­

**Linear Regression: Multiple Approaches Application Study**

**Student Assessment Number: J** (Add your **J number** here NOT your name or student number)

**Module Code**: CO7024

**Module Title**: Statistical Programming

**Word count**: (add YOUR word count here)

**Date of Submission:** (add the submission date here)

# Introduction

The technological advancement known as machine learning (ML) emerged within artificial intelligence (AI) and data science as an essential element of transformation. As a computational framework, machine learning enables systems to learn by processing data to make decisions automatically while lacking explicit programming for individual operational tasks. By analysing large datasets, ML algorithms develop predictive patterns that help multiple fields like healthcare, finance, and engineering. Engineers utilize machine learning to drive predictive toxicology since it enables more efficient in-silico approaches that produce improved predictions (Wang et al., 2020). Machine learning has various algorithms that apply multiple statistical techniques. Algorithms such as the Decision Tree which was introduced by Quinlan (1986) who explored the theoretical approach of the trees through the ID3 algorithm, the Support Vector Machine (SVMs) which is a useful technique for both classification and regression tasks creating a hyperplane that best separates data points (Cortes & Vapnik, 1995), and the linear regression.

Linear regression (LR) is a fundamental statistical technique that enables the discovery of relationships between dependent and one or more independent variables. LR functions are mainly used for making predictions and drawing conclusions across various fields of science and engineering. Linear regression models gain widespread use because of their interpretability and simplicity which makes them foundational within statistical analysis as well as machine learning paradigms. That is to say, A wide range of libraries can apply LR in Python including, Statsmodels, Scikit-learn, SciPy, TensorFlow/Keras, and NumPy

This study aims to apply LR in Python programming language using two different libraries such as SciPy and Scikit-learn and compare the data preprocessing tools that they provide, the ease of model fitting and interpretation, the output evaluation metrics they both offer, and the limitations of each one of them.

# Methods

LR is one of the most popular and widely used statistical techniques that can build a model using one or more variables. It predicts the values depending on the type or the variation of LR that the problem requires. LR has many variations like Simple LR, Multiple LR, Ridge regression, Lasso regression, and Elastic Net regression. Each of these variations applies the LR equation but with some differences. For example, the LASSO regression adds a regularization term to the original equation so that it can shrink the estimation of coefficients to decrease them which will reduce the model’s variance (Czajkowski et al., 2023).

## Simple Linear Regression

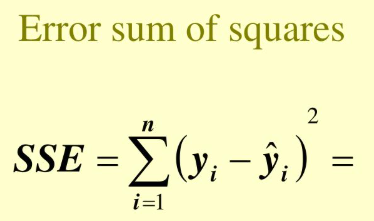
Simple Linear Regression (SLR) is used for one vs one variable prediction which is represented by the following equation:

A diagram of equations and numbers

AI-generated content may be incorrect.

(Fig 1 Simple Linear Regression VS Multiple Linear Regression)

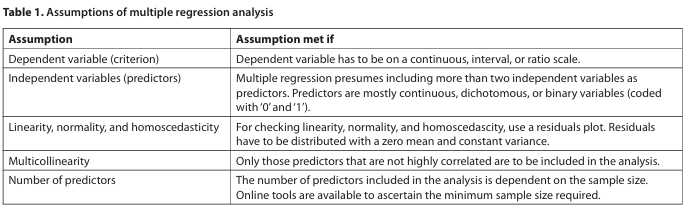
Y’ is the predicted value (the outcome), B1 is the regression coefficient (the slope), B0 is the intercept or the constant, and x1 is the variable or the predictor. Applying the equation will give Y’ as an output. The error between the predicted and actual values can be calculated using mathematical methods such as square error sum (seen in Fig 2). The minimal value of the sum of squared errors shows the best-fitting line of prediction for the dataset.



(Fig 2: Sum of Square Error)

## Multiple Linear Regression

Although SLR is an efficient method for predicting data, one of the disadvantages is it's limited to one variable or predictor. Multiple linear regression (MLR) is a way to overcome this problem. The multiple linear regression model is a statistical method that adapts the SLR model to use multiple predictors. The MLR equation (found in Fig 1.) has the coefficient (b) which reveals that any unit change in the independent variable affects the dependent variable when all other values remain unchanged. Furthermore, as a part of the SLR, minimizing the sum of squared differences serves the model fitting and the coefficient estimation (Fransiska et al., 2022)

MLR requires a linear relationship between the variables, the predictor or the dependent variable (Y value) to be continuous, and some other assumptions. Bazdaric et al (2021) made some assumptions which can be found in Table 1. Failure to meet these assumptions could result in unreliable and possibly biased regression coefficients.

The evaluation of linear regression (both simple and multiple linear regression) model fit requires examination of the coefficient of determination (R^2) because this metric shows the percentage of independent variable influence on dependent variable variation (Ng et al., 2018). The assessment of a regression model's validity through (R^2) does not guarantee a proper and useful model so researchers need to perform additional verification tests to establish assumption compliance (Flatt & Jacobs, 2019). Yan et al (2016) discuss doing a residual analysis which involves plotting residuals against predicted values and independent variables to check for assumption violations.

## The Dataset

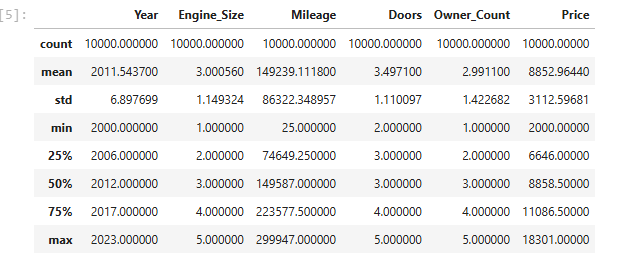
The car price dataset (2002–2023) is ideal for linear regression because it contains modern real-world information. The model can directly predict car values because its requirement for numerical outcomes matches the continuous target variable (Price). The dataset contains 10,000 records, providing faster computation and supporting a simple training process.  Diverse variables such as mileage, year, and car brand allow for interpretable feature engineering. The system maintains practical application for used-car valuation, presenting actionable insights that technicians and non-technical stakeholders can understand.

Preprocessing and exploratory data analysis (EDA) are critical for ensuring model accuracy. The dataset expresses no null values, duplicates, spelling errors, or incorrect data types throughout the entire dataset. Feature engineering including binning the 'Mileage' column into five groups helped with both model interpretation and performance. Despite the risk of high dimensionality, it is practical to use one-hot encoding on this dataset due to the limited number of categorical variables (Cerda & Varoquaux, 2022). A Shapiro-Wilk test revealed a non-normal distribution (p-value = 0) of 'Price' data, so a log transformation is applied after removing outliers. finally, a correlation matrix helped the identification of the 10 most important features for training, ensuring a strong effective linear regression model.

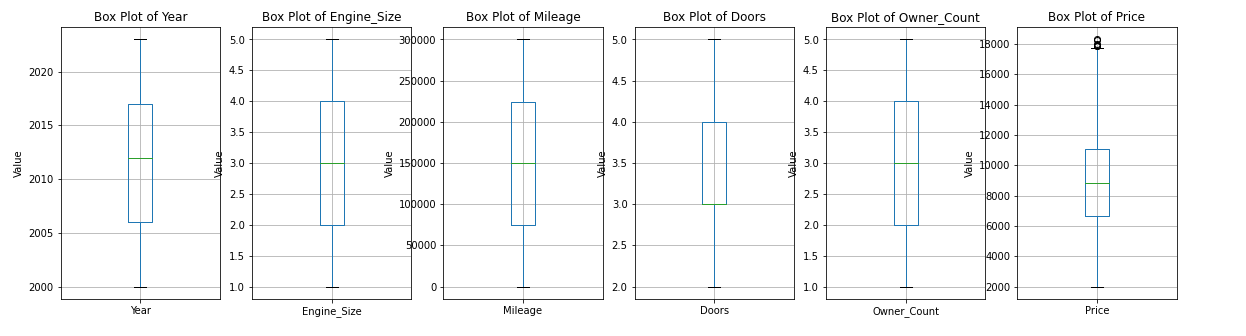
# Results

Close examination of the dataset integrity and balance was performed during extensive EDA testing (Fig 3–8). Summary statistics (Fig 3) showed stable column data except for ‘Price’ which had an outlier distribution leading to observations in Fig 4. Right-skewness in the target variable 'Price' appears in Figure 6 before the application of the interquartile range and log transformation corrected it. The analysis revealed that electric-engine vehicles hold greater average prices than combustion-engine ones (Fig 7) furthermore, the correlation matrix (Fig 8) displayed 'Year', 'Engine Size', and 'Mileage' as key price determinants. The findings indicate that after performing one-hot encoding the analysis discovered positive correlations between price and the transmission type (automatic) as well as hybrid and electric engine cars, so the results match observations made in Cerda & Varoquaux (2022) about encoding categorical variables. These observations are utilized to derive a better understanding and improve the dataset for stable linear regression analysis.

The scikit-learn model displayed high success by demonstrating an 88% mean R² score through five-fold cross-validation thus explaining 88% of the data variance. The SciPy implementation using default parameters demonstrated an R² value of 55% according to Fig 10 and 11. The performance gap between the two models seems to be due to their varying methodologies because scikit-learn employs cross-validation which delivers results across multiple splits, but SciPy's default method uses a single train-test split, of 80% as training data and 20% as testing data, that might be affected by data imbalance or overfitting. Moreover, Scikit-learn optimized implementations such as feature scaling and categorical encoding would fit the dataset better than the SciPy defaults.



(Fig 3. Summary statistics of the dataset)



(Fig 4. Box plot of the outliers)

A green bar code and bar code

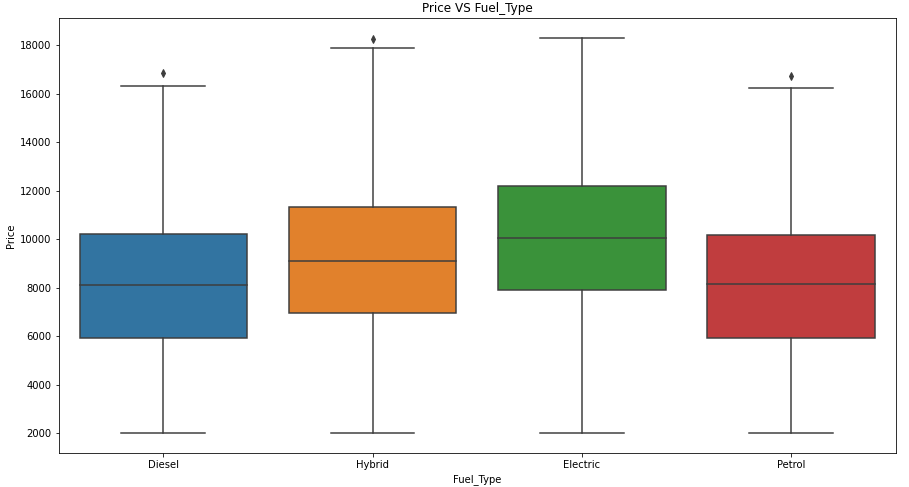
AI-generated content may be incorrect.

(Fig 5. Frequency of the values in discrete columns)

A comparison of red and black graphs

AI-generated content may be incorrect.

(Fig 6. Distribution plot of ‘Price’ and ‘Mileage’ columns)



(Fig 7. Price of the car VS Fuel\_Type)

A screenshot of a computer screen

AI-generated content may be incorrect.

(Fig 8. Correlation matrix of the numerical values)

A screen shot of a computer

AI-generated content may be incorrect.

(Fig 9. Most Co-related values after One-Hot encoding application)

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect. (Fig 10. Scikit-learn model scores)

(Fig 11. SciPy model scores)

# Discussion

The superior performance of scikit-learn’s linear regression model (88% R² with 5-fold cross-validation, Fig 10) over SciPy’s (55% R², Fig 11) underscores critical methodological and practical distinctions between the libraries. The power of Scikit-learn lies in the automated preprocessing using the built-in functions. It combines feature engineering steps such as encoding categorical variables (one-hot, label, and target encoding), feature scaling, outlier handling, cross-validation for generalization assurance, and more (Vangara et al, 2021; Cerda & Varoquaux 2022). The usage of 5-fold cross-validation ensures the model outputs by iteratively training the model on different subsets of the data generating a reliable performance and avoiding the overfitting problem. Thus, the manual approach of SciPy combined with a single train-test split makes the method susceptible to overfitting problems and data imbalances especially in scenarios with skewed or correlated features (e.g., Engine\_Size vs. Price Fig 8) because it needs manual preprocessing (Bazdaric et al., 2021). The statistical diagnostic abilities (p-values, t-tests) along with the lightweight functionality that SciPy provides make it suitable for specific tasks (Virtanen et al., 2020). but scikit-learn’s ability to manage end-to-end ML operations and feature engineering and scalability positions makes it better for picking car prices through prediction. Real-world models work best with scikit-learn since preprocessing and validation function effectively, but SciPy remains an option that is only suitable for minimal and transparent statistical analysis.

Wang, M. W. H., Goodman, J. M., & Allen, T. E. H. (2020). Machine learning in predictive toxicology: recent applications and future directions for classification models. Chemical Research in Toxicology, 34(2), 217-239. <https://doi.org/10.1021/acs.chemrestox.0c00316>

Quinlan, J. R. (1986). *Induction of decision trees*. Machine Learning, 1(1), 81–106. <https://doi.org/10.1007/BF00116251>

Cortes, C., & Vapnik, V. (1995). *Support-Vector Networks*. Machine Learning, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>

Czajkowski, M., Jurczuk, K., & Kretowski, M. (2023). Steering the interpretability of decision trees using lasso regression - an evolutionary perspective. *Information Sciences*, *638*, 118944-. <https://doi.org/10.1016/j.ins.2023.118944>

Bazdaric, K., Sverko, D., Salaric, I., Martinovic, A. & Lucijanic, M. (2021). The ABC of linear regression analysis: What every author and editor should know. European Science Editing (47, e63780). <https://doi.org/10.3897/ese.2021.e63780>

Fransiska, W., Nugroho, S., & Rachmawati, R. (2022). A comparison of weighted least square and quantile regression for solving heteroscedasticity in simple linear regression. Journal of Statistics and Data Science, 1(1), 19-29. <https://doi.org/10.33369/jsds.v1i1.21011>

Ng, S., Chew, Y., Ch’ng, P., & Pinang, P. (2018). An insight of linear regression analysis. Scientific Research Journal, 15(2), 1. <https://doi.org/10.24191/srj.v15i2.9347>

Yan, L., Sun, Y., Boivin, M., Kwon, P., & Li, Y. (2016). Revealing facts and avoiding biases: a review of several common problems in statistical analyses of epidemiological data. Frontiers in Public Health, 4. <https://doi.org/10.3389/fpubh.2016.00207>

Vangara, V. K. M., Vuddanti, S., & Kakani, B. (2021). An accurate and fast computational Python-based module for linear regression analysis in data science applications. In *2021 IEEE International Conference on Intelligent Systems, Smart and Green Technologies (ICISSGT)* (pp. 167–170). IEEE. <https://doi.org/10.1109/ICISSGT52025.2021.00043>

Cerda, P., & Varoquaux, G. (2022). Encoding High-Cardinality String Categorical Variables. *IEEE Transactions on Knowledge and Data Engineering*, *34*(3), 1164–1176. IEEE Transactions on Knowledge and Data Engineering. https://doi.org/10.1109/TKDE.2020.2992529

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Walt, S. J. van der, Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … Contributors, S. 1 0. (2020). SciPy 1.0—Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

The car price dataset represents an excellent selection for analysis because it presents relevant contemporary data covering the period from 2002 until 2023. Modern market trends and technological advancements are included within this established time frame which works well for modeling real-world scenarios, the target value (the car price) is continuous which makes it a perfect fit for the linear regression model. moreover, it is a real-world business problem which makes it relevant and easy to explain to ordinary users, it has 10 thousand records making it computationally efficient for linear regression, and it has a diversity of variables such as Brand, Mileage, Year, and Price.

The pre-processing and Exploratory data analysis (EDA) phase is crucial for an accurate model. Applying EDA techniques shows the absence of null values, duplicated rows, column spelling mistakes, and wrong column data types. Moreover, binning the ‘Mileage’ column shrinks the values into five different values which helps the model for better use of the column. Cerda & Varoquaux (2022) discuss that one-hot encoding results in a high dimensionality feature vector. However, it is a popular approach for datasets with small column numbers. The ‘Price’ column was found to have some outliers which are discarded. Furthermore, a normality test was conducted on the dataset using Shapiro statistical method which results in ’p-value’ of 0 presenting that the dataset doesn’t follow a normal distribution, log transformation on target value (Price) was applied as a solution to the normality problem. Correlation matrix was explored to select the most effective 10 columns for model training.

The describe function (shown in Fig 3) shows a balanced dataset in terms of values and most of the columns don’t have outliers. Fig 4 is a box plot of the outliers and shows no outliers except in the target value (Price), Distribution graph ensures the skewness of the price column (Fig 6). The dataset seems to have consistent values in most of the columns (Fig 5). Furthermore, it is found that the average car price goes up if it contains an electric engine (Fig 7). Fig 8 represents a correlation matrix that demonstrates the values related to the price which are year, engine size, and mileage as expected. Yet after the encoding, the cars that have automatic transmission type and cars with electric or hybrid engines show a correlation to the price.

The scikit-learn model produced a model with a mean R square value of 88% using 5 folds of cross-validation which means that the model explains 88% of the variance of the data given to it. While, when using SciPy with default values it shows R square scores of 55% according to Figure 11

Disscusion

Vangara et al. (2021) developed a new linear regression model which was compared to the SciPy and scikit-learn linear regression model meaning that both libraries' models are accurate, dependable, and relatively fast. The usage of 5-fold cross-validation ensures the model outputs by iteratively training the model on different subsets of the data. This generates a reliable performance and avoids the overfitting problem. Unlike SciPy which only can use single train-test-split, manually or provided by Scikit-learn, which makes the model prone to overfitting the training data and fails to generalize if the data is imbalanced. Therefore, the Scikit-learn model scored a significantly higher score than the SciPy model (88% for sklearn vs 55% for SciPy). Likewise, the Scikit-learn library supports a lot of feature engineering methods such as feature scaling, categorical encoding (label, target, and one-hot encoding), outlier handling, and more. On the other hand, Scipy assumes that the data meets linear regression assumptions (Bazdaric et al (2021) for example the normality and homoscedasticity of the data without pre-processing. It also assumes that there is no multicollinearity making it suffer with the relationships between variables (seen in correlation matrix, fig 8).

Virtanen et al (2020) reported that SciPy is a base library for Scikit-learn and Scikit-image and is used for complex statistical problems. Statistical tests such as t-tests, and chi-square tests can be done by SciPy. moreover, doing N-dimensional interpolation which involves the estimation of unknown points using known points. SciPy applies linear regression manually which makes it efficient for small and medium-sized datasets. it gives the user the ability to easily plot the line fitting the predicted data and retrieves the values of the slope, interception, and standard deviation error. It also gives the user the ability to tune more hyperparameters compared to Scikit-learn.