Applications of Mathematics in Supervised Learning

Alinda Rolland Mucunguzi[1,2,⋆] and Laure Gouba[1,2,⋄]
[1] *African Institute for Mathematical Science (AIMS-Ghana)*
Accra, Ghana.
[⋆] *Email: ralinda@aims.edu.gh*
[2] *The Abdus Salam International Centre for Theoretical Physics (ICTP)*
Strada Costiera 11, I-34151 Trieste Italy.
[⋄] *Email: lgouba@aims.edu.gh*

March 24, 2025

Abstract

In this work, we explore some applications of mathematics in the development and usage of supervised learning algorithms with a strong focus on linear regression models. Subsequently, we look at the mathematical foundations essential for supervised learning, which include linear algebra, probability theory, calculus, optimization, statistics, and geometry. For a concrete illustration of the applications of mathematics in supervised learning, this work employs simple and multiple linear regression models using data that is about pH of pure water. Through these examples, we demonstrate how mathematical techniques are applied in formulating, estimating and evaluating linear regression models. Key processes such as least squares estimation and statistical inference are highlighted to show their critical application in parameter estimation and model validation. The findings underscore the importance of mathematical rigor in ensuring accuracy and interpretability of supervised learning models.

1 Introduction

Machine learning employs techniques from mathematics and related computational resources to create models that learn from experience, adapt to new data provided by a user and generalize patterns after a successful training process. The motivation for the interest in the connection between mathematics and supervised learning is highlighted below;

- Mathematics provides a theoretical foundation upon which machine learning algorithms are built. Concepts from linear algebra, calculus, optimization, statistics, probability theory, graph theory and geometry are essential for understanding the inner operations of algorithms.
- Selecting and customizing a given model. Mathematics plays a vital role in choosing a model with an algorithm that can handle a unique problem at hand.
- Debugging code when things go wrong. Mathematics helps us understand errors and diagnose them effectively. Here, it is like being a detective in the world of code.

- Knowledge of mathematics enables us to effectively interpret the predictions made by a given model that we choose to implement.
- Mathematics offers us rigorous tools for analyzing the performance, behavior and limitations of machine learning models. Through mathematical analysis, we gain insights into why certain algorithms work well in specific scenarios.
- Innovation and research. Creating new machine learning algorithms and improving existing ones require a great deal of mathematical insights and principles.

In subsequent sections of this chapter, we look briefly at different areas of mathematics and their applications in supervised learning. Section 2 is about the areas of mathematics and their applications in supervised learning. In section 3, we present as applications, some linear regression tasks. The conclusion is given in section 4.

2 Areas of mathematics and their applications in supervised learning

2.1 Linear algebra

Linear algebra deals with the study of vectors, vector spaces and mappings that are required to perform linear transformations between given vector spaces. Linear algebra provides tools needed to effectively represent, modify, draw insights and improve machine learning models. A lot of potential in machine learning is put forth using linear algebra objects such as vectors, tensors, matrices and operations like addition, multiplication, inversion, transposition and decomposition of matrices [1]. Various tools are unveiled to us through the knowledge of linear algebra;

1. Vector spaces: A vector space can be visualized as a set of vectors along with vector addition and scalar multiplication satisfying a given set of axioms. Vector spaces provide us a foundation for understanding characteristics and connections between vectors.
2. Vectors: Vectors are elements of a vector space. Vectors are generally objects with magnitude and direction. In the light of machine learning, we use vectors to represent data points and features of data. Addition, subtraction and scalar multiplication are all possible operations with vectors.
3. Matrices and tensors: Matrices are rectangular arrays of objects in rows and columns. Tensors are multidimensional arrays of data. We organize tabular data by way of matrices or tensors and various operations come to life when we organize data using matrices or tensors. Matrices can undergo scaling, addition, rotation, translation and other operations. Tensors can undergo tensor contraction, reshaping, broadcasting and other operations.
4. Linear transformations: These transfer vectors between vector spaces taking into account the underlying properties of the vector spaces. In machine learning, linear transformations give us the ability to reshape and alter data in appropriate ways.

2.1.1 Linear Algebra and Machine Learning Algorithms

Some algorithms in machine learning that utilize various principles from linear algebra include;

- Linear regression: We employ linear regression in creating a linear connection between a predictor and a target variables. For example, matrix operations are employed in solving systems of linear equations which enables the determination of ideal coefficients that minimize error and thus allow us to obtain the best linear fit for data.

- Principle components analysis(PCA): In machine learning, principle component analysis is used to decrease the dimensionality of high-dimensional datasets. Using PCA, we can transform a high dimensional feature space into a lower-dimensional one while retaining the most relevant aspects of a dataset. Here, we use the analysis of eigenvalues and eigenvectors to dissect a given covariance matrix.

- Support vector machine(SVM): For the determination of an ideal hyperplane that categorizes data points into multiple classes, support vector machines heavily rely on linear algebra. SVMs can work with complicated decision boundaries and classify new instances by representing data points as vectors and employing dot products coupled with matrix operations for their classification tasks.

- Neural networks: At the core of deep learning are neural networks that heavily employ linear algebra computations. Matrix multiplication and activation functions are employed in forward and backward propagation in neural networks to handle weights and biases in the network.

2.2 Calculus

Calculus forms a mathematical foundation for several machine learning algorithms and models. Calculus has two branches, that is; integral calculus and differential calculus. Integral calculus focuses on the concept of integration, which is essentially the process of finding the accumulated total of quantities. Differential calculus emphasizes creating an understanding of rates of change and steepness of curves [2]. Machine learning models that employ vast concepts from calculus include;

- Gradient descent: This is a first-order optimization algorithm employed when finding the minimum point of a differentiable function. The gradient descent algorithm repeatedly changes the parameter of a function being optimized in a direction opposite to the slope of the function and continuously reduces the value of the function. The gradient descent algorithm iteratively takes steps in the direction opposite to that of the gradient until convergence. The update rule for gradient descent is:

$$x_{i+1} = x_i - \eta((\nabla f)(x_i))^T, \tag{1}$$

  with η the step size, x_i a data point and ∇f the gradient vector.

- Convolutional neural networks: Here, we observe calculus in deep learning. Calculus enables convolution and pooling operations in convolutional neural networks. During convolution, we apply filters to extract meaningful information from data. During pooling, we employ calculus to decrease feature space dimensionality with minimal information loss. These convolutional neural networks are employed in computer vision to perform certain tasks such as image recognition.

- Recurrent neural networks: This is still calculus in deep learning. Recurrent neural networks are used in the analysis of sequential data. These networks employ gradients to learn over time enabling them to detect patterns and make predictions about sequences.

2.2.1 Calculus and Regularization

Regularization techniques are used to prevent the model from over-fitting a given dataset. During regularization, extra terms are added to a loss function. The extra terms serve as penalties that encourage the model to find solutions that are not too complicated and can easily be generalized on new datasets. For λ, λ_1, λ_2 as scalars and β_i as model coefficients, some regularization techniques are;

- Lasso regularization with a penalty term: $\lambda \sum_{i=1}^{n} |\beta_i|$.
- Ridge regularization with a penalty term: $\lambda \sum_{i=1}^{n} {\beta_i}^2$.
- Elastic net regularization with a penalty term: $\lambda_1 \sum_{i=1}^{n} |\beta_i| + \lambda_2 \sum_{i=1}^{n} {\beta_i}^2$.

2.3 Optimization

Optimization is the process of finding the best solution from a set of feasible solutions. There are two kinds of optimization, continuous optimization and combinatorial optimization. When we use computers to implement machine learning algorithms, then, inevitably mathematical formulations are expressed as numerical optimization methods. In machine learning, the process of optimization involves obtaining an ideal set of model parameters that minimize a given cost function. By convention, most machine learning optimization problems are handled as minimization problems.

A cost function calculates the discrepancy between the predicted output values and the actual output values. In an optimization lens, training a machine learning model boils down to simply finding a good set of model parameters. If we have n data points, y_i as the actual value and $\hat{y}_i$ as the predicted value, some common cost functions can be defined as;

- Mean squared error(MSE): This is usually used in regression tasks,

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2. \tag{2}$$

- Mean absolute error(MAE): This is also used in regression tasks,

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|. \tag{3}$$

- Cross-entropy loss: This is also known as the log loss function. It is used in binary classification tasks where it measures the performance of a classification model whose output is a probability between 0 and 1.

$$LogLoss = -\frac{1}{n} \sum_{i=1}^{n} [y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)]. \tag{4}$$

- Cosine similarity loss:

$$\ell(\theta) = 1 - cos(\theta) = 1 - \frac{A \cdot B}{||A||||B||}, \tag{5}$$

with A as the true vector, B as the predicted vector, and θ is the angle between A and B.

2.3.1 Optimization and Machine Learning Algorithms

In machine learning to find the minimum of a cost function, we use optimization algorithms that iteratively modify model parameters until a desired estimate is obtained. Examples of these algorithms include gradient descent, stochastic gradient descent and Adam.

2.3.2 Convex and Non-convex Optimization

A real-valued function is said to be convex if the line segment joining any two distinct points on its graph lies above the graph between the two points. Convex optimization involves convex objective functions with global minima. Many of the machine learning problems are non-convex resulting in the occurrence of multiple minima. Techniques for convex optimization such as quadratic programming and Langrage multipliers provide foundational tools for the more complex non-convex optimization problems.

2.4 Probability and Statistics

In machine learning, we have probabilistic models and a probabilistic framework. The rigorous probabilistic framework describes how to represent and manipulate uncertainty in models and their predictions. Here, learning can be seen as an attempt to make predictions about future data and the consequences of decisions taken. Observed data can have consistency with different models and thus an appropriate model is uncertain. Further still, predictions about future data and the consequences of future actions are uncertain.

In a probabilistic framework for supervised machine learning, the goal is still one of approximating the relationship between input features,X and labels,Y. For regression tasks, we assume that $Y \in [0, 1]$ and we assume that $Y \in \{0, 1\}$ for classification tasks. The relationship between X and Y is encapsulated by an unknown probability measure μ on $Z = X \times Y$. The elements $(x, y) \in Z$ can be called labeled examples. Learning takes place when we present such labeled data of the form $z \in Z^m$ to a machine learning algorithm which must return $h : X \to [0, 1]$ for a fixed set of possible hypotheses. The level at which the output hypothesis $\mathcal{A}(z)$ correctly represents the hidden hypothesis μ is quantified with the aid of a loss function $\ell : [0, 1] \to [0, 1]$. We always aim at having an $\mathcal{A}(z)$ with a small value of the loss [3].

2.4.1 Categories of Probabilistic Models

These are models that quantify the inherent uncertainty in data and integrate it into their predictions. These models are applied in speech and image recognition systems, recommendation systems, natural language processing and more.

- Generative models: These aim at modeling the joint distribution of the predictor and the target variable. Generative models create new datasets by utilizing the probability distribution of the original dataset. Examples of these models include; Bayesian networks, variational auto-encoders, pixel recurrent neural networks, Markov chains and

diffusion models. A good application of these models is Chat Generative Pre-trained Transformer(ChatGPT) which is a type of large language model that uses probabilistic methods to generate text.

- Discrimination models: These aim at discriminating the conditional distribution of the target variables given the predictor variables. Discrimination models learn appropriate decision boundaries which separates different classes of target variables. Examples of these models include support vector machines, logistic regression and some neural networks.

- Graphical models: Graphical probabilistic machine learning models determine the conditional dependence between variables using graphical representations. They include Bayesian networks which are directed graphical models and Markov networks that utilize undirected graphs.

2.4.2 Various Statistical Tools in Machine Learning

Different machine learning models fall in a class of statistical learning models in which we employ vast statistical techniques to develop models that can learn from data and make appropriate predictions. Statistics also provides us with powerful tools in the form of descriptive statistics that enable us to visualize data, summarize data, identify patterns and detect outliers in a dataset. Furthermore, the principles of statistics form pillars upon which we construct machine learning models, interpret results, and validate models. These include;

- Measures of central tendency.
- Measures of spread.
- Sampling.
- Estimation.
- Hypothesis testing.
- Cross validation.

Machine learning models that employ vast concepts from statistics include;

1. Linear regression: This is a supervised machine learning algorithm we employ to establish the relationship between predictor variables and target variables.

2. Logistic regression: This employs the logistic function to estimate the probability of a categorical outcome basing on the input variables. Logistic regression performs binary classification using the logistic function also known as the sigmoid function to map outcomes to their probabilities. The logistic function is defined as;

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \tag{6}$$

   where z is a linear combination of input features.

3. Decision tree: This employs statistical tools to split data based on the features in the dataset creating a tree-like structure. For regression tasks, we can employ a loss function whereas for classification tasks, we can use the Gini index and the entropy

function to show how "pure" leaf nodes are. The Gini index and the entropy function are respectively given by;

$$G(t) = 1 - \sum_{i=1}^{c} {p_i}^2 \quad \text{and} \quad E = -\sum_{i=1}^{n} p_i log(p_i). \tag{7}$$

In the Gini index expression, t is a node with n_t data points and p_i is the proportion of data points in the node t that belongs to a class i. In the entropy formula, E is entropy over all classes, p_i is the proportion of training data points with a class label i.

4. Random forest: This is an ensemble machine-learning model that improves the accuracy of predictions by using several decision trees. Random forests sample randomly selected subsets of features to build trees. Each tree in a decision forest makes a forecast for an output and the final prediction is obtained from the majority vote for all the trees in the forest.

$$Random\ forest = Decision\ trees + Bagging + Bootstraping + Random\ split. \tag{8}$$

- Bagging is a learning method that employs several different models built by training them on different sample subsets then we either aggregate, average or take the majority of the predictions.
- Bootstrapping is a resampling technique used to estimate a model's parameters by repeatedly sampling with replacement from a training dataset.

Remarks

1. We have a special class of trees called boosted trees. The process of boosting involves combining several poorly performing classifiers to obtain a better classifier. For example, instead of using an equal probability for all items in a dataset, we can give a higher weight to items that have been misclassified by the model. XGBoost which stands for extreme gradient boosting is one of the models that employs boosted trees in its algorithm.

2. The majority of the statistical machine learning models such as decision trees, random forests, and logistic regression incorporate vast ideas from probability theory. At this point, we can also observe that a single machine-learning algorithm can employ concepts from different areas of mathematics.

2.5 Geometry

Geometry in mathematics deals with shapes and structures. Other than the usual Euclidean geometry, other geometries are also relevant in machine learning such as the elliptic geometry of Riemann. However, non-Euclidean data suffers the curse of dimensionality, we need an exponential number of samples to approximate even a basic multidimensional function. Principal component analysis as a technique for dimensionality reduction was described in Subsection 2.1.1, however, sometimes losing important information about the data is inevitable. Thus, we employ the geometric structure of the input data to overcome this problem and this is formalized as geometric priors. For example, in geometric deep learning, functions used must respect two priors;

- Symmetry: This is respected by functions that leave an object invariant. These functions must be composable, and invertible and their collection should contain the identity map.
- Scale preparation: This deals with the stability of a function under slight deformation of its domain.

3 Applications

Linear regression is a fundamental supervised learning algorithm that is employed on data with continuous input and output variables. In linear regression, we seek to obtain a combination of inputs that best explains the output by assuming a linear connection between the input and output variables [4]. Simple linear regression models involve one independent variable and one dependent variable. The words; independent variable, predictor and explanatory variable can be used interchangeably with input variable. Also, the words; labels, dependent variable and target variable are used interchangeably with the output variable. The simple linear regression model estimates the intercept and the slope of a line of best fit between the input and output variables.

3.1 A Simple Linear Regression Task

We begin with an example to motivate an understanding of simple linear regression. In chemistry and various industries, pH measurements play a key role in the success of experiments and the formation of desired products. Various factors affect the pH of a given solution. The data in Table 1 shows how pH of pure water varies with increasing temperature.

Temperature($°C$)	0	10	15	20	25	30	40	50	100
pH	7.47	7.27	7.17	7.08	7.00	6.92	6.77	6.63	6.14

Table 1: Shows how pH of pure water varies with changes in temperature.

The goal here is to predict the pH of pure water when given a particular temperature.

3.1.1 The Approach

Using matplotlib.pyplot library in python, we create a scatter plot for visualization of the trend of the data using 8 of the data points leaving out the third data point and this is as shown in the first subplot in Figure 1. We fit a simple linear regression model using linear regression model from sklearn.linear_model provided by the scikit-learn library in Python. This process fits a model to the data and we obtain the optimal estimates of the intercept and the slope. The second subplot of Figure 1 shows the fitted linear regression model trend line. Generally, the equation of a straight line is $y = mx + c$ which we use in this work as $y = \beta_1 x + \beta_0$, where $m = \beta_1$ is the gradient and $c = \beta_0$ the y-intercept.

Thus, the values of parameters are $\hat{\beta}_0 = 7.3585$ and $\hat{\beta}_1 = -0.01305$. The equation of model is

$$y = 7.3585 - 0.01305x. \tag{9}$$

Prediction: The model was tested on its capacity to generalize using $x = 15°C$ and the corresponding prediction was a pH value, $\hat{y} = 7.16279$ while the actual value was $y = 7.17$.

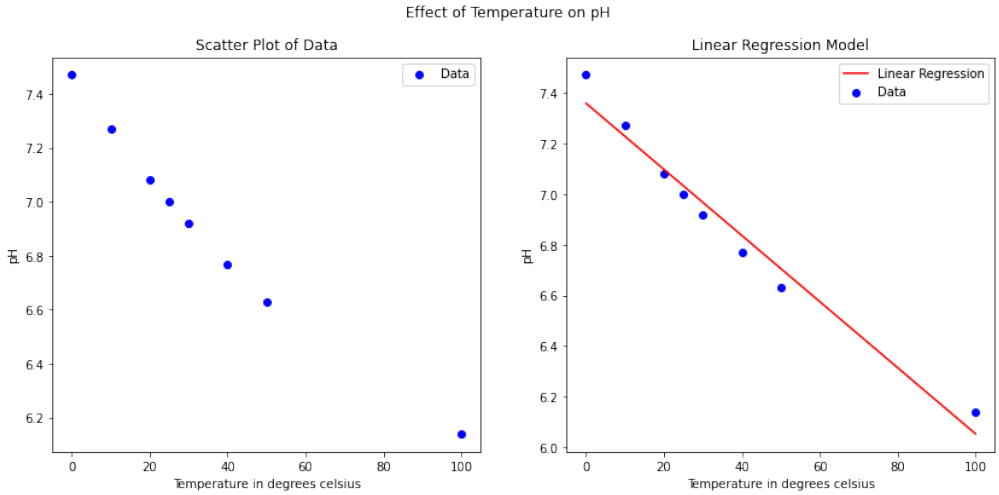

Figure 1: The left plot shows a scatter plot of the data points. The right plot shows the linear regression model line fitted to the data.

3.1.2 Interpretation: A Chemistry Perspective

When temperature increases, pH reduces, however, this does not necessarily imply that a given sample of pure water has become acidic. Increasing temperature increases molecular vibrations within water molecules causing more dissociation of the water molecules thus yielding more hydroxyl and hydrogen ions. The increased hydrogen ions cause a decrease in the pH of water as pH is simply a measure of the hydrogen ion concentration of a solution. However, the hydroxyl and hydrogen ions are liberated in equimolar proportions and this ensures that the pure water remains neutral. The implication is that when one sets out to measure the pH of pure water or any other solution, the temperature too should be measured. Notably, a pH value without a corresponding temperature is incoherent. Using a pH sensor that has an automatic temperature compensation stands as a viable alternative.

Remark: We would like to know how accurate the values of the model parameters obtained are, gauge the closeness of the predicted to the actual values and also we want to know about the general performance of the entire model. This calls for rigorous mathematical formulations which are covered in the proceeding sections of this chapter.

3.2 The Least Squares Regression Approach

We consider a case where we have a single real-valued feature as input and a real-valued output. Let us denote the input-output pairs as (x_i, y_i) with $x_i, y_i \in \mathbb{R}$ and $i \in \{1, 2, \ldots, n\}$. The model here is from a family of straight line functions mapping from $\mathbb{R}$ to $\mathbb{R}$. As earlier noted, we use the equation of a straight-line as $y = \beta_1 x + \beta_0$ and in linear regression, the task is to find the best combination of β_0 and β_1 that best describes the trend observed in the data [5].

To obtain a straight line that best describes the trend in a dataset, we use a loss function that tells us how far off we are from the true value y when we use its approximation $\hat{y}$. We denote the loss function as $\ell(y, \hat{y})$ such that $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. There are different kinds of loss functions, we use the quadratic loss function also known as the squared loss function defined as;

$$\ell(y, \hat{y}) = (y - \hat{y})^2. \tag{10}$$

The cost obtained using the squared loss function is non-negative and it increases quadratically, for example, every time we double $y - \hat{y}$, the cost increases by a factor of four. Figure

2 shows the distances we wish to minimize on the left subplot and how the losses vary quadratically when using the squared loss function on the right subplot for an arbitrary dataset.

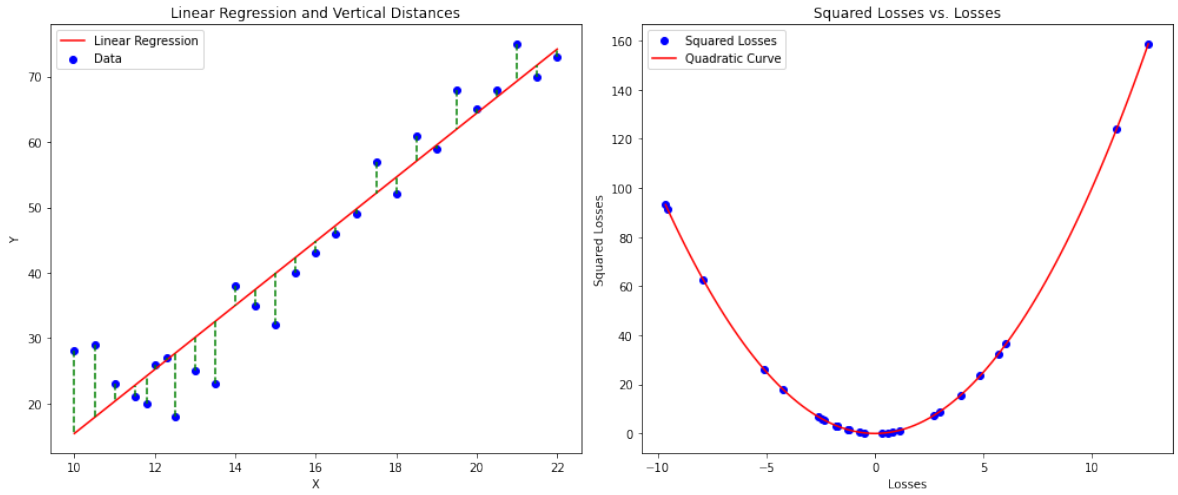

Figure 2: The first subplot shows vertical distances from the data points to the regression line. The second subplot shows how losses are varying with the squared loss function.

Using the equation of a straight line and working with the data points (x_i, y_i), we have;

$$\hat{y}_i = \beta_1 x_i + \beta_0.$$

The left subplot of Figure 2 can aid in the visualization of labels predicted by the fitted regression line and the true labels as by the data. Thus, for any datum, we have the squared loss as;

$$\ell(y_i, \hat{y}_i) = (y_i - \beta_1 x_i - \beta_0)^2.$$

If we take n data points which are usually the number of rows in a dataset and consider the average loss also known as the empirical loss, then, we have that;

$$\begin{aligned} \mathcal{L}(\beta_0, \beta_1) &= \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \ell(y_i, \hat{y}_i), \\ &= \tfrac{1}{n} \textstyle\sum_{i=1}^{n} (y_i - \beta_1 x_i - \beta_0)^2, \\ \mathcal{L}(\beta_0, \beta_1) &= \tfrac{1}{n} \textstyle\sum_{i=1}^{n} (\beta_1 x_i + \beta_0 - y_i)^2. \end{aligned} \tag{11}$$

The empirical loss function, $\mathcal{L}(\beta_0, \beta_1)$ can be minimized with respect to the parameters β_0 and β_1 in order to obtain optimal estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize it;

$$\begin{aligned} \hat{\beta}_0, \hat{\beta}_1 &= arg\min_{\beta_0, \beta_1} \mathcal{L}(\beta_0, \beta_1), \\ \hat{\beta}_0, \hat{\beta}_1 &= arg\min_{\beta_0, \beta_1} \tfrac{1}{n} \textstyle\sum_{i=1}^{n} (\beta_1 x_i + \beta_0 - y_i)^2. \end{aligned} \tag{12}$$

Arguments that minimize the empirical loss function or empirical risk function, $\mathcal{L}(\beta_0, \beta_1)$ are denoted in Equation (12) as *"arg min"*. The problem of obtaining estimates of β_0 and β_1 that minimize $\mathcal{L}(\beta_0, \beta_1)$ is the *empirical risk minimization* and we accomplish it with the use of partial derivatives of the empirical risk function concerning the parameters β_0 and β_1. Taking the partial derivative of $\mathcal{L}(\beta_0, \beta_1)$ with respect to β_0, we have that;

$$\begin{aligned}
\frac{\partial}{\partial\beta_0}\mathcal{L}(\beta_0,\beta_1) &= \tfrac{\partial}{\partial\beta_0}\Big(\tfrac{1}{n}\textstyle\sum_{i=1}^{n}(\beta_1 x_i+\beta_0-y_i)^2\Big),\\
&= \tfrac{1}{n}\textstyle\sum_{i=1}^{n}\tfrac{\partial}{\partial\beta_0}(\beta_1 x_i+\beta_0-y_i)^2,\\
&= \tfrac{2}{n}\textstyle\sum_{i=1}^{n}(\beta_1 x_i+\beta_0-y_i),\\
\frac{\partial}{\partial\beta_0}\mathcal{L}(\beta_0,\beta_1) &= 2\beta_1\Big(\tfrac{1}{n}\textstyle\sum_{i=1}^{n}x_i\Big)+2\beta_0-2\Big(\tfrac{1}{n}\textstyle\sum_{i=1}^{n}y_i\Big). \qquad (13)
\end{aligned}$$

Also, taking the partial derivative of $\mathcal{L}(\beta_0,\beta_1)$ with respect to β_1, we have that;

$$\begin{aligned}
\frac{\partial}{\partial\beta_1}\mathcal{L}(\beta_0,\beta_1) &= \tfrac{\partial}{\partial\beta_1}\Big(\tfrac{1}{n}\textstyle\sum_{i=1}^{n}(\beta_1 x_i+\beta_0-y_i)^2\Big),\\
&= \tfrac{1}{n}\textstyle\sum_{i=1}^{n}\tfrac{\partial}{\partial\beta_1}(\beta_1 x_i+\beta_0-y_i)^2,\\
&= \tfrac{2}{n}\textstyle\sum_{i=1}^{n}x_i(\beta_1 x_i+\beta_0-y_i),\\
\frac{\partial}{\partial\beta_1}\mathcal{L}(\beta_0,\beta_1) &= 2\beta_1\Big(\tfrac{1}{n}\textstyle\sum_{i=1}^{n}{x_i}^2\Big)+2\beta_0\Big(\tfrac{1}{n}\textstyle\sum_{i=1}^{n}x_i\Big)-2(\tfrac{1}{n}\textstyle\sum_{i=1}^{n}x_i y_i). \qquad (14)
\end{aligned}$$

The expressions in Equations (13) and (14) involve four averages and for convenience, these can be simplified as;

$$\begin{aligned}
\bar{y} &= \tfrac{1}{n}\textstyle\sum_{i=1}^{n}y_i \quad , \quad \bar{x}=\tfrac{1}{n}\textstyle\sum_{i=1}^{n}x_i, \qquad (15)\\
\lambda &= \tfrac{1}{n}\textstyle\sum_{i=1}^{n}{x_i}^2 \quad , \quad \alpha=\tfrac{1}{n}\textstyle\sum_{i=1}^{n}x_i y_i. \qquad (16)
\end{aligned}$$

For minimum values, we set the partial derivatives in Equations (13) and (14) to zero. We obtain a system of two equations in two unknowns $\hat{\beta}_0$ and $\hat{\beta}_1$;

$$\begin{aligned}
\frac{\partial}{\partial\beta_0}\mathcal{L}(\beta_0,\beta_1) &= 0,\\
2(\hat{\beta}_1\bar{x}+\hat{\beta}_0-\bar{y}) &= 0,\\
\hat{\beta}_1\bar{x}+\hat{\beta}_0-\bar{y} &= 0, \qquad (17)
\end{aligned}$$

Also, we have;

$$\begin{aligned}
\frac{\partial}{\partial\beta_1}\mathcal{L}(\beta_0,\beta_1) &= 0,\\
2(\hat{\beta}_1\lambda+\hat{\beta}_0\bar{x}-\alpha) &= 0,\\
\hat{\beta}_1\lambda+\hat{\beta}_0\bar{x}-\alpha &= 0, \qquad (18)
\end{aligned}$$

We multiply the Equation (17) by $\bar{x}$ and then we subtract it from Equation (18). After such an operation, we obtain that;

$$\begin{aligned}
\hat{\beta}_1 &= \tfrac{\alpha-\bar{x}\bar{y}}{\lambda-\bar{x}^2}=\frac{\tfrac{1}{n}\sum_{i=1}^{n}x_i y_i-\Big[\Big(\tfrac{1}{n}\sum_{i=1}^{n}x_i\Big)\Big(\tfrac{1}{n}\sum_{i=1}^{n}y_i\Big)\Big]}{\tfrac{1}{n}\sum_{i=1}^{n}{x_i}^2-\Big(\tfrac{1}{n}\sum_{i=1}^{n}x_i\Big)^2}, \qquad (19)\\
\hat{\beta}_0 &= \bar{y}-\hat{\beta}_1\bar{x}=\tfrac{1}{n}\textstyle\sum_{i=1}^{n}y_i-\tfrac{\hat{\beta}_1}{n}\textstyle\sum_{i=1}^{n}x_i. \qquad (20)
\end{aligned}$$

If $\hat{y}_i = \hat{\beta}_1 x_i + \hat{\beta}_0$ is the predicted value of y based on some ith value of x, then, $e_i = y_i - \hat{y}_i$ is the value of the ith residual. Thus, e_i is the difference between the value of the ith observed response value and the value of the ith prediction made by the linear regression model. The *residual sum of squares* (RSS) is defined as;

$$RSS = {e_1}^2 + {e_2}^2 + \ldots + {e_n}^2 = \sum_{i=1}^{n} {e_i}^2. \tag{21}$$

This can equivalently be represented as;

$$\begin{aligned} RSS &= (y_1 - \hat{\beta}_1 x_1 - \hat{\beta}_0)^2 + (y_2 - \hat{\beta}_1 x_2 - \hat{\beta}_0)^2 + \ldots + (y_n - \hat{\beta}_1 x_n - \hat{\beta}_0)^2, \\ RSS &= \textstyle\sum_{i=1}^{n} (y_i - \hat{\beta}_1 x_i - \hat{\beta}_0)^2. \end{aligned} \tag{22}$$

3.2.1 Assessing the Accuracy of Coefficient Estimates

When we have data about the input and the output variables, the true relationship between the two is unknown. We assume that the true relationship is of the form $y = f(x) + \epsilon$ with f some an unknown function and ϵ is a normally distributed independent error term that is a catch-all for what we miss in the model [6]. We can approximate f by a linear function between x and y as;

$$y = \beta_1 x + \beta_0 + \epsilon. \tag{23}$$

The Equation (23) provides us with the best linear approximation to the true relationship between variables x and y which is also known as the population regression line. The least squares regression coefficient estimates in Equations (19) and (20) are characteristic of the least squares regression line. The distinction is visualized by Figure 3 using a randomly generated set of 20 data points for a hypothetical model $y = 3 + 4x + \epsilon$ with epsilon randomly generated from a normal distribution.

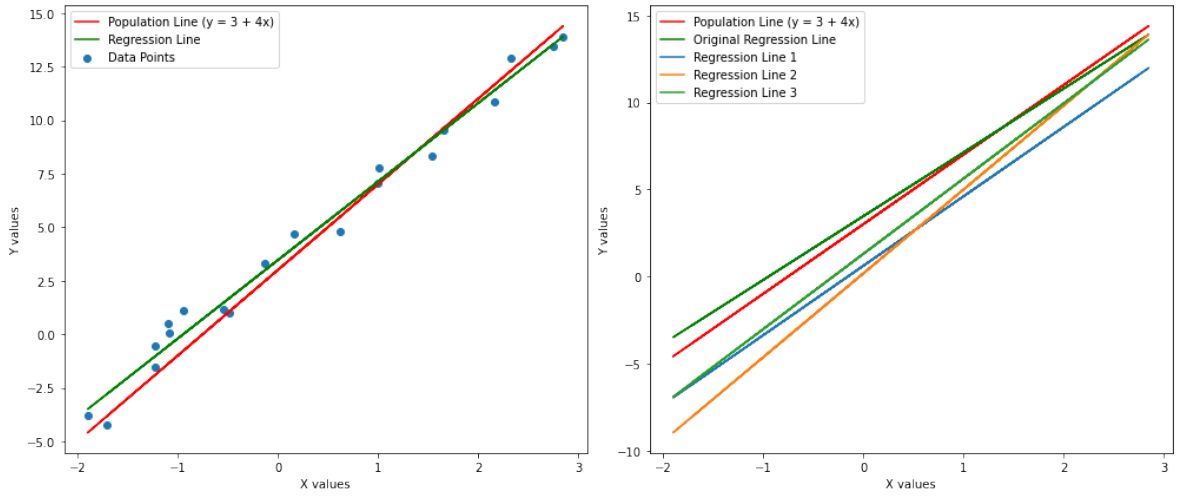

Figure 3: In the first subplot, the green line represents the linear regression line and the red line the population regression line. The second subplot includes three more regression lines for more random sets of observations.

The red line in Figure 3 shows the true relationship $f(x) = 3 + 4x$ whereas the green line shows the least squares estimate based on the input data. In real-life applications, we are

given a set of data from which we compute the least squares regression coefficients, however, the population regression lines remain unknown.

We are interested in assessing the accuracy of the least squares regression coefficients, $\hat{\beta}_0$ and $\hat{\beta}_1$. A good analogy is the sample mean and the population mean for n observations, $Z = z_1, z_2, \ldots, z_n$. An unbiased estimate for the population mean,μ is the sample mean,$\hat{\mu}$ with

$$\hat{\mu} = \bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i. \tag{24}$$

Similarly, the true population coefficients β_0 and β_1 remain unknown but, $\hat{\beta}_0$ and $\hat{\beta}_1$ provide their unbiased estimates and this is proved in Section 3.3.3. If we separately average the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ over a large number of datasets, then, the average of the estimates would be very close or even equal to the actual β_0 and β_1 of the population.

3.2.2 How close $\hat{\beta}_0$ and $\hat{\beta}_1$ are to β_0 and β_1

We again uses the μ and $\hat{\mu}$ analogy. A single $\hat{\mu}$ can underestimate or overestimate μ. Thus, the standard error,$SE(\hat{\mu})$ of a single $\hat{\mu}$ can be obtained from its variance that is given by;

$$var(\hat{\mu}) = SE(\hat{\mu})^2 = \frac{\sigma^2}{n}. \tag{25}$$

Equation (25) tells us how a single estimate $\hat{\mu}$ on average differs from μ and how the variance shrinks concerning n. For least squares regression, the estimate of the standard error is;

$$SE(\hat{\beta}_i) = RSE = \sqrt{\frac{RSS}{n-2}}, \tag{26}$$

where RSE is the residual standard error and the denominator is $(n-2)$ because we would have already estimated β_0 and β_1. The estimators $\hat{\beta}_0$ and $\hat{\beta}_1$ follow the distributions;

$$\hat{\beta}_0 \sim N\Big(\beta_0, \sigma^2\Big[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\Big]\Big), \qquad \hat{\beta}_1 \sim N\Big(\beta_1, \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\Big). \tag{27}$$

Therefore, the standard errors $SE(\hat{\beta}_0)$ and $SE(\hat{\beta}_1)$ are;

$$SE(\hat{\beta}_0) = \sqrt{\sigma^2\Big[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\Big]}, \qquad SE(\hat{\beta_1}) = \sqrt{\frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}, \tag{28}$$

where σ^2 is $Var(\epsilon)$.

3.2.3 Confidence Intervals

The standard errors in Equations (28) can be used to compute confidence intervals within which we can capture the unknown values of the population parameters β_0 and β_1. A 95% confidence interval in simple linear regression takes the form;

$$\hat{\beta}_1 \pm 2SE(\hat{\beta}_1).$$

This is to say that we have 95% chance that the interval, $[\hat{\beta}_1 - 2SE\hat{\beta}_1, \hat{\beta}_1 + 2SE\hat{\beta}_1]$ captures the true value of β_1. A similar reasoning applies to $\hat{\beta}_0$.

Null hypothesis,H_0	Alternative hypothesis,H_a
There is no relationship between x and y.	There is a relationship between x and y.

Table 2: The table highlights the null and alternative hypothesis on β_1.

3.2.4 Hypothesis Testing

In simple linear regression, standard errors are also used for hypothesis testing on coefficients. For a predictor,x and a tareget variable y, let us have a null hypothesis,H_0 and an alternative hypothesis,H_a as;

Which corresponds to testing; $H_0 : \beta_1 = 0$ versus $H_a : \beta_1 \neq 0$.

If $\beta_1 = 0$, then we have that $y = \beta_0 + \epsilon$ and this gives an implication that x is not associated with y. To test the null hypothesis, we need to determine if $\hat{\beta}_1$ is sufficiently large and far from zero so that we are sure β_1 is not zero. If $SE(\hat{\beta}_1)$ has a small value, then, even relatively small values of $\hat{\beta}_1$ provide sufficient evidence that β_1 is not zero. If $SE(\hat{\beta}_1)$ has a large value, then, $\hat{\beta}_1$ should to be large enough such that β_1 is not zero.

In practice, we use the *t-statistic* from the *t-distribution* to measure the number of standard deviations that $\hat{\beta}_1$ is away with from zero. The *t-statistic* is defined as;

$$t = \frac{\hat{\beta}_1}{SE(\hat{\beta}_1)}.$$

We compute the probability of having any number say $q \in \mathbb{R}$ such that $q \geq |t|$ assuming $\beta_1 = 0$. This probability is what we call the *p-value*. If the *p-value* is small, then we can conclude that there is an association between x and y then we reject the null hypothesis. Using the data on pH of pure water in Table 1, we have the following results;

	$coefficient$	*standard error*	*t-statistic*	*p-value*
Intercept	7.35850	0.04223	174.21542	2.41301e-12
Temperature	-0.01305	0.00094	-13.86829	8.75248e-06

Table 3: The table shows the coefficients from the simple linear model in Equation **??**, the *standard errors*, *t-values* and *p-values*.

- The intercept,$\hat{\beta}_0$ is a positive coefficient and has a very high *t-value*, indicating that it is significantly different from zero.
- The temperature coefficient,$\hat{\beta}_1$ is negative with a big negative t-value indicating that there is a significant inverse relationship between temperature and pH. The associated *p-value* is very low, confirming the statistical significance of this inverse relationship.

3.2.5 The Residual Standard Error(RSE)

When we reject the null hypothesis and accept the alternative hypothesis, then we would like to know the extent to which the model fits the data. We assess the quality of the simple linear regression fit using the *residual standard error*(RSE) and the R^2-*statistic* [7].

By the Equation (23), the presence of ϵ tells us that we are not able to accurately predict the output,y when we have the input feature,x. The *residual standard error* is an estimate of the standard deviation of ϵ which is typically the average amount by which the estimates $\hat{y}$'s deviate from the true regression line of the population. It is given by the formula;

$$RSE = \sqrt{\frac{1}{n-2}RSS}. \tag{29}$$

As we saw in Section 3.2, the *residual sum of squares*, $RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ and thus;

$$RSE = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{30}$$

- Residual standard *error* (RSE): This is a measure of the lack of fit of a given linear regression model to the data. For a model that fits the data well, the RSE value is small and for a model that does not fit the data well, the RSE value is large indicating that the estimates $\hat{y}_i$'s are far from the ground truths y_i's.

- The R^2-*statistic*: This is a measure of fit that quantifies the proportion of variance in the response variable,y that is explained by the predictor variable,x. The R^2-*value* is calculated by the formula;

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}.$$

  where total sum of squares, $TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$ and RSS is as in Equation (21),

Now, using the data on the variation of pH of pure water with temperature in Table 1, we can observe the following about the entire constructed model;

Statistical Quantity	Value
RSE	0.0768
R^2	0.9697

Table 4: The table shows the *residual standard error* and the R^2 value based on the data.

- The residual standard error, RSE tells us that, on average, the predicted pH values deviate from the actual pH values by about 0.07862 units and this is a small *residual standard error* indicating a better fit of the model to the data.

- The R^2-*value* of approximately, 0.9697 indicates that about 96.97% of the variability in pH of pure water can be explained by the linear relationship with temperature. This tells us that the model provides a good fit for the observed trend between pH values of pure water and temperature.

3.3 The Foundation of Multiple Linear Regression

In practice, we usually have more than one predictor for the output variable(s). Instead of fitting separate linear regression models for each predictor, we extend the model to take in more predictors [8]. The multiple linear regression model takes the form;

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m + \epsilon, \tag{31}$$

where each β_i represents an association between some predictor,x_i and the response,y, m is the total number of features and ϵ is the error term. Using the data in the Table, we can have the following multiple linear regression model;

$$pH = \beta_0 + \beta_1(Temperature) + \beta_2(\delta_w) + \beta_3(K_w). \tag{32}$$

We can have some of the predictors in the model as; powers of the original predictor, interactions of the predictors or else as a mixture of powers and interactions of the predictors;

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_m x^m + \epsilon, \tag{33}$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \cdots + \beta_m x_m + \epsilon, \tag{34}$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} {x_1}^2 + \beta_{12} x_1 x_2 + \cdots + \beta_m x_m + \epsilon. \tag{35}$$

We hold the following assumptions which apply to both multiple and simple linear regression;

$$Cov(\epsilon_i, \epsilon_j) = 0, \quad \text{for} \quad i \neq j, \quad \text{with} \quad i, j \in \{0, 1, 2, ..., n\}.$$
$$\mathbb{E}(\epsilon_i) = 0 \quad \text{and} \quad Var(\epsilon_i) = \sigma^2 \quad \forall i \quad \text{with} \quad i \in \{0, 1, 2, ..., n\}.$$

The three Equations (33), (34) and (35) are all multiple linear regression models, here, linearity is in terms of the regression coefficients. If we consider a sample of size n, then the sample version of the Equation (31) is;

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_m x_{im} + \epsilon_i, \quad \text{with} \quad i \in \{0, 1, 2, ..., n\}. \tag{36}$$

For each of the n observations from a given dataset, we have the following system of equations;

$$\begin{aligned} y_1 &= \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \cdots + \beta_m x_{1m} + \epsilon_1, \\ y_2 &= \beta_0 + \beta_1 x_{21} + \beta_2 x_{22} + \cdots + \beta_m x_{2m} + \epsilon_2, \\ &\vdots \qquad\qquad\qquad\qquad \vdots \\ y_n &= \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_m x_{nm} + \epsilon_n. \end{aligned} \tag{37}$$

Expressing the above system of equations in matrix form yields;

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1m} \\ 1 & x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}.$$

The above matrix form can be condensed to;

$$\underbrace{y}_{n\times 1} = \underbrace{X}_{n\times p} \cdot \underbrace{\beta}_{p\times 1} + \underbrace{\epsilon}_{n\times 1}, \tag{38}$$

with $p = m + 1$.

3.3.1 A Multiple Linear Regression Task

We extend the Table 1 to include two more features that is; conductivity values of pure water,δ_w in $\mu S/cm \times 10^{-2}$ and dissociation constants of pure water,K_w in $mol^2dm^{-6} \times 10^{-14}$.

Temperature(C)	$\delta_w(\mu S/cm)$	$K_w(mol^2dm^{-6})$	pH value
0	0.1162	0.114	7.47
10	2.312	0.293	7.27
20	4.205	0.681	7.08
25	5.512	1.008	7.00
30	7.105	1.471	6.92
40	11.298	2.916	6.77
50	17.071	5.476	6.63
100	77.697	51.300	6.14

Table 5: Shows temperature, conductivity values, dissociation constants and pH values of pure water.

For the data in the Table 5, the multiple linear regression line is of the form;

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3, \tag{39}$$

where; $\hat{\beta}_1$,$\hat{\beta}_2$ and $\hat{\beta}_3$ are gradients corresponding to the features; temperature(x_1), electric conductivity(x_2), dissociation constant(x_3) respectively and $\hat{\beta}_0$ is a constant. The dependent variable estimate,$\hat{y}$ denotes the pH value of pure water depending on the three input features, x_1, x_2 and x_3.

The Equation (39) can be rewritten as;

$$\hat{y} = \sum_{i=0}^{3} \hat{\beta}_i x_i = \hat{\beta}_0 x_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3, \tag{40}$$

with $x_0 = 1$.

We convert the Equation (40) to matrix form. Then insert feature values and output values;

$$y = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{bmatrix},$$

$$\underbrace{\begin{bmatrix} 7.47 \\ 7.27 \\ 7.08 \\ 7.00 \\ 6.92 \\ 6.77 \\ 6.63 \\ 6.14 \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} 1 & 0 & 0.1162 & 0.114 \\ 1 & 10 & 2.312 & 0.293 \\ 1 & 20 & 4.205 & 0.681 \\ 1 & 25 & 5.512 & 1.008 \\ 1 & 30 & 7.105 & 1.471 \\ 1 & 40 & 11.298 & 2.916 \\ 1 & 50 & 17.071 & 5.476 \\ 1 & 100 & 77.697 & 51.300 \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{bmatrix}}_{\hat{\beta}}.$$

Given the feature matrix,X, we can find the values of parameters $\hat{\beta}_i$ for $i \in \{0, 1, 2, 3\}$ as;

$$\hat{\beta} = (X^T X)^{-1} X^T y, \tag{41}$$

where $\hat{\beta} = [\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3]^T$ with T for transpose. The theorem in subsection 3.3.2 shows us how we can arrive at the Equation (41). Using a python library,NumPy, we utilize the NumPy arrays to handle the data which is in two dimensions. If we had a large dataset, we would split it into training and validation sets. Then, we implement multiple linear regression using the *LinearRegression* model from the *sklearn.linear_model* which is provided by the *scikit-learn* library in python. This process fits a multiple linear regression model to data and yields the optimal parameters as;

$$\hat{\beta}_0 = 7.464836, \quad \hat{\beta}_1 = -0.02479, \quad \hat{\beta}_2 = 0.23045, \quad \hat{\beta}_3 = 0.019051, \tag{42}$$

Thus, the equation of the model becomes;

$$y = 7.464836 - 0.02479x_1 + 0.23045x_2 + 0.019051x_3. \tag{43}$$

Prediction: After creating the model, we test its power to generalize using data it has not seen before. New $[x_0, x_1, x_2, x_3] = [1, 15, 4.833, 0.895]$. The algorithm predicted a pH value, $\hat{y} = 7.2213$.

3.3.2 The Least Squares Estimation

We can still employ the least squares approach to fit a model to data with multiple predictors. The coefficients β_0, β_1, $\cdots$, β_m are unknown and we would like to estimate them in such a way that they minimize the squared loss function,

$$\min_{\hat{\beta}} \ \ell(\hat{\beta}) = \min_{\hat{\beta}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \min_{\hat{\beta}} \sum_{i=1}^{n} {e_i}^2 \quad \text{with} \quad i \in \{0, 1, 2, ..., n\}. \tag{44}$$

If we let $e = e_i \in \mathbb{R}^n$ and $\hat{y} = \hat{y}_i = X\hat{\beta} \in \mathbb{R}^n$, then, e is obtained as $e = y - \hat{y}$. By the Equation (38), the minimization problem becomes;

$$\min_{\hat{\beta}} \ \ell(\hat{\beta}) = \min_{\hat{\beta}} ||e||^2 = \min_{\hat{\beta}} ||y - X\hat{\beta}||^2. \tag{45}$$

Theorem: If $X^T X$ is nonsingular, then the least square estimate of β is [9]

$$\hat{\beta} = (X^T X)^{-1} X^T y. \tag{46}$$

Proof: For the proof of this theorem, the following ideas about gradient of a function of multiple variables are required;

$$\begin{aligned}
\frac{\partial}{\partial X}(X^T a) &= \tfrac{\partial}{\partial X}(a^T X) = a, \\
\frac{\partial}{\partial X}(||X||^2) &= \tfrac{\partial}{\partial X}(X^T X) = 2X, \\
\frac{\partial}{\partial X}(X^T A X) &= 2AX, \\
\frac{\partial}{\partial X}(||AX||^2) &= \tfrac{\partial}{\partial X}(X^T A^T A X) = 2A^T A X,
\end{aligned}$$

where X and a are vectors of same dimension,n and A is an $n \times n$ symmetric matrix.
For the proof of the theorem, we use the identity $||U - V||^2 = ||U||^2 + ||V||^2 - 2U^T V$. Using Equation (45) and the stated identity, we have that;

$$\begin{aligned}
\ell(\hat{\beta}) &= ||y||^2 + ||X\hat{\beta}||^2 - 2(X\hat{\beta})^T y, \\
&= y^T y + \hat{\beta}^T X^T X \hat{\beta} - 2\hat{\beta}^T X^T y.
\end{aligned}$$

We compute the derivative of the loss function with respect to $\hat{\beta}$ utilizing the ideas in the equations stated after the theorem statement;

$$\frac{\partial \ell}{\partial \hat{\beta}} = 0 + 2X^T X\hat{\beta} - 2X^T y. \tag{47}$$

We set the gradient function,$\frac{\partial \ell}{\partial \hat{\beta}}$ to zero and this yields the least squares normal equations as;

$$X^T X\hat{\beta} = X^T y. \tag{48}$$

Multiplying both sides of Equation above by $(X^T X)^{-1}$, we obtain $\hat{\beta}$ as;

$$\hat{\beta} = (X^T X)^{-1} X^T y. \tag{49}$$

Remark: The very first normal equation of $X^T X\hat{\beta} = X^T y$ is;

$$n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} x_{i1} + \hat{\beta}_2 \sum_{i=1}^{n} x_{i2} + \cdots + \hat{\beta}_m \sum_{i=1}^{n} x_{im} = \sum_{i=1}^{n} y_i. \tag{50}$$

Dividing through by n yields;

$$\hat{\beta}_0 + \hat{\beta}_1 \bar{x}_1 + \hat{\beta}_2 \bar{x}_2 + \cdots + \hat{\beta}_m \bar{x}_m = \bar{y}. \tag{51}$$

This shows us that the centroid of the data is on the least squares regression plane. This signifies that the plane is a good representation of the central tendency of the data.

Remark

The fitted values of the least squares model are;

$$\hat{y} = X\hat{\beta} = \underbrace{X(X^T X)^{-1} X^T}_{H} y = Hy. \tag{52}$$

By the Equation (EQN:squares), the residuals can be visualized as;

$$e = y - \hat{y} = y - Hy = (I - H)y. \tag{53}$$

The matrix, $H \in \mathbb{R}^{n \times n}$ is known as the *hat matrix* and it has the following characteristics;

- It is symmetric, that is; $H^T = H$.
- It is idempotent, that is; $H^2 = H$.
- $H(I - H) = O$ where O is the zero matrix.

3.3.3 Point Estimation in Multiple Linear regression

The least squares estimator, $\hat{\beta}$ is an unbiased linear estimator for β. This still holds for a simple linear regression case. Under the assumptions of multiple linear regression,

$$\mathbb{E}(\hat{\beta}) = \beta. \tag{54}$$

That is, $\hat{\beta}$ is a component-wise unbiased estimator for β; $\mathbb{E}(\hat{\beta}_i) = \beta_i \quad \forall i \in \{0, 1, 2, ..., m\}$. By the theorem in the subsection 3.3.2, we have that;

$$\hat{\beta} = (X^T X)^{-1} X^T y. \tag{55}$$

However, $y = X\beta + \epsilon$, thus we have that;

$$\begin{aligned} \hat{\beta} &= (X^T X)^{-1} X^T \cdot (X\beta + \epsilon), \\ &= (X^T X)^{-1} X^T \cdot X\beta + (X^T X)^{-1} X^T \cdot \epsilon, \\ &= (X^T X)^{-1}(X^T X)\beta + (X^T X)^{-1} X^T \cdot \epsilon, \\ &= I\beta + (X^T X)^{-1} X^T \cdot \epsilon, \\ &= \beta + (X^T X)^{-1} X^T \cdot \epsilon. \end{aligned} \tag{56}$$

Taking the expectation,$\mathbb{E}$ on both sides of Equation **??** yields;

$$\begin{aligned} \mathbb{E}(\hat{\beta}) &= \mathbb{E}(\beta + (X^T X)^{-1} X^T \cdot \epsilon), \\ &= \beta + (X^T X)^{-1} X^T \underbrace{\mathbb{E}(\epsilon)}_{0}, \\ \mathbb{E}(\hat{\beta}) &= \beta. \end{aligned} \tag{57}$$

Remark

1. Similar to the simple linear regression case, in multiple linear regression we can; assess the accuracy of coefficient estimates, estimate confidence intervals, carry out hypothesis testing and also assess the accuracy of the entire model.

2. When we fit a linear regression model to a given dataset, some problems may arise and these need to be handled carefully. Some of these problems can be dealt with during the data pre-processing stage of machine learning. They include [10];

   - Outliers.
   - High leverage points.
   - Collinearity of predictors.
   - Correlation of error terms.
   - Non-constant variance of error terms.
   - Non-linearity of the response-predictor relationship.

4 Conclusion

In this work, we have explored the profound interconnection between mathematics and machine learning, specifically through the lens of linear regression models. We began the study with a comprehensive overview of machine learning and how it is important to our world

today. We looked at various mathematical topics that underpin machine learning algorithms and we established a foundation for understanding how mathematical principles are indispensable to the field of machine learning.

Both simple and multiple linear regression, served as a focal point to aid us in illustrating the theoretical as well as practical applications of mathematical principles and concepts in machine learning. Using data on the pH of pure water, we demonstrated how machine learning models are employed to derive meaningful insights and predictions from data with the vast utility of mathematical frameworks. Through the examples provided, we showed how linear algebra facilitates the representation and manipulation of data, how calculus is used to optimize model parameters and how statistics provides the framework for making inferences about data.

In our detailed examination of simple and multiple linear regression, we showcased the step-by-step mathematical processes involved in; model formulation, parameter estimation, model evaluation and interpretation of results. Through these examples, we emphasized that mathematics is not just a theoretical foundation but also a practical tool that drives the development and refinement of machine learning models. The mathematical rigor involved in linear regression models ensures that the predictions are not only accurate but also interpretable, which is essential for scientific and industrial applications.

The applications of mathematics in machine learning extend far beyond the examples covered in this work. From the simplest of algorithms to the most complex neural networks, mathematics forms the backbone of machine learning enabling the creation of models that can learn from data and make useful predictions. As the field of machine learning continues to evolve, the integration of advanced mathematical techniques will undoubtedly lead to more sophisticated powerful algorithms and this is a possible orientation for future studies.

Acknowledgments

The authors would like to acknowledge support from AIMS Ghana. The authors would like to thank Dr. Sara Abdelazeem Hassan Abass for useful comments.

References

[1] Jason Brownlee. Basics of linear algebra for machine learning. Machine Learning Mastery.

[2] Jason Brownlee, Stefania Cristina, and Mehreen Saeed. Calculus for machine learning. Machine Learning Mastery, 2022.

[3] Martin Anthony. Aspects of discrete mathematics and probability in the theory of machine learning. Discrete applied mathematics, 156(6):883-902, 2008.

[4] Hasan Halit Tali and Ceren Çelti. An approach towards the least-squares method for simple linear regression. Advances in Artificial Intelligence Research, 2(2):38-44, 2022.

[5] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer, 2009.

[6] Bruce D McCullough. Assessing the reliability of statistical software: Part i. The American Statistician, 52(4):358-366, 1998.

[7] Frank Emmert-Streib and Matthias Dehmer. Evaluation of regression models: Model assessment, model selection and generalization error. Machine learning and knowledge extraction, 1(1):521-551, 2019.

[8] Dastan Maulud and Adnan M Abdulazeez. A review on linear regression comprehensive in machine learning. Journal of Applied Science and Technology Trends , 1(2):140-147, 2020.

[9] David A Freedman. Statistical models: theory and practice. cambridge university press, 2009.

[10] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. An introduction to statistical learning , volume 112. Springer, 2013.