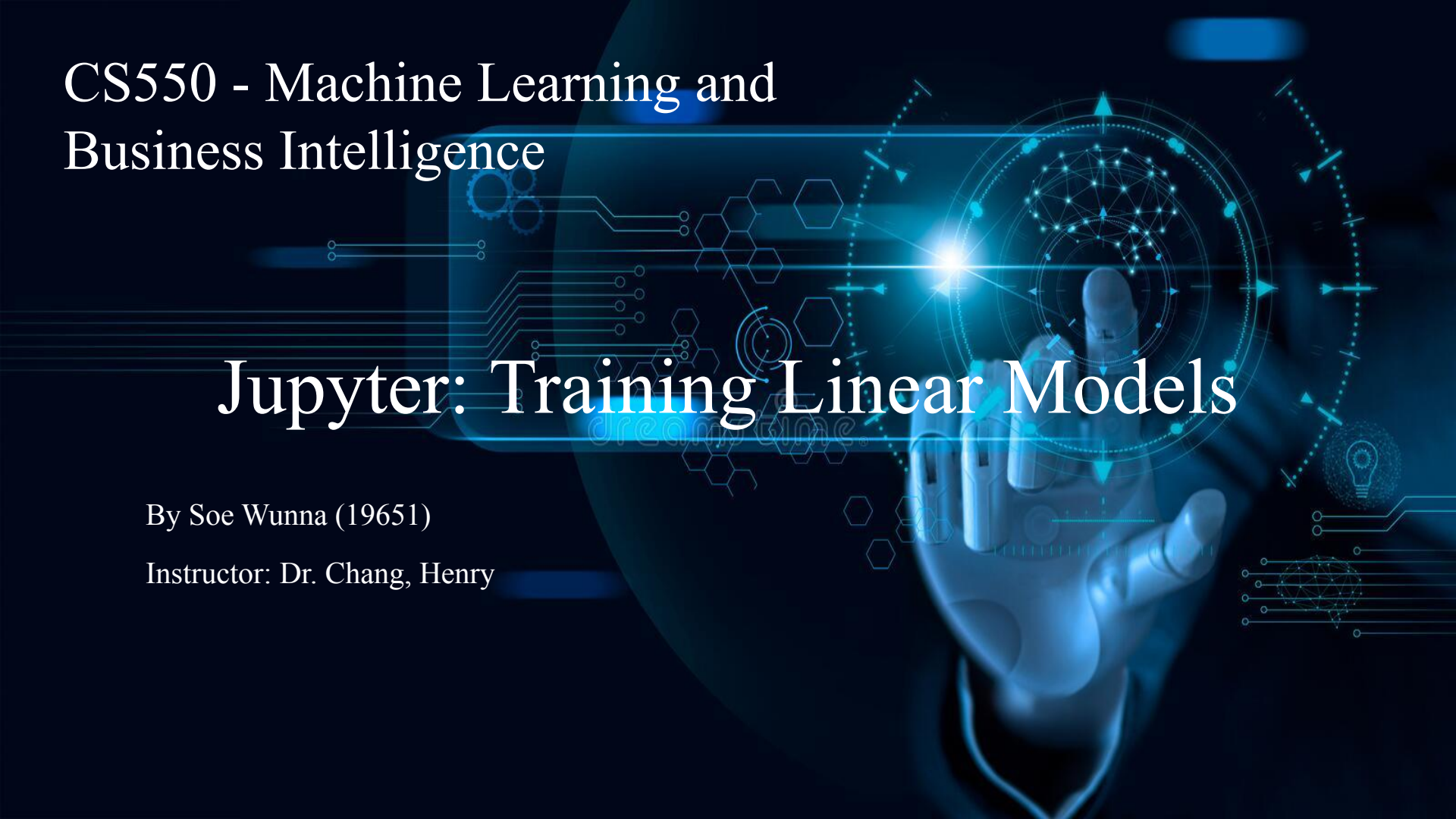
CS550 - Machine Learning and Business Intelligence

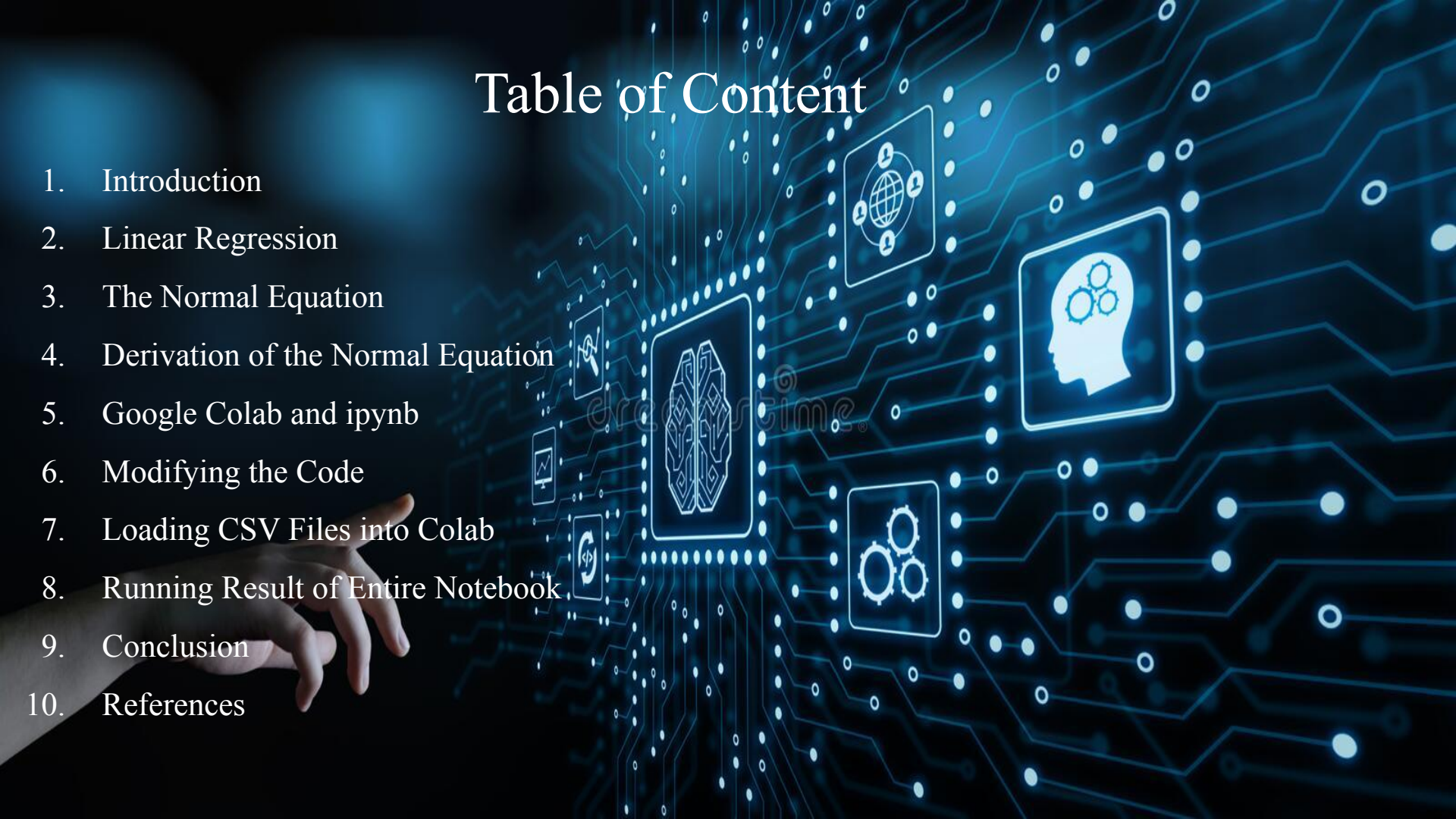
Jupyter: Training Linear Models



By Soe Wunna (19651)

Instructor: Dr. Chang, Henry

Table of Content

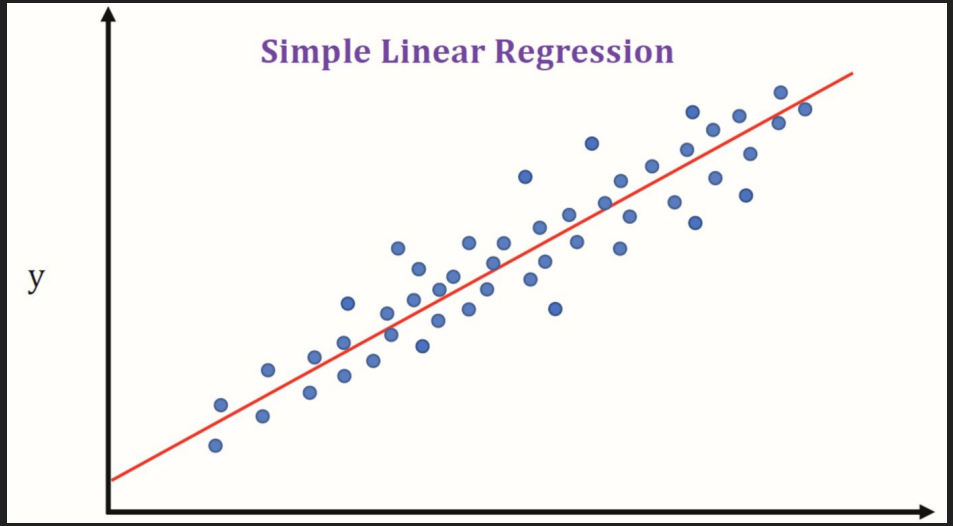


1. Introduction

In this presentation, we will discuss about Linear Regression Model and Normal Equation for Machine Learning, and apply normal equation to the python notebook file.

2. Linear Regression

Linear Regression is the supervised Machine Learning model in which the model finds the best fit linear line between the independent and dependent variable i.e it finds the linear relationship between the dependent and independent variable.



3. Normal Equation

The normal equation is a method to find the optimal solution for linear regression, which is a statistical model that aims to find the best linear relationship between a dependent variable and one or more independent variables. It is used to determine the parameters (coefficients) of the model that minimize the sum of the squared differences between the observed values and the values predicted by the model. The normal equation provides a closed-form solution to this problem, which can be calculated directly from the training data without the need for iterative optimization. The equation is given by:

$$\theta = \left(X^T X\right)^{-1} . \left(X^T y\right)$$

4. Derivation of the Normal Equation

$$\theta = \left(X^T X\right)^{-1} . \left(X^T y\right)$$

where θ is the vector of coefficients, X is the design matrix that contains the independent variables, y is the vector of dependent variables, and T is the transpose operator. The normal equation has a computational cost of O(n^3), where n is the number of features, making it more computationally efficient than some iterative optimization methods when the number of features is small. However, it can be impractical to use when the number of features is very large, as the computational cost and memory requirements become prohibitively high.

The normal equation is derived by finding the parameters that minimize the mean squared error (MSE) between the observed values and the values predicted by the model. The MSE is defined as:

MSE = (1/m) * ∑(y_i - h_θ(x_i))^2

where m is the number of training examples, y_i is the observed value for the i-th training example, h_θ(x_i) is the predicted value for the i-th training example, and θ is the vector of coefficients. The goal is to find the values of θ that minimize the MSE.

To do this, we take the partial derivative of the MSE with respect to each coefficient in θ and set it equal to zero. This results in the normal equation:

XTXθ = XTy

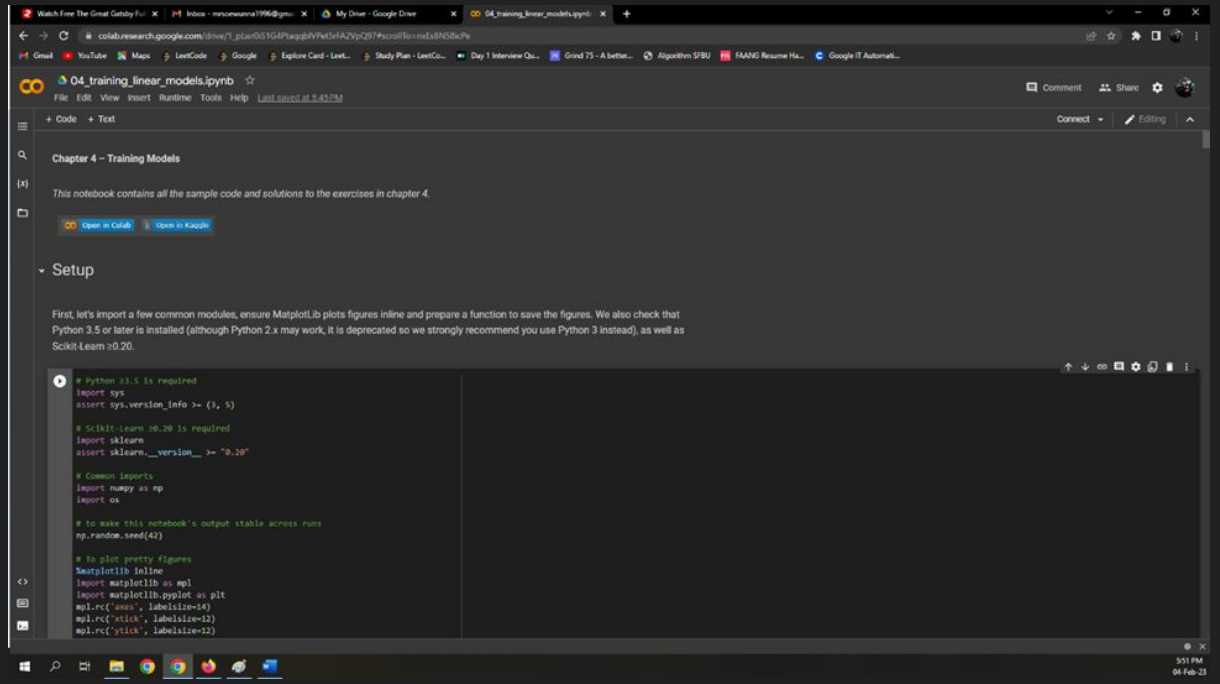
where X is the design matrix that contains the independent variables, T is the transpose operator, and y is the vector of dependent variables. Solving for θ, we get:

θ = (XTX)^(-1)XTy

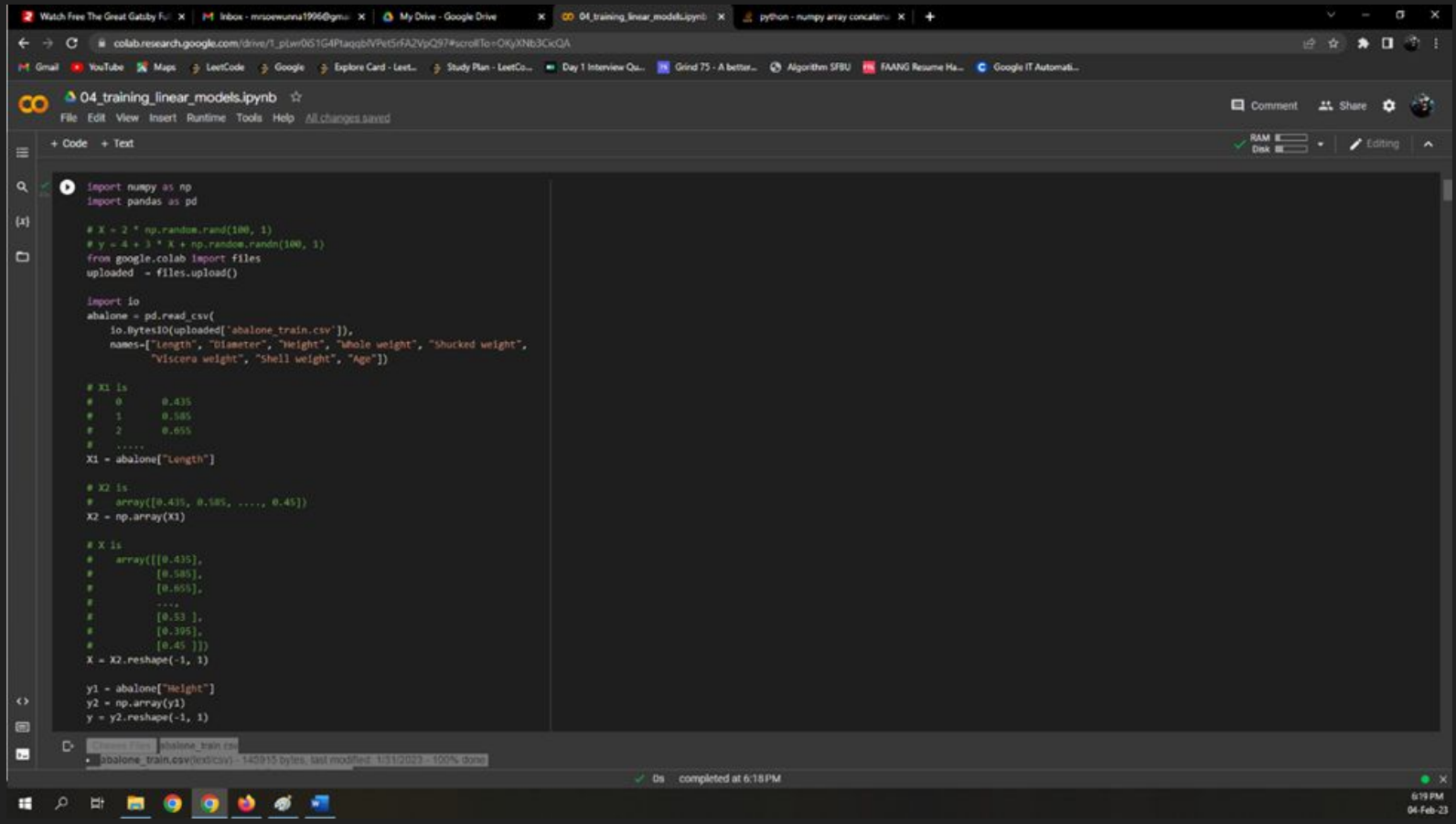
This equation provides the optimal solution for the linear regression problem in a closed-form manner.

5. Google Colab and ipynb

Google Colab is a free online platform for machine learning research and education. It provides access to a Jupyter Notebook-style environment that runs in the cloud, with access to high-performance computing resources, including GPUs and TPUs. This makes it an ideal platform for prototyping and experimenting with machine learning models, as users can quickly implement and train models on large datasets without having to worry about setting up their own hardware infrastructure.



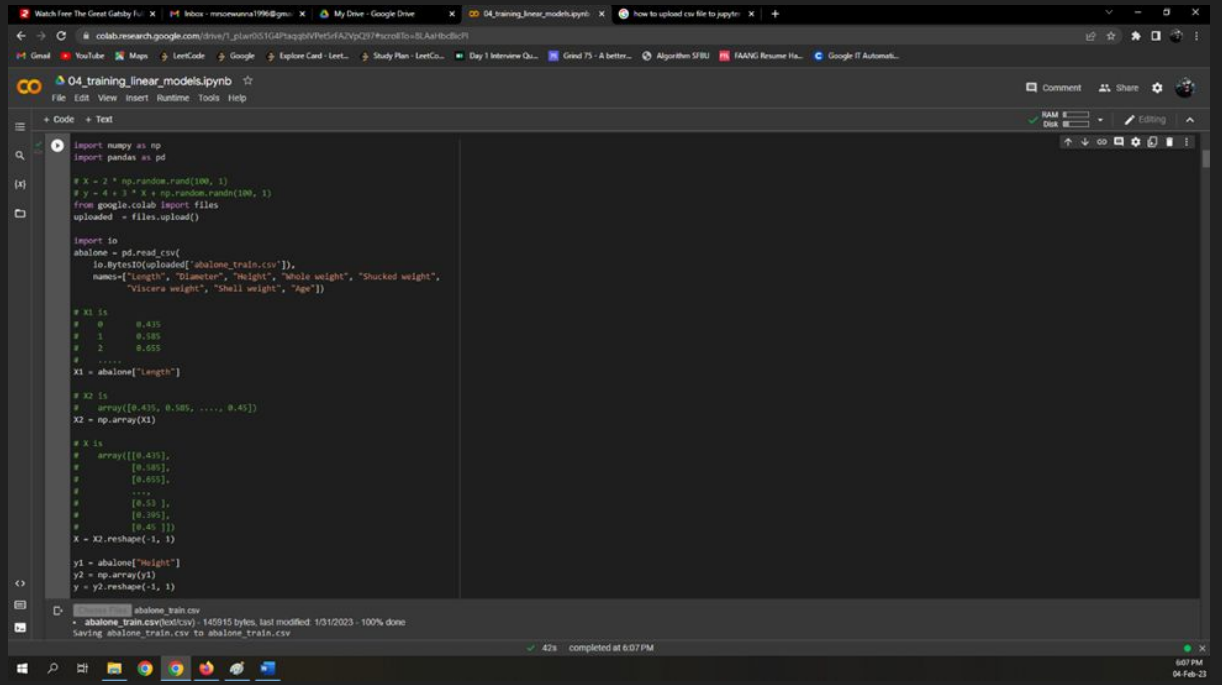
6. Modifying the Code



7. Loading CSV File into Colab

CSV (Comma-Separated Values) files are a common format for storing and exchanging data, and they are widely used in machine learning notebooks. The primary use of CSV files in machine learning is to load and process data for training and testing models.

In a machine learning notebook, CSV files can be loaded using various libraries such as Pandas, NumPy, or native Python functions.



8. Running Result of Entire Notebook

Before running the entire notebook, we have to do either one of the following modifications.

```
X_b = np.c_[np.ones((3320, 1)), X]  # add x0 = 1 to each instance
theta_best = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)
```

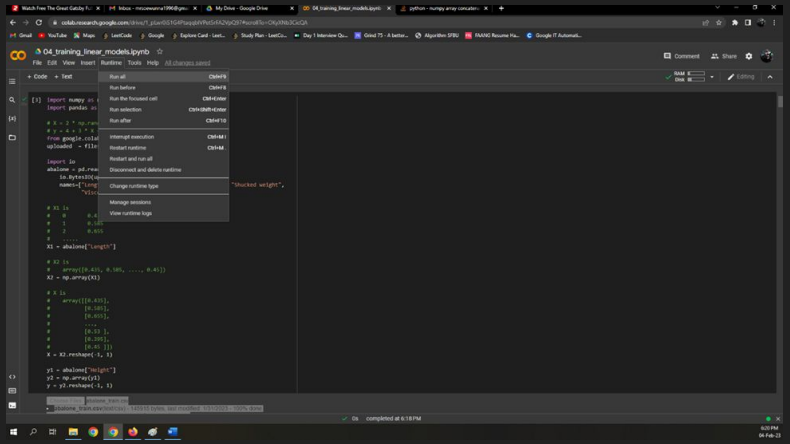
Changing 100 to 3320 and the problem has been solved.

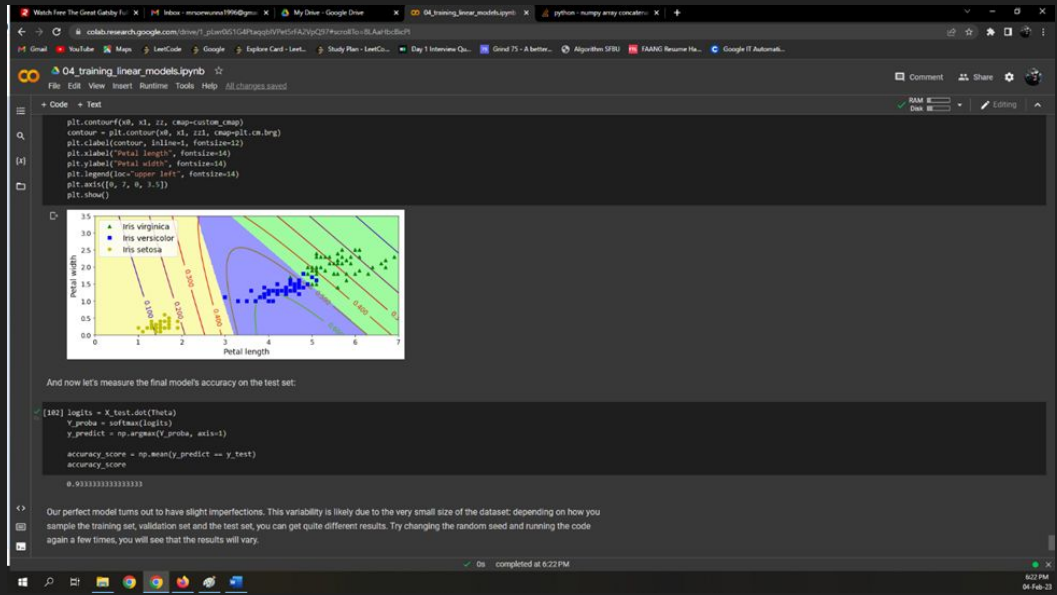
(OR)

```
import io
abalone = pd.read_csv(
    io.BytesIO(uploaded['abalone_train.csv']),
    names=["Length", "Diameter", "Height", "Whole weight", "Shucked weight",
           "Viscera weight", "Shell weight", "Age"], nrows = 100)
```

We need to define nrows = 100.

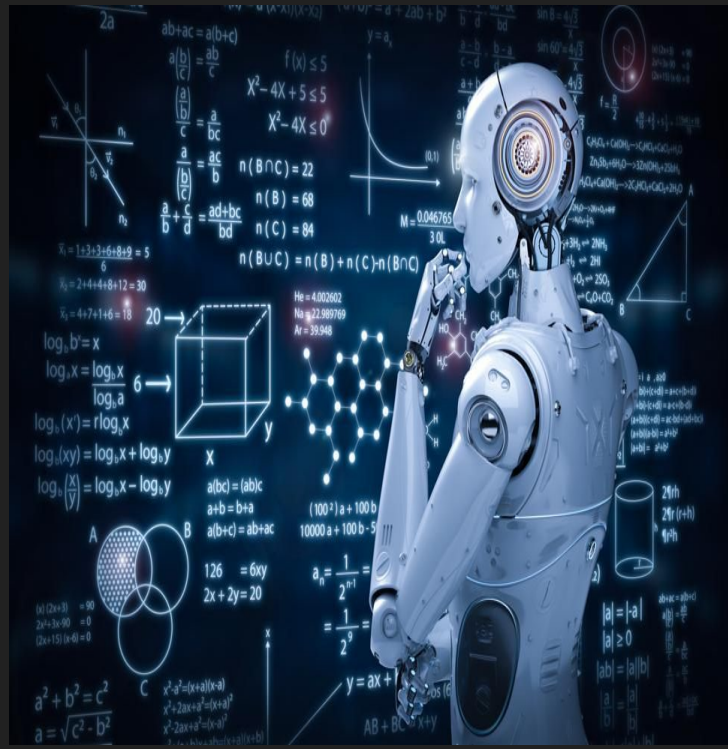
8. Running Result of Entire Notebook





9. Conclusion

In conclusion, the normal equation is a useful tool in linear regression, a popular machine learning algorithm used to model the relationship between a dependent variable and one or more independent variables. The normal equation provides a closed-form solution to finding the optimal coefficients for the linear regression model that minimize the mean squared error between the observed and predicted values. The normal equation is computationally efficient when the number of features is small, and it has the advantage of providing a direct solution without the need for iterative optimization. However, its computational cost and memory requirements increase rapidly with the number of features, making it less practical for large-scale problems. Despite these limitations, the normal equation remains an important concept in the field of machine learning and is widely used in practice for solving linear regression problems.



10. References

1. https://hc.labnet.sfbu.edu/~henry/sfbu/course/data_science/algorithm/slide/linear_regression_example.html#lf

2. https://www.geeksforgeeks.org/ml-normal-equation-in-linear-regression/

3. https://www.datacamp.com/tutorial/tutorial-normal-equation-for-linear-regressionhttps://www.datacamp.com/tutorial/tutorial-normal-equation-for-linear-regression

4. https://stackoverflow.com/questions/65545279/linear-regression-why-does-normal-equation-give-huge-error