

CS4372 Project 1 Report

Salma Khalfallah
SMK210009

Hana Al-Jaser
HZA220000

September 21, 2025

1 Introduction

Linear regression is a simple approach to supervised learning that simply assumes a linear relationship between the predictor and the response variables in a model. For example, $f(x) = 2x$ indicates that the response variable, $y = f(x)$, is twice the value of the predictor variable x . Although a simplistic model with strong assumptions, this model is a powerful model with high-inference properties.

$$f(x) = w_0 + w_1x_1 + \dots + w_px_p + \epsilon$$

An issue arises in this methodology in that it is simply difficult to find the true function f of any relationship between response and predictor variables. This raises a motivation to *estimate* what the function f might look like through its weights $[w_0 \dots w_p]$. In the following report, our goal is to delve into two different ways to perform this estimation (Stochastic Gradient Descent, Ordinary Least Squares) and compare the results that stem from the two methodologies.

The dataset with which we chose to work is a "Facebook Metrics" dataset, which looks at observed performance metrics of a renowned cosmetic brand's Facebook page. In this situation, the learning goal is to understand what social media metrics best predict page performance. Our target variable is "Total interactions", which is the sum of likes, comments, and shares of the post.

2 Pre-Processing

During preprocessing, to prepare the data for the OLS regression, we first had to identify any issues with our initial data set. Specifically, we had to check for null values, missing values, and data inconsistencies. The data set contained no null values, missing values, or data inconsistencies that would require manipulation of the data.

Afterwards, we decided to explore the data and its attributes. As previously stated, the target for this learning task is the "Total interactions" attribute.

Other attributes for this dataset include social media metrics such as lifetime post metrics, details on the post, and general page metrics. Most of the attributes on the page are numeric, save for a couple of categorical variables which are appropriately encoded into factor variables. Upon initial analysis of attributes, it is important to note that the attributes in the dataset, including the target variable, do not follow a normal distribution but rather some version of a Gamma distribution (see: Figure 1)

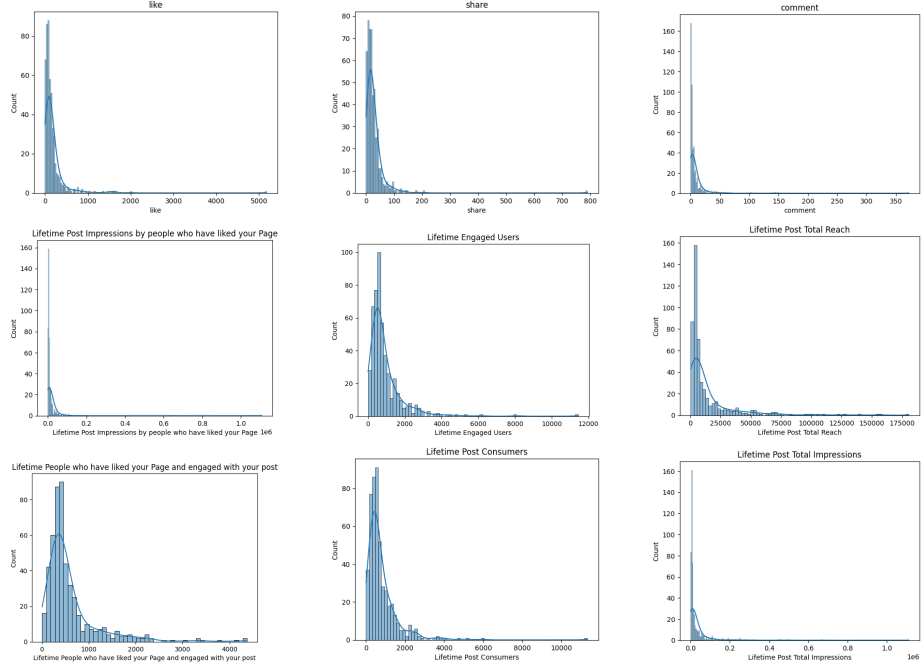


Figure 1: Feature Distributions

It makes sense for the attributes to not be normally distributed, since we are working with real-world data where normality assumptions are often violated. This does not hinder our model, since linear regression does not hinge on the attributes in the data set being normal.

Continuing the exploration of data, we would next like to explore the correlation between features in the feature space. This is helpful in identifying the most highly correlated values to the target variable as well as potential feature multicollinearities.

After performing feature selection, we extracted 9 of the most statistically significant variables from the dataset to perform regression upon. (see: Figure 1) These variables include: shares, likes, comments, as well as various lifetime



Figure 2: Feature Correlation Matrix

post engagement metrics. It is also important to note potential multicollinearities; as highlighted in the heatmap, the shares, likes, and comments are highly relatively correlated while lifetime engaged users and post consumers are highly correlated at a correlation value of 0.968.

In order to continue with the regression, we had to scale our attributes. This is especially important for the second computational method we are working with, Stochastic Gradient Descent. Feature scaling in gradient descent is critical for performance for a number of reasons, but the main idea is this: convergence in the algorithm becomes difficult to obtain due to the difference in scale of different attributes. By appropriately scaling features in gradient descent, the algorithm is able to identify appropriate parameters more efficiently and converge quicker than before.

The final pre-processing step before our model construction and result analysis occurs is simply dividing the dataset into a training and testing dataset. This way, we are able to test our models' generalization against new data points using the training and testing MSE accordingly. We chose a split of 80% training data and 20% testing data.

In summary, various pre-processing steps were performed to prepare the initial data for modeling and analysis. Potential missing and/or null values were identified and accounted for, initial feature and target distributions were analyzed and discussed, feature selection and scaling was performed, and final data was split into training and testing splits for modeling and analysis. The data is

now - finally - ready for the next step.

3 Model Construction

3.1 Introducing Ordinary Least Squares

In linear regression, our objective is often to minimize the error mean squared error such that:

$$MSE = \frac{1}{N} \sum_{i=1}^N (f(x_i) - y_i)^2$$

where the MSE simply represents the difference between the estimated response value and the true response value. Intuitively, the smaller the MSE, the lower the error in the model. By minimizing the error function, we approximate the true function f .

One way in which this problem could be solved is by using simple calculus: calculating the partial derivative of the MSE with respect to weights w_0, \dots, w_p and setting that derivative equal to zero. The intuition lies in that, when the partial derivative equals zero, the MSE would be at its minimal value. Therefore, the weights could be solved for using algebra.

$$\frac{\delta MSE}{\delta w_0} = 0, \dots, \frac{\delta MSE}{\delta w_p} = 0$$

This solution is called the "Ordinary Least Squares" solution. Although this is a relatively slower solution ($O(p^3)$), this solution will return a *global* solution. This means that we know that MSE from this solution will be the lowest in the error space.

3.2 Result Analysis

After fitting the data to the Ordinary Least Squares Regression model, the function outputted the summary below (see: Figure 3)

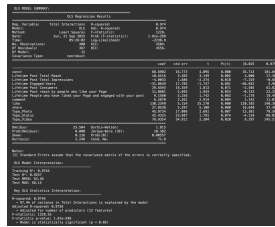


Figure 3: Ordinary Least Squares Summary Output

The first thing to identify are the p-values for each feature in the model summary table. The p-values of each individual feature roughly lie around ~ 0 . This value simply represents the strength of your model in rejecting the null hypothesis H_0 that your attribute is insignificant to the model and can be removed. A small p-value represents a low likelihood of the attribute occurring by chance, and suggests that the null hypothesis can be rejected. In this case, most of the individual p-values are relatively larger and suggest a weaker alternative hypothesis. On the other hand, the total model p-value $p = 1.81e^{-299}$. This is an *extremely* small p-value and suggests that our overall model is statistically significant.

Another model interpretation metric to consider is the R^2 metric. The R^2 metric simply the percentage of the variance in the data that is explained by any model. The OLS model has an adjusted training $R^2 = 0.9736$ and a testing $R^2 = 0.9257$. As well as this, the model had a test $RMSE = 63.46$ and a test $MAE = 50.15$. This data suggests that the OLS fitted model accounts for 92% of the model variance.

Given the relatively weaker individual attribute p-values as well as the testing error values, the OLS regression model is not a completely terrible model fit, but stands to be possibly improved using other methods of regression.

3.3 Introducing Stochastic Gradient Descent

Recalling our linear regression learning objective, we want to minimize the mean squared error formula (MSE):

$$MSE = \frac{1}{N} \sum_{i=1}^N (f(x_i) - y_i)^2$$

The previous solution, although a global solution, is a very slow computational solution. For larger datasets, the OLS solution is simply not viable and requires an alternative: this is where we pivot into the Stochastic Gradient Descent algorithm.

The idea for Gradient Descent is to simply take iterative steps in the opposite direction of a function f at any point in the space. By initializing the weights (w_0, \dots, w_p) to random values and then looking for the local minimum of a specific MSE curve through an iterative function:

$$w_p^{new} = w_p^{old} - \mu \left(\frac{\delta MSE}{\delta w_p} \right) \forall p$$

where w_p is the weight for predictor p , μ is a pre-assigned learning rate, and $\frac{\delta MSE}{\delta w_p}$ is the gradient of the error function with respect to the weight w_p .

This algorithm does not guarantee a global minimum by any means; gradient descent is simply going down a specific curve on the MSE function landed on through the randomization of weights. However, this is an almost linear function that works to mitigate the previous time-complexity problem arising from the OLS solution.

3.4 Result Analysis

When analyzing the results for the Stochastic Gradient Descent regression algorithm, it is first important to tune hyperparameters in the model. Since these parameters are set pre-modeling, one way in which a model could be improved is by identifying the best value for any hyperparameter in a parameterized model to obtain the most optimal results. In this case, we would be tuning the optimal learning rate, μ , such that algorithm converges at the quickest rate. The model converged the fastest at 257 iterations when $\mu = 0.001$.

With the optimized SGD model, both the training and testing $R^2 = 0.9998$, indicating almost all of the variance in the data can be explained by the model. This statistic is significant due to the lack of drop in R^2 during the testing stage; this suggests high generalizability in the SGD regression model, a key property of a successful linear model.

The high R^2 value, as well as other key test metrics such as the test $RMSE = 2.9$ and the test $MAE = 1.47$ suggests a well-fit, highly generalizable model.

4 Conclusion