European Transport Congress

19th European Transport Congress of the EPTS Foundation e.V

CONFERENCE PROCEEDINGS

ZUM d.o.o.

CARE 4CLIMATE

University of Maribor

European Platform of Transport Sciences

19th European Transport Congress of the EPTS Foundation e.V

European Green Deal Challenges and Solutions for Mobility and Logistics in Cities

October 7 - 8, 2021, Maribor, Slovenia

Conference Proceedings

Editor

Tomislav Letnik

October 2021

Title 19th European Transport Congress: European Green Deal Challenges and Solutions for Mobility and Logistics in Cities

October 7 – 8, 2021, Maribor, Slovenia, Conference Proceedings

Editor doc. dr. Tomislav Letnik, PhD
(University of Maribor, Faculty of Civil Engineering, Transportation Engineering and Architecture)

Reviewers Prof. Laurent Guihery (CY Cergy Paris University), Prof. Dr. Sönke Reise (University of Applied Sciences, Business and Design Wismar), Assoc. Prof. Borna Abramovic, PhD (University of Zagreb, HRV), Ing. Bc. Vladimír Faltus, PhD (Czech Technical University in Prague (CTU))

Technical editors Metka Dernovšek, MSc.
(University of Maribor)

Cover designer Metka Dernovšek, MSc.
(University of Maribor)

Graphics material Authors

Conference 19th European Transport Congress: European Green Deal Challenges and Solutions for Mobility and Logistics in Cities

Location and date Maribor, October 7 – 8, 2021

Organizing committee Tomislav Letnik (University of Maribor), Zdravko Kačič (University of Maribor), Stane Božičnik (University of Maribor), Eva Schmidt (European Platform of Transport Sciences), Mateja Kukovec (ZUM d. o. o., LIFE IP CARE4CLIMATE), Sebastian Belz (European Platform of Transport Sciences), Katja Hanžič (University of Maribor), Maršenka Marksel (University of Maribor), Florian Polterauer (Plasser & Theurer), Mitja Klemenčič (University of Maribor)

Published by Zum urbanizem, planiranje, projektiranje d.o.o.
Grajska ulica 7, 2000 Maribor, Slovenia

Co-published by University of Maribor
Slomškov trg 15, 2000 Maribor, Slovenia

Edition 1st

Publication type E-Book

Available at www.fgpa.um.si/etc/downloads/

Published Maribor, October 2021

CIP - Kataložni zapis o publikaciji
Univerzitetna knjižnica Maribor

656.1:502/504(082)(0.034.2)

EUROPEAN Transport Congress of the EPTS Foundation e.V (19 ; 2021 ; Maribor)
19th European Transport Congress of the EPTS Foundation e.V [Elektronski vir] : European green deal challenges and solutions for mobility and logistics in cities : conference proceedings : October 7-8, 2021, Maribor, Slovenia / editor Tomislav Letnik. - 1st ed. - E-zbornik. - Maribor : Zum urbanizem, planiranje, projektiranje : University, 2021

Način dostopa (URL): https://www.fgpa.um.si/etc/downloads/
ISBN 978-961-95633-0-4 (ZUM)
COBISS.SI-ID 90733315

ISBN 978-961-95633-0-4 (pdf)

Price Free copy

For publisher Andreja Kuzmanić, ZUM urbanizem, planiranje, projektiranje d.o.o.

The Use of Machine Learning to Predict Diesel Fuel Consumption in Road Vehicles 207
Artur Budzyński, Aleksander Sładkowski

INNOVATIONS IN URBAN / REGIONAL MOBILITY AND FREIGHT

Role of Railway Transport in Green Deal 2050 Challenge – Situation in Czechia 225
Václav Lauda, Vojtěch Novotný

The New Franco-Genevan Rail Service Léman Express: The Challenge of Mobility in the Cross-Border Metropolis of Greater Geneva 239
Laurent Guihéry

Quality Evaluation of Timetables in the Non-Metropolitan Area: A Case Study of the South Bohemia Region 257
Vladimír Ľupták, Ondrej Stopka, Ladislav Bartuška, Martin Jurkovič

Using Advanced Technologies to Improve Urban Mobility/ Accessibility of People with Disabilities 279
Metka Dernovšek, Nataša Rebernik, Matej Brumen, Tomislav Letnik, Katja Hanžič

Adaptation of European Railways to the Digital Economy in the Era of Energy and Climate Transformation 295
Bartosz Grucza, Wojciech Paprocki

Saturation Flow at Nested Signalized Intersection: A Case Study in Niğde 303
Hatice Göçmen Demir, Yusuf Kağan Demir, Cansu Zorlu

DIGITALISATION, AUTOMATIZATION AND MODELLING

Digitalisation in Public Transport as an Opportunity and Threat for Speciffic Target Groups: An Analyses of Several Use-Cases 315
Elmar Fürst, Gerald Lamprecht, Bernhard Landrichter

Integration Opportunities of Drones into the Document Handling and Transporting 327
Dr. Krisztián Bóna, Dr. Gábor Kovács, Cintia Párizs

19TH EUROPEAN TRANSPORT CONGRESS
EUROPEAN GREEN DEAL CHALLENGES AND SOLUTIONS FOR MOBILITY AND LOGISTICS IN CITIES
OCTOBER 7 – 8, 2021, MARIBOR, SLOVENIA, CONFERENCE PROCEEDINGS

The Use of Machine Learning to Predict Diesel Fuel Consumption in Road Vehicles

ARTUR BUDZYŃSKI, ALEKSANDER SŁADKOWSKI

Abstract The article is devoted to the issues of using machine learning methods with the scikit-learn library to predict diesel fuel consumption for trucks. Its main goal was to find a solution that allows predicting the fuel consumption of a truck with the least error. The data for the study was collected using the GPS tracking system of a small transport company from Poland. The problem of data adaptation for analysis was presented. 13 different forecasting models were evaluated on three success indicators. Finally, the models became available on the git-hub platform, which each user can use to predict fuel consumption in their fleet.

Keywords: • Machine Learning • Fuel Consumption Prediction • Scikit-Learn

CORRESPONDENCE ADDRESS: Artur Budzyński, MSc, Aleksander Sładkowski, Prof., DSc, Silesian University of Technology, Krasiński 8, 40-019 Katowice, Poland, e-mail: artur.budzynski@polsl.pl, e-mail: aleksander.sladkowski@polsl.pl

 ISBN 978-961-95633-0-4

INTRODUCTION

Fuel consumption is one of the main cost items in the operation of road transport. For every transport company, actions to optimize fuel consumption are essential. This is dictated by both environmental and economic factors.

For example, in the article [1], models for predicting fuel consumption based on data from a smartphone and an onboard diagnostic (OBD) system in a taxi were proposed. Models of neural networks, backpropagation errors, carrier vector regression, and random forests were used. The inputs were: average speed, average acceleration, average deceleration, percentage of acceleration time, percentage of deceleration time. The random forest model turned out to be the best.

Artificial neural networks are used relatively often to predict fuel consumption in vehicles of energy companies, for example, [2]. At the same time, the following is analyzed: maintenance that excludes downtime, periodic service and installation supervision. For the analysis in the network, the following data are entered, among other things: the number of cylinders of the car engine, engine displacement, number of valves, vehicle model and weight.

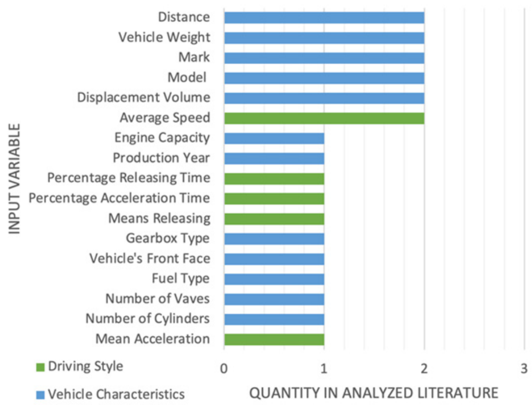
Scientists from Turkey have built 3 models to predict instantaneous and total fuel consumption. To do this, they used multiple linear regression, an artificial neural network, and a support vector machine. The type of fuel, the working volume of the cylinder, the frontal surface of the car, the mass of the car, the distance, and the average speed were used as the initial data. Support vector machine turned out to be the best forecasting tool [3].

Canadian scientists decided to build prediction models based on large data sets. The input data included: car segment, model, year, gearbox type, engine volume, driving time and distance. They used a carrier vector regression model and artificial neural networks, which proved to be better [4].

After analyzing the literature, we can conclude that individual projects are focused either on the vehicle and its characteristics, or on the driver and his driving style. Information about how often each of the variables from the projects from the above literature appeared is shown in Fig. 1. In the article, both types of data were taken into account in the modeling. The data for the analysis comes from a small transport company in southern Poland that uses the Lontex system to monitor its vehicles. The data is uploaded in the form of daily vehicle mileage reports. The surveyed vehicle fleet consists of 8 trucks (set: a tractor with a semitrailer). The analyzed data refers to the period from the beginning of May 2019 to the end of May 2020. This includes

1797 cases where the daily mileage for a given vehicle was non-zero. The aim of this article is to investigate whether the models from the scikit-learn libraries are useful for predicting the fuel consumption of trucks.

Figure 1. Input variables in the analyzed literature



When analyzing the literature, no example of shared knowledge was found on how to build a machine learning model step by step so that you can repeat the experiment and implement it in the company. In this work, it was decided to make available the code written during the analyzes. It is assumed that this is a phenomenon beneficial for entrepreneurs, especially small ones. The argument is that keeping an analyst full-time for such a company is too much of an expense. An analyst who is a machine learning specialist is especially expensive.

A Jupyter Notebook [5] was used to perform the tests. This is a tool that allows you to write code and comment on it in your browser. It is convenient for managing ongoing projects and documents. It also allows you to freely share the code referred to in the publication. The Python programming language was used during the calculations. It is a language used successfully for scientific calculations and reflection. It is attractive for exploratory data analysis and for the development of algorithms. There are many useful libraries for Python [6]. One of them is Scikit-Learn, which allows the implementation of many machine learning algorithms [7]. The Pandas library was also used for analyzes, which facilitates working with data sets [8]. The NumPy library was also used, which allows many numerical calculations in Python and is an open access software [9]. The survey data is collected by the GPS vehicle tracking system. A comprehensive library for creating static, animated, and interactive visualizations in Python named Matplotlib was used for data visualization [10].

PREPARATION OF DATA FOR RESEARCH

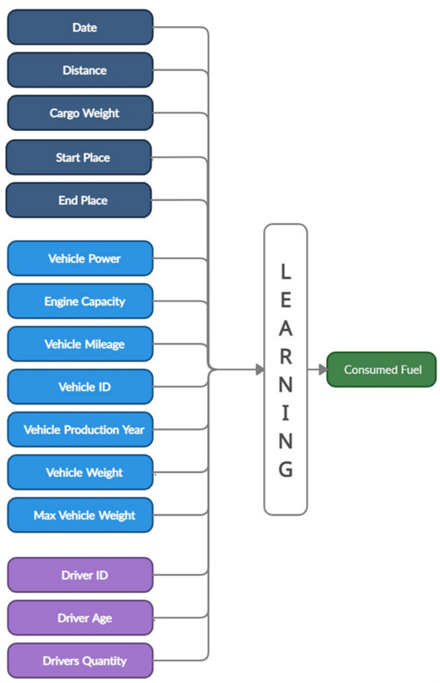
The data for the study is taken from the Lontex monitoring fleet management system. The lines with zero kilometers have been deleted. Based on the date, new features were created by specifying separately: year, month, day of the week. Vehicle mileage data in the Lontex system is stored intermittently. It is visually convenient when a person is looking at the table. However, it is not good for the machine and it needs to be changed. Using the code, the gaps were removed and the variable type changed to be useful for the model. The model diagram is shown in Fig. 2.

Missing data was supplemented with the mean value of a given feature based on existing data; it is common practice. In this case, it was considered better than filling the gaps with a value of 0 or -1. Data for the test for the period from the beginning of May 2020 to the end of May 2021. The training data is 70 %, the test data is 30 %, which corresponds to 1257 training and 540 test cases.

SUCCESS METRICS

3 success metrics were selected to assess the quality of the models. Calculating them will allow you to compare individual models with each other. The first is MSE (mean square error), this is the difference of sums of squares. The lower the value, the better the model performs. The pattern is shown below.

Figure 2. Variables for prediction model



$$MSE = \frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2 \ , \quad (1)$$

where: $\hat{y}_t$ – predicted value of the dependent variable; y_t – actual value of the dependent variable.

The second metric is RMSE (root mean square error), i.e., the root of MSE, and R^2, i.e., the coefficient of determination. It illustrates errors better than MSE. The lower the value, the better the model performs. The pattern is shown below.

$$RMSE = \sqrt{MSE}. \quad (2)$$

The last metric is the coefficient of determination R^2. It ranges from 0 to 1. The formula is presented below.

$$R^2 = \frac{SS_M}{SS_T} = \frac{\sum_{t=1}^{n}(\hat{y}_t - \bar{y})^2}{\sum_{t=1}^{n}(y_t - \bar{y})^2} \ , \quad (3)$$

where $\bar{y}$ – mean value of the dependent variable.

PREDICTIONS USING VARIOUS MODELS

13 regression models were selected: 2 classical linear regression, 7 regression with the selection of variables, 2 Bayess regressors, 1 outlier regressor, 1 generalized regression. The formula for the generalized linear model is shown below:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + \cdots + w_p x_p \ , \quad (4)$$

where: $\hat{y}$ – forecasted value; w_0 – regression coefficient for an absolute term; w_1, …,w_p – regression coefficients; x_1, … , x_p – variables.

LinearRegression

LinearRegression is a linear regression model. A linear model with coefficients adapts to minimize the residual sum of squares between the observed target values in the data set and the target values predicted by the linear approximation. The problem is described mathematically by the pattern:

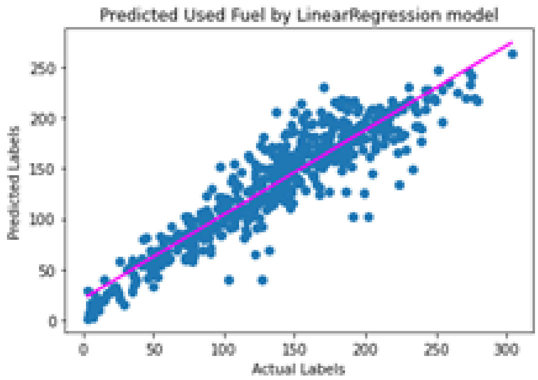
$$\min_{w} ||Xw - y||_2^2, \quad (5)$$

where: w – $(w_1, \ldots, w_p)$ – coefficient; y – output variable.

The calculated metrics during training are shown below, and the model predictions during testing are shown in Fig. 3.

MSE: 560,67; $RMSE$: 23,68; R_2: 0,85

Figure 3. Predictions with model LinearRegression



Similar figures could be presented for each calculation method (each model).

RidgeRegression

RidgeRegression is also a linear regression model. It solves some of the problems of ordinary least squares by imposing a penalty on the size of the coefficients. The formula below describes the problem to be solved and the calculated metrics. Model predictions when testing in Fig. 4.

$$\min_{w} ||Xw - y||_2^2 + \alpha ||w||_2^2 \ , \tag{6}$$

where α - complexity parameter.

MSE: 560,68; $RMSE$: 23,68; R^2: 0,85

ElasticNet

It is a linear regression model with two regulations of the norms of coefficients. This allows you to learn a rare model in, which few weights are non-zero as in the case of the Lasso. The formula below describes the problem to be solved and the calculated metrics. The model predictions during testing are shown in Fig. 5.

$$\min_{w} \frac{1}{2n_{\text{samples}}} ||Xw - y||_2^2 + \alpha\rho ||w||_1 + \frac{\alpha(1-\rho)}{2} ||w||_2^2 \ , \tag{7}$$

where ρ - $l1\_ratio$.

MSE: 580,56; $RMSE$: 24,09; R^2: 0,85

Lars

Least-angle regression is a regression algorithm for high-dimensional data. At each step, the algorithm finds the function that most correlates with the goal. If there is a situation where there are many features with equal correlation, it does not continue along the same feature but runs in an equilateral direction between these features. The calculated metrics are shown below. The model predictions during testing are shown in Fig. 6.

MSE: 563,33; $RMSE$: 23,73; R^2: 0,85

Lasso

Lasso is a linear model that estimates rare coefficients. This is useful in some contexts because it tends to prefer solutions with fewer non-zero coefficients, effectively reducing the number of features on, which a given solution depends. The calculated metrics are shown below. The model predictions during testing are shown in Fig. 7.

MSE: 578,64; $RMSE$: 24,06; R^2: 0,85

LassoLars

It is a Lasso model with an implemented Lars algorithm. The calculated metrics are shown below. The model predictions during testing are shown in Fig. 8.

MSE: 2062,24; $RMSE$: 45,41; R^2: 0,46

LassoLarsIC

It is a Lasso model with an implemented Lars algorithm. It uses BIC and AIC for this. BIC is a Bayessian evaluation criterion. AIC is the Akaike evaluation criterion. The calculated metrics are shown below. The model predictions during testing are shown in Fig. 9.

MSE: 567,87; $RMSE$: 23,83; R^2: 0,85

OrthogonalMatchingPursuit (OMP)

Orthogonal Matching Pursuit (OMP) implements the OMP algorithm to approximate the fits of a linear model with a limited number of non-zero elements. If this number is not specified then the default value of 10 % of the features is used.

$$\underset{w}{argmin}||y - Xw||_2^2 \ subject\ to\ ||w||_0 \leq n_{nonzero\backslash\_coefs}\ , \tag{8}$$

where $n_{nonzero\backslash\_coefs}$ – restriction of non-zero elements.

This model predictions during testing are shown in Fig. 10.

MSE: 716,03; $RMSE$: 26,76; R^2: 0,81

OrthogonalMatchingPursuitCV

It is an OMP algorithm with cross validation. The calculated metrics are shown below. The model predictions during testing are shown in Fig. 11.

MSE: 605,12; $RMSE$: 24,60; R^2: 0,84

BayesianRidge

Bayesian techniques can be used to include regularization parameters in the estimation procedure. The regularization parameter shall be fine-tuned to the available data. BayesianRidge estimates the probabilistic model of the regression problem. The predictions during testing are shown in Fig. 12.

$$p(w|\lambda) = \mathcal{N}(w|0, \lambda^{-1} I_p)\ , \tag{9}$$

where: p – predictive distribution; λ – regularization parameter; I_p – identity matrix $M \times M$.

MSE: 585,76; $RMSE$: 24,20; R^2: 0,85

ARDRegression

ADR (Automatic Relevance Determination) is very similar to BayesianRidge but can lead to rarer ratios.

$$p(w|\lambda) = \mathcal{N}(w|0, A^{-1})\ , \tag{10}$$

where $A = \lambda = \{\lambda_1, \dots, \lambda_p\}$ – diagonal matrix.

MSE: 716,05; $RMSE$: 26,76; R^2: 0,81

RANSACRegressor

The RANSAC model (RANdom Sample Consensus) is an iterative algorithm for the reliable estimation of parameters from a subset of intermediate data from a complete dataset. The calculated metrics are shown below.

MSE: 704,72; $RMSE$: 26,55; R^2: 0,81

TweedieRegressor

This model is used for analyzing various GLM (Generalized Linear Regression) models. The calculated metrics are shown below.

MSE: 936,21; $RMSE$: 30,60; R^2: 0,75

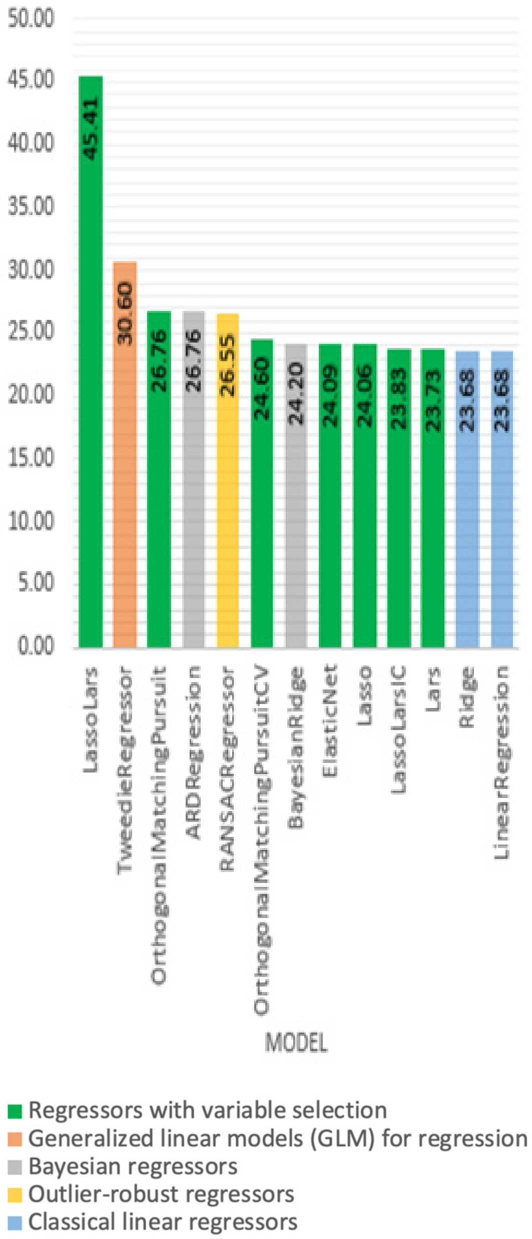
Summary of training and testing

Thirteen models were trained and tested in the work. 3 success metrics were calculated for each: MSE, $RMSE$, and R^2. A breakdown of the models according to the $RMSE$ metric is shown in Fig. 4. The MSE metrics are not plotted as they are simply squared $RMSE$ values. The $RMSE$ metric was found to better visualize the results than the MSE. The summary of the R^2 metric is shown in Fig. 5.

VALIDATION

For the validation process, data was collected from the Lontex vehicle tracking system. The data comes from the same company. The time range covers the entire month of June 2021. In summary, the conditions are the same as for testing.

Figure 4. Testing with metric RMSE



The new thing is that the data on new data reflects the conditions in, which the model is to be implemented and operated, i.e., predict fuel consumption in the future. Figs. 6 and 7 show the $RMSE$ and R^2 metrics of all tested models. The $RMSE$ metric is simply the root of the MSE metric. It was considered sufficient to plot one of them, i.e., $RMSE$, as the one that better visually presents the results. The LARS was the best performing model. The LassoLars model was the weakest. There are also visible differences between the values obtained during testing and validation. This is an argument for validating after testing is complete. This is valuable and answers the question of how the model deals with completely new data under real conditions when deployed.

Figure 5. Testing with metric R² (hereinafter column colors match the description in the previous figure)

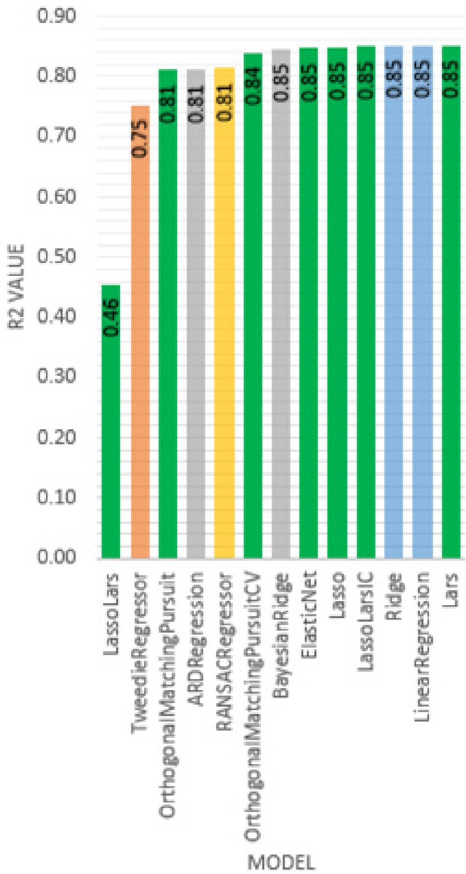


Figure 6. Validation with metric RMSE

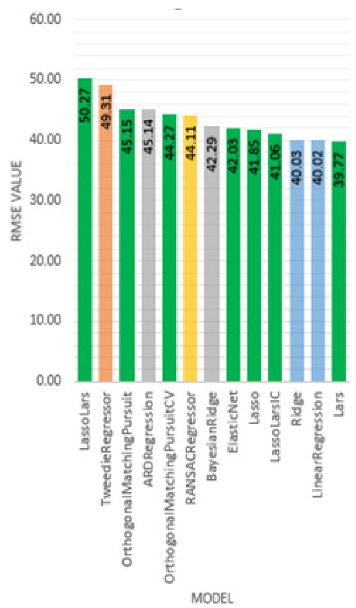
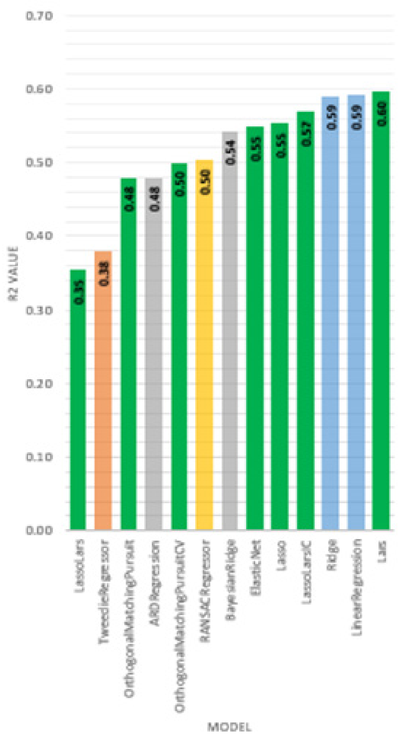


Figure 7. Validation with metric R²



IMPLEMENTATION

At the beginning of the article, the subject of the software language, tool and software was discussed. Creating the right environment to do such a project on your own is quite a complicated process. The solution to this problem is to use the ready-made Anaconda package [12].

It should be downloaded from the manufacturer's website. The software is available for the most popular operating systems: Windows, Linux and MacOs. After downloading, we have access to Jupyter Notebook, which is a very convenient tool for programming in Python. All necessary libraries are also configured: Scikit-Learn, Pandas and NumPy.

The next step is to create your own "csv" file that will be used to train the models.

Then download the ready code from the project website [14] and run it on your computer in Jupyter Notebook. In case of problems with implementation, you can use the "Step by Step" guide also available on the project website [14]. There, each of the steps above is presented with a detailed description of each click and a visualization of what it looks like in Windows. The tutorial is adapted to a person who had no contact with programming before.

CONCLUSIONS

Machine learning can be used to predict fuel consumption in heavy goods vehicles. Many variables affect fuel consumption. These are the variables related to the driver, his driving style, the vehicle and its properties, and the connection route with the specification of the goods transported. One of the goals of this article is to increase machine learning deployment possibilities for the road haulage business. This is in line with the strategies of research and government units aimed at the widest possible use of artificial intelligence in the economy by generating positive value. The Lars was the model that performed best during the research. LassoLars was the weakest.

GitHub allows you to make your code available to a wide audience. This is beneficial for improving collaboration and learning [13]. The code from testing, validation and all saved models were made available on the GitHub platform to enable implementations [14]. Using the information contained in this publication and the provided code, you can make a prediction or build your own machine learning model. Using the provided code, it can be done by a person who does not have much experience with machine learning and programming in Python. It is assumed that the use of machine learning will allow fleet managers to increase awareness of the dependencies affecting fuel consumption. This knowledge can help you make better fuel-saving decisions. This effect is beneficial for economic and environmental reasons, both for the enterprise and for the state economy as a whole.

REFERENCES

[1] Yao, Y., Zhao, X., Liu, C., Rong, J., Zhang, Y., Dong, Z., Su, Y. & Chen, F. (2020). Vehicle Fuel Consumption Prediction Method Based on Driving Behavior Data Collected from Smartphones. Journal of Advanced Transportation, 2020 (9263605), 1-11.

[2] Zargarnezhad, S., Dashti, R. & Ahmadi, R. (2019). Predicting vehicle fuel consumption in energy distribution companies using ANNs. Transportation Research Part D: Transport and Environment. 74, 174-188.

[3] Çapraz, A.G., Özel, P., Şevkli, M. & Beyca, Ö.F. (2016). Fuel Consumption Models Applied to Automobiles Using Real-time Data: A Comparison of Statistical Models, Procedia Computer Science. 83, 774-781.

[4] Moradi, E. & Miranda-Moreno, L. (2020). Vehicular Fuel Consumption Estimation Using Real-World Measures Through Cascaded Machine Learning Modeling. Transportation Research Part D: Transport and Environment. 88(102576), 1-17.

[5] Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S. & Willing, C. (2016). Jupyter Notebooks—a Publishing Format for Reproducible Computational Workflows. Positioning and Power in Academic Publishing: Players, Agents and Agendas - Proceedings of the 20th International Conference on Electronic Publishing, ELPUB 2016 (pp. 87-90).

[6] Pérez, F. & Granger, B.E. (2007). IPython: A System for Interactive Scientific Computing. Computing in Science and Engineering. 9(3), 21-29.

[7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. 12, 2825-2830.

[8] McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, SCIPY 2010 (pp. 56-61).

[9] Van Der Walt, S., Colbert, S.C. & Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science and Engineering. 13(2), 22-30.

[10] Hunter, J.D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science and Engineering. 9(3), 90-95.

[11] Dabbish, L.A., Stuart, H.C., Jason, T. & Herbsleb, J.D. (2012). Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository. Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW (pp. 1-11).

[12] Budzyński, A. (2021). FCPS – Fuel Consumption Predicting System. URL: https://github.com/BudzynskiA/FCPS.