**Özyeğin University**

**CS551 – Data Science with Python**

**Assignment-1**

**Car Pricing Prediction by Linear Regression**

Furkan Cantürk

21.11.2020

# 1. INTRODUCTION

Linear regression is easy to understand, and to model. It might be implemented as a preliminary model to interpret linear relation between dependent variables (i.e., features) and an independent variable (i.e., target, respondent). Our car price dataset more than one feature, so, multiple linear regression is used in this report.

Multiple linear regression models can be depicted by the following equation.

(1)

y is a dependent variable which is subject to predict. In linear regression, y is always a numerical variable and cannot be categorical. 𝑥𝑖’s are independent variables which are taken into the consideration to predict y, where k is the number of independent variables. βi’s are coefficients which determine how much a unit change in 𝑥𝑖 changes y while other variables remain constant. β0 is the intercept value which is the model output if there is no effect of any independent variables on the dependent variable. 𝜀𝑖’s are the random errors, i.e., residuals.

There are some assumptions for linear regression:

*Linearity*: The true relationship between the dependent variable and independent variables is linear.

*Multivariate Normality*: Residuals are normally distributed around a zero mean with a constant variance. Also, they are independent, i.e., not dependent on any independent variable.

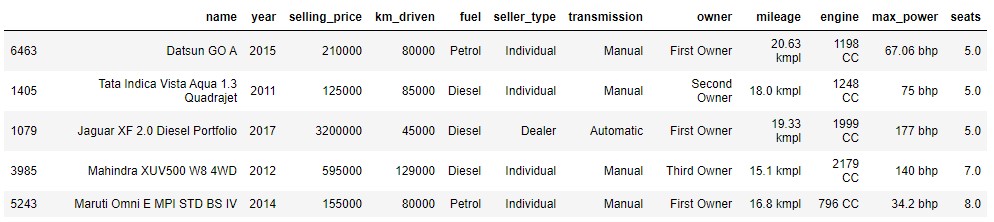
*Homoscedasticity*: Regression line has the same variance over all values of an independent variable.

*Multicollinearity*: Independent variables are not highly correlated with each other.

Unless data is consistent with the assumptions, performance of linear regression models deteriorates.

# 2. EXPLORATORY DATA ANALYSIS

**Table 1. Sample from Car Pricing Data Set**

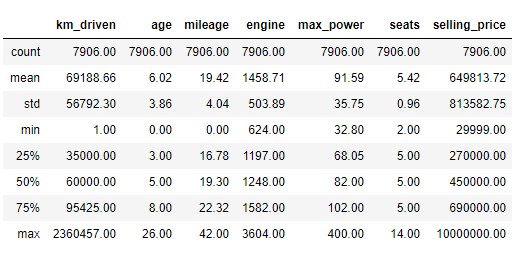


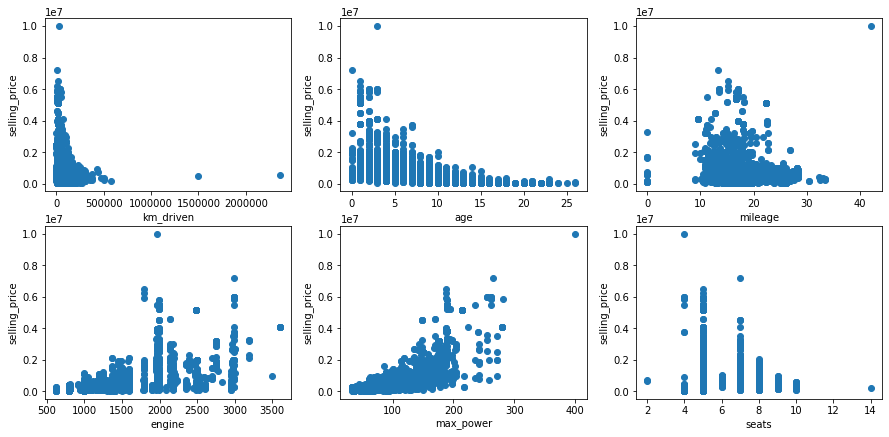
# Table 2. Number of non-missing values in each feature and the target variable A screenshot of a computer program Description automatically generated

There are 8128 observations in the dataset and 6717 observations are unique. 11 features are listed in Table 1 and 2. Some observations have missing values for some features. Total number of these observations is 221, which is about 2.72% of the data. After dropping them, 7907 left.

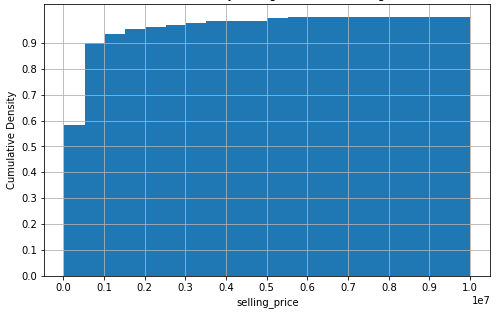
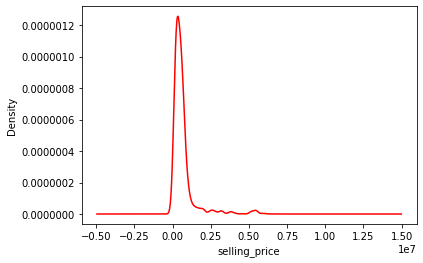
*km\_driven*, *age*, *mileage*, *engine*, *max\_power*, and *seats* are the numerical features where *age* the converted feature, equivalent to 2020 - *year.* *name, fuel, seller\_type, transmission,* and *owner* are the categorical features.

**Table 3. Statistics of the Numerical Features and the Target Variable**





# Figure 1. Distributions of Numerical Features Over Selling Price

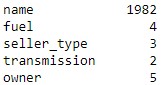


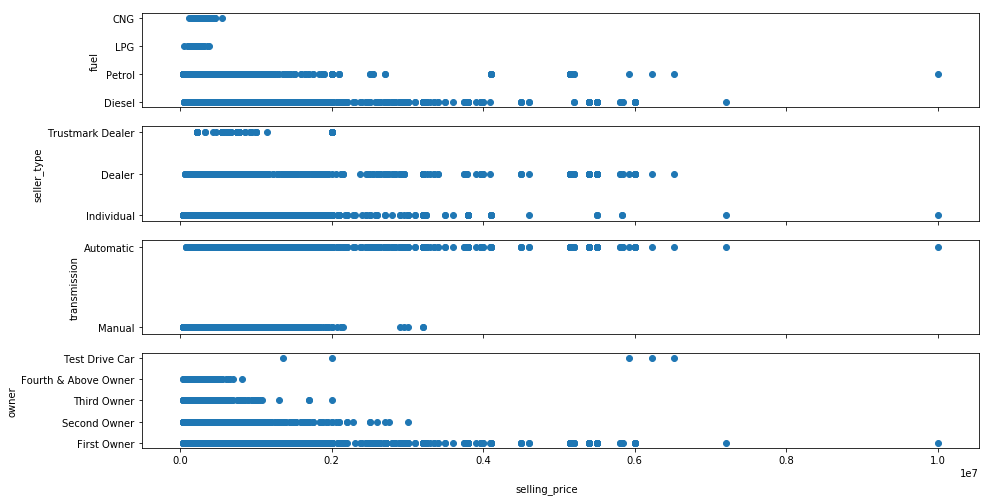
# Figure 2. (i) Density Distribution and (ii) Cumulative Density Histogram of Car Selling Price

90% of the observations is accumulated up to 0.1 \* 107 selling price according to Figure 2(i). Besides, they are not normally distributed based on the target. So, these show that a linear regression model to be fitted on this data is expected to yield high errors. Besides, *seats*, *mileage*, and *km\_driven*,are not linearly correlated with the target.

Outliers should be removed, or Standard Normalization should be applied to data but there is only one outlier whose *selling\_price* value is 107, which is ineffective alone to change regression fit against 7907 observations.

# Table 4. Values Count of Categorical Features

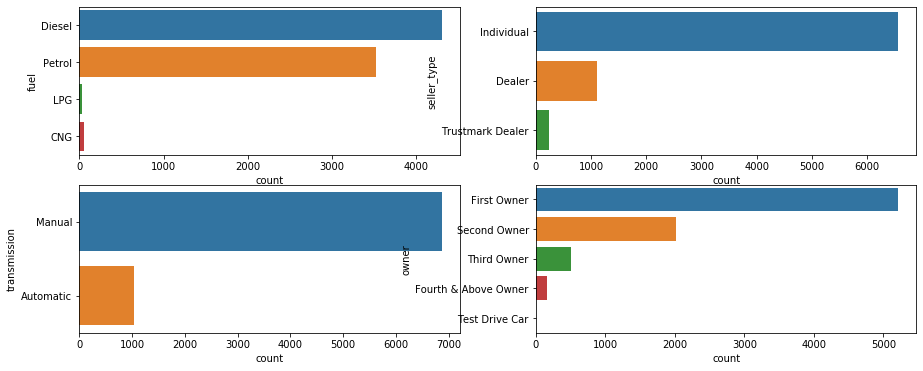




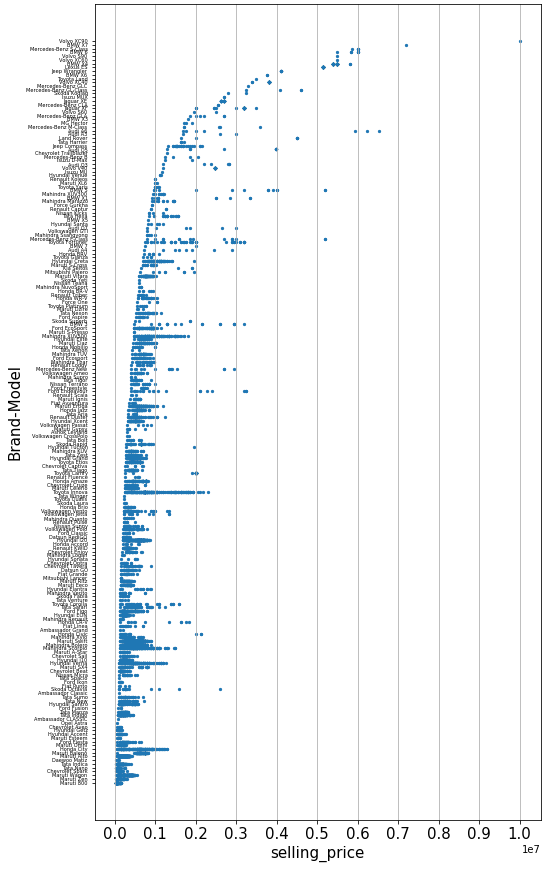
# Figure 3. Price Distributions of Each Level of Each Categorical Feature Except “name”

To use categorical features in a linear regression model, they are needed to be convert to a numerical form. So, I encode each categorical feature as a group of bits representation with One-Hot Encoding. For example, fuel feature is encoded as [0,1,0,0] if its value is *LPG*. So, it is converted to 4 features to be used in the model.

*name* has 1982 different values. It means 1982 new features (after One-Hot Encoding) if *name* is chosen to use in car price prediction. However, using *name* directly in the modelling is problematic with 2 reasons. Firstly, this model, using *name* as a feature, is harder to understand individual effect of each feature on car price among thousands of features. Secondly, more importantly, this model will become quickly impractical as time passes since each new car represents a new feature with its unique name. So, for each a new car name, we will need to update the model by adding a new variable of new name. Also, a general model is aimed to predict the price of any car, not using a separate feature for each specific car name.

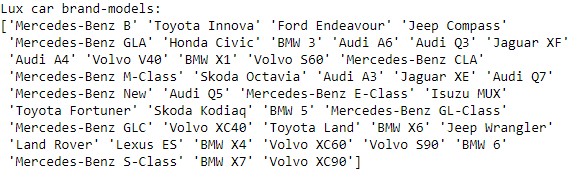


# Figure 4. Value Counts of the Categorical Features Except “name”

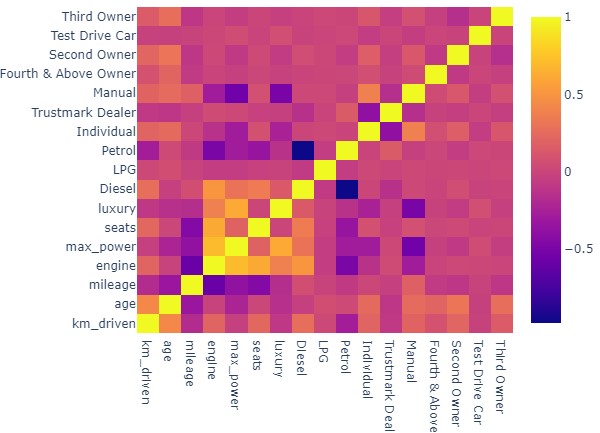


**Figure 5. Price Distribution of Car Brand-Model Levels**

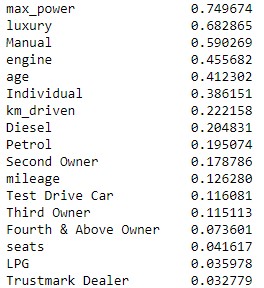
Due to high cardinality of *name,* its value levels should be reduced if it is used in regression of car price. So, a new feature named *brand\_model* is created from *name* by getting first two words in it. *brand\_model* has 200 unique values. When price distribution of brand-model levels is inspected over Figure 5, we see that the brand-model levels whose prices are less than 0.2\*107 are distributed around an almost linear curve although some brand-model levels have a high variance in their prices. So, *brand\_model* can be clustered in 2 with a new feature named *luxury.* Accordingly, the list of brandmodel types whose all observations’ prices are larger than 0.2\*107 is extracted. If *brand\_model* of an observation is in that list, its *luxury* value takes 1, otherwise 0.



I note that this clustering might be done in more than 2 clusters and different split values (e.g. 0.1\*107, 0.3\*107 etc.) should be experimented, but optimizing the clustering of dataset is beyond the scope of report. So, this optimizing phase is skipped, and the split value was determined by just observation.



# Figure 6. Heatmap of Correlation Matrix



# Table 5. Correlations with Car Price

*engine*, *Diesel, luxury, seats,* and *max\_power* are the variables which highly correlated with each other according to the correlation matrix in Figure 6.

In Table 5, the first 6 variables have a correlation with the target at rates larger than 0.38. Correlation rates of the rest is less than 0.23. So, there is a clear separation of highly correlated and low correlated variables with the target.

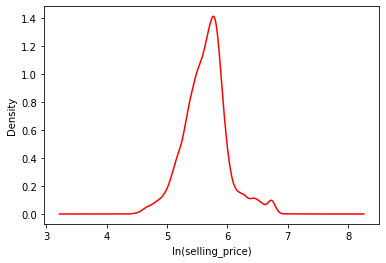
# 3. MODELLING AND RESULTS

Considering the exploratory data analysis, 5 linear regression models are built. Before fitting the regression lines, all numerical inputs are normalized with min-max scaling. For every feature, the minimum value of that feature is transformed into 0, the maximum value is transformed into 1, and every other value gets transformed between 0 and 1 with formula: (value – min\_value) / (max\_value – min\_value). Besides, all categorical features are one-hot encoded, as said in the previous section, but the first bit in the binary representation is dropped to avoid multicollinearity. For example, *transmission* is represented with only one bit, named *Manual.* If *Manual*=0 for an observation, then it means *transmission* value of that observation is *Automatic*. If *transmission* is represented with two bits, i.e. two separate features as *Manual* and *Automatic,* each bit correlates with each other since 0-1 is the reversed of 1-0 representation.

The transformed data set is splitted into training and test datasets where proportion of test data to all is 33%. Each model is trained with the same training dataset and tested with the same test dataset.

# Table 6. Variables, Target, and Test Scores of the Models

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Model-1** | **Model-2** | **Model-3** | **Model-4** | **Model-5** |
| **x1** | *max\_power* | *max\_power* | *max\_power* | *max\_power* | *max\_power* |
| **x2** | *Manual* | *Manual* | *Manual* | *Manual* | *Manual* |
| **x3** | *engine* | *engine* | *engine* | *engine* | *Petrol* |
| **x4** | *age* | *age* | *age* | *age* | *age* |
| **x5** | *Individual* | *Individual* | *Individual* | *Individual* | *Individual* |
| **x6** |  | *luxury* |  | *luxury* | *luxury* |
| **y** | *selling\_price* | *selling\_price* | ln(*selling\_price*) | ln(*selling\_price*) | ln(*selling\_price*) |
| **R-Square** | 0.678 | 0.743 | 0.823 | 0.875 | 0.908 |
| **RMSE** | 464,055.111 | 414,022.749 | 343,496.846 | 289,219.81 | 248,503.33 |
| **MAE** | 273,839.53 | 238,327.976 | 155,315.837 | 139,399.806 | 129,646.441 |



**Figure 7. Density Distribution of ln(*selling\_price*)**

Model-1 is constructed with the first 5 variables in Table 5 except *luxury*, and contribution of *luxury* to prediction success is observed with Model-2.

Density distribution of prices was observed in Figure 2 (i), which is highly skewed. Logarithmic transformation is applied on the target so that a more normalized distribution is obtained as in Figure 7 to be more consistent with the linear regression assumptions. Performance difference of Model-1 and Model-3 shows the contribution to the prediction success made by this transformation.

Both transformation of *name* into *luxury* and logarithmic transformation of *selling\_price* is applied with Model-4, which achieves better performance than the previous ones.

To build the last model, all variables are evaluated: Highly correlated variables with target are selected and the variables causing high multicollinearity are removed. The only difference of Model-5, compared to Model-4, is that *Petrol* isselected rather than *engine* since *engine* causes high multicollinearity as seen in Figure 6.

# 4. IMPLEMENTATION

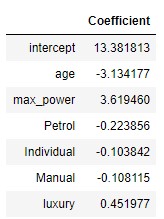
All data visualization, pre-processing, and modelling is done with Python libraries such as Pandas, Numpy, Seaborn, Plotly, and Scikit-learn. The code is at Appendix as a Jupyter Notebook.

# 5. CONCLUSION

For a transformed observation (i.e. encoded categorical variables, and min-max scaled numerical variables), coefficients for the linear regression equation formula of Model-5 is demonstrated in Table

7. *Diesel, engine, seats, mileage, km\_driven, LPG, Third Owner, Second Owner, Test Drive Car, Trustmark Dealer,* and *Fourth & Above Owner* are the discarded variables in Model-5, i.e. their coefficients are 0’s.

**Table 7. Coefficients of Model-5**



Linear equation of Model-5 as follows.

ln (price) = 13.381813 − 3.134177 ∗ age + 3.619460 ∗ 𝑚𝑎𝑥\_𝑝𝑜𝑤𝑒𝑟 − 0.223856 ∗ Petrol

− 0.103842\*Individual − 0.108115 ∗ Manual + 0.451977 ∗ luxury (2)

where 13.381813 is the intercept value, *age* and *max\_power* are the scaled variables and *Petrol*, *Individual*, *Manual*, and *luxury* are binary variables indicating categorical features of observation accordingly.

To calculate a predicted car price with some error, the linear predictor becomes the power of e:

(3)

which is equal to

(4)

(4) provides the following information about car price prediction.

One-unit change in (scaled) *age* changes car price by a factor of 0.043536, which is equal to 𝑒 − 3.134177, while other variables remain same (w.o.v.r.s).

One-unit change in (scaled) *max\_power* changes car price by a factor of 37.317423 w.o.v.r.s.

If *fuel* is *Petrol,* it changes car price by a factor of 0.799431 w.o.v.r.s.

If *seller\_type* is *Individual*, it changes car price by a factor of 0.901368 w.o.v.r.s.

If *transmission* is *Manual*, it changes car price by a factor of 0.897524 w.o.v.r.s.

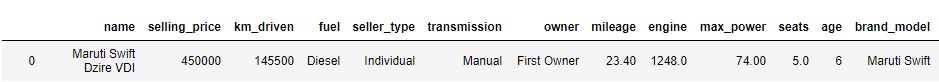
If *brand\_model* is in *luxury,* it changes car price by a factor of 1.571415 w.o.v.r.s.

If all variables are 0, then the predicted price is the intercept value 648,108.078, which is equal to

𝑒13.381813 .

# Example

# Table 8. An Observation in the Test Dataset



For the observation represented in Table 8, the inputs are *age*=6, *max\_power*=74, *Petrol*=0, *Individual*=1, *Manual*=1, and *luxury*=0 since its brand-model is not in the lux car brand-model list.

After *age* and *max\_power* values are multiplied with some scaling factor, all inputs are plugged in (3). This price prediction result gives 527,548.22. If brand-model of this observation was in the lux car list, the predicted price would be 828,997.36 which is equal to 527,548.22 \* 1.571415 (1.571415 is the luxury factor as stated above).