ISSN: 2754-6705

Journal of Mathematical & Computer Applications

SCIENTIFIC Research and Community

Research Article

Open Access

From Equations to Predictions: Understanding the Mathematics and Machine Learning of Multiple Linear Regression

Vesna Knights[1]*and Marıja Prchkovska[2]

[1]University "St Kliment Ohridski"- Bitola, Faculty or Technology and Technical Science -Veles, 7000, Bitola, Republic of North Macedonia

[2]Mother Teresa University, Faculty of Computer Science, Informatics, 1000,Skopje, Republic of North Macedonia

ABSTRACT

In this paper, the core concepts of multiple linear regression are explored, with a focus on its mathematical foundations and integration with machine learning principles. The objective is to bridge the gap between theory and practical application, providing readers with a comprehensive understanding of this versatile method and highlighting its synergy with traditional statistical approaches and modern computational methods. The paper begins by applying multiple linear regression to predict wine quality based on physicochemical attributes, using a comprehensive dataset. The least squares method is used to estimate regression coefficients, facilitating the construction of a predictive model. The study also encompasses the testing of assumptions such as homoscedasticity and normality of residuals, along with the assessment of autocorrelation to ensure model robustness. To illustrate the practical implementation of multiple linear regression, a demonstration using PyTorch, a popular deep learning framework, is provided. A linear model is defined, and the significance of gradient descent in optimizing model parameters is elucidated. Additionally, the paper covers topics such as data preprocessing, model evaluation, and insights into interpreting regression results.

Furthermore, the performance of linear regression is evaluated in comparison to decision trees, random forests, and support vector regression, showcasing the versatility of this classic technique. By presenting a holistic view of multiple linear regression, emphasizing its mathematical foundations, practical implementation, and integration with machine learning, researchers and practitioners are empowered to leverage the potential of linear regression across various domains.

***Corresponding author**

Vesna Knights, University "St Kliment Ohridski"- Bitola, Faculty or Technology and Technical Science -Veles, 7000, Bitola, Republic of North Macedonia.

Received: March 19, 2024; **Accepted:** March 21, 2024, **Published:** April 03, 2024

Keywords: Linear Regression, Machine Learning, Mathematical Foundations, Model Implementation, Predictive Modeling

Introductıon

Multiple linear regression, a foundational statistical technique, plays a pivotal role in modeling the intricate relationships that exist between a dependent variable (response) and one or more independent variables (predictors) [1-3]. This method involves fitting a linear equation to observed data, enabling us to comprehend, quantify, and predict associations among variables. Its versatility extends across a multitude of domains, including economics, marketing, and scientific research, where it serves as an invaluable tool for making predictions and unraveling intricate variable connections [4-6].

At its core, multiple linear regression is a supervised learning algorithm. It's particularly adept at handling continuous real-numbered target variables [7-9]. This method establishes relationships between the dependent variable, denoted as 'y,' and one or more independent variables, collectively represented as 'x,' through the creation of a best-fit line. This process operates under the fundamental principle of ordinary least squares (OLS) or mean square error (MSE) [10-12]. OLS serves as a method to estimate the unknown parameters of the linear regression function, with its primary objective being the minimization of the sum of squared differences between the observed dependent variable and the values predicted by the linear regression function [10,11].

This paper embarks on an exploration of the intricate world of multiple linear regression, aiming to bridge the chasm between theoretical understanding and practical application. The following sections delve into the mathematical foundations of this method, in alignment with the insights presented by Kutner, Nachtsheim, Neter, and Li [13]. The discussion extends further, encompassing the synergistic relationship between traditional statistical approaches and contemporary computational methods. Our journey begins with the practical application of multiple linear regression to predict wine quality based on physicochemical attributes, employing an extensive dataset [14]. Leveraging the least squares method, we estimate regression coefficients, paving the way for the construction of a predictive model [15]. Assumptions, such as homoscedasticity and normality of residuals, are rigorously tested. Additionally, we assess autocorrelation, ensuring the robustness of our model.

On the practical implementation of multiple linear regression, this paper provides a hands-on demonstration using PyTorch, a well-regarded deep learning framework [16-18]. Within this context, a linear model is defined, emphasizing the critical role of gradient descent in optimizing model parameters [18]. Subsequent sections of the paper delve into essential topics such as data preprocessing, model evaluation, and insightful approaches for interpreting regression results [19].

Furthermore, this study broadens its scope by evaluating the performance of linear regression against other contemporary machine learning techniques, including decision trees, random forests, and support vector regression [17,20,21]. This comparative analysis underscores the enduring adaptability of this time-honored method within the domain of predictive modeling. By offering a comprehensive perspective on multiple linear regression, emphasizing its mathematical foundations, practical applications, and integration with modern machine learning, this work aims to empower researchers and practitioners, equipping them to leverage the substantial potential of linear regression across various fields [22].

Materıal and Methods

Material

For the purpose of this study, a database from Cortez et al. (2009) was utilized [14]. The dataset includes the following attribute information:

Input variables (based on physicochemical tests):

Input variables (based on physicochemical tests)
1 - fixed acidity (tartaric acid - g / dm^3)
2 - volatile acidity (acetic acid - g / dm^3)
3 - citric acid (g / dm^3)
4 - residual sugar (g / dm^3)
5 - chlorides (sodium chloride - g / dm^3
6 - free sulfur dioxide (mg / dm^3)
7 - total sulfur dioxide (mg / dm^3)
8 - density (g / cm^3)
9 - pH
10 - sulphates (potassium sulphate - g / dm3)
11 - alcohol (% by volume)
Output variable (based on sensory data):
12 - quality (score between 0 and 10)

Methods

The Collection of the Data

The data for this study were obtained from the dataset provided by Cortez et al. in 2009 [14]. Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 47(4), 547-553]. The dataset contains information on physicochemical attributes of wine, making it suitable for the analysis and implementation of multiple linear regression.

Statistical Analysis

The statistical analysis in this study primarily involves the implementation of Multiple Linear Regression.

Implementation of Multiple Linear Regression

Objective: The objective of Multiple Linear Regression is to find the estimates of the regression coefficients (β0, β1, β2, ..., βp) that minimize the sum of the squared differences between the observed values (y) and the values predicted by the linear regression model.

Loss Function: Multiple Linear Regression employs a loss function that measures the squared differences between the observed and predicted values. The ultimate goal is to minimize the sum of squared residuals.

Assumptions

Multiple Linear Regression assumes that the errors (residuals) are normally distributed with constant variance (homoscedasticity) and does not require a specific probabilistic model for the errors.

Linear Regression Model

In simple linear regression, with one independent variable (X) and one dependent variable (Y), the model is defined as:

$Y = \beta 0 + \beta iX$

For Multiple Linear Regression, where there are multiple independent variables (x1, x2, ..., xp), the model is represented as:

$Y(yi)=\beta 0+\beta 1x1+\beta 2x2+...+\beta pxp$

where Y(yi) presents the observed value

İn order to make predictions, the model is expressed as:

$Y` = \beta 0 + \beta 1X1 + \beta 2X2 + ... + \beta pXp + \varepsilon$

Y` represents the predicted value of the dependent variable Y for a given set of independent variables.

β0 is the y-intercept, representing the expected value of Y when all independent variables are 0.

β1, β2, ..., βp are the coefficients (slopes) for the independent variables.

ε (Error or Residual) is the difference between the actual observed value (Y(yi)) and the predicted value (Y'). Matematically:

$$\varepsilon = yi-\hat{y}i$$

The primary objective of linear regression is to determine the coefficients that minimize the sum of squared errors (SSE) and provide an accurate model for predicting the target variable based on the input features. This is achieved through methods like the least squares approach, optimizing the coefficients to create a predictive model.

In the context of machine learning, this approach allows us to find the best-fitting linear model that captures the relationship between the independent variables and the dependent variable, facilitating accurate predictions on new, unseen data.

Results

The dataset comprises m = 1599 examples and n = 11 independent variables (Table 1). The target variable, 'quality,' falls within a range of 0 to 10, while the remaining eleven variables represent various physicochemical attributes. Given the presence of multiple independent variables, we are tasked with fitting a multiple linear regression model.

The equation for multiple linear regression can be expressed as:

$(Y(yi)) = \beta 0 + \beta 1 *$ fixed acidity $+ \beta 2 *$ volatile acidity $+ \beta 3 *$ citric acid $+ \beta 4 *$ residual sugar $+ \beta 5 *$ chlorides $+ \beta 6 *$ free sulfur dioxide $+ \beta 7 *$ total sulfur dioxide $+ \beta 8 *$ density $+ \beta 9 *$ pH $+ \beta 10 *$ sulphates $+ \beta 11 *$ alcohol (1)

Table 1: The Dataset of Wine

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	Y(yi)
	fixed_ acidity	volatile_ acidity	citric_ acid	Residual _sugar	chlorides	free_ sulfur_ dioxide	total_ sulfur_ dioxide	density	pH	sulphates	alco-hol	quality
y0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
y1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
y2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	6
…	…	…		…	…	…	…	…	…	…	…	…
y1599	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	6

Before making predictions with linear regression, it's essential to estimate the coefficients β0 and βi from the available data. The estimation of βj, representing the coefficients, can be calculated using the following formula:

$$\beta j = \sum_{i=1}^{n}(x_{ij} - \bar{xj})(\text{yi} - \bar{\text{y}}) / \sum_{i=1}^{n}(x_{ij} - \bar{xj})^2 \quad (2)$$

Where, xij is the value of the j-th feature for the i-th data point (e.g., fixed acidity, volatile acidity, citric acid, etc.).
$\bar{x}$j is the mean of the j-th feature across all data points.

Y is the mean of the dependent variable (quality) across all data points.

The intercept term (β0) can be computed as:
Intercept

$$\beta_0 = \bar{\text{y}} - \sum_{i=1}^{n} \beta_j \bar{xj} \quad (3)$$

Instead of performing complex calculations manually using the given formulas to estimate the coefficients, (β0, β1, β2, β3, ..., β11) we leveraged machine learning techniques and libraries to automate this process. The coefficients were computed using the following code:

Table 2: Code for Computed Coefficients

Code

import statsmodels.formula.api as smf
# Update the formula to encompass the relevant variables 	
formula = "quality ~ fixed_acidity + volatile_acidity + citric_acid + residual_sugar + chlorides + free_sulfur_dioxide + total_sulfur_ dioxide + density + pH + sulphates + alcohol"
# Fit the regression model 	
est = smf.ols(formula=formula, data=data).fit()
# Display the summary of the regression analysis 	
print(est.summary())

By utilizing this approach, we achieved a more efficient and automated means of estimating the coefficients, allowing us to focus on the interpretation and insights drawn from the results

The results of the multiple linear regression analysis are summarized in the following table:

Table 3: The Results of The Multiple Linear Regression Analysis

Variable	Coefficient	P-value
Intercept	21.9652	0.300
Fixed Acidity	0.0250	0.336
Volatile Acidity	-1.0836	0.000
Citric Acid	-0.1826	0.215
Residual Sugar	0.0163	0.276
Chlorides	-1.8742	0.000
Free Sulfur Dioxide	0.0044	0.045
Total Sulfur Dioxide	-0.0033	0.000
Density	-17.8812	0.409
pH	-0.4137	0.031
Sulphates	0.9163	0.000
Alcohol	0.2762	0.000

These coefficients represent the estimated associations between each independent variable and the dependent variable, quality. For instance, the coefficient for volatile acidity (-1.0836) indicates that an increase in volatile acidity is correlated with a decrease in wine quality. Conversely, the coefficient for alcohol (0.2762) suggests that a higher alcohol content tends to be associated with higher wine quality.

This comprehensive analysis contributes valuable insights into the collective impact of these physicochemical attributes on wine

The next step is preparing data for a machine-learning model by performing:

- Separating the features (X) and the target variable (y- quality).
- Standardizing the features using `**StandardScaler**`, by performing the following transformations on each feature: It calculates the mean (μ) and standard deviation (σ) of each feature in the training data.
- For each feature, it subtracts the mean (μ) and then divides by the standard deviation (σ):

$$X_{\text{standardized}} = (X-\mu)/\sigma$$

Where: X is the original feature value, $X_{\text{standardized}}$ is the standardized feature value.
Splitting the data into training and testing sets using train_test_split (X, y, random_state = 0, test_size=0.25).

Once these coefficients have been calculated, they can be used to make predictions for new data points by plugging in the values of the independent variables into the linear regression equation.

ŷi - Predicted values based on the linear model

$$\hat{y}i = \beta 0 + \beta jX + ei \quad (4)$$

The error term (e) is known as a residual, represents the difference between the actual observed values (yi) and the predicted values (ŷi) for each data point (i).

Table 4: Code for Calculated Residuals

Code
residuals = y_train. values-y_pred mean_residuals = np. mean(residuals) print ("Mean of Residuals {}". format(mean_residuals)) Mean of Residuals 1.2741174994864182e-16

Residuals are calculated by subtracting the predicted values (y_pred) from the actual values (y_train). These residuals represent the differences between the observed (actual) values and the values predicted by linear regression model for each data point in your training dataset.

mean_residuals calculates the mean (average) of the residuals.

The output tha is provided, "Mean of Residuals 1.2741174994864182e-16," indicates that the mean of the residuals is extremely close to zero but not exactly zero. The value is approximately, which is a very small number.

In theory, the mean of residuals should ideally be exactly zero for a well-fitted linear regression model. Indicating that the linear regression model is reasonably well-calibrated on the training data, and, on average, it does not exhibit systematic bias in its predictions.

In the context of regression analysis, homoscedasticity,indicates that the residuals exhibit consistent or nearly consistent variance along the regression line. To assess this, we can create a scatter plot of the error terms against the predicted values, ensuring that there is no discernible pattern in the residuals.

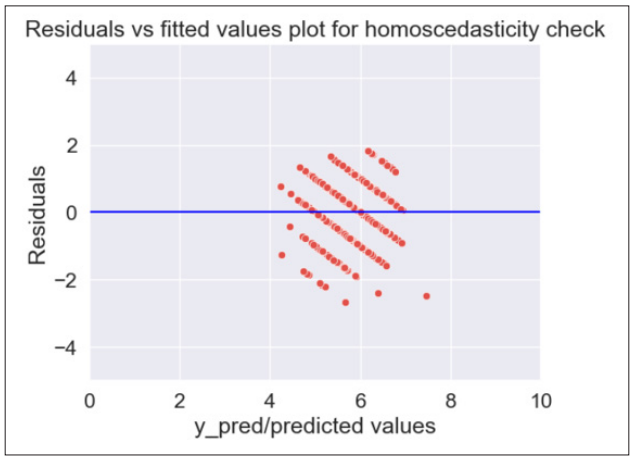

Figure 1: Presence of Heteroscedasticity in the Regression Analysis

The graphical method involves visualizing the relationship between the error terms and predicted values to identify any patterns that may indicate the presence of heteroscedasticity in the regression analysis.

By Using Goldfeld-Quandt test, heteroscedasticity is tested.

Table 5: Code for Testing Heteroscedasticity

Code
residuals = y_train. values-y_pred mean_residuals = np. mean(residuals) print ("Mean of Residuals {}". format(mean_residuals)) Mean of Residuals 1.2741174994864182e-16

In statistical analysis, the Goldfeld-Quandt test is commonly employed to assess homoscedasticity, a concept denoting the assumption that the variance of errors (residuals) in a regression model remains consistent irrespective of the levels of independent variables. Homoscedasticity holds significance in regression analysis as it signifies that the model's errors exhibit uniform variability, thereby contributing to the reliability of the model's performance.

When interpreting the Goldfeld-Quandt test results, the pivotal element is the p-value. In the context of the obtained p-value in wine analysis (0.9197664304253765), it signifies the following hypotheses:

Null Hypothesis (H0): The error terms exhibit homoscedasticity, implying they possess a constant variance.

Alternative Hypothesis (Ha): The error terms display heteroscedasticity, indicating varying variance.

In our specific case, the calculated p-value (0.9197664304253765) significantly exceeds the conventional significance level of 0.05. When the p-value surpasses the significance level, it implies that there is insufficient evidence to support the conclusion that the error terms exhibit heteroscedasticity. The null hypothesis implies that the error terms maintain homoscedasticity.

Homoscedasticity is a fundamental assumption in linear regression models. When this assumption is met, it signifies that the errors in the model consistently vary across different levels of the independent variables. This uniformity ensures that the model's predictions maintain reliability across the entire spectrum of predictor values. This uniformity facilitates a clearer interpretation of the relationship between the dependent and independent variables.

The Goldfeld-Quandt test's outcome [('F statistic', 0.8906577345903255), ('p-value', 0.9197664304253765)] is considered favorable as it supports the fundamental assumption of homoscedasticity in linear regression. This assumption is crucial for ensuring the model's reliability, interpretability, and validity of statistical inferences derived from the model.

Checking for the normality of error terms (residuals) is an important step in regression analysis to assess whether the residuals follow a normal distribution, which is one of the assumptions of linear regression. The normality of residuals implies that the errors are normally distributed around zero, indicating that the model is appropriate for the data.

Check for Normality of error terms/residuals
p = sns. distplot (residuals, kde=True)
p = plt. title ('Normality of error terms/residuals')

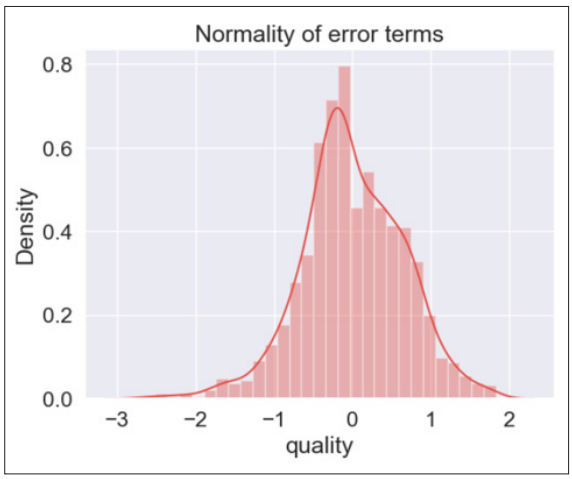

Figure 2: KDE Histogram of Normality of Error Terms (Residuals)

The unit on the x-axis of the histogram and KDE (Kernel Density Estimation) plot for the normality of error terms/residuals depends on the values of the residuals themselves. The x-axis represents the range of values that the residuals are taken.

In the context of your specific analysis, the x-axis likely represents the range of residual values. These residual values are the differences between the actual observed values (yi) and the predicted values (ŷi) for each data point in your dataset.

For example, our regression problem where the dependent variable (quality) has values ranging from 0 to 10, and the model predictions (ŷi) also fall within this range, then the residuals on the x-axis would typically be centered around zero (representing the cases where the model predictions are close to the actual values), and the range would extend to both positive and negative values, depending on how much the predictions deviate from the actual values.

Autocorrelation is another statistical concept used to analyze and understand patterns in data. It is a statistical measure that assesses the linear relationship between a time series and its lagged values (previous observations). It is often used to detect patterns or correlations within a time series data. Autocorrelation can help identify periodicity, trends, or seasonality in time series data.

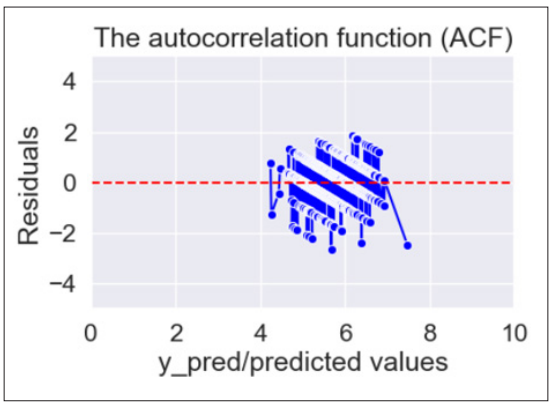

Figure 3: Autocorrelation Function

The autocorrelation function (ACF) is used to plot the correlation between a time series and its lagged values at various lags.

The Ljung-Box test is a statistical test used to check for the presence of autocorrelation in time series data or in the residuals of a regression model. It assesses whether the past values of a series (lags) are correlated with the current values.

The null hypothesis of the Ljung-Box test is that there is no autocorrelation in the data, meaning that the values at different lags are not significantly correlated. The alternative hypothesis is that there is autocorrelation present, meaning that the values at different lags are correlated.

Minimum lb_stat is value, 2.091432890259537, of the Ljung-Box statistic calculated for a specific lag or set of lags. It indicates the magnitude of autocorrelation in the residuals at those lags.

lb_pvalue is greater than chosen significance level, as it is in the results (0.07947300165019978 > 0.05), it suggests that the Ljung-Box statistic is not statistically significant. This means that there is no strong evidence to conclude that autocorrelation is present in the residuals at the specified lags.

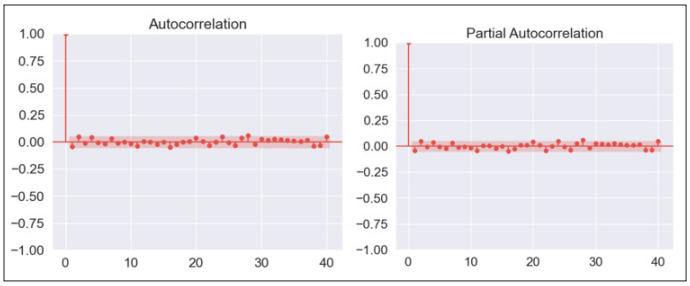

Figure 4: Autocorrelation **Figure 5:** Partial Autocorrelation

Autocorrelation function (ACF) plot and partial autocorrelation function (PACF) plot are commonly used in time series analysis to understand the autocorrelation structure of a time series or the residuals of a time series model. The plots help identify the presence of autocorrelation at different lags and can guide the selection of appropriate models for time series data.

The observed pattern in the plot indicates the presence of autocorrelation because there is a spike that extends beyond the red confidence interval region. This suggests that there may be underlying dependencies or patterns in the data, possibly related to seasonality or other factors. It's important to further investigate and consider these autocorrelations when analyzing the time series data.

In the domain of linear regression analysis, a paramount component is the Loss Function. This integral element plays a pivotal role in evaluating the model's performance in terms of its ability to capture the underlying relationship between the independent variable (often denoted as X) and the dependent variable (Y).

The Sum of Squared Errors (SSE) is employed as an essential indicator of the overall goodness of fit of the linear regression model. It quantifies the collective magnitude of squared residuals, offering valuable insights into the model's ability to accurately represent the observed data.

Least squares method, which minimizes the sum of squared differences between the observed Y values and the predicted $\hat{Y}$ values:

The relationship between ε and SSE is expressed by the formula for SSE:

$$\text{ESS} = \sum_{i=1}^{N}(\text{yi} - \hat{\text{y}}\text{i})^2$$

Here's how ε and SSE are related:

$$\text{SSE} = \sum_{i=1}^{N}(\text{Y} - \hat{Y}_i)^2 = e_1^2 + e_2^2 + e_3^2 + e_4^2 \ldots + e_{11}^2 = \sum_{i=1}^{N} e_i^2 \quad (5)$$

And SST is diference differences betweean the observed values yi, and $\bar{y}$ main of tte valye of yi

$$\text{SST} = \sum_{i=1}^{N}(\text{Y} - \bar{Y}_i)^2 = e_1^2 + e_2^2 + e_3^2 + e_4^2 \ldots + e_{11}^2 = \sum_{i=1}^{N} e_i^2 \quad (6)$$

Mean Absolute Error (MAE), measures the average absolute difference between the actual (observed) values and the predicted values.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|Y - \hat{Y}_i| \quad (7)$$

The central objective of the Loss Function is to quantify the error inherent in the model's predictions. In practice, it measures the extent of disparity between the observed values of the dependent variable (Y) and the corresponding predicted values ($\hat{Y}$) for each data point (i). A widely adopted metric within this context is the Mean Squared Error (MSE), defined as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(Y - \hat{Y}_i\right)^2 \quad (8)$$

According to multiple linear regression for the given wine dataset calculations were made for mean_absolute_error (MAE), mean_squared_error (MSE) and root_mean_squared_error (RMSE), for the model trained set but also for the tested set. The results are presented in Table 6

For model evaluation is used R^2, statistical measure of how close data are to the fitted regression line.

$$R^2 = 1 - \frac{SSE}{SST} \quad (9)$$

Where SSE is Sum of Square Error and SST is Sum of Square Total

Table 6: Multiple Linear Regression Model

Loss Function	Multiple Linear Regression	
	y_train	y_test
MAE	0.48949	0.53303
MSE	0.38888	0.490888
RMSE	0.62360	0.700634
R2	0.38123	0.303635
VIF	1.6161	1.43602

A lower MAE, MSE and RMSE, indicates a very good fit of the model to the data because it the predicted values are closer to the actual values.

The Variance Inflation Factor (VIF) is a measure that helps us understand how much the variance of an estimated regression coefficient is inflated due to the presence of multicollinearity in the dataset. Multicollinearity occurs when predictor variables in a regression model are highly correlated with each other.

$$VIF = \frac{1}{1-R^2} \quad (10)$$

R-squared (R2) values are statistical measures that indicate how well the regression model explains the variability in the data. A higher R2 value, closer to 1, suggests that the model is better at explaining the variability. In the results, R2 values of 0.38123 and 0.303635. A VIF value greater than 1 and less than 5 indicates moderate correlation. These values indicate that the explains some of the variability in the data, but there is still a substantial amount of unexplained variability.

VIF (Variance Inflation Factor) is a metric used to assess multicollinearity in a regression model. Multicollinearity occurs when predictor variables in the model are highly correlated with each other, which can lead to unstable coefficient estimates. A high VIF value (typically greater than 1) suggests that a predictor variable is highly correlated with other predictors in the model, indicating multicollinearity. In the results, VIF values of 1.6161 and 1.43602, which are relatively low. Lower VIF values are generally better because they indicate lower levels of multicollinearity.

In summary, R2 values suggest that regression models explain some but not all of the variability in the data. Additionally, your VIF values are relatively low, indicating lower levels of multicollinearity, which is generally a positive outcome in regression analysis.

In the realm of machine learning, choosing the right algorithm is paramount for achieving accurate and reliable predictions. In this analysis, we have evaluated the performance of three distinct regression models: the DecisionTreeRegressor, RandomForestRegressor, and Support Vector Machine (SVM). Each of these models brings its own strengths and characteristics to the table.

The DecisionTreeRegressor is known for its ability to capture complex relationships within the data, potentially leading to a high level of accuracy on the training set. However, it may also be prone to overfitting, where it performs exceptionally well on the training data but struggles to generalize to new, unseen data.

The Random Forest Regressor, on the other hand, employs an ensemble approach, combining multiple decision trees to enhance prediction accuracy. This model often strikes a balance between complexity and generalization, making it a popular choice for various regression tasks.

Lastly, the Support Vector Machine, or SVM, is a powerful algorithm that excels in capturing intricate patterns within data. While it may exhibit a lower accuracy on the training set compared to other models, it can provide robust predictions and is particularly adept at handling non-linear relationships.

In this comparative analysis, we present the results of these models based on metrics such as accuracy, R-squared, and various error measures. By understanding the strengths and limitations of each model, we aim to guide the selection process towards the algorithm best suited for the specific nuances of our dataset and objectives.

Table 7: Comparising Linerar Regression Problem Solwing with Diferent Type of Machine Learning Models for Wine Dataset

Model	Performance Comparison of Regression Models				
	Accuracy	R2	MAE	MSE	RMSE
DecisionTreeRegressor	1.0	1.0	0.000	0.000	0.00
RandomForestRegressor	0.929	0.929	0.158	0.047	0.217
SVM	0.556	0.556	0.380	0.295	0.543

To illustrate the practical implementation of multiple linear regression, a demonstration using PyTorch, as a popular deep learning framework, is provided in purpose to see the primary similarity and differences between traditional linear regression analysis and a linear regression model using the PyTorch deep learning framework.

Initialization of Variables
Initially, random values are assigned to the coefficients (β0, β1, ..., β11) that will be learned during training. These coefficients are declared as PyTorch tensors with `requires_grad=True` to enable gradient computation.

Linear Model Function
The linear model function `mylnmodel` is defined, which takes in the independent variables (e.g., fixed acidity, volatile acidity, etc.) as tensors and computes the predicted value for `quality` using the learned coefficients. It is a simple linear equation with coefficients and variables.

The Mean Squared Error function (MSE) is implemented to calculate the loss between the predicted values and the actual "quality" values in the dataset.

Gradient Calculation
After predicting the values and computing the loss, the gradients of the loss function with respect to the coefficients are calculated using `loss. backward () `. This step enables the model to update the coefficients in the direction that minimizes the loss.

Gradient Descent is used for optimization. The code runs for 5,000 iterations, updating the coefficients with small steps in the direction of gradient descent. This process iteratively refines the coefficients to improve the model's accuracy.

Dıscussıon and Conclusion
This paper has provided a comprehensive exploration of multiple linear regression, shedding light on its foundational principles and seamless integration with contemporary machine learning techniques. By bridging the theoretical underpinnings with practical applications, we have aimed to equip readers with a holistic understanding of this versatile statistical method.

Three key outcomes emerge from this study. Firstly, we demonstrate the formulation of independent and dependent variables in linear regression, providing a structured framework for modeling. Secondly, analyze model performance using essential metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2). These metrics offer insights into model accuracy and its explanatory power. Also validate a linear regression model, it is essential to assess several key assumptions to ensure the model's reliability and suitability for the data. These common assumptions for Linear Regression are as follows:

Normality of Residuals
The first assumption involves examining whether the residual errors follow a normal distribution.

Mean of Residuals Close to Zero
The second assumption requires that the mean of the residual errors should ideally be close to zero or approach zero. (A non-zero mean may indicate a systematic bias in the model which is not the case in our study).

Multivariate Normality
Linear regression assumes that all variables are multivariate normally distributed.

Homoscedasticity
Which means that the variance of the residuals should remain constant across the regression line. To assess homoscedasticity, a scatter plot of residuals against fitted values can be examined. If the plot exhibits a consistent spread of points, homoscedasticity is met; otherwise, a funnel-shaped pattern may indicate heteroscedasticity.

Multicollinearity Check
The last assumption pertains to multicollinearity refers to high correlations among independent variables. To detect multicollinearity, the Variance Inflation Factor (VIF) is often used. VIF measures the correlation and strength of correlation between independent variables. A VIF value greater than 1 and less than 5 indicates moderate correlation, while a VIF less than 5 is considered a critical level of multicollinearity.

These assumptions collectively help ensure that a multiple linear regression model is appropriate for the given data and that the model's predictions are reliable. Violations of these assumptions may require further analysis or potential model adjustments.

Lastly, we conduct comparative assessments with alternative regression models, including decision trees, random forests, and support vector regression. Also illustrate the practical implementation of multiple linear regression, a demonstration using PyTorch, as a popular deep learning framework

Conflict of Interest
The authors have declared that there is no conflict of interest.

Author Contributions
Research Conceptualization: VAK; Machine Learning Model Comparison: VAK, MP. PyTorch Implementation: VAK, MP; Results and Discussion: VAK; Manuscript Writing: VAK, MP; References and Citations: VAK, MP; Figures and Tables: VAK, MP; Proofreading and Finalization: VAK.

References

1. Montgomery DC, Peck EA, Vining GG (2012) Introduction to Linear Regression Analysis. John Wiley & Sons 821.
2. Wooldridge JM (2019) Introductory Econometrics: A Modern Approach. Cengage Learning https://au.cengage.com/c/isbn/9781337558860/.
3. Fox J (2016) Applied Regression Analysis and Generalized Linear Models. SAGE Publications 816.
4. Gujarati DN, Porter DC (2009) Basic Econometrics. McGraw-Hill Education https://www.cbpbu.ac.in/userfiles/file/2020/STUDY_MAT/ECO/1.pdf.
5. Greene WH (2018) Econometric Analysis. Pearson Education https://www.ctanujit.org/uploads/2/5/3/9/25393293/_econometric_analysis_by_greence.pdf.
6. Sun Y, Wang X, Zhang C, Zuo M (2023) Multiple Regression: Methodology and Applications. Highlights in Science, Engineering and Technology AMMSAC 49: 542.
7. James G, Witten D, Hastie T, Tibshirani R (2013) An Introduction to Statistical Learning. Springer https://link.springer.com/book/10.1007/978-1-4614-7138-7.
8. Hastie T, Tibshirani R, Friedman J (2001) The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
9. Gelman A, Hill J (2006) Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge University Press https://www.cambridge.org/highereducation/books/data-analysis-using-regression-and-multilevel-hierarchical-models/32A29531C7FD730C3A68951A17C9D983#overview.
10. Hastie T, Tibshirani R, Friedman J (2009) The Elements of Statistical Learning. Springer https://link.springer.com/book/10.1007/978-0-387-84858-7.
11. Murphy KP (2012) Machine Learning: A Probabilistic Perspective. MIT Press 1104.
12. Iwasaki M (2020) Multiple Regression Analysis from Data Science Perspective 131-140.
13. Kutner MH, Nachtsheim CJ, Neter J, Li W (2004) Applied Linear Statistical Models. McGraw-Hill Education https://users.stat.ufl.edu/~winner/sta4211/ALSM_5Ed_Kutner.pdf.
14. Cortez P, Cerdeira A, Almeida F, Matos T, Reis J (2009) Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems 47: 547-553.
15. Least-Squares Method (2008) In: The Concise Encyclopedia of Statistics. Springer, New York, NY 304-306.
16. PyTorch (2023) https://pytorch.org./.
17. Bishop CM (2006) Pattern Recognition and Machine Learning. Springer https://link.springer.com/book/9780387310732.
18. Chen J, Song L, Wainwright MJ, Jordan MI (2018) Learning to explain: An information-theoretic perspective on model interpretation. In: Proceedings of the 35th International Conference on Machine Learning 80: 883-892.
19. Ribeiro MT, Singh S, Guestrin C (2016) Why should I trust you? Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining 1135-1144.
20. Lee H, Wang J, Leblon B (2020) Using Linear Regression, Random Forests, and Support Vector Machine with Unmanned Aerial Vehicle Multispectral Images to Predict Canopy Nitrogen Weight in Corn. Remote Sensing 12: 2071.
21. Jana M (2023) Exploring Machine Learning Models: A Comprehensive Comparison of Logistic Regression, Decision Trees, SVM, Random Forest, and XGBoost. Medium https://medium.com/@malli.learnings/exploring-machine-learning-models-a-comprehensive-comparison-of-logistic-regression-decision-38cc12287055.
22. Knights V, Kolak M, Markovikj G, Gajdoš Kljusurić J (2023) Modeling and Optimization with Artificial Intelligence in Nutrition. Applied Sciences 13: 7835.