Capstone Project

Lucas Saffi

Machine Learning Engineer Nanodegree

April 14th, 2019

Definition

Project Overview

Pricing the most appropriate value for a product is a challenge for any company, because of the many market variables and its costs. Depending on the market this challenge can be even bigger because of the complexity, and certainly, the automotive segment is one of them.

Although technologies of vehicle sharing like Uber and Didi are becoming more common, the number of units of vehicles sold has grown continuously. According to Technavio, until 2021 the CAGR of used vehicles in the world will be 7%, reaching the number of 128 million of units. Considering this information and market competition, only prepared dealerships will excel and thrive. Therefore, it is crucial for them understand the market behavior to make better decisions, specially about price, which is fundamental to guarantee market volume and a good margin.

According to Listiani, in the research paper "Support Vector Regression Analysis for Price Prediction in a Car Leasing", the used car segment in Germany in 2004 generated 52 billion revenue and yields just 0.03 billion (1%), although it had a profit potential of 3%. In the past, it was hard to have a hardware good enough to compute fast and enable machine learning applications to effectively predict which price should be applied to maximize result, but today it is easy.

Vehicle pricing depends on factors like car model, brand, mileage, fuel type, power and time. Pricing correctly can help a dealership sell faster, with less cost, and with a more predictable margin. So, predict more accurately the most likely price of a car in market is certainly a solution with big impact for dealerships. Another reason for my interest about this project is that I am currently working in a dealership network, having knowledge of this can differentiate me to propose projects to achieve better and faster results for used cars.

The dataset was offered on Kaggle with the name "Used cars database" and generated by Orges Leka, which collected 370.000 samples from ebay. It has the following information: date that the ads was created and last seen on ebay, mileage, type of seller, year of registration, offer type, gearbox, power, price, postal code, if vehicle is from control or test group (AB test) and if it was repaired or not.

The link to access the dataset is: https://www.kaggle.com/orgesleka/used-cars-database

Problem Statement

The problem to be solved is the price prediction of used automotive vehicles, which is a regression problem. Therefore, the model has the objective to determine the price a car will have based on past car ads samples. For example, if a car is a Volkswagen, model Golf, manual, 128 HP, regular gasoline and is going to be sold during a certain period, it is likely that will cost € 10.000,00, that is what de model should predict.

In order to solve this problem, regression models such as Support Vector Machines with different kernels ("rbf", "linear", "poly") and a Random Forest Regressor will be developed and, the best model will be tuned and compared to the benchmark, which is a Stochastic Gradient Descent algorithm. Therefore, each model will be fitted to the training samples from dataset and, after determining the best predictor, the one selected will have its parameters tuned with GridSearchCV, in order to achieve an even better result. This solution is a good option, because can evaluate algorithms with different properties, advantages and performance. It enables us to better understand the complexity of the problem and choose the one which better fits the project.

Metrics

The evaluation metrics used will be the score of each model, which is the R^2 of the prediction and the range value is from 0 to 1. This metric was chosen because measures how well the dependent variables explain the independent variable variance. So, it will measure how well future samples are likely to be predicted by the model. This metric is composed by the correct value price, prediction, mean and number of samples. The equation of this metric is:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n_{\text{samples}}-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{\text{samples}}-1}(y_i - \bar{y})^2}$$

Analysis

Data Exploration

As mentioned above the data was taken from Kaggle and have in total 20 features of more than 370.000 used vehicles sold in Germany:

- dateCrawled: when the advertising was first crawled. All field-values are taken from this date
- name: "name" of the car
- seller: private or dealer
- offerType
- price: the price on the advertising to sell the car
- abtest
- vehicleType
- yearOfRegistration: at which year the car was first registered
- gearbox
- powerPS: power of the car in PS
- model
- kilometer: how many kilometers the car has driven

- monthOfRegistration: at which month the car was first registered
- fuelType
- brand
- notRepairedDamage: if the car has a damage which is not repaired yet
- dateCreated: the date for which the ad at ebay was created
- nrOfPictures: number of pictures in the advertising. As this field only contains the value 0, it is not going to be used.
- postalCode
- lastSeenOnline: when the crawler saw this ad last online

This data needs to be well preprocessed, because of values wrongly filling some cells and the presence of outliers which are, in some cases, certainly a mistake due to its discrepancy to the mean. The table below present general numerical information about the dataset which easily present some weird values like maximum and minimum price:

Information	price	powerPS	kilometer
count	3.692050e+05	369205.0	369205.0
mean	1.730462e+04	115.3	125713.05
std	3.599177e+06	192.5	40044.9
min	0.000000e+00	0.0	5000.0
25%	1.150000e+03	69.0	125000.0
50%	2.950000e+03	105.0	150000.0
75%	7.100000e+03	150.0	150000.0
max	2.147484e+09	20000.0	150000.0

Besides that, information like the date that the advertising was created and last seen can be subtracted to create another feature, which would be the time this used car was available. So, using two start features it can be created another much more useful for analysis.

Therefore, for analyzing data, it will be necessary to remove values that do not make sense to this project, decrease outliers' effects, normalize data, since the absolute values of features are different and create other features from the previous on the dataset.

In the table below are 5 samples of the dataset:

dateCrawled	name	seller	offerType	price	abtest	vehicleType	yearOfRegistration	gearbox	powerPS	model	kilometer	monthOfRegistration	fuelType	brand	notRepairedDamage	dateCreated	nrOfPictures	postalCode	lastSeen
24/03/2016 11:52	Golf_3_1.6	privat	Angebot	480	test		1993	manuell	0	golf	150000	0	benzin	volkswagen		24/03/2016 00:00	0	70435	07/04/2016 03:16
24/03/2016 10:58	A5_Sportback_2.7_Tdi	privat	Angebot	18300	test	coupe	2011	manuell	190		125000	5	diesel	audi	ja	24/03/2016 00:00	0	66954	07/04/2016 01:46
14/03/2016 12:52	Jeep_Grand_Cherokee_"Overland"	privat	Angebot	9800	test	suv	2004	automatik	163	grand	125000	8	diesel	jeep		14/03/2016 00:00	0	90480	05/04/2016 12:47
17/03/2016 16:54	GOLF_4_1_4__3TÜRER	privat	Angebot	1500	test	kleinwagen	2001	manuell	75	golf	150000	6	benzin	volkswagen	nein	17/03/2016 00:00	0	91074	17/03/2016 17:40
31/03/2016 17:25	Skoda_Fabia_1.4_TDI_PD_Classic	privat	Angebot	3600	test	kleinwagen	2008	manuell	69	fabia	90000	7	diesel	skoda	nein	31/03/2016 00:00	0	60437	06/04/2016 10:17

Exploratory Visualization

In this dataset are present numerical values, which can be directly related to price, such as power, mileage (kilometer) and time online. So, to better understand each feature, including the price, a histogram of each of them were generated:

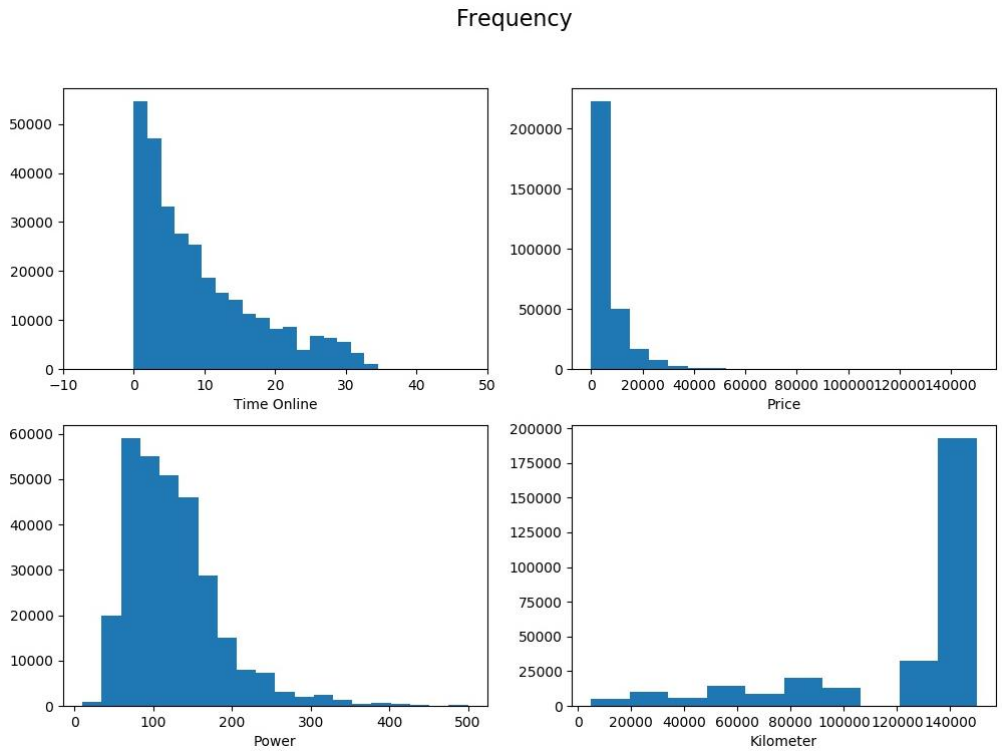

Figure 1 - Frequency of the main numerical features of the used cars dataset

Before creating the charts above, significant outliers were removed from dataset in order to visualize a more real situation.

It is possible to perceive the most common values of the majority of the used cars in relation to time online, price, mileage and power. Time online and mileage histograms show that the major part of the cars have their advertising between 0 to 10 days online and 120.000 to 150.000 kilometers. The majority vehicles have power between 50 HP and 200 HP and price below € 20.000,00.

Algorithms and Techniques

The regressors chosen to predict price values are a Support Vector Regressor (SVR) and an ensemble model, Random Forest Regressor. The SVR model was chosen because of the different kernels (linear, polynomial and rbf) which can be applied, so it is possible to verify if some of them excels predicting. Random Forest Regressor was considered a good option because uses more than one decision tree for prediction to calculate the average result of them, that way, the accuracy is likely to be higher and helps against over-fitting.

Support Vector Regressor tries to define a hyperplane and boundary lines according to the kernel chosen that better fit data considering a certain threshold. The image below from “medium.com” website illustrates how it works:

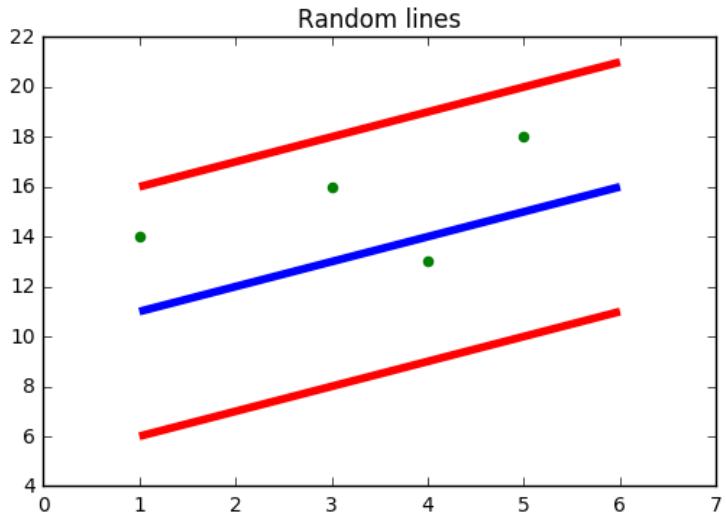

Figure 2 - Examples of the Support Vector Machine functionality.

The blue line on image above is the hyperplane and the red lines are the boundaries. The red lines are the margin of tolerance, so only data that are between those lines will be considered to fitting model.

Random Forest Regressor, as mentioned previously, uses multiple decision trees for prediction, but it is called “Random” because of the random split of data and features between the decision trees and their nodes. The figure below, which is also from “medium.com”, exemplifies it:

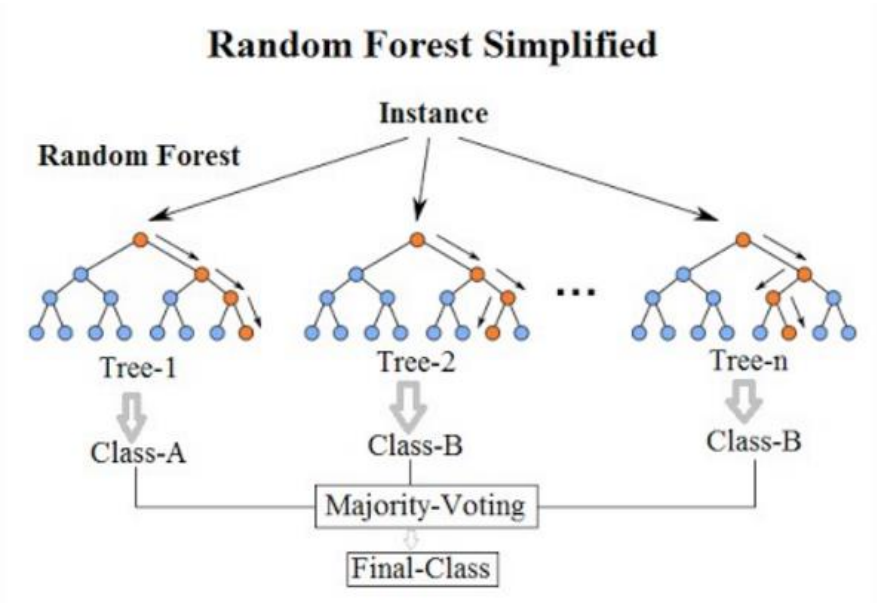

Figure 3 - Exemple of how the random forest model works

Benchmark

The model chosen to be the benchmark is the Stochastic Gradient Descent Regressor (SGD Regressor). It was selected because of its great performance for big datasets and to support different loss functions and penalties to fit linear regression models. In this project the default parameters, including loss function and penalty will be used to fit data, which are the “squared_loss” and “l2” respectively. The score obtained with this model will be considered the benchmark to the Support Vector Regression and Random Forest models.

Methodology

Data Preprocessing

The preprocessing part was composed by the following steps:

- Correct words misspelled;
- Remove samples that were not correctly read to read_csv function;
- Remove time from date features;
- Change data type of those features which were not correctly identified by read_csv function;
- Remove duplicate rows and outliers;
- Create “time_online” feature using “lastSeen” and “dateCreated” columns;
- Remove unnecessary columns such as: “name”, ”nrOfPictures”, ”postalCode”, ”dateCrawled”, ”dateCreated”, “lastSeen”;
- Apply log transformation for skewed features;
- Normalize numerical data for better comparison between features;
- Transform categorical features into numbers.

After executing the steps above the data is ready to be processed by the models.

Implementation

After preprocessing data, it was possible to implement the SVG, SVR and Random Forest models. The first step is to split data between train and test and separate the dependent and independent variables (X_train, y_train, X_test, y_test). The rate of test chosen was of 20% and the final absolute values of training and test sets was of 249.322 samples and 62.331 respectively. After that, models with default parameters were created, fitted data and predicted the X_test samples. During this step it was possible to notice the difference of speed to fit and predict depending on the model. It was not possible to use all data with SVR algorithms due to the processor’s limitations of the computer used to develop this project. If all data were used it would take too long for training. Because of that, it was chosen different quantities of samples to train models: 1.000, 10.000 and 70.000.

The R^2 score, as mentioned before, represents how well future samples are likely to be predicted by the model, so the algorithm with the higher score, which was the Random

Forest Regressor, was chosen to be boosted defining better parameters using GridSearchCV function from Sklearn library. Besides the higher R^2 score, this model was also the one that performed the process faster as show the charts below:

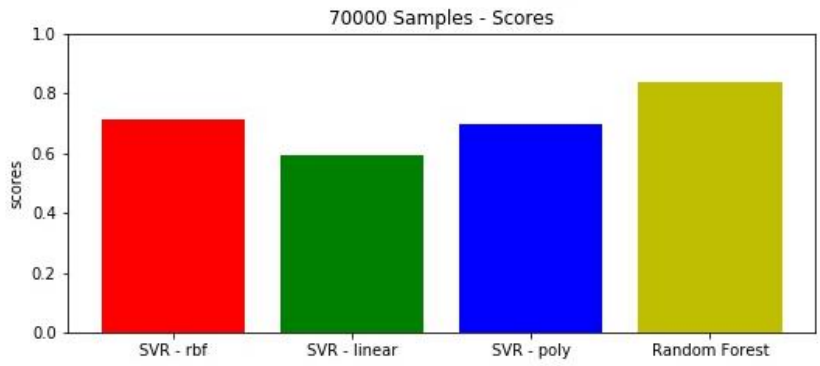

Figure 4 - Score values of all the models

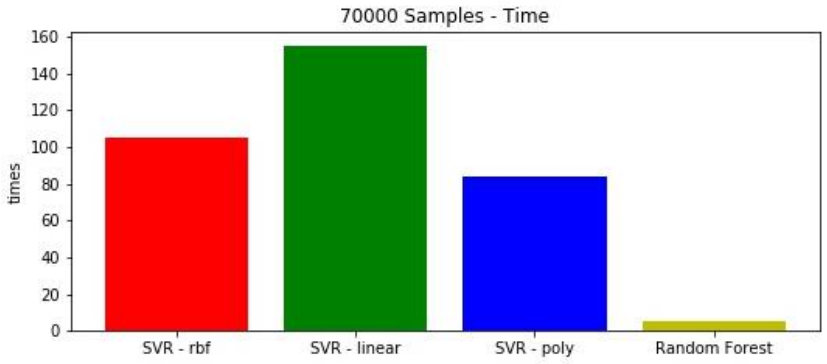

Figure 5 - Time values of all models

Refinement

After fit and determine which model has the highest R^2 score, it is possible to find better parameters to reach an even higher score in price prediction. So, since the Random Forest Regressor got the best result in the implementation step, GridSearchCV function was used with the following parameters represented in the code line below:

```
parameters = {'n_estimators':[100,200],'criterion':['mse','mae'],'min_samples_leaf':[2,5],'min_samples_split':[2,4]}
```

Therefore, four parameters were chosen to search for a better model performance. The "n_estimators" is the number of trees in the forest, "criterion" the function to measure the quality of split, "min_samples_leaf" which is the minimum number of samples required to be at a leaf node and "min_samples_split", which is the minimum number of samples required to split an internal node.

Due to hardware limitations it was necessary to use just 3.000 data samples to determine the best model parameters with GridSearchCV. The model with the best parameters determined was:

```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
        max_features='auto', max_leaf_nodes=None,
        min_impurity_decrease=0.0, min_impurity_split=None,
        min_samples_leaf=2, min_samples_split=2,
        min_weight_fraction_leaf=0.0, n_estimators=200, n_jobs=None,
        oob_score=False, random_state=None, verbose=0, warm_start=False)
```

Since in the implementation step the maximum number of samples used to fit models due to time consuming of SVR algorithms were 70.000, the next step, which had the goal to compare R^2 score of Random Forest Regressor and SVG model score, used all the 249.322 samples.

Results

Model Evaluation and Validation

The model being developed to beat benchmark performance, the Random Forest Regressor, had its parameters defined in the previous step. Besides the higher score, the random forest was chosen due to the advantage of avoiding overfitting using multiple decision trees and the excellent performance when working with a big dataset. That is why the number of estimators is so important as a parameter and also the minimum samples leafs and split, they are directly related to how well the model can predict prices without overfitting.

In order to evaluate the model, it was used the cross validation function from sklearn library. The function divided the test samples in 4 parts and for each of them were determined the score the model would achieve in prediction. That way, it was possible to understand how well the model deals with small perturbations in data.

Split	**Score**
1	0.845
2	0.844
3	0.843
4	0.848

Based on values obtained for each of the 4 parts from split, it is possible to notice that the variation between them are really small, which indicates that the model is very robust.

Justification

Using all training set samples to fit models, the following results were obtained:

Models	Scores	Time (seconds)
Stochastic Gradient Descent (BenchMark)	0.592	0.954
Random Forest Regressor	0.866	329.81

Despite the great performance in relation to time of the benchmark, the random forest regressor achieved a considerable higher score. The difference is of 0.274 in absolute values. The R^2 of 0.866 can certainly help to better price used cars, but it did not achieve a value which it is possible to consider only model's prediction as the final result. Other features maybe relevant to achieve an even more accurate regressor.

Conclusion

Free-form Visualization

After predicting values with benchmark model and Random Forest Regressor, it was created charts, which could help us to better understand and visualize the performance in price prediction of each of them:

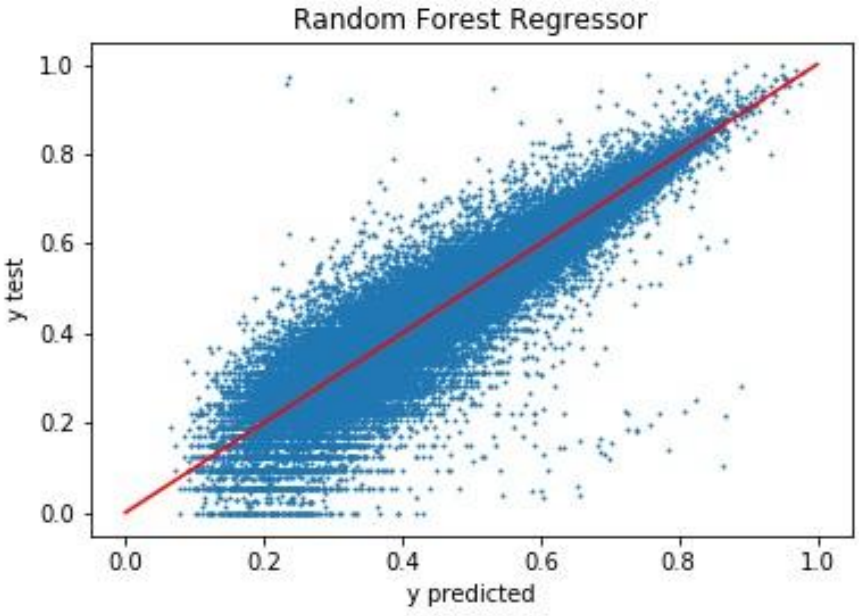

Figure 6 - Comparison of y predicted by Random Forest Regressor and true test values

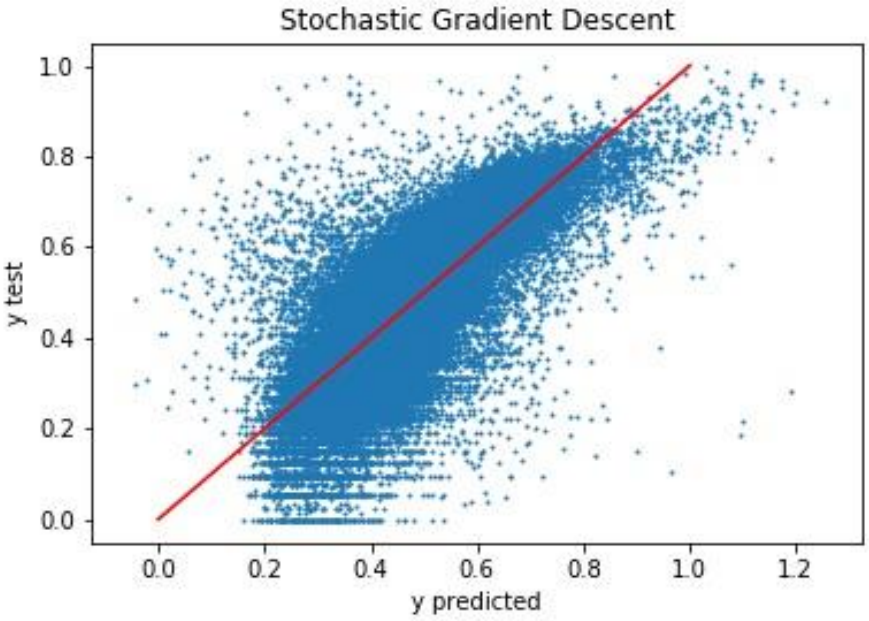

Figure 7 - Comparison of y predicted by Benchmark and true test values

Each blue point on the charts above has the X axis values as the y predicted and the Y axis as the true y values from the test set. The red line represents where the points should be located if the y predicted were equal to true y. Analyzing both charts, it is possible to visualize that data are more dispersed in the second figure, which refers to the benchmark model, justifying the lower R^2 score.

Reflection

The process to develop this project had the following steps:

1. Preprocessing data: Error values and data transformation was necessary to prepare data for analysis and to train models.
2. Data visualization: Visualizations were developed to better understand the distribution of the dataset in relation to price, time online, power and mileage.
3. Models training and prediction: Models were created with default parameters and compared to determine which of them was worth more to tune parameters.
4. Model tuning: It was searched better parameters to the Random Forest Regressor achieve better results in price prediction.
5. Comparisons and conclusions: Final scores and performance of the Stochastic Gradient Descent model (benchmark) were compared to the tuned Random Forest Regressor model.

The most difficult step was the preprocessing and model training and prediction. Both steps are directly related, because the way the preprocessing step is developed influence directly model's score. The first approach attempted was the transformation of categorical values into columns, which resulted in low score values. But, after change the approach, transforming the categorical values into numbers and including this

numbers into the process of log transformation and normalization, the results got much better for all models.

Improvement

Due to hardware limitations all dataset could not be used to fit models, since after trying to and let the program process for a long time, the program could not retrieve any result. In order to surpass this limitation, two possibilities could be done: run the process on GPU or in the cloud.

Another attempt to develop a better model for price prediction is one based on deep neural networks, which has been used in different fields and in both categorical and regression problems. In order to develop one, it would be necessary to test different architectures and define the best optimizer to achieve good results.

References

Leka, Orges. “Used cars database”. www.kaggle.com/orgesleka/used-cars-database. Acessed 01 Apr.2019

Koehrsen, Will.”An Implementation and Explanation of the Random Forest in Python”, www.towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76. Acessed 5 Apr.2019

Bhattacharyya, Indresh.” Support Vector Regression Or SVR”. www.medium.com/coinmonks/support-vector-regression-or-svr-8eb3acf6d0ff. Acessed 13 Apr. 2019

Koehrsen, Will.” Random Forest Simple Explanation”, https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d. Acessed 13 Apr. 2019

Scikitlearn.”RandomForestRegressor”.www.scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html. Acessed 5 Apr.2019

Scikitlearn.”Sthocastic Gradient Descent”. www.scikitlearn.org/stable/modules/sgd.html#regression. Acessed 5 Apr.2019