Python Data Science

...

Project work - Machine Learning VS Neural Network

Saša Arsić

Novembre 2020

PROJECT DESCRIPTION:

First step:

Research the best model that can be used in car price prediction. Also, exploring the best parameters of chosen models, and applying of models on data set with best parameters.

Comparing results of model trough R2 score.

Second step:

Building NN for Tabular problem, and estimation of car price prediction by NN.

Third step:

Conclusion with results from ML models and DL, Fastai tabular.

First step

- Getting data set for regression problem (Car Price)
- Data Overview
- Exploratory of data
- Preprocessing
- Building models for our prediction
- Predicting the price of car trough 10 different machine models, and comparing their R score.
- First on raw data, then on std raw data and finally on norm raw data.
- Secondly repeat the same procedure on transformed data, transformation is based on feature engineering.
- Comparing R Scores of models.
- Trying to improve our best models with Grid search, Random search and hyperparameter tuning.
- Let’ see what will say Feature importance

First impression:

After our visual observation, we could see that some features have small or any affect on our dependent variable, we will first include all variables, later we will reduce that number and observe how it affects our models.

We may say that we're dealing with regression problem, but our dataset consist more categorical than numerical variables.

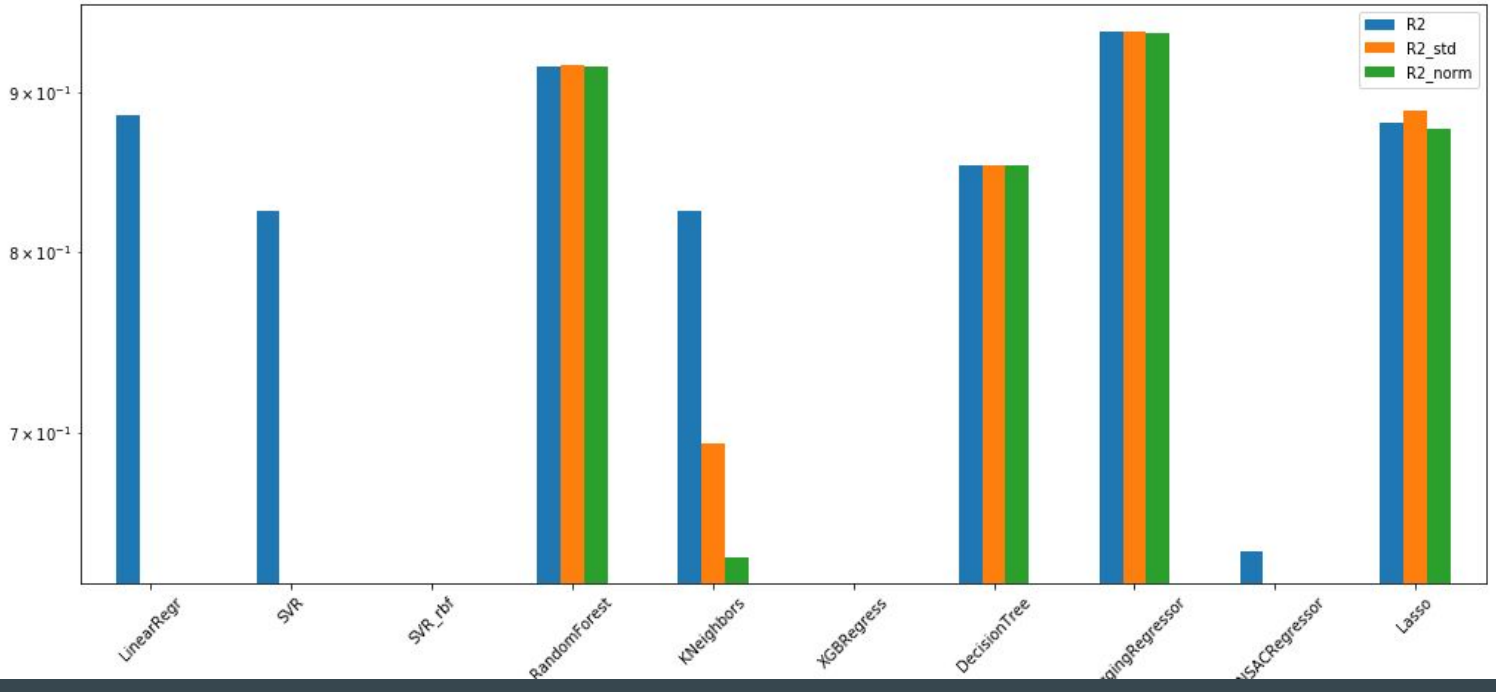
The linear dependence is in most cases represented between numerical variables and the dependent variable.

Will 205 samples be enough to train and predict the price of a car?

It remains to be seen.

First try on raw data:

We first took the raw data without any processing other than necessary for further work (nan, dummi), we passed them through our regression models and obtained the following results.



We can conclude that Linear Regression has a good score only over raw data, as well as Svr_linear, while Svr_rbf and Xgboost proved to be so bad for this type of prediction.

The best score has Bagging regressor.

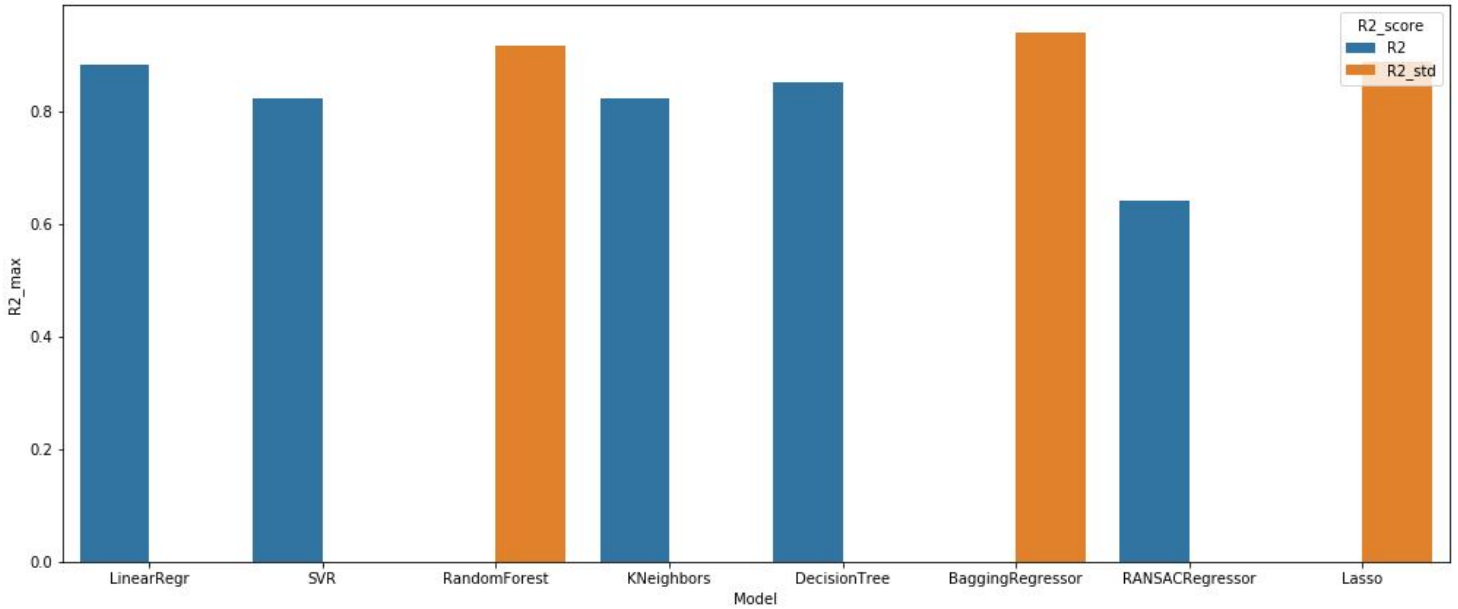
Best pass through all types of data are Tree-based regression models.

They behave most stably with each data transformation.

Interestingly, the KNN is declining gradually.

* the results of the models that are not presented on the graph are negative or their values deviate too much, so we left them out of the visual presentation.

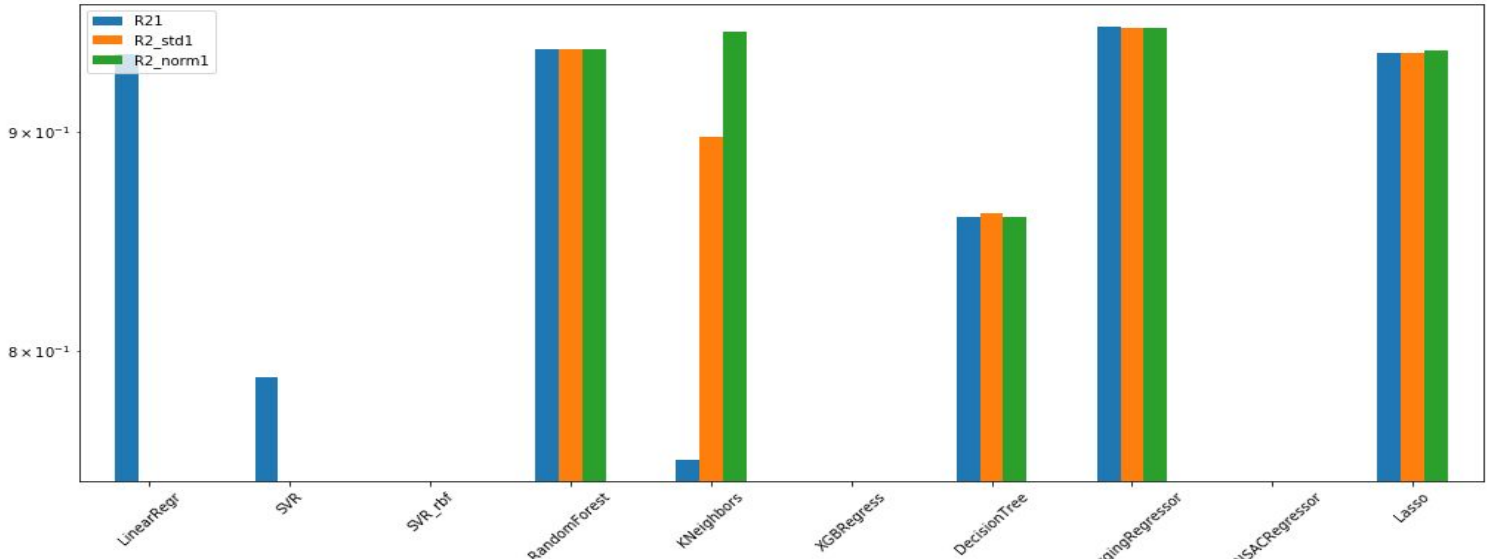
By comparing R scores, we came to the conclusion that the models reacted best to data without standardization and normalization, the biggest R2 error is with std data (bagging regressor) but untransformed data have a wider application. Normalization did not make any progress, nor did it fit into our Rmax score.



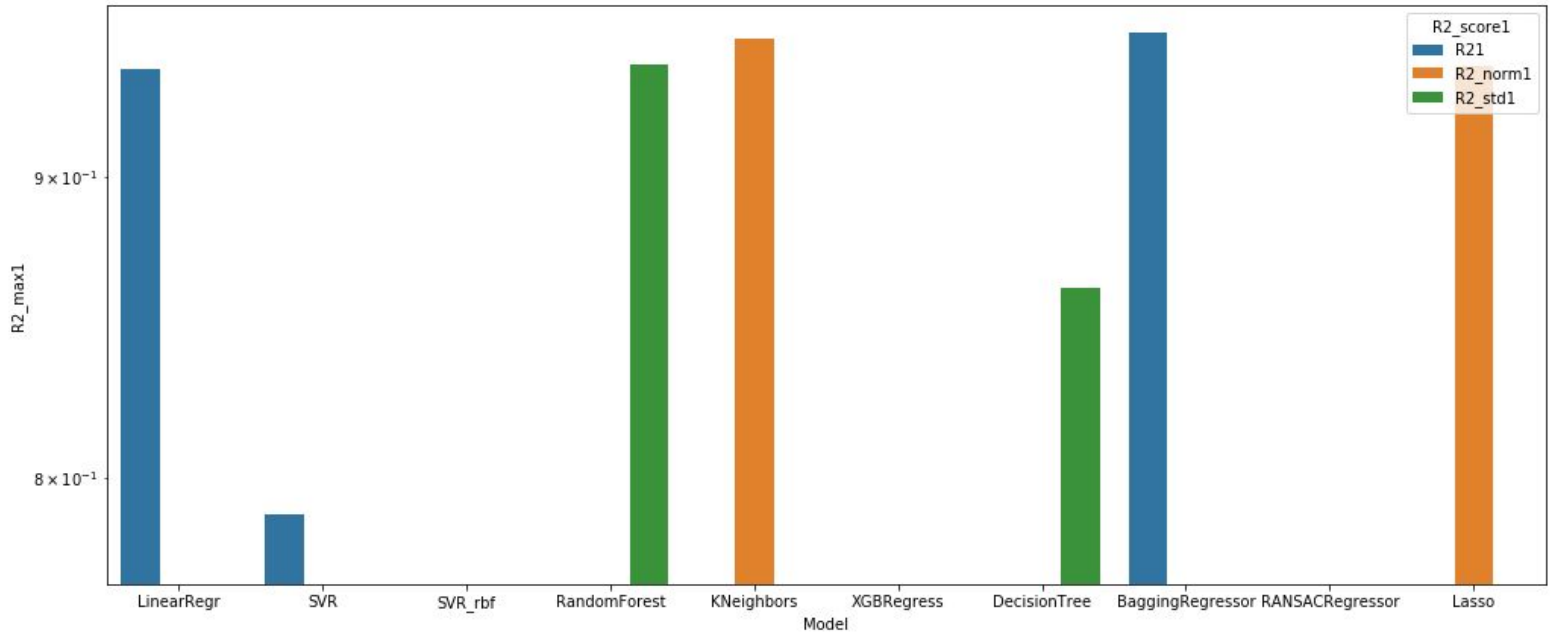
Now transformed data set

The processing of the raw data set is complete: Let's try to improve the results by combining and selecting the most important features according to the visualization. We made a new feature ‘avgmpg’ by a combination of two features ‘citympg’ and ‘highwaympg’, which both has negative correlation with our dependent variable ‘’Price. Then we divided the feature ‘Carbrand’ into 4 classes, in relation to the price. We will run this transformed data set again through the models and follow the R score.

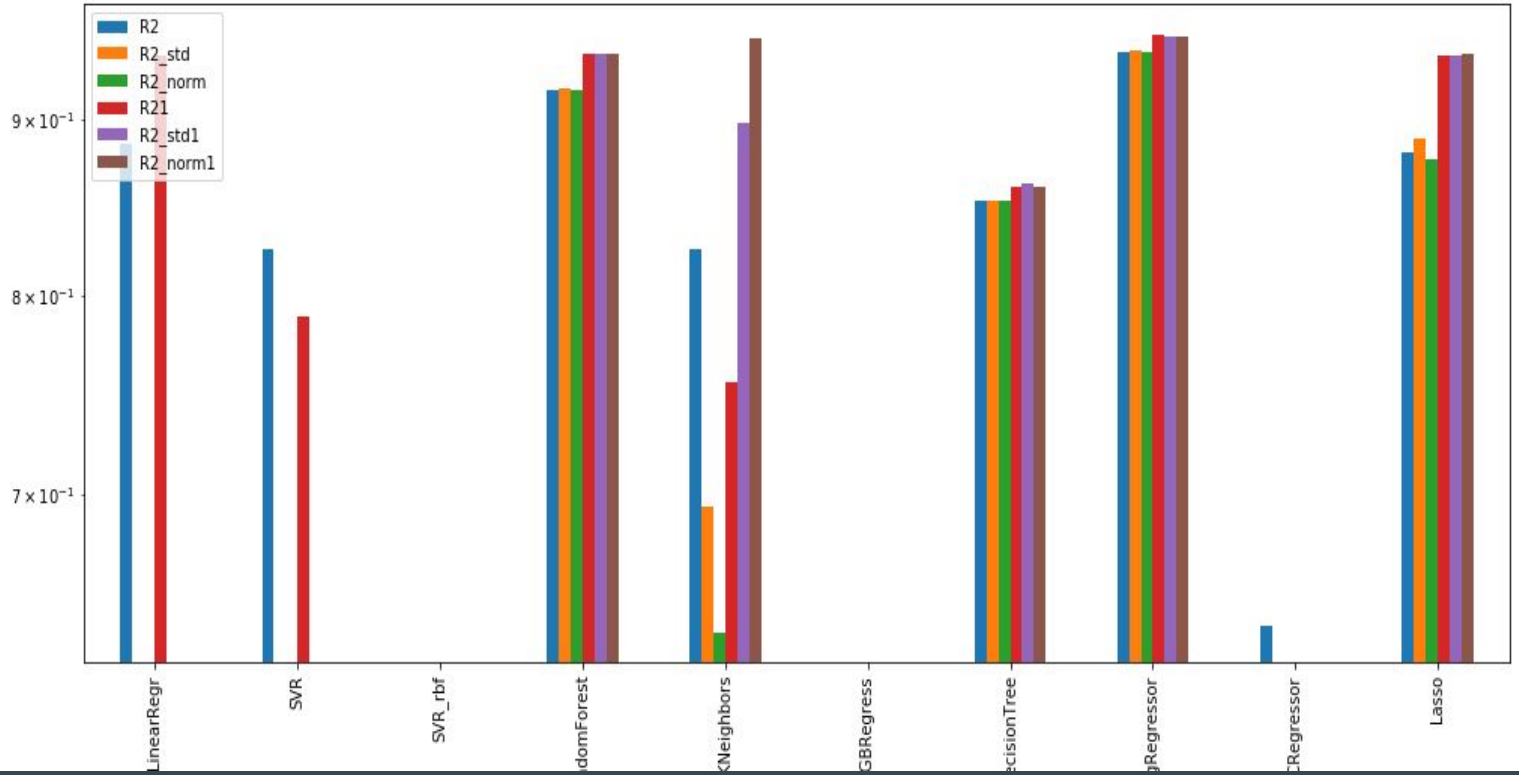
Let's see how our R2 score reacted on transformed data set.



There was a change, all models reacted on data transformation, for some the result increased, for some it dropped. Maximum R2 increased to 95 with Bagging Regressor. Models that had a bad performance during std and normalization, still have a bad effect here.



The biggest change happened with the KNN model through norm data, Analyzing the R max score, the KNN model now climbs to the second place, after the Bagging Regressor.



Now that we have the two best models, we tried to improve them with Grid search, Random search, changing parameters, but we did not get better results with neither Bagging nor KNN Model.

When we run our dataset through the fastai tabular, through two minutes, without any transformations,without comparing the scores, deciding which model should or should not be used we get following results.

The Metrics used In Evaluating The Network:

The calculated RMSE & R-Squared For The Training Set : [tensor(0.0578), tensor(0.9880)]

The calculated RMSE & R-Squared For The Validation Set : [tensor(0.1533), tensor(0.7603)]

Was all our work on ML in vain?

I would not say, we have learned how which model reacts to this type of problem and next time we narrow down the choice of models we want to use.
Since our first results were very good, we could not expect any major improvements, what's more, as we got worse results through Grid search, I think we have a case of overfitting, which was confirmed by the difference between the error of the validation and training set in NN.
We managed to get into the soul of our data set, to get to know it to the fullest. and I think that's important.

While ML models do become progressively better, at whatever their function is, they still need some guidance.
Whit deep learning model, an algorithm can determine on his own if a prediction is accurate or not trough its own neural network.

For me, someone who has met this science field for first time, NN although with greater possibilities and more complex processes within themselves, they appear simpler to use at first glance.