Momentum-Enhanced Linear Regression for Faster Convergence in Real-World Predictions

Hussein Al-Bazzaz[1]

[1]Affiliation not available

April 18, 2024

Abstract

This paper investigates the efficacy of linear regression enhanced by gradient descent with momentum for predicting real-world outcomes. The introductory sections establish the significance of machine learning in driving sustainable innovations and detail the foundational aspects of linear regression, a statistical technique pivotal for modelling relationships between variables using a least squares approach. Enhanced with momentum-based gradient descent, these models achieve faster and more stable convergence, which is particularly beneficial in complex, real-life data scenarios. Through empirical analysis using the Boston and California housing datasets, we demonstrate that linear regression, when optimized, can effectively predict housing values with high R 2 scores, indicating robust predictive power across socioeconomic and geographic variables. Our findings underscore the model's utility as a forecasting tool in today's data-driven landscape. Future research directions include optimizing momentum coefficients and learning rates and potentially incorporating adaptive methods to enhance convergence efficiency. This study provides insights into the continuous improvements required in predictive analytics to maintain accuracy and reliability in diverse applications.

Momentum-Enhanced Linear Regression for Faster Convergence in Real-World Predictions

Hussein Al-Bazzaz

***Abstract*—This paper investigates the efficacy of linear regression enhanced by gradient descent with momentum for predicting real-world outcomes. The introductory sections establish the significance of machine learning in driving sustainable innovations and detail the foundational aspects of linear regression, a statistical technique pivotal for modelling relationships between variables using a least squares approach. Enhanced with momentum-based gradient descent, these models achieve faster and more stable convergence, which is particularly beneficial in complex, real-life data scenarios. Through empirical analysis using the Boston and California housing datasets, we demonstrate that linear regression, when optimized, can effectively predict housing values with high R^2 scores, indicating robust predictive power across socio-economic and geographic variables. Our findings underscore the model's utility as a forecasting tool in today's data-driven landscape. Future research directions include optimizing momentum coefficients and learning rates and potentially incorporating adaptive methods to enhance convergence efficiency. This study provides insights into the continuous improvements required in predictive analytics to maintain accuracy and reliability in diverse applications.**

***Index Terms*—Momentum Gradient Descent, Linear Regression, Predictive Analytics, Convergence Efficiency, Housing Market Analysis, Machine Learning Optimization**

I. Introduction

The transformative capability of science now harnessed through machine learning, is at the forefront of identifying complex patterns and developing efficient solutions for a sustainable future. This capability has played a pivotal role in ensuring global food security and overcoming challenges previously deemed impossible. Such advancements have been propelled by the relentless efforts of scientists, whose contributions have profoundly altered our societal landscape. As we embark on this new chapter, machine learning stands to deepen our comprehension of the natural world further, improve the efficiency of resource use, and reduce our ecological impact, leading us toward a thriving and sustainable future.

Linear regression is a foundational statistical technique used to predict a dependent variable's value from one or more independent variables [1], where a linear equation characterizes the relationship between the variables. This method is instrumental in forecasting outcomes, as it minimizes the variance between actual and predicted results by fitting a linear model through the data, typically using the "least squares" method for its simplicity and efficiency in finding the best-fit line. When enhanced with gradient descent optimization algorithms, incorporating momentum, linear regression models can gain improved convergence rates, which is especially beneficial in complex, real-life data that present ravines and noisy gradients. The momentum in these optimization algorithms smoothes the path toward the optimum by considering the direction and magnitude of prior updates, thereby enhancing the stability and speed of convergence in finding the optimal solution for the linear regression model.

Recent studies highlight the fundamental aspects of linear regression, emphasizing its utility in modelling relationships between dependent and independent variables through least squares and generalized linear models (GLM). The diverse applications of these techniques in fields such as image quality assessment and fire detection underscore their significance. For instance, the research in [2] presents a self-supervised approach for No-Reference Image Quality Assessment (NR-IQA) using linear regression to map image representations to quality scores, demonstrating improved data efficiency and generalization capabilities. Similarly, the research in [3] proposes a different approach to image quality assessment, leveraging a mixture of experts' approaches to learn image quality features in an unsupervised setting, further training a linear regression model for accurate quality assessment. Moreover, the research in [4] discusses the effectiveness of linear regression over logistic regression in a differential private setting for image classification, highlighting the lower computational burden and improved privacy-performance trade-offs. The research in [5] utilizes linear regression in the Recurrent Trend Predictive Neural Network (rTPNN) for multi-sensor fire detection, showcasing its capability to process and predict from multivariate time series data. Additionally, the research in [6] presents an embedding model for knowledge graph link prediction, conceptualizing the task as a simple linear regression problem to capture diverse connectivity patterns and relation properties, achieving significant performance improvements. Furthermore, linear regression's application in predicting rainfall from environmental variables illustrates its adaptability and efficiency in various fields, merging traditional and computational methods to enhance prediction accuracy [7].

These examples illustrate linear regression's broad applicability and effectiveness in addressing complex, real-world problems across various domains, from enhancing the accuracy of image quality assessments to advancing fire detection technologies and knowledge graph embeddings. The continued evolution of these methodologies, incorporating machine learning techniques, exemplifies their critical role in developing innovative solutions and improving decision-making processes. The remainder of this paper is organized as follows: Section II delves into the intricacies of linear regression. Section III presents a comprehensive demonstration

This research received no funding.

Hussein Al-Bazzaz is with the Communication Technical Engineering Department, College of Engineering Technical, Ashur University, Baghdad, Iraq (e-mail: hussein.al-buzzuz@au.edu.iq).

of linear regression applied to real-world datasets. Finally, Section IV concludes this paper's findings.

II. Methodology

In linear regression models, input data serve as independent variables; their corresponding targets are the dependent variables. Equation 1, fundamental to linear regression, delineates the relationship between dependent and independent variables as follows:

$$f(\vec{x}_i|\boldsymbol{\omega}) = \boldsymbol{\omega}_0 + \sum_{j=1}^{D} \boldsymbol{\omega}_j \vec{x}_{ij} \tag{1}$$

In this context, $\vec{x}_I$ denotes the feature vector for the ith observation, while $\boldsymbol{\omega}$ symbolizes the model's coefficients, encompassing the intercept. The intercept term, denoted by $\boldsymbol{\omega}_0$, is essential for setting the regression line or hyperplane's position in relation to the data points, ensuring it does not necessarily pass through the origin. This term serves as the baseline prediction when input features are zero. To integrate the intercept seamlessly with other coefficients, a feature $\vec{x}_{i0}$ with a constant value of one is introduced, preserving the intercept's effect without modification. The linear regression equation is efficiently expressed in matrix form as follows:

$$f(\mathcal{X}|\boldsymbol{\omega}) = \mathcal{X}\boldsymbol{\omega} \tag{2}$$

where $\boldsymbol{\omega}$ includes the intercept $\boldsymbol{\omega}_0$. The input matrix $\mathcal{X}$ features a first column of ones, aligning with the intercept to ensure its effect is considered in the model. This format facilitates handling multiple variables and data points, with the intercept enabling vertical adjustments of the regression line for optimal data fit, referred to as the "bias term" or "intercept column." In regression modelling, selecting the most suitable coefficients is crucial and involves the application of a loss function. This function calculates the discrepancy between the predicted outcomes of the model and the actual data. The least squares method, which aims to minimize this discrepancy, is essential for improving the model's accuracy in capturing the patterns of real-world data. Although the least squares method is foundational in linear regression for determining the best coefficients, several alternative metrics can be used as loss functions for optimization. These alternatives provide a variety of approaches for evaluating model performance, accommodating different data requirements and modelling objectives. Below, we describe the most commonly used evaluation metrics in linear regression, each offering a unique framework for optimizing coefficients to achieve the most accurate model predictions.

1) Least Squares Method: - Summary: Aims to reduce the sum of squared residuals, the squared differences between observed and predicted values. - Formula: Minimize $\sum_{i=1}^{n}(y_i - f(x_i))^2$
2) Ridge Regression (L2 Regularization): - Summary: Incorporates a penalty on the sum of squared coefficients to the least squares criterion. - Formula: Minimize $\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \sum_{j=1}^{p} \omega_j^2$
3) Lasso Regression (L1 Regularization): - Summary: A penalty is applied to the absolute value of the coefficients to the least squares objective. - Formula: Minimize $\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \sum_{j=1}^{p} |\omega_j|$
4) Elastic Net: - Summary: Merges ridge and lasso regression penalties into the least squares function. - Formula: Minimize $\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda_1 \sum_{j=1}^{p} \omega_j^2 + \lambda_2 \sum_{j=1}^{p} |\omega_j|$
5) Mean Squared Error (MSE): - Summary: Calculates the average of the squared differences between observed and predicted values. - Formula: MSE $= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

Subsequently, we will delve into the optimization methods, focusing mainly on the least squares solution. The least squares method is fundamental for several reasons. It is notably simple and computationally efficient. Despite its apparent simplicity, the least squares approach is highly effective and versatile, offering reliable outcomes across various contexts. Furthermore, the underlying mathematical principles of the least squares method serve as a vital basis for discussing critical issues such as overfitting, underfitting, and model variance. The associated loss function is expressed as follows:

$$L(\boldsymbol{w}) = \sum_{i=1}^{N}(y_i - f(\vec{x}_i|\boldsymbol{w}))^2 \tag{3}$$

Here, y_i represents the target variable. We also express this equation in matrix form as follows:

$$L(\boldsymbol{w}) = (\mathcal{Y} - f(\mathcal{X}|\boldsymbol{\omega}))^T(\mathcal{Y} - f(\mathcal{X}|\boldsymbol{\omega})) \tag{4}$$

In this representation, $\mathcal{X}$ is an $N \times (D+1)$ matrix, and $\boldsymbol{w}$ is a $(D+1) \times 1$ matrix, yielding an $N \times 1$ matrix where each entry predicts the output for each corresponding input. Using the target variable y to find the best coefficients in the equation above highlights that this optimization method falls under supervised learning. To determine the model coefficients that minimize the loss function detailed in Equation 4, we take the derivative of the loss function with respect to the coefficients and equate it to zero. This approach is grounded in the principle that a function's minimum is located where its derivative equals zero. Solving this system of linear equations yields the values of the coefficients that minimize the loss function, providing the optimal parameters for the linear regression model and thus minimizing the squared differences between the predicted outputs and the actual observed values. Starting from the linear regression algorithm's loss function, we differentiate this function with respect to the coefficients. The expanded form of the loss function is:

$$L(\boldsymbol{\omega}) = \mathcal{Y}^T\mathcal{Y} - \mathcal{Y}^T\mathcal{X}\boldsymbol{\omega} - \boldsymbol{\omega}^T\mathcal{X}^T\mathcal{Y} + \boldsymbol{\omega}^T\mathcal{X}^T\mathcal{X}\boldsymbol{\omega} \tag{5}$$

Differentiation of Equation 5 against $\boldsymbol{\omega}$ requires evaluating each term separately. The derivative of the constant term $\mathcal{Y}^T\mathcal{Y}$ is zero. The derivatives of $-\mathcal{Y}^T\mathcal{X}\boldsymbol{\omega}$ and $-\boldsymbol{\omega}^T\mathcal{X}^T\mathcal{Y}$ lead to $-2\mathcal{X}^T\mathcal{Y}$ by applying the derivative rule of Ax with respect to x as A^T. The last term, $\boldsymbol{\omega}^T\mathcal{X}^T\mathcal{X}\boldsymbol{\omega}$, follows the differentiation rule of x^TAx to x, resulting in $2\mathcal{X}^T\mathcal{X}\boldsymbol{\omega}$. Thus, the derivative of the loss function is:

$$-2\mathcal{X}^T\mathcal{Y} + 2\mathcal{X}^T\mathcal{X}\boldsymbol{\omega} = 0 \tag{6}$$

Solving for $\boldsymbol{\omega}$ in Equation 6 yields the estimated coefficients:

$$\hat{\boldsymbol{\omega}} = (\mathcal{X}^T\mathcal{X})^{-1}\mathcal{X}^T\mathcal{Y} \tag{7}$$

The estimated coefficient vector, $\hat{\boldsymbol{\omega}}$, is critical for making predictions with the linear regression model, using either training or new data. These coefficients reveal the influence of each independent variable on the dependent variable, indicating the expected change in the dependent variable for a one-unit increase in an independent variable, holding others constant. The size of a coefficient shows its effect's strength, and its sign indicates the direction (positive for direct and negative for inverse relationships). Coefficients close to zero suggest minimal influence. This analysis is critical to understanding and predicting the dependent variable's response to changes in independent variables. The move towards superintelligent AI calls for a unified global effort, highlighting two critical areas in AI research. First, the focus on ensuring model reproducibility is vital for collaborative advancement in the field. Replicating AI models and their outcomes is fundamental to building scientific cooperation and trust. Such clarity in research methods and results is also essential for navigating safely toward superintelligent AI. Second, it is crucial for researchers to comprehensively outline the computational complexity of their AI models in academic papers. An in-depth complexity analysis is essential to a system's manageability and viability. Grasping the scalability and practicality of AI solutions is crucial for their alignment with human well-being and applicability under realistic conditions. Emphasizing these factors is critical to supporting the development of superintelligent AI as a beneficial and secure advancement for humankind.

Computational complexity refers to the amount of computational resources needed by an algorithm to solve a problem, which typically varies with the input data size. This complexity is commonly expressed using Big O notation, $O(\cdot)$, which estimates the maximum time or space (memory) an algorithm might require relative to the size of its input. For instance, $O(n)$ complexity suggests that the algorithm's resource needs increase linearly with the input size n. In contrast, $O(n^2)$ complexity indicates a quadratic increase, with resources growing proportionally to the square of the input size. Lower complexities like $O(n)$ or $O(\log n)$ are preferred for handling large datasets, as opposed to higher complexities like $O(n^2)$ or $O(n^3)$.

For linear regression, the computational complexity mainly involves matrix multiplication and inversion. Considering $\mathcal{X}$ as an $N \times D$ matrix, the complexity of multiplying $\mathcal{X}^T$ by $\mathcal{X}$ is $O(ND^2)$, assuming each element of the resultant matrix necessitates N multiplications and summations. Conversely, inverting the $D \times D$ matrix, $(\mathcal{X}^T\mathcal{X})^{-1}$, which is the most computationally intensive step, exhibits cubic complexity, $O(D^3)$. Thus, the overall complexity for computing the model coefficients is $O(D^3 + ND^2)$, with the final multiplication of $\mathcal{X}^T\mathcal{Y}$ having a minimal impact on total computational cost.

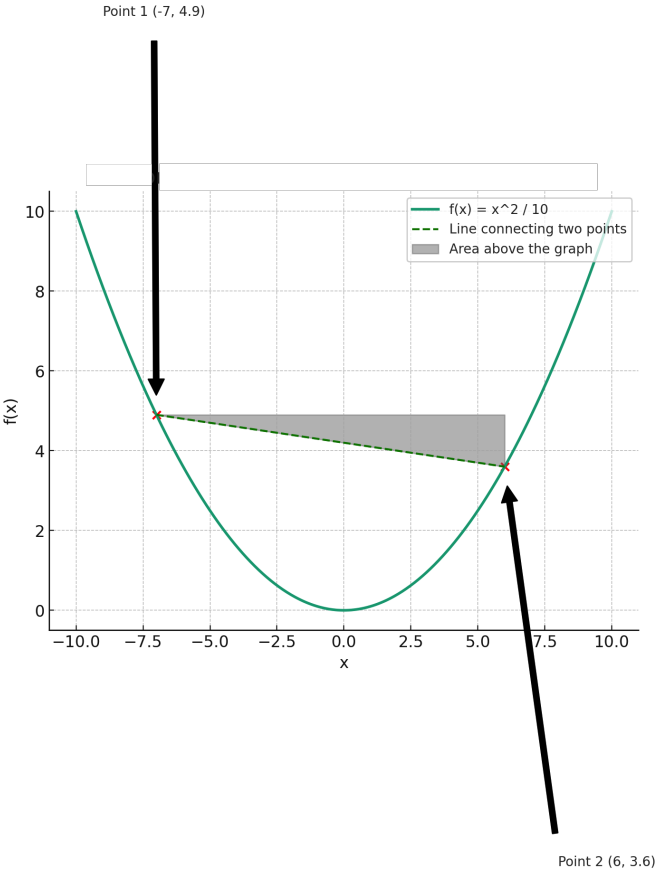

Fig. 1. Convex Function Demonstration

This complexity mirrors the structure of a polynomial function $f(x) = c_k x^k + c_{k-1}x^{k-1} + \ldots + c_1 x + c_0$, where x is the variable and $c_k, c_{k-1}, \ldots, c_0$ are constants, with k being a non-negative integer. Although polynomial complexities are manageable for medium-sized datasets, they become challenging for vast datasets. In order to mitigate the aforementioned challenge, methods like Gradient Descent are applied, which are particularly effective for large datasets. We will next explore the mathematical framework of the linear regression learning process via Gradient Descent, emphasizing its adaptability and efficiency for large-scale data analysis.

Within the realm of model optimization, the gradient of the loss function indicates the direction and magnitude of adjustments required for the model parameters to reduce error. The gradient, a vector of partial derivatives of the loss function concerning each model coefficient, signifies the necessary adjustments per parameter to lower the loss, maintaining other parameters constant. The gradient direction points towards the steepest increase in error, and its components suggest whether to increase or decrease a parameter (positive values indicate a decrease is needed, while negative values suggest an increase) and the urgency of this change.

The expansion of the least squares loss function yields a quadratic formula in the model coefficients $\boldsymbol{\omega}$, akin to a polynomial with degree two, $f(x) = ax^2 + bx + c$, where a, b, and c are constants with a not equal to zero. This quadratic nature

deems the least squares solution a convex function, implying that any local minimum is a global minimum, facilitating a straightforward optimization path.

To rigorously prove the function's convexity, we examine the second derivative or the Hessian. Convexity is confirmed if the Hessian is positive semidefinite, indicating that the second derivative across all directions is non-negative, a characteristic ensuring the presence of a global minimum.

The first derivative of the loss function with respect to the coefficients is previously outlined as:

$$\frac{\partial L(\boldsymbol{\omega})}{\partial \boldsymbol{\omega}} = -2\mathcal{X}^T\mathcal{Y} + 2\mathcal{X}^T\mathcal{X}\boldsymbol{\omega} \tag{8}$$

In matrix calculus, matrix and vector terms are differentiated by specific rules. For scalar products, the derivative Ax with respect to x yields A^T, while for matrix-vector products, the rule adapts accordingly. Thus, the second derivative of the loss function, indicative of its curvature and convexity, is:

$$\frac{\partial^2 L(\boldsymbol{\omega})}{\partial \boldsymbol{\omega}^2} = 2\mathcal{X}^T\mathcal{X} \tag{9}$$

The Hessian matrix's role is paramount in determining function curvature and convexity in optimization. In linear regression, the positive semi-definiteness of the Hessian, and thus the convexity of the function, is assured if the input data matrix $\mathcal{X}$ is of full column rank, indicating linear independence among columns. To maintain $\mathcal{X}^T\mathcal{X}$'s positive semi-definiteness, certain prerequisites regarding the data matrix $\mathcal{X}$'s structure must be met:

1) **Sufficient Data Points:** There should be at least as many observations (rows in $\mathcal{X}$) as there are variables (columns in $\mathcal{X}$). Excessive observations over variables enhance the chances of achieving full column rank in the matrix.
2) **Independence of Features:** It is crucial that the variables (columns in $\mathcal{X}$) are independent of each other, meaning no variable is a linear combination of the others.
3) **Data Preparation:** Applying data preparation techniques such as feature selection or extraction can aid in eliminating feature dependencies, ensuring the matrix attains full column rank.

After recognizing the necessity of a full-rank input matrix $\mathcal{X}$ for ensuring the convexity of the objective function in linear regression, we explore the QR factorization technique. This method verifies the full rank status of $\mathcal{X}$ and provides insights into the data's structure. If $\mathcal{X}$ lacks full rank, the QR factorization helps identify steps to preprocess $\mathcal{X}$ and achieve full rank, essential for preserving the convexity of the objective function during optimization. The QR factorization decomposes a matrix into Q (an orthogonal matrix) and R (an upper triangular matrix), offering a strategic approach to understanding and modifying the data matrix. Researchers gain crucial insights by applying QR factorization to the training data matrix $\mathcal{X}$. First, they can ascertain whether $\mathcal{X}$ possesses full rank, indicating that its columns are linearly independent. Should $\mathcal{X}$ not exhibit full rank, this suggests the presence of linear dependency among its features; a scenario QR factorization helps identify by pinpointing the redundant features. A full-rank status for $\mathcal{X}$ is confirmed when all diagonal elements of the R matrix from the QR factorization are non-zero. Moreover, QR factorization aids in detecting linearly dependent features within $\mathcal{X}$; if any diagonal element of the R matrix is zero or approaches zero, it signals linear dependence between that column and its predecessors. Specifically, a zero value in the j'th diagonal element of R indicates that the j'th column of $\mathcal{X}$ is a linear combination of the preceding $j-1$ columns.

We will explore the application of gradient descent in the linear regression learning process, where the aim is to iteratively refine the model's coefficients to minimize the loss function. Each iteration updates the coefficients to lower the loss function's value than the last, progressively reducing the overall loss. This iterative improvement can be represented as:

$$L(\boldsymbol{\omega^{(0)}}) > L(\boldsymbol{\omega^{(1)}}) > L(\boldsymbol{\omega^{(2)}}) > L(\boldsymbol{\omega^{(3)}}) > \dots \tag{10}$$

Here, $\omega^{(k)}$ refers to the coefficient set at the k'th iteration. This sequence illustrates that with each iteration, the loss associated with the current coefficient set should be lower than that of the previous set, demonstrating advancement in the optimization process. To effectively leverage gradient descent for optimization, it is critical to grasp the gradient of a function, the concept of a tangent in this scenario, and the strategy of moving against the gradient (towards the negative gradient direction). The gradient represents the direction of the steepest increase at any point on a function's surface. Drawing a tangent line or plane in the gradient's direction would indicate where the function's value rises most sharply. Thus, advancing in the direction opposite to the gradient is fundamental for systematically finding the function's minimum, embodying the essence of gradient descent optimization. The steps for this learning approach are detailed in Algorithm 1.

Algorithm 1 Linear Regression using Gradient Descent

Require: Learning rate α, convergence threshold ϵ
Ensure: Optimal parameters $\boldsymbol{\omega}$
1: Initialize the parameters $\boldsymbol{\omega}$ randomly
2: Initialize iteration pointer as follows: $k = 0$
3: **while** $|L(\boldsymbol{\omega^{(k)}}) - L(\boldsymbol{\omega^{(k-1)}})| < \epsilon$ **do**
4: Calculate the gradient $\nabla L(\boldsymbol{\omega})$
5: Update the Parameters as follows: $\boldsymbol{\omega} = \boldsymbol{\omega} - \alpha\nabla L(\boldsymbol{\omega})$
6: Increment iteration pointer as follows: k=k+1
7: **end while**
8: **return** $\boldsymbol{\omega}$

Selecting an appropriate learning rate α involves trial and error, necessitating empirical adjustment and insight gained from experience.

Linear regression via gradient descent can substantially benefit from incorporating the least squares method with a momentum term in the optimization algorithm as demonstrated in Algorithm II.

The addition of momentum, denoted by β, helps to accelerate convergence in the gradient descent process, which is crucial when dealing with uneven objective function landscapes

Algorithm 2 Linear Regression using Gradient Descent

Require: Learning rate α, convergence threshold ϵ
Ensure: Optimal parameters $\boldsymbol{\omega}$
1: Initialize the parameters $\boldsymbol{\omega}$ randomly
2: Initialize iteration pointer as follows: $i = 0$
3: **while** $|L(\boldsymbol{\omega}^{(i)}) - L(\boldsymbol{\omega}^{(i-1)})| < \epsilon$ **do**
4: Calculate the gradient $\nabla L(\boldsymbol{\omega})$
5: Update the velocity: $\nu = \beta\nu - \alpha\nabla L(\boldsymbol{\omega})$
6: Update the Parameters as follows: $\boldsymbol{\omega} = \boldsymbol{\omega} + \nu$
7: Increment iteration pointer as follows: i=i+1
8: **end while**
9: **return** $\boldsymbol{\omega}$

that exhibit steep ravines or suffer from noisy gradients due to stochastic variability in the data. By updating the velocity at each iteration, which blends the momentum of past updates (β) with the current gradient scaled by the learning rate (α), the algorithm can navigate more smoothly toward the optimal solution. This smoother path is attributed to the momentum term's ability to retain a fraction of the previous update, thereby applying a form of "friction" that lessens oscillations and avoids erratic swings in parameter updates.

Choosing an appropriate learning rate, α is critical and usually requires empirical tuning to achieve the best performance. It often starts with a small value and is adjusted based on the model's validation performance. Advanced techniques like learning rate scheduling or adaptive learning rate methods further refine the process by dynamically adjusting α, thus ensuring a balance between fast convergence and the stability of the learning process. The algorithm iteratively adjusts the parameters, moving closer to the optimum as long as the change in loss between iterations is above a convergence threshold, ϵ. This iterative process continues until the loss stabilizes within the desired threshold, indicating that the parameters have converged to an optimal set.

III. Analysis of Model Performance and Validation Results

Evaluating the effectiveness of a linear regression model entails examining its predictive capabilities to ascertain its proficiency in forecasting the values of the dependent variable from the independent variables. Several indicators offer perspectives on distinct facets of the model's performance, encompassing the precision of its predictions, the detection of anomalies, and the model's competence in reflecting the variability inherent in the dataset.

The R^2 score, also known as the coefficient of determination, quantifies the fraction of variance in the dependent variable that can be predicted from the independent variables. This metric ranges from 0 to 1, with higher values denoting a more accurate alignment of the model with the observed data. The R^2 score plays a pivotal role in gauging the extent to which the linear regression model accounts for the variability observed in the dataset. Moreover, it is instrumental in evaluating and comparing the efficacy of various models applied to the same dataset [8], [9].

The Mean Absolute Error (MAE) measures the average magnitude of errors within a collection of predictions, disregarding the direction of these errors. It is determined by computing the mean of the absolute discrepancies between forecasted and observed values. MAE serves as an intuitive metric for gauging prediction accuracy, facilitating an evaluation of the proximity between the model's predictions and the actual results on average [10].

The Mean Squared Error (MSE) calculates the average squared discrepancies between the predicted and actual values. This approach of squaring the errors amplifies the significance of more significant discrepancies, rendering MSE exceptionally responsive to outliers. It is an effective tool for detecting the model's tendency to commit substantial errors. However, it does so with the caveat of potentially exaggerating the impact of outliers in the assessment [11].

The Root Mean Squared Error (RMSE) is derived by taking the square root of the Mean Squared Error. This calculation results in a metric presented in the same units as the dependent variable, enhancing its interpretability compared to the MSE. By imposing heavier penalties on more significant errors, RMSE offers an insightful gauge of the model's predictive accuracy, concurrently emphasizing the influence of substantial outliers on the model's overall performance [11].

The Median Absolute Error (MedAE) represents the median value of all absolute deviations between the predicted outcomes of the model and the actual observed values. Distinct from Mean Absolute Error (MAE) and Mean Squared Error (MSE), MedAE remains unaffected by outliers, offering a sturdy indicator of the model's predictive precision. It captures the median error magnitude, making it especially valuable in analyzing datasets characterized by substantial outliers, thereby reflecting a more central tendency of the model's predictive discrepancies [12].

In validating the performance of linear regression models, adopting metrics such as the R^2 score, MAE, MSE, RMSE, and MedAE is pivotal for a comprehensive assessment. The R^2 score is crucial for understanding the proportion of variance the model explains, offering a measure of fit that facilitates comparison across different models. MAE provides an intuitive gauge of average prediction error magnitude, focusing on accuracy without influencing error direction. MSE and RMSE are particularly valuable for highlighting the impact of more significant errors, with RMSE improving interpretability by matching the units of the dependent variable. MedAE offers robustness against outliers, presenting a median error particularly useful in datasets with significant anomalies. These metrics provide a nuanced view of model performance, encompassing accuracy, sensitivity to outliers, and the ability to capture variability, thereby enabling a thorough evaluation and comparison of predictive models within research contexts.

A. Analysis of the Boston Housing Dataset

The dataset from the Boston Standard Metropolitan Statistical Area [13] of 1970 is explored to analyze the influence of various factors on the median value of owner-occupied homes. The dataset comprises various features of housing areas around

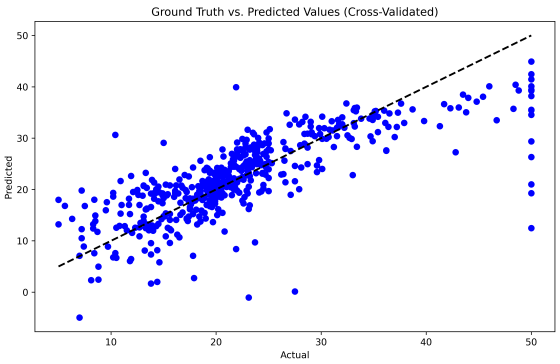

Fig. 2. The comparison of ground truth versus predicted values for the Boston housing dataset

the Boston suburb, such as crime rate, average number of rooms, accessibility to highways, and others, alongside the median value of homes. Our primary objective is to predict the median value of homes based on these features using a Linear Regression model.

We employed a comprehensive experimental analysis to assess the linear regression model's performance in our investigation into the Boston Standard Metropolitan Statistical Area dataset. We evaluated the model's efficacy based on various metrics using the KFold cross-validation method with ten splits and ensuring data shuffling with a fixed random seed for reproducibility. These metrics were calculated through cross-validation, with their average values and standard deviations reported to evaluate the model's performance comprehensively.

Figure 2 presents a scatter plot comparing the ground truth against the predicted values for the regression model using the Boston Standard Metropolitan Statistical Area dataset. As demonstrated in Table I, the average R^2 score is 0.717 with a standard deviation of 0.075, suggesting that, on average, the model explains about 71.7% of the variance in the dataset, which indicates a relatively strong predictive power. The MAE and the MedAE average at 3.377 and 2.619, respectively, signifying the average deviation of the predictions from the actual values. The MSE and RMSE provide information on the average squared error and its square root, with the RMSE averaging 4.793. The standard deviation in these scores reflects variability in the model's performance across different cross-validation folds or test sets. The model is reasonably well-fitted, although there is room for improvement, especially for higher-value predictions where the plot shows a greater dispersion of points from the prediction line.

B. the California Housing dataset

This dataset is a variant of the California Housing dataset, originally procured from Luís Torgo at the University of Porto [14]. Their research utilized data from the 1990 California census, formatting it into a dataset where each row represents a census block group, a geographical unit defined by the U.S. Census Bureau as the smallest for which sample data is

Metric	Average Score	Standard Deviation
R^2 Score	0.717	0.075
Median Absolute Error (MedAE)	2.619	0.279
Mean Absolute Error (MAE)	3.377	0.294
Mean Squared Error (MSE)	23.364	6.199
Root Mean Squared Error (RMSE)	4.793	0.626

TABLE I
ANALYSIS OF REGRESSION MODEL PERFORMANCE ON THE BOSTON HOUSING DATASET

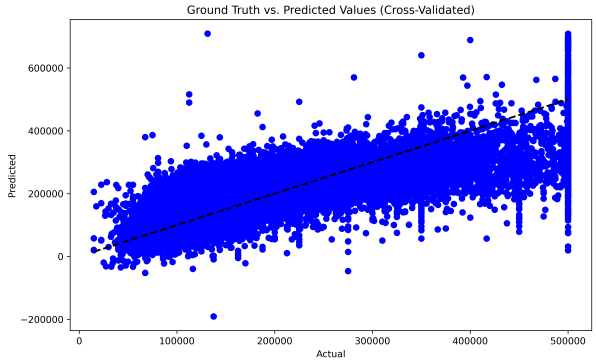

Fig. 3. Caption

published, generally encompassing a population ranging from 600 to 3,000 individuals.

The primary dependent variable in the California Housing dataset commonly used for regression analysis is the median house value, representing the central housing value within each block group. The independent variables, which serve as predictors for the median house value, include median income, median house age, average number of rooms per household, average number of bedrooms per household, population, average household occupancy, and geographical coordinates (Latitude and Longitude). These predictors are selected to analyze the influence of economic, demographic, and locational factors on housing prices across California, offering insights into how various attributes contribute to real estate valuation.

TABLE II
ANALYSIS OF REGRESSION MODEL PERFORMANCE ON THE CALIFORNIA HOUSING DATASET

Metric	Average Score	Standard Deviation
R^2 Score	0.618	0.026
Median Absolute Error (MedAE)	39681.544	555.972
Mean Absolute Error (MAE)	51722.390	896.286
Mean Squared Error (MSE)	5079307182.379	344122565.648
Root Mean Squared Error (RMSE)	71229.000	2395.163

The scatter plot demonstrated in Figure 3 and Table II substantiate the model's efficacy in predicting housing prices within the Californian market. The table comprehensively analyzes a regression model's performance on the California housing dataset. The R^2 score, with an average of 0.618 and a low standard deviation of 0.026, suggests that the model reasonably predicts approximately 61.8% of the variance in housing prices, showing consistency across different model

evaluations. The MedAE and MAE are considerably high, averaging 39,681.544 and 51,722.390, respectively, with relatively low standard deviations, indicating a consistent but significant difference between predicted values and actual prices. The MSE and RMSE present exceptionally high values, with averages of about 5,079,307,182 and 71,229, respectively. These large values and their considerable standard deviations imply substantial variability in the model's predictions, particularly highlighting the impact of outliers or extreme values in the dataset. This analysis underscores the model's effectiveness in capturing general trends while indicating areas where accuracy could be improved, particularly in handling data variability and outliers. The scatter plot provides a visual affirmation, displaying a concentration of data points around the line of best fit, which denotes predictions closely aligned with actual values, notwithstanding some variability at higher values, which is typical in real-world data. These performance metrics collectively validate the model's reliability and predictive power, rendering it a valuable tool for stakeholders in the housing domain. The high values of performance metrics such as MedAE, MAE, MSE and RMSE in the regression model are amplified by the large number of test observations, which increase the aggregate of squared errors and, consequently, the magnitude of these metrics.

IV. Conclusion

In this paper, we explored the effectiveness of linear regression models in predicting real-world outcomes across diverse domains. The study highlighted linear regression's broad applicability and robustness in fields such as real estate, underscoring its relevance and adaptability in today's data-driven landscape.

Our detailed analysis using the Boston and California housing datasets demonstrated that linear regression models offer significant predictive power when optimized and applied correctly. The model's ability to explain a substantial portion of the variance in housing prices and its consistent performance across different datasets emphasize its utility as a forecasting tool. Notably, the R^2 scores from both datasets—0.717 for Boston and 0.618 for California—indicate that the models can reliably predict housing values based on various socio-economic and geographic features.

The findings from this study affirm the potential of linear regression as a valuable tool in predictive analytics. While the model demonstrates substantial efficacy in interpreting and predicting outcomes, continuous improvements and adaptations are necessary to enhance its accuracy and reliability.

Future research on linear regression using gradient descent with momentum offers promising avenues for enhancing the efficiency and stability of convergence in predictive modelling. Critical areas for further investigation include optimizing the choice of the momentum coefficient (β) and the learning rate (α) to adapt dynamically to different data characteristics and objective landscapes. Experimental work could explore adaptive momentum methods that adjust β in response to changes in the gradient's direction and magnitude, potentially reducing the number of iterations needed to achieve convergence. Additionally, integrating second-order derivatives or incorporating adaptive learning rate schedules could address challenges associated with steep ravines and noisy gradients, which are common in high-dimensional data sets. By systematically evaluating these enhancements on diverse datasets, researchers can develop more robust guidelines for practically implementing this algorithm, thereby improving its applicability and performance across various predictive tasks.

Funding

This research received no funding.

Acknowledgment

I would like to express my deepest gratitude to my mother and father for their unwavering support and immense sacrifices, which have made my journey in engineering and artificial intelligence research possible. I am profoundly thankful for the opportunity they have given me to pursue my education and passion in this field.

References

[1] IBM, "Linear regression," https://www.ibm.com/topics/linear-regression, 2024.

[2] L. Agnolucci, L. Galteri, M. Bertini, and A. Del Bimbo, "Arniqa: Learning distortion manifold for image quality assessment," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 189–198.

[3] A. Saha, S. Mishra, and A. C. Bovik, "Re-iqa: Unsupervised learning for image quality assessment in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5846–5855.

[4] H. Mehta, W. Krichene, A. Thakurta, A. Kurakin, and A. Cutkosky, "Differentially private image classification from features," *arXiv preprint arXiv:2211.13403*, 2022.

[5] M. Nakip, C. Güzelíş, and O. Yildiz, "Recurrent trend predictive neural network for multi-sensor fire detection," *IEEE Access*, vol. 9, pp. 84 204–84 216, 2021.

[6] Y. Peng and J. Zhang, "Lineare: Simple but powerful knowledge graph embedding for link prediction," in *2020 IEEE international conference on data mining (ICDM)*. IEEE, 2020, pp. 422–431.

[7] C. M. Liyew and H. A. Melese, "Machine learning techniques to predict daily rainfall amount," *Journal of Big Data*, vol. 8, pp. 1–11, 2021.

[8] R. G. D. Steel, J. H. Torrie *et al.*, "Principles and procedures of statistics." *Principles and procedures of statistics.*, 1960.

[9] N. R. Draper and H. Smith, *Applied regression analysis*. John Wiley & Sons, 1998, vol. 326.

[10] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.

[11] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature," *Geoscientific model development*, vol. 7, no. 3, pp. 1247–1250, 2014.

[12] T. Pham-Gia and T. L. Hung, "The mean and median absolute deviations," *Mathematical and computer Modelling*, vol. 34, no. 7-8, pp. 921–936, 2001.

[13] Vikrishnan, "Boston house prices," https://www.kaggle.com/datasets/vikrishnan/boston-house-prices, 1970.

[14] H. Wang, "Housing," https://www.kaggle.com/datasets/harrywang/housing, 1990.

H. Al-Bazzaz Hussein Al-Bazzaz, a Baghdad College High School alumnus, holds a Ph.D. in Information and Systems Engineering from Concordia University and a Master's in Computer Engineering from the University of Balamand. His multifaceted expertise spans Advanced Computing, Artificial Intelligence, and Systems Design, bolstered by teaching experience gained during his graduate studies. Affiliated with IEEE Montreal and a Concordia International Tuition Award recipient, he brings a multicultural perspective and a deep commitment to technology advancement. He aims to infuse innovative teaching methods and foster a diverse, globally-aware academic landscape.