**Multiple Linear Regression**

SV42/2020 Danilo Babić, SV61/2020 Jelena Miković

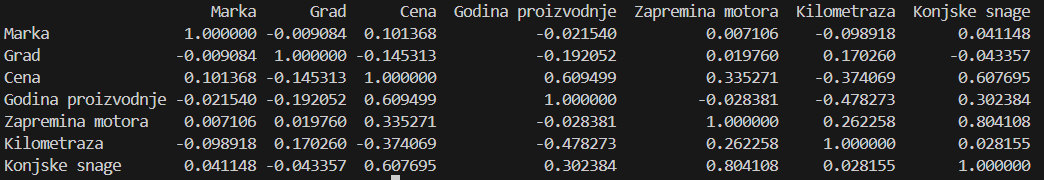
# Problem

The goal is to predict the price of used cars based on given attributes such as brand, city, year of production, body type, fuel type, engine displacement, mileage, horsepower, and transmission type.

# Data and Correlation matrix

The dataset used in this project consists of around 1200 used cars and their prices. Most of the cars were fairly priced, but there were a few outliers that needed to be removed for more accurate representation.

Bellow is the table that represent matrix of correlation of other attributes and how they affect the price. Most of the values were high and needed to be taken in considiration when calculating the price, but the city and brand did not stand out as much and when removing them the result was slightly better.



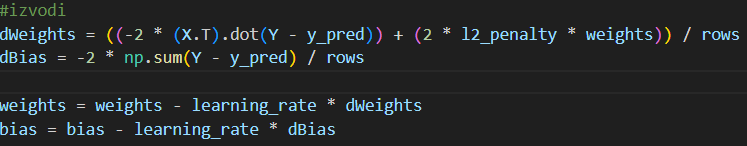
# Encoding

This project required usage of both one-hot encoding and label encoding. Since there are a huge number of cities and brands, we used label encoding for those to keep our dimensions to a minimum. We already knew what kind of fuels, body type and transmission type are possible so we used one-hot encoding, while making sure the test data also has all of the columns that are used. The number of columns ended up being 24.

# Ridge Regression

The primary objective of ridge regression is to minimize the sum of squared residuals (difference between actual and predicted values) while also penalizing the size of the coefficients (weights) associated with predictor variables.

Calculated on the formula below, in each iteration the weights and bias are adjusted. And based on final results the predictions are made.

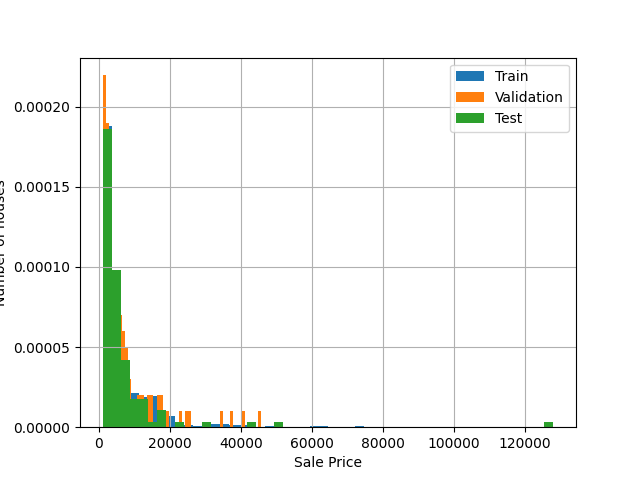


**Parameters that gave best result:**

* Learning Rate: 0.1
* Iterations: 1000
* Penalty: 1

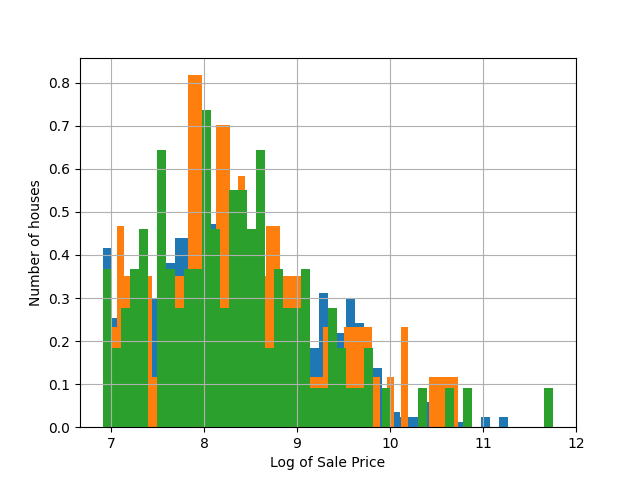
# Log Transformation and Standardization

On the left picture we can see that the data is skewed to the left, so by using the logarithmic transformations we get a much nicer distribution of data.



Bat

Bat



Additionaly, features with larger scales might have a disproportionate impact on the model's parameters. For example, if one feature has values in the thousands while another has values in the tens, the algorithm might assign a larger weight to the former, even if it's not necessarily more important. Both datasets are scaled based on the distribution of the training data.

mean = np.mean(X\_train, axis=0)

std = np.std(X\_train, axis=0)

X\_train\_scaled = (X\_train - mean) / std

X\_test\_scaled = (X\_test - mean) / std

# RMSE and Results Analysis

Root Mean Squared Error (RMSE) is a commonly used metric to evaluate the performance of regression models. It measures the average magnitude of the errors between the predicted and actual values, with lower values indicating better model performance.

The final score we got on the testing data was around 2000. And here we can see how the model predicted the prices for some of the examples.

# References

* <https://www.analyticsvidhya.com/blog/2020/03/one-hot-encoding-vs-label-encoding-using-scikit-learn/>
* <https://www.ibm.com/topics/ridge-regression>
* <https://kenbenoit.net/assets/courses/me104/logmodels2.pdf>
* <https://en.wikipedia.org/wiki/Root-mean-square_deviation>

Note: ChatGPT helped format some of the sentances above, we chose everything we wanted to say, he just made it sound better. ☺