OTOMOTO CAR PRICE PREDICTION

1.INTRODUCTION :

In any Data Science Environment, Time is of the essence. We need good results in a shorter span of time. I made my decisions based around this notion.

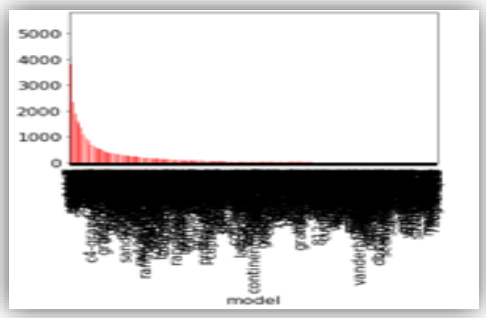
As I started off with the dataset, I looked first and foremost for the number of instances. Having, huge a dataset more than 200000 columns gave me an idea on how to proceed. I made a point to look for:

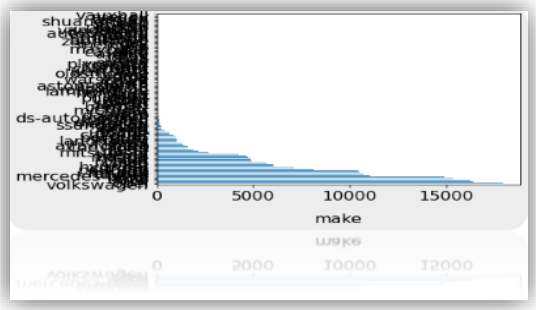
1. Null values
2. Look at categorical columns.
3. Look at continuous columns.
4. Formulate my own hypothesis before embarking on any modelling.

2.COLUMN DESCRIPTION:

Talking about the columns,

- The columns “Make”, and “Model” were not included for the simple purpose that they introduced a high level of complexity to the model and had no importance to the whole environment. Both the columns as can be observed are highly imbalanced and should play no part.





- "vehicle_year" column being an int datatype plays an important role in the whole setup as the year determines the price of the vehicle.

- Column's "mileage", "engine_power" and "engine_capacity" are all continuous variable columns and as such , they also played a important role in regression. Out of these 3, "engine_power" and "engine_capacity" were highly correlated to each and I proceeded to remove one ,i.e., "engine_capacity".

- "gearbox" and "fuel_type" columns were also string categorical and were converted to labels of int64 datatype.

- The columns "damaged" and "is_business" were already label encoded and talked about if the car is damaged and the person selling is an individual or a business.

- And finally, we come to the Target variable that is called the "Target_price" column.

3.INITIAL THOUGTS :

- After performing some exploratory analysis, for "damaged" column, a lot of vehicles that came in were not damaged as compared to the damaged ones. Model would take this into account and possibly predict the not damaged types much better than the damaged ones. This can be a problem.

- Although not by much, in "is_business" column also, a lot of companies were readily selling the cars instead of individuals. Model could predict one better than the other.

- Same is the case with "fuel_type" column where, Petrol and Diesel types occupied more than 70% of all the rows telling again the same story of the dataset being imbalanced as such.

- The situation was much worse for the "gearbox" column as approximately 90% of the columns were filled with "Automatic" and "Manual" transmission types.

4.OUTLIERS:

- After performing EDA, it was time to treat the outliers. "mileage" column was the most affected one out of all of them as could be seen by the graph. "engine_power" and "engine_capacity" also did not lag behind much in this aspect. Removing "engine_capacity" solved one part of the problem.

- But, owing to the already imbalanced columns in the dataset, I went against the notion of replacing the outliers with log or median values as it predict on an artificial data as compared to the real one.

- We ofcourse cannot use mean to replace the outliers as they play a huge role in columns with outliers.

- The dataset is already, so losing on some outliers would not affect the accuracy so much as to when we replace them with artificial values.

5.FEATURE SCALING :

- After comparing the results with both MinMax Scalar and Standard Scalar, I could observe that not much difference in accuracy was obtained and as such I decided to go with Standard Scaler as it gave an added benefit of having a normal distribution.

- Most people would decide against scaling the target variables too, but for me, when it comes to regression columns, I emphasis on having all continuous columns scaled as it helps in the computation time and makes the model learn a bit quicker. We can always use inverse_transform to make it come back to its original values.

6.K-FOLD CROSS VALIDATION :

- While performing 10-Fold Cross Validation, a best score of 0.29 was guaranteed, and that was not at all a good score, somehow after doing repeated adjustments, the same score was being obtained, So I forego the task completely to focus on Train and Test split completely as we had enough data to back on.

- Ideally, I would always use Cross Validation techniques for samples which does not have much data and needs to be used judiciously.

7.CHOOSING MODEL:

A. **Support Vector Regression** :

- Due to higher number of columns and rows, Linear Regression in my experience would not work much over here, So decided against employing Linear Regression altogether and instead started of with Support Vector Regression.

- As rightly thought, even SVR was not able to give satisfactory results with a very low r2 score of just 0.39. No Data Scientist would ever in their right minds would believe the work is done here.

B. **Convolutional Neural Network:**

- I decided on increasing the complexity of the model and went for Deep learning approach. I made sure that while looking for hyper parameter tuning, not a lot of changes were made as I wanted to have all my models based on the standard structure to have a common ground for me to compare them.

- A single Cov1D layer and Flatten layer along with 2 Dense layers were utilized over here for this purpose.

- And surely, I got a very improved r2 score of 0.81. Although not enough, the signs were looking promising now.

C. **Polynomial Regression** :

- Polynomial Regression with a degree 4 increases the complexity all the more and owing to this we got a good r2 score of 0.85.

- Increasing the degree would also increase the accuracy but only up until a certain point after which the model would start throwing all sorts of inaccurate results. 4

D. **Random Forest Regression** :

- Saving best for the last, Employing Random Forest Regression gave excellent benefits and we finally a very competitive r2 score of 0.87.

- Looking at the computation time, I chose Random Forest Regression as the preferred model of choice of such a dataset.

E. **XG Boost Regression** :

- Although finally a satisfiable result was achieved, I still did not understand how good the Random Regressor Model could be.

- The only way was to stack up against another power model and thus used XG Boost.

- After increasing the n_estimator parameter to 100, only was I able to replicate the results given by Random Forest.

- This gave me a pretty good idea as the number of trees in Random Forest was only kept as 10.

8.CLOSING THOUGHTS:

- While the Random Forest Regression was able to create good accuracy, preferably it would have been better if my Line Manager or my Head would be able to say how strictly adhered should my accuracy be.

- With regression, it is very difficult to get that exact continuous value and as such a threshold of +1000 or -1000 should be talked about so that it helps to better gauge the model.

- Imbalance was a huge issue and if the dataset would have been cleaner, it could have made the process smooth.

- Data Augmentation could have been a viable option but that would have created much more artificial values.

9.FUTURE WORK :

- First and foremost, I will add in total 2 more columns: “Distance_Travelled” and “Type” Column The first column talks about the total distance travelled by a particular car during its lifetime. It is a huge factor while deciding any price and should be considered. The second column is the type of car, Is it family oriented, bachelor oriented, a sports car. Something of that sort.

- Next, I would like to make changes to the “model” and “make” column. Introducing a categorical variation to these columns would have helped factoring in these columns also to better understand the model.

- With increase in the number of columns, we should also increase the number of rows complement it. Thankfully, due to less number of NA values, that did not impact the accuracy much.

- This would seem a bit controversial, but instead of having regression we could convert this into a categorical problem itself by deciding the range of values across row. For eg:- 10000-20000 Zlotys would be 0, 20000-30000 Zlotys would 1, and so on.

- Treat the Imbalance in the “is_business” column and the “model” column. This has affected the r2 score for sure and as such even a score of more than 0.9 just does not seem right as it would predict one category better than the other.