_1 - 82) + \ldots + 2(\theta_0 + 92\theta_1 - 95)\} \\ &= \frac{2(17\theta_0 + 1541\theta_1 - 1525)}{17} \end{aligned}$$

$$\begin{aligned} \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1} &= \frac{1}{17}\{2(\theta_0 + 95\theta_1 - 92) \times 95 + 2(\theta_0 + 80\theta_1 - 82) \times 80 + \ldots + 2(\theta_0 + 92\theta_1 - 95) \times 92\} \\ &= \frac{1}{17}\sum_{i=1}^{17} 2(\theta_0 + \theta_1 x_i - y_i) \times x_i = \frac{1}{17}\sum_{i=1}^{17} 2(h(x_i) - y_i) \times x_i \\ &= \frac{2(1541\theta_0 + 140157\theta_1 - 138673)}{17} \end{aligned}$$

As both of the partial derivatives are equal to 0, I'm able to use two different equations to solve for θ_0 and θ_1.

$$\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0} = \frac{2(17\theta_0 + 1541\theta_1 - 1525)}{17} = 0$$

$$17\theta_0 + 1541\theta_1 - 1525 = 0$$

$$17\theta_0 + 1541\theta_1 = 1525$$

$$\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1} = \frac{2(1541\theta_0 + 140157\theta_1 - 138673)}{17} = 0$$

$$1541\theta_0 + 140157\theta_1 - 138673 = 0$$

$$1541\theta_0 + 140157\theta_1 = 138673$$

System of Equation:

$$17\theta_0 + 1541\theta_1 = 1525$$

$$1541\theta_0 + 140157\theta_1 = 138673$$

Gaussian Elimination is a different method of solving a system of linear equations. This process uses matrices to represent the linear equations. Each column represents the coefficient of a variable while the last column represents the output of the linear equations. When put together, each row represents a unique linear equation. In my case, the matrix to be solved can be represented as $\begin{bmatrix} 17 & 1541 & 1525 \\ 1541 & 140157 & 138673 \end{bmatrix}$. My goal is to transform this matrix so that in the first row, the coefficient of θ_0 is 1 while the coefficient of θ_1 is 0. On the other hand, in the second row, I'd have to transform the matrix so that the coefficient of θ_0 is 0 and the coefficient of θ_1 is 1. This particular form is called the reduced row echelon form. Thus, the linear equations would be transformed into an equation such as $0\theta_0 + 1\theta_1 = a$ and $1\theta_0 + 0\theta_1 = b$ where $a, b \in \mathbb{R}$. I noticed that in order to do this, there are two operations that I must execute. The first one is to add rows with each other and the second is to multiply the rows by a constant. These operations are called the elementary row operations. In fact, I figured that this is the exact same operations we do with a system of linear equations, except that the gaussian elimination algorithm is done in the perspective of linear algebra.

Gaussian Elimination Process:

First Row Operation: Multiply the 1st row by $\dfrac{1}{17}$

$$\begin{bmatrix} 17 & 1541 & 1525 \\ 1541 & 140157 & 138673 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & \frac{1541}{17} & \frac{1525}{17} \\ 1541 & 140157 & 138673 \end{bmatrix}$$

Second Row Operation: Add the 1st row multiplied by -1541 to the 2nd row

$$\begin{bmatrix} 1 & \frac{1541}{17} & \frac{1525}{17} \\ 1541 & 140157 & 138673 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & \frac{1541}{17} & \frac{1525}{17} \\ 0 & \frac{7988}{17} & \frac{7416}{17} \end{bmatrix}$$

Third Row Operation: Multiply 2nd row with $\dfrac{17}{7988}$

$$\begin{bmatrix} 1 & \frac{1541}{17} & \frac{1525}{17} \\ 0 & \frac{7988}{17} & \frac{7416}{17} \end{bmatrix} \rightarrow \begin{bmatrix} 1 & \frac{1541}{17} & \frac{1525}{17} \\ 0 & 1 & \frac{1854}{1997} \end{bmatrix}$$

Fourth Row Operation: Add the 2nd row multiplied by $-\dfrac{1541}{17}$ to the 1st row

$$\begin{bmatrix} 1 & \frac{1541}{17} & \frac{1525}{17} \\ 0 & 1 & \frac{1854}{1997} \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & \frac{11083}{1997} \\ 0 & 1 & \frac{1854}{1997} \end{bmatrix}$$

$$\theta_1 = \frac{1854}{1997} \approx 0.928$$

$$\theta_0 = \frac{11083}{1997} \approx 5.550$$

$$\therefore f(x) = 0.928x + 5.550$$

Although I've obtained a value for the coefficients, the partial derivatives says that the rate of change in respect to each coefficients is 0. Thus, I need to further check whether these coefficients are the values of the error function at local minimum or local maximum. I can use the second partial derivatives test that was explained previously to confirm whether the values of the coefficient is at the local minimum.

Second Partial Derivatives of $\dfrac{\partial J(\theta_0, \theta_1)}{\partial \theta_0}$

$$J_{\theta_0\theta_0}(5.550{,}0.928) = \frac{2(17)}{17} = 2$$

$$J_{\theta_0\theta_1}(5.550{,}0.928) = \frac{2(1541)}{17} \approx 181.294$$

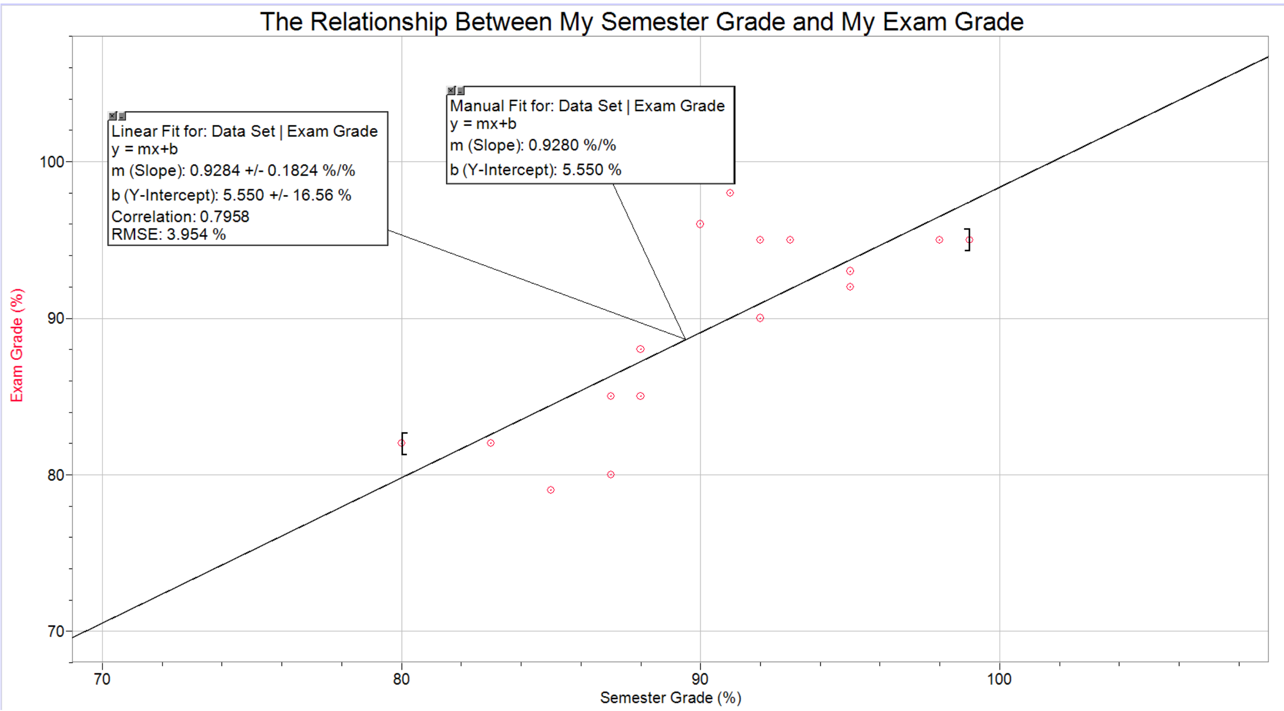
Second Partial Derivatives of $\dfrac{\partial J(\theta_0, \theta_1)}{\partial \theta_1}$

$$J_{\theta_1\theta_0}(5.550{,}0.928) = \frac{2(1541)}{17} \approx 181.294$$

$$J_{\theta_1\theta_1}(5.550{,}0.928) = \frac{2(140157)}{17} \approx 16489.059$$

As I get $H = 2 \times 16489.059 - (181.294)^2 \approx 110.604$ which is $H > 0$, $(5.550{,}0.928)$ is a critical point. Since $J_{\theta_0\theta_0}(5.550{,}0.928) > 0$, the values of the coefficients are at the local minimum due to the Second partial derivatives test.

Validating the Regression Line



Now that I know the value of θ_0 and θ_1 are 5.550 and 0.928 respectively, the regression line $f(x)$ is now complete: $f(x) = 5.550 + 0.928x$. If I graph this linear model, it is shown as the graph below.

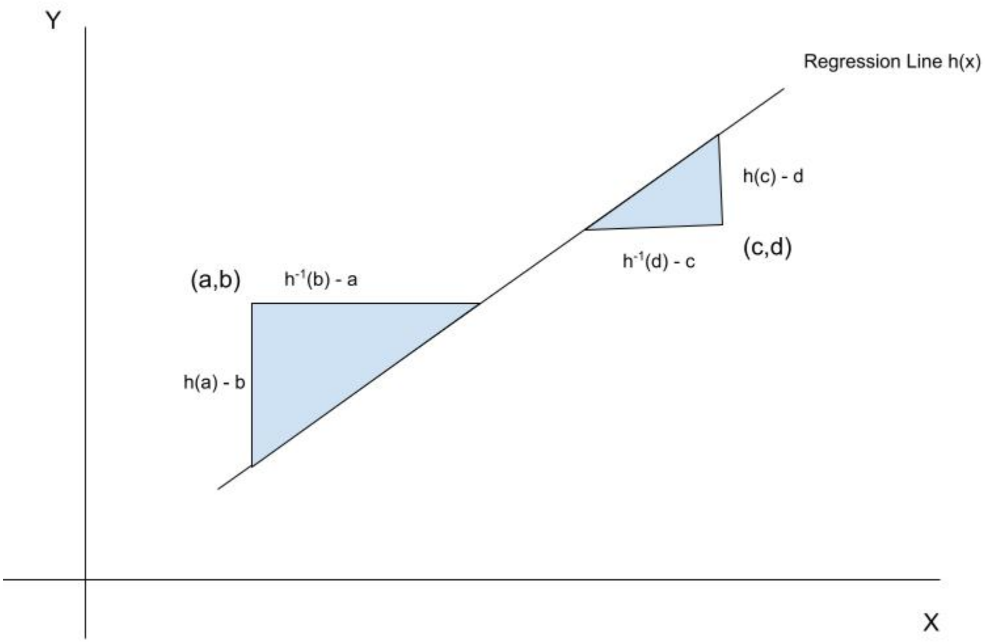
As the graph shows, the linear model that I've found using Means Squared Error seems to be extremely close to what the graphing software (Logger Pro) has generated. The y-intercepts seems to be identical and the gradient of the regression line is smaller than that of the line of best-fit generated by Logger Pro. However, the smaller gradient is likely due to the fact that I rounded the value of the gradient only up to its third decimal place. Therefore, I would say that my attempt to find the linear model of the relationship between semester grades and exam grades worked out to be significantly accurately. Furthermore, one interesting aspect I found about the regression line was that when x is equal to the mean of the semester grade (approximately 90%), the regression line outputs the mean value of the exam grade (approximately 89%).

My Custom Error Function

For my regression line using my custom error function, I will represent the regression line as $h(x)$ to prevent confusion with $f(x)$ that was used for MSE error function.

My other aim is to create my own custom error function and apply it to my exam grades dataset. Therefore, I have tried many different methods. At first, I thought of finding the vertical distance from the mean exam grade to the regression line. However, the result came out to be disappointing with an equation of $h(x) = \bar{y}$. Then, I thought of using the area under the curve. I realized that the area under the equation $h(x) = \bar{y}$ should essentially be equal to the area under the regression line. Even after these few trials, I kept getting horizontal lines that had no correlation with my dataset. As I looked over my previous failures, I realized that I made a mistake of not using my data points individually but as a whole. Thus, my error function was essentially comparing the regression line with a single datapoint.

IB Math HL teacher always says to start from the basics whenever we meet a problem. Therefore, I started to look at the simple things. One of them was $h(x_i) - y_i$, which represents the vertical distance from the regression line and the data points at a specific value of x. I then wondered about the horizontal difference between the regression line and the x-component of a datapoint. I wondered whether the average squared horizontal distance from the data points to the regression line would be the same as the vertical distance between them. It turns out that they aren't. Using the average horizontal distance between the regression line and x-component of the data points, the regression line was $h(x) = -43.184 + 1.466x$. Due to this difference, I believed that incorporating both the vertical and the horizontal distance in my error function will significantly increase the correlation of my regression line. Hence, I decided to use the average area bounded by the regression line, horizontal and the vertical segment between the regression line and the data points as shown below.



Unlike Mean Squared Error, I noticed that squaring the area is not necessary. There are three possible places where a datapoint can be located. The datapoint can be located above the regression line, on the regression line or below the regression line. When it is on the regression line, the area bounded by the vertical and horizontal segment between the regression line and the datapoint and the regression line it self is 0. However, when the datapoint is not on the regression line, the area is always negative. For example, for point (a, b) in the image above is located above the regression line, I noticed $h^{-1}(b) - a > 0$ and $h(a) - b < 0$. On the other hand, when the point is located below the regression line, such as (c, d) from the image above, I noticed $h^{-1}(d) - c < 0$ and

$h(c) - d > 0$. Therefore, I observed that the bounded area is always negative. I found that this method is only viable for when I am certain that the gradient will be greater than 0 ($\theta_1 > 0$) which I was from looking at the scattering plot shown previously. Thus, I came up with an error function $J(\theta_0, \theta_1) = \dfrac{\sum_{i=1}^{n} \frac{-(h^{-1}(y_i) - x_i)(h(x_i) - y_i)}{2}}{n}$. I have calculated the inverse of the regression line $h(x)$ where $h^{-1}(y) = \dfrac{y - \theta_0}{\theta_1}$.

However, I was hindered by the massive amount of complex computation that was required to expand this to a more mathematically friendly way. Therefore, using a web-application called Jupyter Notebook, I learned to use pandas, a python library for manipulating datas, and SymPy, another python library to carry out execute complex mathematical tasks.

Custom Error Function Calculations on Jupyter Notebook

Expanding out $J(\theta_0, \theta_1)$

```
In [342]: o,z = sy.symbols('o z', real=True)
          f = z + o*df['x']
          h = (df['y'] - z)/o
          c = -((f-df['y'])*(h-df['x'])/2)

In [343]: x = c

In [344]: x

Out[344]: 0     -(-95 + (92 - z)/o)*(95*o + z - 92)/2
          1     -(-80 + (82 - z)/o)*(80*o + z - 82)/2
          2     -(-88 + (85 - z)/o)*(88*o + z - 85)/2
          3     -(-93 + (95 - z)/o)*(93*o + z - 95)/2
          4     -(-91 + (98 - z)/o)*(91*o + z - 98)/2
          5     -(-98 + (95 - z)/o)*(98*o + z - 95)/2
          6     -(-99 + (95 - z)/o)*(99*o + z - 95)/2
          7     -(-85 + (79 - z)/o)*(85*o + z - 79)/2
          8     -(-83 + (82 - z)/o)*(83*o + z - 82)/2
          9     -(-90 + (96 - z)/o)*(90*o + z - 96)/2
          10    -(-92 + (90 - z)/o)*(92*o + z - 90)/2
          11    -(-98 + (95 - z)/o)*(98*o + z - 95)/2
          12    -(-88 + (88 - z)/o)*(88*o + z - 88)/2
          13    -(-87 + (80 - z)/o)*(87*o + z - 80)/2
          14    -(-87 + (85 - z)/o)*(87*o + z - 85)/2
          15    -(-95 + (93 - z)/o)*(95*o + z - 93)/2
          16    -(-92 + (95 - z)/o)*(92*o + z - 95)/2
          dtype: object

In [395]: j = (x.sum())/17

In [396]: j
```

Out[396]:

$$-\frac{\left(-99 + \frac{95-z}{o}\right)(99o + z - 95)}{34} - \frac{\left(-98 + \frac{95-z}{o}\right)(98o + z - 95)}{17} - \frac{\left(-95 + \frac{92-z}{o}\right)(95o + z - 92)}{34} - \frac{\left(-95 + \frac{93-z}{o}\right)(95o + z - 93)}{34}$$
$$- \frac{\left(-93 + \frac{95-z}{o}\right)(93o + z - 95)}{34} - \frac{\left(-92 + \frac{90-z}{o}\right)(92o + z - 90)}{34} - \frac{\left(-92 + \frac{95-z}{o}\right)(92o + z - 95)}{34} - \frac{\left(-91 + \frac{98-z}{o}\right)(91o + z - 98)}{34}$$
$$- \frac{\left(-90 + \frac{96-z}{o}\right)(90o + z - 96)}{34} - \frac{\left(-88 + \frac{85-z}{o}\right)(88o + z - 85)}{34} - \frac{\left(-88 + \frac{88-z}{o}\right)(88o + z - 88)}{34} - \frac{\left(-87 + \frac{80-z}{o}\right)(87o + z - 80)}{34}$$
$$- \frac{\left(-87 + \frac{85-z}{o}\right)(87o + z - 85)}{34} - \frac{\left(-85 + \frac{79-z}{o}\right)(85o + z - 79)}{34} - \frac{\left(-83 + \frac{82-z}{o}\right)(83o + z - 82)}{34} - \frac{\left(-80 + \frac{82-z}{o}\right)(80o + z - 82)}{34}$$

$j = J(\theta_0, \theta_1),\ \ o\ = \theta_1, z\ = \theta_0,\ \ df['x'] =$ the data value of semester grade,

$df['y'] =$ the data value of the exam grade, j is the error function.

In order to minimize the error function, I must differentiate is with respect to both θ_0 and θ_1 and solve for these coefficients when the derivative equals 0.

Partial Derivatives of $J(\theta_0, \theta_1)$ with respect to each coefficients

```
In [347]: do = sy.diff(j, o)
```

```
In [348]: do
```

Out[348]:

$$\frac{140157}{34} - \frac{5(79-z)}{2o} - \frac{87(80-z)}{34o} - \frac{163(82-z)}{34o} - \frac{175(85-z)}{34o} - \frac{44(88-z)}{17o} - \frac{46(90-z)}{17o} - \frac{95(92-z)}{34o} - \frac{95(93-z)}{34o} - \frac{240(95-z)}{17o} - \frac{45(96-z)}{17o} - \frac{91(98-z)}{34o} + \frac{(79-z)(85o+z-79)}{34o^2} + \frac{(80-z)(87o+z-80)}{34o^2} + \frac{(82-z)(80o+z-82)}{34o^2} + \frac{(82-z)(83o+z-82)}{34o^2} + \frac{(85-z)(87o+z-85)}{34o^2} + \frac{(85-z)(88o+z-85)}{34o^2} + \frac{(88-z)(88o+z-88)}{34o^2} + \frac{(90-z)(92o+z-90)}{34o^2} + \frac{(92-z)(95o+z-92)}{34o^2} + \frac{(93-z)(95o+z-93)}{34o^2} + \frac{(95-z)(92o+z-95)}{34o^2} + \frac{(95-z)(93o+z-95)}{34o^2} + \frac{(95-z)(98o+z-95)}{17o^2} + \frac{(95-z)(99o+z-95)}{34o^2} + \frac{(96-z)(90o+z-96)}{34o^2} + \frac{(98-z)(91o+z-98)}{34o^2}$$

```
In [349]: dz = sy.diff(j, z)
```

```
In [350]: dz
```

Out[350]:

$$\frac{1541}{34} - \frac{79-z}{34o} - \frac{80-z}{34o} - \frac{82-z}{17o} - \frac{85-z}{17o} - \frac{88-z}{34o} - \frac{90-z}{34o} - \frac{92-z}{34o} - \frac{93-z}{34o} - \frac{5(95-z)}{34o} - \frac{96-z}{34o} - \frac{98-z}{34o} + \frac{80o+z-82}{34o} + \frac{83o+z-82}{34o} + \frac{85o+z-79}{34o} + \frac{87o+z-85}{34o} + \frac{87o+z-80}{34o} + \frac{88o+z-88}{34o} + \frac{88o+z-85}{34o} + \frac{90o+z-96}{34o} + \frac{91o+z-98}{34o} + \frac{92o+z-95}{34o} + \frac{92o+z-90}{34o} + \frac{93o+z-95}{34o} + \frac{95o+z-93}{34o} + \frac{95o+z-92}{34o} + \frac{98o+z-95}{17o} + \frac{99o+z-95}{34o}$$

$$dz = \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0} \qquad do = \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1}$$

As the partial derivatives are found, these derivatives must be equated to 0 to find the local minima. Fortunately, the SymPy library can help me find out the coefficients through their way of solving a non-linear system of equations.

```
In [398]: an = sy.nonlinsolve([do, dz], (o, z))
```

```
In [399]: an
```

Out[399]:

$$\left\{\left(-\frac{3\sqrt{603094}}{1997}, \frac{1525}{17} + \frac{4623\sqrt{603094}}{33949}\right), \left(\frac{3\sqrt{603094}}{1997}, \frac{1525}{17} - \frac{4623\sqrt{603094}}{33949}\right)\right\}$$

However, I obtained two possible values of these coefficients. Therefore, to further confirm which solution is the correct solution, the second partial derivatives test will tell me which one is the local minimum.

```
In [440]: a = an.args[0]
          a
```

Out[440]: $\left(-\frac{3\sqrt{603094}}{1997}, \frac{1525}{17} + \frac{4623\sqrt{603094}}{33949}\right)$

```
In [433]: doz = sy.diff(do, z)
          doo = sy.diff(do,o)
          dzo = sy.diff(dz,o)
          dzz = sy.diff(dz, z)

In [434]: doz = sy.N(doz.subs([(o, a[0]), (z,a[1])]), 9)

In [435]: doo = sy.N(doo.subs([(o, a[0]), (z,a[1])]), 9)

In [436]: dzo = sy.N(dzo.subs([(o, a[0]), (z,a[1])]), 9)

In [437]: dzz = sy.N(dzz.subs([(o, a[0]), (z,a[1])]), 9)

In [438]: doo*dzz - (dzo)**2
```

Out[438]: 20.3080778

```
In [439]: doo
```

Out[439]: −7066.92005

$$doz = J_{\theta_1\theta_0} \quad doo = J_{\theta_1\theta_1} \quad dzo = J_{\theta_0\theta_1} \quad dzz = J_{\theta_0\theta_0}$$

For the first solution $(-\frac{3\sqrt{603094}}{1997}, \frac{1525}{17} + \frac{4623\sqrt{603094}}{33949})$, I obtained a value $H \approx 20.3$. Thus, as $H > 0$, the first solution can be a local maximum or a local minimum. However, it is shown that $J_{\theta_1\theta_1} \approx -7066.9$ which is smaller than 0. Thus, the first solution is a local maximum. Hence, it is not the solution that I am looking for.

In [441]:
```
a = an.args[1]
a
```
Out[441]: $\left(\frac{3\sqrt{603094}}{1997}, \frac{1525}{17} - \frac{4623\sqrt{603094}}{33949}\right)$

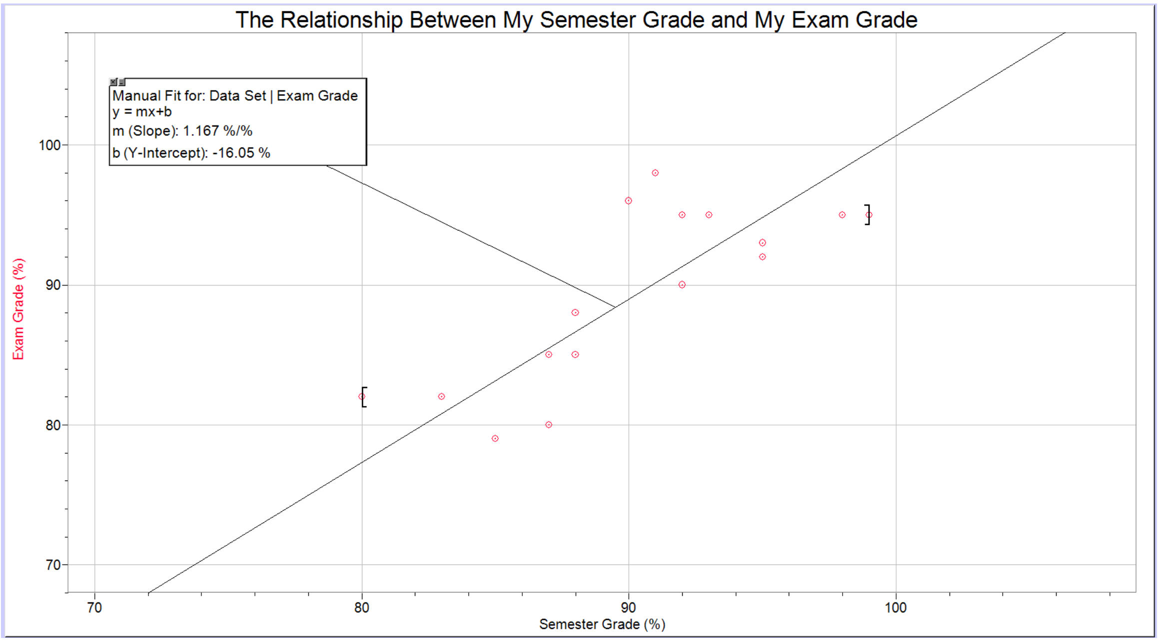
In [442]:
```
doz = sy.diff(do, z)
doo = sy.diff(do,o)
dzo = sy.diff(dz,o)
dzz = sy.diff(dz, z)
```
In [443]:
```
doz = sy.N(doz.subs([(o, a[0]), (z,a[1])]), 9)
```
In [444]:
```
doo = sy.N(doo.subs([(o, a[0]), (z,a[1])]), 9)
```
In [445]:
```
dzo = sy.N(dzo.subs([(o, a[0]), (z,a[1])]), 9)
```
In [446]:
```
dzz = sy.N(dzz.subs([(o, a[0]), (z,a[1])]), 9)
```
In [447]:
```
doo*dzz - (dzo)**2
```
Out[447]: 20.3080778

In [448]:
```
doo
```
Out[448]: 7066.92005

For the second solution $(\frac{3\sqrt{603094}}{1997}, \frac{1525}{17} - \frac{4623\sqrt{603094}}{33949})$, I obtained a value $H \approx 20.3$. Thus, as $H > 0$, it is a critical point. Unlike the first solution, $J_{\theta_1\theta_1} \approx 7066.9$ is greater than 0. Thus, the second solution is a local minimum.

$(\frac{3\sqrt{603094}}{1997}, \frac{1525}{17} - \frac{4623\sqrt{603094}}{33949}) \approx (1.167, -16.046)$. Therefore, using my own method, I get the regression line, $f(x) = -16.046 + 1.167x$

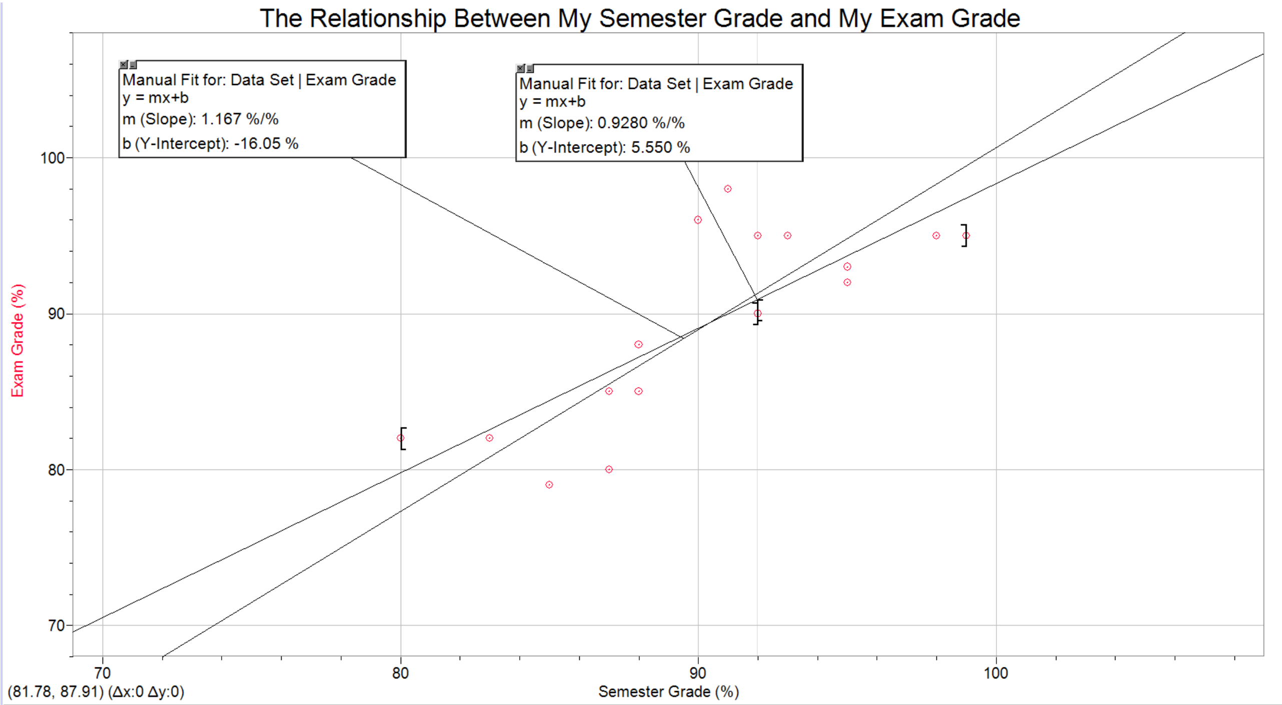
Validating My Regression Line Using My Custom Error Function



Predictions Using the Two Regression Line

Currently, I have a grade of 96% as of right now. Using both my regression line, I can predict that my exam grade this semester would be around 96% from the regression line retrieved from my custom error function, or around 95% from the regression line retrieved from using the MSE. In fact, I had gotten exactly 95% on my exam for this particular subject. While the regression line obtained using the MSE error function accurately predicted my exam grade, my custom regression line was also incredibly accurate.

Conclusion



I've found two regression line that attempt to represent the relationship between my semester grades and my exam grades. The regression line found through MSE came out to be $f(x) = 5.550 + 0.928x$ while the regression line found using my own algorithm came to be $h(x) = -16.046 + 1.167x$. For $h(x)$, I've obtained a greater gradient for my regression line found from my custom error function than the regression line found from MSE. Furthermore, the y-intercept of $h(x)$ resulted in a negative value while the y-intercept of $f(x)$ was a positive value.

When I calculated the point of intersection between $f(x)$ and $h(x)$, the coordinate of the point of intersection was (90, 89) which is the mean coordinate of the dataset. Given the fact that both regression line passes through the mean coordinate suggests that a regression line of any dataset always passes the mean coordinate of that dataset. As the mean coordinate represents the overall dataset, it would be logical for the regression line to pass through mean coordinate as its goal is to represent the data as well.

Both regression lines represent the dataset with a certain amount of accuracy. However, a certain aspect of the regression line that I created using my own error function seems to be highly unrealistic. The y-intercept of $h(x)$ is the predicted semester exam grade when my semester course grade is 0%. However, the range of exam grade is from 0% to 100%. Thus, it is impossible to get a negative percent grade (-16.046%). Therefore, for $h(x)$, it should be used with range restrictions where $0 \leq x \leq 100$.

Although, my dataset didn't have any outliers, I wondered how the regression lines would change if there were to be some. Having an outlier in my particular situation means that the distance between the outlier and the regression line is greater than its neighboring data points. Hence, the average squared vertical distance between the data points and the regression line would be greater. I presume this would be the case for my custom error function. As it multiplies the horizontal and vertical distance between the datapoint and the regression line, the average area bounded by the regression line and the horizontal and vertical segment between the datapoint and the regression line would be greater as well. This results in inaccurate regression lines. This also makes sense theoretically, for the presence of an outlier indicates the lack of reliability or validity of obtained results. Hence, a linear model that encompasses all the data points would also be less reliable.

Further Exploration:

While linear regression is a great method of creating models of datasets, it's not always the most practical to use. This is because often in reality things don't have a linear relationship nor do they depend on a single variable. Now that I've explored and investigated the basics of the mathematics behind machine learning such as univariate linear regression, I want to try stepping up the game by trying out multivariate linear regression or even other regressions such as logarithmic. This might allow for a better and more diverse range of knowledge in machine learning, where I would essentially carry out the same investigative process of using a well known error function and creating my own error function.

MLA Citation:

Amey, Stephen. 3 - Second-Order PDs, economics.uwo.ca/math/resources/calculus-multivariable-functions/3-second-order-partial-derivatives/content

“Introduction to Partial Derivatives (Article).” Khan Academy, Khan Academy, www.khanacademy.org/math/multivariable-calculus/multivariable-derivatives/partial-derivative-and-gradient-articles/a/introduction-to-partial-derivatives.

“Maxima, Minima, and Saddle Points (Article).” Khan Academy, Khan Academy, www.khanacademy.org/math/multivariable-calculus/applications-of-multivariable-derivatives/optimizing-multivariable-functions/a/maximums-minimums-and-saddle-points.

“Second Partial Derivatives (Article).” Khan Academy, Khan Academy, www.khanacademy.org/math/multivariable-calculus/multivariable-derivatives/partial-derivative-and-gradient-articles/a/second-partial-derivatives.

“Second Partial Derivative Test (Article).” Khan Academy, Khan Academy, www.khanacademy.org/math/multivariable-calculus/applications-of-multivariable-derivatives/optimizing-multivariable-functions/a/second-partial-derivative-test.