Journal of Modern Technology and Engineering
Vol.5, No.2, 2020, pp.157-166

ESTIMATION OF THE SECOND HAND CAR PRICES FROM DATA EXTRACTED VIA WEB SCRAPING TECHNIQUES

Resmiye Nasiboglu[1]*, Adem Akdogan[2]

[1]Department of Computer Science, Dokuz Eylul University, Izmir, Turkey
[2]The Graduate School of Natural and Applied Sciences, Dokuz Eylul University, Izmir, Turkey

Abstract. In the countries with high car prices, the used car market attracts a lot of attention. Accurate determination of vehicle prices on behalf of the buyer and seller is important in terms of selling the vehicle in a shorter time and therefore keeping the market alive. However, there are difficulties in determining the right price due to the rapid change of the market. In order to prevent this problem, the idea of an artificial intelligence-supported price estimation mechanism has emerged, which has not been so wide used, especially in the used car industry. This system, which is the main subject of this study, creates an optimal decision support structure for both vehicle buyer and vehicle dealer. In order to keep the changing prices up-to-date, it is important that the current information is automatically drawn from online car sales sites. For this purpose, web scraping technique, especially Beautiful Soup and Selenium libraries were used in our study. 100,000 vehicle data were excavated during the training. There are 12 attributes in each data. Several algorithms, such as Linear Regression, Ridge, Lasso, Elastic Net, KNN, Random Forest, XGBoost and Gradient Boosting Machine were used. As a result of optimization processes, the best model is chosen for each vehicle type. According to the chosen best model results, Gradient Boosting Machine - 19, Extreme Gradient Boosting (XGBoost) - 11, Random Forest - 7, Ridge - 5, Lasso - 2 and Elastic Net - 1 have performed as the best models.

Keywords: Machine learning, web scraping, prediction model.

Corresponding author: Resmiye Nasiboglu, Department of Computer Science, Dokuz Eylul University, Izmir, Turkey, e-mail: *resmiye.nasiboglu@deu.edu.tr*

Received: 20 February 2020; Accepted: 15 July 2020; Published: 30 August 2020.

1 Introduction

The used car market attracts a lot of attention in the countries with high car prices. It is important to determine the optimal prices for buyers and sellers. The inexperienced sellers often cannot determine the ideal price for their vehicles. For this reason, sellers have to spend the most of their time researching the used car industry in order to avoid harm. The same process also happens for people who want to buy a car. The buyer, who wants to get the desired vehicle features at the optimum price, makes an intense market analysis. Due to volatile in the sector, the buyer or seller may remain unstable as a result of this intensive work. The development of an artificial intelligence supported prediction mechanism can solve these problems. Especially non-experts can determine the optimum price using this estimation model. This structure can also work as a decision support mechanism for experts. Machine learning models are widely used as predictive mechanisms. However, a lot of data is required in order for machine learning models to work effectively. This data is more than enough on online car sales sites. This data must be obtained from these sites. The manual operation of this process is unrealistic in terms of workload. For this reason, web scraping methods can be used that enable automatic extraction of data from websites.

The data used were obtained by web scraping techniques in this work. The Beautiful Soup and Selenium libraries were used in this process. First of all, the data obtained were edited and converted into training and test data. After that, trainings were carried out using models such as Linear Regression, Ridge, Lasso, Elastic Net, KNN, Random Forest, XGBoost and Gradient Boosting Machine. The whole data set is classified according to vehicle brands. The special training models have been created for each brand.

The rest of the article is as follows. An information about used web scraping techniques, data preparation techniques, and prediction models are given in chapter 2. Training results and comments are given in chapter 3. Finally the last chapter, includes conclusion and general evaluation of the work.

2 Material and Method

2.1 Web Scraping

There are developments in artificial intelligence technologies owing to the hardware strengthening of computers. The amount of data required for artificial intelligence is increasing day by day. The Internet is the largest resource used to access this data. It is necessary to obtain this data quickly, structurally and systematically. However, it is a very costly process to handle these works with manpower. For this reason, web scraping techniques are used to overcome these problems.

The web scraping is basically obtaining information autonomously using software from the internet (Figure 1). The libraries are used for web scraping such as Storm Crawler, Jauntium, Jaunt, Scrapy, Norconex, Apify, Colly, Selenium, Beautiful Soup and Grablab. Selenium and Beautiful Soup libraries were used in this work.

HTML website
Web scraping
Extracted data
SQL
XLS
CSV

Figure 1: Web scraping process in general

Although the Selenium library is slower, it is very useful on websites have a static URL. On the other hand, although the Beautiful Soup library is quite fast, it is inadequate in websites with embedded page structures

Selenium is a library that can perform automatic web browsing through the web driver (Chaulagain et al., 2017). The HTML and CSS codes of the page are accessed as a result of the mechanical processes performed. Web scraping takes place as a result of manipulating them. Then, the data obtained is cleaned and made structural. The Selenium library is also used for testing operations (Gojare et al., 2015). All elements on the page can be manipulated with the functions to be written in these languages. Selenium converts these written request functions into Json Wire Protocol commands and sends them to the web drive of that browser, which browser is desired. After this point, the operations take place between the web driver and the browser (Figure 2).

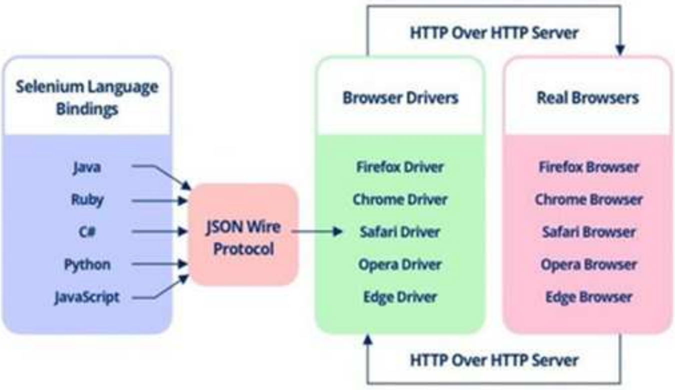

Figure 2: Web scraping Process with Selenium (Sadiq, 2020)

Another library used in our work is the Beautiful Soup (Hajba, 2018). This library is generally used to analyze HTML and XML structures. Unlike the Selenium library, it does not need any web driver structure. It reads the source codes of the target site and parses those source codes. Thus, it provides data to be obtained very quickly compared to the Selenium library. In our work, it took only 2 hours when the Beautiful Soup library was used, although it took 18 hours while using the Selenium library. The computations were carried out on Intel i7, 8 GB RAM, 256 GB SSD laptop.

2.2 Data Preprocessing

After the web scraping process, the data pre-processing phase is performed. Firstly, it is checked whether there is a deficiency in the attributes of the data at this stage. In our work, these missing data are removed from the system since the identified deficiency is very low compared to the total data (24 missing data).

In the second stage, string type numbers are determined and converted to integer type such as price and kilometer. After this process, outlier observations are cleared in the structure. The biggest source of these outlier observations is erroneous entries originating from the user (e.g. entering 300 in kilometers instead of 300.000) and the vehicles that the sellers put on the advertisement much higher than the market price (e.g. determining the price of the vehicle with 50.000 TL market price as 150.000 TL). After these outlier observations are removed from the structure, the data becomes ready for training.

2.3 Linear Regression

The Linear Regression model is a machine learning model used to predict a numerical dependent variable according to independent variables (Mekparyup et al., 2014):

$$y_{pred} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n \tag{1}$$

where, y_{pred} is the regression prediction value, β_0 is the regression constant, and $\beta_i, i = 1, .., n$, are independent variables' coefficients.

In this work the Multiple Linear Regression model was used because there were more than one independent variables. Ridge, Lasso and Elastic Net models, which are derivatives of Linear Regression, were also used in this study. While the L2 regularization is performed in Ridge Regression model, the L1 regularization is performed in Lasso Regression model.

2.4 K-Nearest Neighbors

Although the K-nearest Neighbors algorithm (KNN) is generally used in classification problems, it also gives very good results in regression problems (Guo et al., 2003). The KNN algorithm is an algorithm that estimates the result of the testing data using relevant values of the nearest k neighbors according to all sample data. Different methods can be used to calculate these distances:

Euclidean

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{2}$$

Manhattan

$$\sum_{i=1}^{k}|x_i - y_i| \tag{3}$$

Minkowski

$$(\sum_{i=1}^{k}(|x_i - y_i|)^q)^{1/q} \tag{4}$$

Although the KNN algorithm creates a very effective machine learning model especially against noise containing data, recalculating distances in each test case can pose problems for big data.

2.5 Random Forest

The Random Forest model can be used for both regression and classification problems. This model is based on the decision tree model (Breiman, 2001). One of the biggest problems of decision trees is overfitting problem. Random Forest gives a solution to this problem against simple decision tree model. Random Forest method is formed by random combination of different decision trees. Each tree has different degrees of weights in the model. Out-Of-Bag (OOB) error rate determination approach is used to determine the weights. The data set is divided into 2/3 of the training and 1/3 of the test sets in this method. The tree with the lowest error will have the highest weight, while the tree with the highest error will have the lowest weight. When forming the decision tree, the homogeneity of the classes in the nodes is very important. For this reason, some methods are used to make the most appropriate classification. The Gini Index method provides the best classification by ensuring the homogeneity of the classes (Menze et al., 2009).

$$Gini(D) = 1 - \sum_{j=1}^{n} p_j^2 \tag{5}$$

where $p_j, j = 1, .., n$, is the ratio of the count of the j^{th} class data to the total data, $Gini(D)$ is the amount of information. The value of Gini index determined the homogeneity of the dataset. While the high $Gini(D)$ value indicates that the class has a homogeneous structure, the decreasing in this value means that the homogeneity of the classes is impaired. A model is created by determining the tree structure according to the attributes giving the maximum descent of Gini index after splitting for each decision node.

2.6 Gradient Boosting Machine

The predictors in boosting operations are not made independently, as in the Random Forest model. Boosting is an iterative technique so the estimated value at each step is based on the errors of the prediction made before it. For this reason, in the recent step, estimations closer to the real values are made as follows (Friedman, 2001):

$$y'_{pred} = y_{pred} - a \cdot 2 \cdot \sum (y_i - y_{pred,i}) \tag{6}$$

where y'_{pred} is a new predicted value, y_i is the real value, $y_{pred,i}$ is the previous predicted value, and a is the learning rate parameter. Assuming that the mean error squares (MSE) method is used to determine the error values in the estimates made, equation (6) is obtained to update the new predictions. According to this equation, when the total value of the residuals is very close to 0 or equal to 0, the updates to the model will not be made.

2.7 XGBoost (Extreme Gradient Boosting)

Although Extreme Gradient Boosting model is based on the Gradient Boosting Machine (GBM) algorithm, it has serious differences according to this algorithm. One of these differences is parallelization. While estimating the classical GBM model, a tree is created for the current estimation. After the estimation is completed, the error rate is determined. According to this determined error, a tree is created again for the new prediction. However, while creating a tree in the XGBoost model, the branches of the attributes can be performed simultaneously according to different conditions. Thanks to this structure, parallel operations can be performed for the values of the same attribute in different situations that will produce independent results. For this reason, XGBoost has a faster model (Chen & Guestrin, 2016). In addition, it can produce solutions for big data with Out-Of-Core optimization and the model tries to prevent overfitting by regularizing it.

3 Training Results

In our work, the data were taken from online car sales sites. Intel i7, 8 GB RAM, 256 GB SSD laptop was used in the processing. It took only 2 hours when the Beautiful Soup library was used, although it took 18 hours while using the Selenium library. All the machine learning models mentioned above were trained on the available data and hyper-parameter optimizations were also performed in these models. Root Mean-Square Error (RMSE) obtained using equation (7) and Mean Absolute Error (MAE) obtained using equation (8) were used in order to compare these results with each other (Chai & Draxler, 2014).

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - y_{pred,i})^2} \tag{7}$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - y_{pred,i}| \tag{8}$$

The results for each vehicle class and for each method are given in Table 1 and Table 2. Abbreviations used indicate the following meanings in Table 2:
- learning_rate = Learning rate value selected for GBM model,
- max_depth =Maximum depth value selected for GBM model,
- n_estimators = Number of decision trees to be established in GBM model,
- subsample = Proportion of samples to be used for individual estimators,
- adj = Adjusted parameter values,
- def = Default parameter values.

Table 1: RMSE values of all models and MAE of the best model

Car brand	The best model	Elastic Net	GBM	KNN	Lasso	Linear reg.	Random Forest	Ridge	XGBoost	MAE of the best model
alfa-romeo	xgboost	15599,6	12364,2	22532,3	13249,0	13120,6	12176,4	12935,6	12155,5	6944,3
aston-martin	lasso	363364,0	307584,2	491469,9	247602,0	284912,8	295145,6	295431,5	308713,4	202422,2
audi	gbm	167148,6	40021,5	154526,3	104037,1	101300,1	40850,2	99781,1	41113,4	17684,3
bentley	xgboost	483118,4	239319,6	310775,4	423592,2	423114,6	255800,5	421811,3	221009,0	168591,8
bmw	xgboost	89968,4	39105,4	97604,1	55134,8	51756,2	42352,8	51454,0	38207,6	15654,8
chery	random_forest	5843,9	5602,8	6214,2	6475,3	6665,0	5596,2	5925,4	6549,0	4072,6
chevrolet	xgboost	102215,5	33432,0	89530,9	47279,5	47471,2	31237,3	38856,9	30764,2	7936,3
chrysler	gbm	30104,4	17054,4	47680,8	18152,2	18164,0	18120,1	18617,2	17408,9	12859,0
citroen	xgboost	12657,4	6372,4	18736,0	7462,3	7477,0	6594,1	7327,9	6353,0	4440,3
dacia	gbm	9108,9	6518,9	14697,1	7588,8	7572,7	6995,1	7556,6	6666,9	4612,7
daewoo	random_forest	3192,5	3675,7	3429,0	6021,5	6039,0	2672,5	3567,5	4379,1	2478,3
daihatsu	random_forest	7841,1	3548,1	13418,0	7936,4	7941,9	3442,1	4253,2	3979,0	2862,4
ds-automobiles	ridge	14124,6	9834,3	26873,6	9756,2	9756,6	10137,6	9715,6	11656,4	7995,5
ferrari	ridge	449637,8	450803,0	506279,7	415224,4	463616,0	379848,1	364776,8	391804,2	289485,6
fiat	gbm	12378,7	5631,7	18570,2	6670,3	6669,4	6391,9	6552,6	5812,1	3902,8
ford	gbm	47465,9	24378,9	52105,0	33267,5	32732,8	24679,0	25771,9	27987,5	7160,9
geely	ridge	4815,7	4612,8	8733,6	4065,9	4047,0	4526,4	3983,8	5067,9	3253,9
honda	gbm	23168,1	8189,8	25122,3	13044,3	13030,4	9116,9	13098,2	8384,5	5785,4
hyundai	gbm	14713,3	9325,1	17720,8	9840,0	9800,4	9585,8	9791,6	9389,2	4969,1
jaguar	gbm	100153,4	37855,4	111889,2	71277,3	70460,8	38553,9	64782,6	39586,5	21596,1
kia	gbm	18644,3	7447,5	24734,9	8284,2	8209,0	7515,6	7903,1	7476,7	5193,9
lada	gbm	3101,0	2889,9	4419,1	3162,2	3164,5	3067,1	3126,2	3246,4	2025,7
lancia	gbm	284166,0	249575,1	254575,4	272883,3	272910,3	257171,6	261006,5	264394,9	88199,6
maserati	xgboost	205524,1	178195,4	243332,3	193243,2	193114,3	175421,1	196351,8	165886,0	107412,0

Table 1: RMSE values of all models and MAE of the best model (continued)

Car brand	The best model	Elastic Net	GBM	KNN	Lasso	Linear reg.	Random Forest	Ridge	XGBoost	MAE of the best model
mazda	gbm	14950,7	6812,3	21646,1	7992,9	7886,5	7308,7	7993,8	7071,4	4638,8
mercedes-benz	gbm	168048,0	41843,9	150347,1	57343,9	57193,5	53404,2	57735,3	42478,9	16416,4
mini	xgboost	23310,5	11920,3	28235,2	15038,1	14542,6	12132,2	13906,8	11410,7	8698,9
mitsubishi	ridge	24243,6	9018,5	28212,5	10299,4	10351,5	9455,4	8831,2	10306,8	5978,3
nissan	xgboost	92176,5	27858,7	92707,7	59819,3	59834,6	20673,5	35246,7	10878,2	4921,6
opel	gbm	16193,3	8495,4	19484,4	9624,5	9584,9	8922,8	9542,9	8709,7	4982,2
peugeot	gbm	19748,7	6160,5	21078,0	7942,3	7990,6	6586,5	7965,4	6508,9	4391,2
porsche	xgboost	363982,0	227540,2	390145,1	350084,1	348952,0	225588,9	338956,2	217369,6	113726,5
proton	elastic_net	3480,7	4339,7	6984,7	3975,1	3977,0	4130,0	3677,3	4722,5	2775,4
renault	gbm	14809,9	6755,1	19818,4	9960,1	10009,2	7201,8	9918,8	6783,7	4352,3
rover	ridge	7948,7	5710,6	10673,2	5995,4	5782,1	5809,0	5577,9	6119,4	4249,2
seat	gbm	18682,0	10500,7	22445,9	12388,8	12376,1	10962,2	12342,5	10672,6	6685,0
skoda	random_forest	25894,9	12015,0	34754,3	16907,2	16887,8	11630,9	16081,2	17313,3	6651,5
smart	random_forest	14164,0	8328,4	15161,7	13083,7	13686,6	6916,0	11448,8	8521,2	5697,5
subaru	lasso	86341,8	55902,4	98804,3	51465,0	51540,1	61197,7	52371,3	59967,5	20389,3
suzuki	gbm	16376,3	11767,7	26656,6	13493,6	12940,0	11940,1	12559,1	11899,3	5550,3
tata	random_forest	2926,5	2647,3	4475,0	3442,7	3434,6	2456,5	2896,0	2968,6	1988,2
tofas	random_forest	7417,3	6262,9	8363,3	6764,5	6766,7	6164,2	6510,4	6763,0	3704,1
toyota	gbm	14521,4	7224,5	17182,2	8309,2	8298,8	7903,2	8300,6	7323,5	5177,1
volkswagen	xgboost	30623,1	13370,1	33241,5	16508,0	16546,2	13510,8	16336,5	12653,1	7069,5
volvo	xgboost	66578,6	14487,5	75511,2	19290,8	19342,9	15085,8	19162,6	13952,3	9018,7

Table 2: Error values of the GBM model with adjusted vs default hyper-parameters

Car brand	adj_ learning rate	adj _ max depth	adj _n_ estimators	adj _ RMSE	adj _subsample	def_learning_rate	def_max_depth	def_n_estimators	def_ RMSE	def_sub sample
audi	0,1	8	1000	40021,54	0,75	0,1	3	100	46110,14	1
chrysler	0,1	3	500	17054,45	0,5	0,1	3	100	17061,85	1
dacia	0,01	3	1000	6518,86	0,5	0,1	3	100	6688,67	1
fiat	0,1	3	500	5631,68	0,75	0,1	3	100	6321,61	1
ford	0,1	3	500	24378,92	0,75	0,1	3	100	25436,72	1
honda	0,01	5	1000	8189,79	0,5	0,1	3	100	9191,37	1
hyundai	0,1	3	500	9325,11	0,75	0,1	3	100	9837,35	1
jaguar	0,01	5	1000	37855,36	0,75	0,1	3	100	43997,67	1
kia	0,1	5	100	7447,54	0,5	0,1	3	100	7826,27	1
lada	0,1	3	100	2889,95	1	0,1	3	100	2896,40	1
lancia	0,1	50	1000	249575,05	1	0,1	3	100	264410,03	1
mazda	0,01	3	1000	6812,28	0,5	0,1	3	100	7209,72	1
mercedes-benz	0,1	3	1000	41843,90	1	0,1	3	100	55218,59	1
opel	0,01	8	1000	8495,42	0,5	0,1	3	100	9370,25	1
peugeot	0,01	5	1000	6160,49	1	0,1	3	100	7608,30	1
renault	0,01	5	1000	6755,13	0,5	0,1	3	100	7314,76	1
seat	0,01	5	1000	10500,67	0,5	0,1	3	100	10663,25	1
suzuki	0,01	3	500	11767,69	0,5	0,1	3	100	11785,96	1
toyota	0,1	3	500	7224,46	0,5	0,1	3	100	7720,79	1

The RMSE values of all machine learning models are available in Table 1. The model with the lowest error is chosen as our main training model by making a comparison. There is also Mean Absolute Error (MAE) of the model with the lowest error in the column "MAE of best model". If necessary, the data processing steps can be renewed and retrained by comparing the RMSE and MAE values of the selected model. In addition, the hyper-parameter values determined as a result of optimization of the most selected model (GBM) are given in Table 2. Thanks to this table, second degree hyper-parameter optimizations can be performed more easily.

When the results are analyzed, it can be seen which model is selected how many times as the best prediction model:

- Gradient Boosting Machine (GBM): 19
- Extreme Gradient Boosting (XGBoost): 11
- Random Forest: 7
- Ridge: 5
- Lasso: 2
- Elastic Net: 1

4 Conclusion

The Gradient Boosting Machine model was chosen as the best model in 19 different vehicle brands. As a result of the training, other models also were chosen as the best models such as XGBoost (11), Random Forest (7), Ridge (5), Lasso (2) and Elastic Net (1). According to the results, it is seen that the Gradient Boosting Machine model provides an obvious advantage over other models especially with the XGBoost model (30/45).

Another conclusion that can be achieved is that it is better to use many models collectively, rather than using a single model, especially in non-homogeneous data structures. In our work, it was observed that each vehicle brand had its own data distribution. Therefore, the model that provides maximum accuracy was chosen for each vehicle brand.

References

Breiman, L. (2001). Random Forests. *Machine learning, 45*(1), 5-32.

Chai, T., Draxler, R.R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)-Arguments against avoiding RMSE in the literature. *Geoscientific Model Development, 7*(3), 1247-1250.

Chaulagain, R.S., Pandey, S., Basnet, S.R., & Shakya, S. (2017, November). Cloud based web scraping for big data applications. In 2017 IEEE International Conference on Smart Cloud (SmartCloud) (pp. 138-143). IEEE.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery And Data Mining* (pp. 785-794).

Friedman, J.H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics, 29*(5), 1189-1232.

Gojare, S., Joshi, R., & Gaigaware, D. (2015). Analysis and design of selenium webdriver automation testing framework. *Procedia Computer Science*, 50, 341-346.

Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003, November). KNN model-based approach in classification. In *OTM Confederated International Conferences On the Move to Meaningful Internet Systems* (pp. 986-996). Springer, Berlin, Heidelberg.

Hajba, G.L. (2018). *Website Scraping with Python: Using BeautifulSoup and Scrapy.* Apress, Berkeley, California.

Mekparyup, J., Saithanu, K., & Buaphan, M. (2014). Multiple Linear Regression Analysis for Estimation of Nitrogen Oxides in Rayong. *Global Journal of Pure and Applied Mathematics, 10*(5), 769-774.

Menze, B.H., Kelm, B.M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W., & Hamprecht, F.A. (2009). A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC Bioinformatics, 10*(1), 213.

Sadiq, S. `https://hackr.io/blog/complete-guide-selenium-webdriver#comment-link` 22.06.2020

Yildiz, A. https://learn.g2.com/hs-fs/hubfs/what-is-web-scraping.png?width=480&name=what-is-web-scraping.png, 22.06.2020