Car Sales Prediction

Emmanuel Opoku Hinneh   
*Department of Data Science*  
*Carolina University*Winston Salem, NC, USA  
hinnehe@carolinau.edu

*Abstract*—This paper explores how to perform an EDA, target variable selection, choosing an appropriate model, and analyzing the evaluation metrics of a model.

Keywords—eda, mse, rmse.

# Introduction

The machine learning life cycle keeps evolving, but the fundamental steps involve planning, data preparation, model engineering, model evaluation, model deployment, and monitoring and maintenance. Organizations must go through all these steps to make their machine-learning journey successful.

In terms of our preparation, our dataset is car sales data. It consists of 4340 data points with 8 features: **name**, **year**, **selling\_price**, **km\_driven**, **fuel**, **seller\_type**, **transmission**, and **owner**. Our goal is to explore the data, engineer it, and build a model that can predict how much a car might be sold, given these features. We will not be considering deployment and maintenance.

Section II will focus on the EDA; section III will focus on model engineering and the different models we can build as a solution, section IV will discuss the model evaluation, and Section V will conclude.

# Exploratory Data Analysis (EDA)

Improving raw data is one of our priorities for data-driven EDA procedures. This process relies on both data preprocessing and feature engineering. EDA includes, but is not limited to, data integration, analysis, cleaning, transformation, and dimension reduction. Preprocessing prepares the data for feature engineering, which is manipulated during the feature engineering stage. Some of the manipulations are removing irrelevant features, handling missing data, encoding variables, and dealing with categorical variables. Fortunately, the car sales data has all the data and does not have errors. After careful feature engineering, we had these features in our dataset: ***year***, ***selling\_price***, ***km\_driven***, ***fuel***, ***seller\_type***, ***transmission***, ***owner***, ***brand***, and ***model***.

Next, we explore the data visually. We separate the numerical data from the categorical data to make this easier.

The graphs in Fig. 10 and Fig. 11 show a histogram of the features.

A blue and white graph

Description automatically generated

Fig. 1. Year histogram

A comparison of a bar graph

Description automatically generated

Fig. 2. Kilometers Driven histogram

Fig. 3 shows the multivariate comparison of the numerical features, and Fig. 4 shows a heatmap of the features.

A group of blue dots

Description automatically generated

Figure 3. Bivariate analysis of numerical values.

A screenshot of a chart

Description automatically generated

Figure 4. Heatmap of numerical values.

The graph in Fig. 5 shows a bar chart of the categorical data and their respective percentages.

A graph of different types of numbers

Description automatically generated with medium confidence

Fig. 5. Categorical features.

The graph shows almost as many petrol-powered cars (48.9%) sold as diesel-powered cars (49.6%). However, most cars sold had manual transmission (89.7%). The majority of the cars were sold to first-time owners (65.3%), and the majority of the sellers were individuals (74.7%).

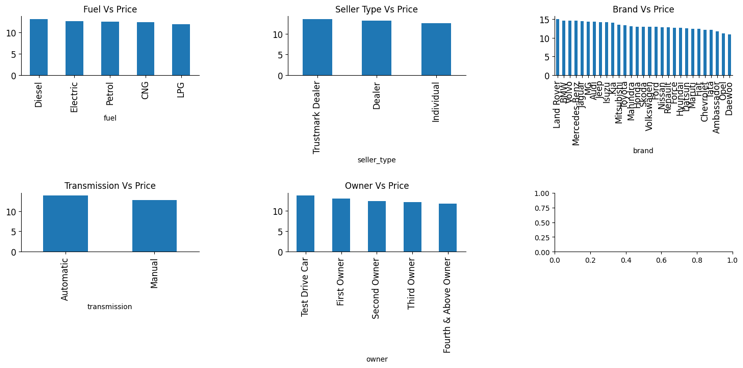


Fig. 6. Bivariate analysis of categorical features.

From Fig. 6:

* Selling Price is negatively correlated with kilometers driven.
* Diesel cars are more expensive.
* Trustmaker dealers sell their cars at an expensive price.
* Automatic transmissions are more expensive.
* Cars that were first test-driven are more expensive.
* The selling price is positively correlated with the year.

Looking at the scale of the kilometers driven and the selling price, it’ll be best to normalize the values so that they do not take more weight during the model training.

# Model engineering

The aim is to build a model that can predict the selling price of a vehicle, given our features. Hence, the target variable in the dataset is the “selling\_price”. The algorithm of choice has to consider the type of value we expect from the model. Since our expected output is a numerical (continuous) value, we can consider linear regression, decision tree, or random forest. Before building our model, we also need to encode our categorical data. One assumption was made concerning the card brand. We’re going to assume that the car brand correlates with the price of the car. Target Encoding was used for car brands while one-hot-encoding was used for all other categorical features. The features to be used in building our model are year, “selling\_price\_log,” “km\_driven\_log,” fuel, “seller\_type,” transmission, owner, and brand.

## Linear Regression

Linear regression models the relationship between variables by fitting a linear equation to the observed data. The predicted variable is the dependent variable, and the ones that help the prediction are independent variables. It has an equation as shown below.

is the independent variable, and is the dependent variable. is the slope of the line, and is the intercept.

After training and testing the model, these are some of the findings.

**Mean Squared Error (MSE): 0.1668**

This metric measures the average squared difference between predicted and actual values. Lower MSE indicates better performance. In our case, the model makes predictions that are, on average, squared 0.1668 away from the actual price.

**Root Mean Squared Error (RMSE): 0.4084**

This is the square root of MSE. It's another way to measure prediction error on the same scale as the actual values. A lower RMSE is desirable. Here, the RMSE is 0.4084, which can be interpreted as an average difference of 0.4084 between predicted and actual prices.

**Mean Absolute Error (MAE): 0.3118**

This metric calculates the average of the absolute differences between the predicted and actual values. It's less sensitive to outliers compared to MSE. A lower MAE indicates better performance. In this case, the average absolute difference between the predicted and actual price is 0.3118.

**R-squared Score (R²): 0.7582**

This metric represents the proportion of variance in the dependent variable that can be explained by the model's independent variable(s). It ranges from 0 to 1, with higher values indicating a better fit. An R² of 0.7582 suggests that the model's predictions can explain 75.82% of the variance in the actual values.

**Explained Variance Score: 0.7583**

This metric is very similar to R-squared and provides the same interpretation.

## Decision tree.

A decision tree generates regression or classification models as a tree structure. It divides a dataset into smaller subsets while gradually developing an associated decision tree. The final output is a tree with both decision and leaf nodes. The algorithm's primary principle is to discover the point in the independent variable at which the mean squared error is minimized. The algorithm executes this repeatedly, resulting in a tree-like structure. Fig. 7. Shows a tree of the model.

A diagram of a company structure

Description automatically generated

Fig. 7. Decision tree.

Similar to the linear regression model, the values of our evaluation metrics are:

* Mean Squared Error: 0.2837
* Root Mean Squared Error: 0.5327
* Mean Absolute Error: 0.4194
* R-squared Score: 0.5886
* Explained Variance Score: 0.5886

## C. Random Forest

Random forest combines multiple decision trees to make a better prediction. It’s an example of an ensembling method that utilizes the bagging method. It reduces the risk of overfitting that the decision tree introduces but consumes much more resources than the decision tree. Fig. 8 shows a portion of the random forest.

A diagram of a company structure

Description automatically generated

Fig. 8. Random Forest.

Below are the evaluation metrics from the random forest regressor.

* Mean Squared Error: 0.1673
* Root Mean Squared Error: 0.4090
* Mean Absolute Error: 0.3134
* R-squared Score: 0.7574
* Explained Variance Score: 0.7575

# Conclusion

The lowest MSE is achieved by Linear Regression (0.1668), followed by Random Forest (0.1673), and then by Decision Tree (0.2837). A lower MSE indicates better performance regarding the model's ability to predict accurately. Similar to MSE, lower RMSE values indicate better model performance. Again, Linear Regression has the lowest RMSE (0.4084), followed by Random Forest (0.4090), and then Decision Tree (0.5327). The lowest MAE is again achieved by Linear Regression (0.3118), followed by Random Forest (0.3134), and then by Decision Tree (0.4194). Lower MAE values suggest better accuracy. Linear Regression has the highest R-squared score (0.7582), which explains the highest proportion of variance in the target variable among the three models. Random Forest comes second (0.7574), followed by Decision Tree (0.5886). Similar to R-squared, Linear Regression has the highest Explained Variance Score (0.7583), followed by Random Forest (0.7575), and then Decision Tree (0.5886).

Overall, Linear Regression generally outperforms both Decision Tree and Random Forest models in terms of these evaluation metrics, with Random Forest closely following and Decision Tree being the least performing among the three models.