Oracle Machine Learning for Python in APEX - Analyzing and Predicting CO2 Emission by private vehicles

Miruna Teleașă[1], Alexandra Teodora Bardici[2]
[1,2]Bucharest Academy of Economic Studies
teleasamiruna19@stud.ase.ro, bardicialexandra19@stud.ase.ro

Nowadays, the global warming threat is a highly discussed matter. One of the factors that accelerates this process is the air pollution that can be caused by cars' emissions. This paper concerns how the size of the engine, the type of fuel, the fuel consumption and the transmission type influence the emission of CO2. In order to understand and predict that variable, we used several machine learning algorithms, such as Regression for Generalized Linear Model or K-Means for Hierarchical Cluster Model. The technology that empowered this analysis was Oracle's Machine Learning for Python (OML4Py) that allowed us to integrate both database and data management concepts and data analysis algorithms. By doing that, we managed to discover a pattern for the emission of CO2 based on the factors previously mentioned and, after that, predict future levels of CO2 emissions for various car models.

Keywords: *Machine Learning Algorithms, Python, Oracle Autonomous Database, Environment, Regression, K-Means*

1 Introduction

The analysis we are going to perform has the goal of estimating the influence of different vehicle characteristics on the carbon dioxide emissions produced by it, with the purpose of providing a guideline for the most environment friendly vehicles. This is done with the hope that, in the future, car manufacturing companies, on the one hand, and vehicle buyers, on the other hand, would make conscious choices when it comes to vehicle characteristics and attributes, providing the best alternatives to eco-friendly vehicles. We chose these data considering that, in Europe, for example, passenger cars are the largest air polluters, having accounted for 60.7% of total road transport emissions in 2016 [1]. Considering also that the entire transport sector accounts for 21% of the total carbon dioxide emission [2], we can safely say that the personal vehicles make up 12.74% of the total emissions worldwide. This is an impactful percent and understanding the way in which these vehicles can be altered to create more eco-friendly ones could lead to a better, safer world [1]. In order to do this, it is important to understand the importance that various technical attributes of the vehicles have when it comes to the carbon dioxide they emit.

To achieve the previous result, we used a data set provided by the Canadian Government, which contains the model-specific estimated carbon dioxide emissions for new light-duty vehicles for retail sale in Canada in the year 2022 [3]. This dataset contains columns for the Make and Model and Class of the vehicle, all three being qualitative variables, considering for the Model variable six different types: Four-wheel drive (4WD), All wheel drive (AWD), Flexible-fuel vehicle (FFV), Short Wheelbase (SWB), Long Wheelbase (LWB) and Extended Wheelbase (EWB). Moreover, a series of quantitative variables were also considered, such as the Engine size measured in litres, the number of Cylinders of the engine, the combined Fuel Consumption measured in litres/100km and, finally, the CO2 Emissions, which is our considered dependent variable, measured in g/km. Last but not least, the Transmission and Fuel Type were also considered, both qualitative variables, first being either Automatic (A), Automated Manual (AM), Automatic with Select shift (AS), Continuously Variable (AV), Manual (M), or measured in the number of gears, a number from 3 to 10. The Fuel Type, however, is considered either Regular (X) or Premium (Z) Gasoline, Diesel (D), Ethanol (E) or Natural Gas (N).

In order for the machine learning algorithms to be applied on our set of data, a software for data analysis was needed that would support large sets of data and would enable us to use a programming language, for example Python, to conduct the analysis. Therefore, we used Oracle's Machine Learning for Python API (OML4Py), which enables the employment of Python commands for data transformation and analysis, while also providing a statistical approach to data examination. This API interprets Python commands and translates them into operations for SQL in-database execution, thus connecting the two programming languages into a complex environment. The main advantage of this software is that it allows users to operate directly onto the database without using SQL commands, as it transparently translates Python functions into SQL. Also, it provides access to preexistent machine learning algorithms, while minimizing the data movement across platforms, keeping it secure in the autonomous database.

OML4Py environment provides direct access to all Python libraries needed for data analysis and machine learning processes. Out of those libraries, we firstly used the *pandas* library, which is one of the open source Python standard libraries for data importing and manipulation through an indexed data structure called DataFrame [5]. Furthermore, we used the *numpy* standard Python library, a library containing various mathematical functions and data structures that support "vectorized" operations for big data [5]. Another Python specific functionality that was used during the machine learning process was the data plotting and data visualization, for better understanding the way in which our dataset is structured, and the impact of the algorithms applied on it [5]. For all this, the matplotlib library was used. Lastly, the module that made possible the integration between the database and the Python commands was the oml module, which allows the manipulation of Oracle Database objects, such as tables and views, by embedded Python specific commands [6].

The Oracle Autonomous Database is a cloud-base database management technology developed and provided by Oracle that is automated to perform routine tasks like backups and security [4]. The purpose of using this database is to store data and information used by the application and to be able to access it from the front-end. There are two main components on which the autonomous database is built upon: the data warehouse and the transaction processing. While first manages the entirety of not only the data, but also the operations performed on it, like provisions, scaling or configuring, the second provides the analytical and security-driven processes that guarantee the smooth functioning of the app that will be developed based on it.

The application that will be developed contains only a fragment of the entirety of capabilities that this analysis offers, with the purpose of demonstrating the hypothesis. However, very complex and useful applications can be followingly developed on the machine learning algorithms' backbone and making use of the discoveries that will be outlined in this article. The final goal might be developing an application that would be available to every user interested in buying an eco-friendly vehicle and that would allow him to use basic information regarding the said vehicle, like the fuel consumption or engine size in order to find out what volume of emissions the vehicle would produce.

Considering this, to develop the application we used Oracles Application Express. Oracle Application Express is one of those tools, hereinafter referred to as APEX, that has emerged as an integrated development environment, providing not only database management, but also web-development functionalities. The enthralling benefit of using APEX in comparison to other web-development tools is that Oracle has embedded in their technologies a Hyper Text Markup Language (HTML) generator, which allows users to create an application without vast knowledge of HTML, JavaScript, PHP, or CSS. Not only this, but the connection between the database and the web application also empowers users to build a front-end and a back-end application without having to use third parties APIs.

2 Algorithms, Results, and Interpretation

The machine learning process is based on the study of existent datasets from various perspectives by using various algorithms and methods. Some authors might say that training the computer into predicting different data is actually synonym with "the systematic study of algorithms and systems that improve their knowledge or performance with experience" [7]. The dissimilarity between the algorithms is derived from the desired predictions and the type of data analyzed, considering that there are algorithms adapted for qualitative data, quantitative data or for both. By way of illustration, an algorithm adapted for qualitative data is the Natural Language Processing, which, based on sentiment analysis, can predict if a news article is fake or can power various Smart Assistants, like Siri or Alexa. However, the results from the data set approached by this project will be quantitative data types, considering that we are interested in predicting the CO2 emissions of different cars, measured in g/km. In the light of that, we used a series of algorithms adapted to this specific type of data. We decided, for that manner, to use three types of data analysis algorithms, all three having slightly different approaches, but similar purposes.

Firstly, we will apply a simple Regression algorithm, on a model that we will generate, a model that will be considered as linear. Secondly, we will apply a K-Means algorithm, suitable for clustered data, which will imply grouping our data and will result in a classification rather than a prediction. Lastly, we will consider the regression process on a Support Vector Machine model, which is marginally more complex than the linear model, but also generates more accurate and detailed results.

Before applying these machine-learning specific algorithms, it is important to point out that we divided the initial data frame into two different data frames, 80% for training the data and 20% to test the results, and we accustomed both data frames around the dependent variable considered, CO2 emissions.

```
emissions_df = oml.sync(table =
'EMISSIONS_2022')
training_df, testing_df =
emissions_df.split(ratio = (0.8,0.2))

training_x =
training_df.drop(['CO2_EMISSIONS'])
training_y =
training_df['CO2_EMISSIONS']

testing_x = testing_df
testing_y =
testing_df['CO2_EMISSIONS']
```

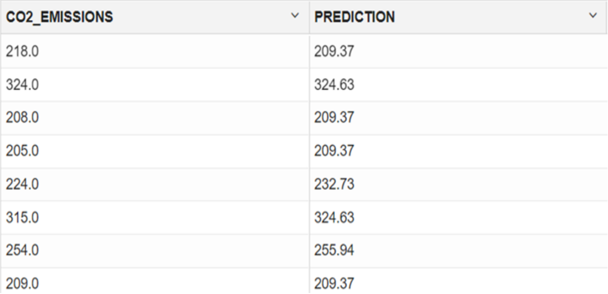
2.1. Regression – Generalized Linear Model (GLM)

A Generalized Linear Model is a certain type of modeling structure adapted to regression analysis, as in this type of analysis, the dependent variable is modeled as a linear function of the independent ones. The main difference between a Linear Model and a Generalized Linear Model is that the latter is more adapted to variables without a specific, normal distribution, thus being more reliable when it comes to big data analysis [8]. In code, the first step of the analysis was to generate the said model, and Oracle's Machine Learning API provided the oml.glm() method, which takes as parameters the type of analysis carried out, the regression in our case, and the settings needed for this analysis. The result of calling this method is a Generalized Linear Model, upon which the fit() method can be applied, so that it considers the training and testing data we desire to analyze. After fitting the model to the desired datasets, we can further investigate its details to see the basic statistical indicators.

The first indicator observed is the adjusted R2, which stands for the coefficient of determination and marks the percentage of variance in the dependent variable that is influenced by the independent variables, adjusted for the number of observations. The resulting coefficient in the conducted analysis is 98.92%, meaning that almost the entire variation of the CO2 emissions in the analyzed data is explainable by the variables used and that the investigation is statistically significant.

Moreover, the regression coefficients of the generated model, corresponding to an Analysis of Variance (ANOVA) table, coefficients determining the exact dependency function between the CO2 emission and the independent variables used, can be accessed. Consequently, we can better understand the influence each variable has on the final predicted model, and what is the direction in which this influence is oriented. As an example, for the coefficient obtained of the fuel consumption of +24.84, we can say that if the fuel consumption increases by 1 L/100km, considering all other independent variables as constant, the CO2 emissions will increase by 24.84 g/km.

To better understand the model significance and the importance of the predicted results, the residual values of the observation were calculated, that is, the difference between the values that our generated formula predicts and the actual values. For this, the data testing fragment was used, on which the predict() method was applied and the following results were obtained, stating that the error between the actual and the predicted values is relatively small, so the model is fairly correct:



CO2_EMISSIONS	PREDICTION
218.0	209.37
324.0	324.63
208.0	209.37
205.0	209.37
224.0	232.73
315.0	324.63
254.0	255.94
209.0	209.37

Fig. 1. Example of the residuals obtained in the Generalized Linear Model's Regression

Another method used to visualize better the error of the predicted model is a scatter plot, constructed with the seaborn Python library, that plots on one axis the predicted values and on the other the actual values of the dependent variable. Taking this into account, the closer the points are to the regression line, the more fitted and significant the constructed model. Moreover, another fact to take in consideration after examining the scatter plot is that the line that marks the relation between the two variables is positively skewed. That shows that there is a direct, linear, positive relation between the two, further embellishing the idea that the actual emissions and the predicted ones are similar, and that the model we constructed is reliable.

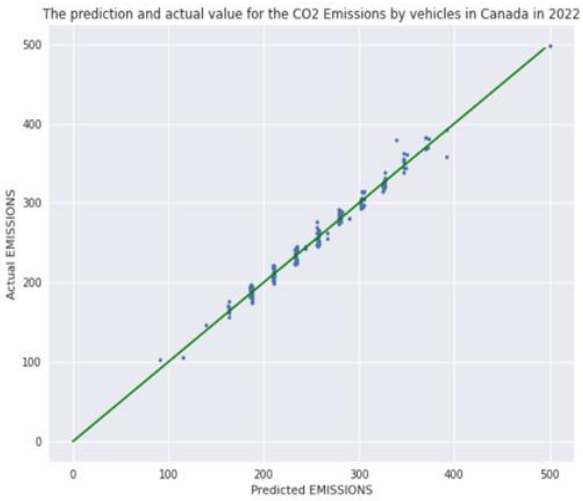


Fig. 2. Scatter plot of the predicted vs the actual values of the CO2 Emissions obtained by regressing the Generalized Linear Model

Finally, the homoscedasticity assumption about the model was tested by a residual plot that graphed the points of the standard residuals obtained and the predicted values. The homoscedasticity is, in broad terms, the constant variance in the errors of the obtained model [8], which is what is to be desired for our analysis. Graphically, in the generated plot, the more randomly distributed along the regression line the points are, the more probable the model is to be homoscedastic and the data to be linear, an assumption that is applicable for our generated residual plot.

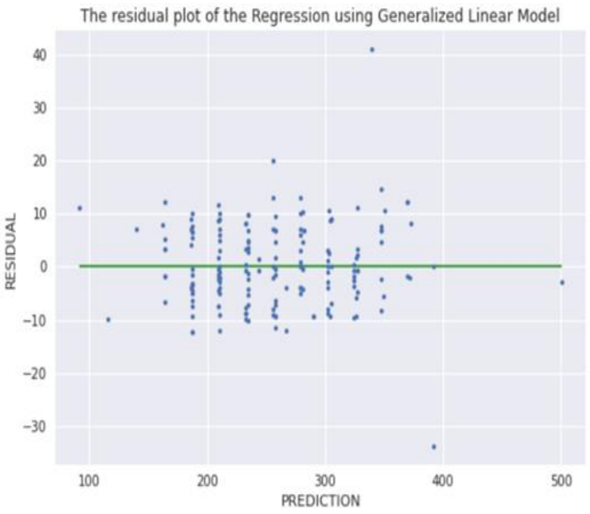


Fig. 3. Residual scatter plot of the CO2 Emissions regression model

2.2. K-means – Hierarchical Cluster Model

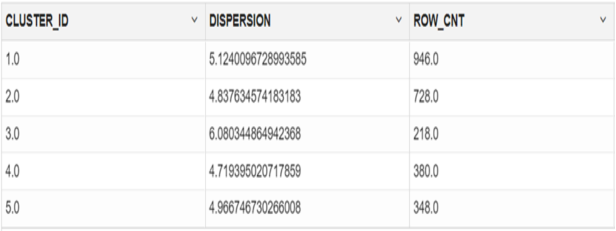
A cluster model is one that divides an unlabeled dataset into different groups based on certain characteristics and maps each observation to a certain group, or cluster, according to specific information subtracted from it. The K-means algorithm is a type of predictive machine learning algorithm that learns a "clustering model from training data that can be subsequently used to assign new (testing) data to clusters" [7]. The said algorithm is a hierarchical, distance-based one, that uses existing observations to predict groups in which future observations will be placed, according to certain variables. For this analysis we considered the groups generated by the Fuel Consumption and the Engine Size variables, considering that both have a positive coefficient in the linear model, thus both positively influencing the CO2 Emissions. This means that the further right a cluster is, the more likely it is to generate higher emissions for vehicles falling within that cluster. For this, we broke down the dataset into a smaller data frame for the predicting set of observations, with the Fuel Consumption and the Engine Size data.

The OML4Py API provides the oml.km class that takes into account a variety of settings with the purpose of generating the clusters. In the context of this analysis, we are using 20 iterations with 3 clusters to be generated, considering that there is a fairly small set of data, with only 946 observations.

```
emissions_df = oml.sync(table =
'EMISSIONS_2022')
training_df, testing_df =
emissions_df.split(ratio = (0.8,0.2))
training_y =
training_df[['FUEL_CONSUMPTION',
'ENGINE_SIZE']]
testing_y =
testing_df[['FUEL_CONSUMPTION',
'ENGINE_SIZE']]
```

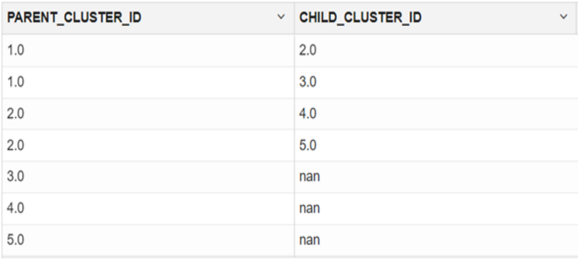
```
setting = {'KMNS_ITERATIONS': 20}
km_mod = oml.km(n_clusters = 3,
**setting).fit(training_df,
model_name="EMISSIONS_KMEANS_CLUSTER_M
ODEL")
```

The generated model can be further analyzed by invoking the clusters and its taxonomy, displayed as a data frame. While the taxonomy table shows the hierarchy of the child clusters in relation to their parents, the clusters table presents various attributes regarding the model, like the dispersion of clusters and the number of observations in each of them.



CLUSTER_ID	DISPERSION	ROW_CNT
1.0	5.1240096728993585	946.0
2.0	4.837634574183183	728.0
3.0	6.080344864942368	218.0
4.0	4.719395020717859	380.0
5.0	4.966746730266008	348.0

Fig. 4. K-Means generated Clusters' Table



PARENT_CLUSTER_ID	CHILD_CLUSTER_ID
1.0	2.0
1.0	3.0
2.0	4.0
2.0	5.0
3.0	nan
4.0	nan
5.0	nan

Fig. 5. K-Means generated Clusters' Taxonomy Table

Here, it can be remarked that the dispersion value, for example, is a variable that measures how spread the observations are inside the cluster, which means that, in our analysis, Cluster 3 is more dispersed than Cluster 4, which is the most compact of the three clusters generated. These certain clusters are observed in the taxonomy table, displayed as having no children, because they are the leaves of the hierarchical cluster tree.

The same conclusion is drawn by looking at the ROW_CNT column of the cluster table, which displays the smallest row count for the 3rd cluster, and the highest for the 4th. However, it is important to be considered that all the three clusters contain similar numbers of observations, which means that the clustering process was a reliable and precise one.

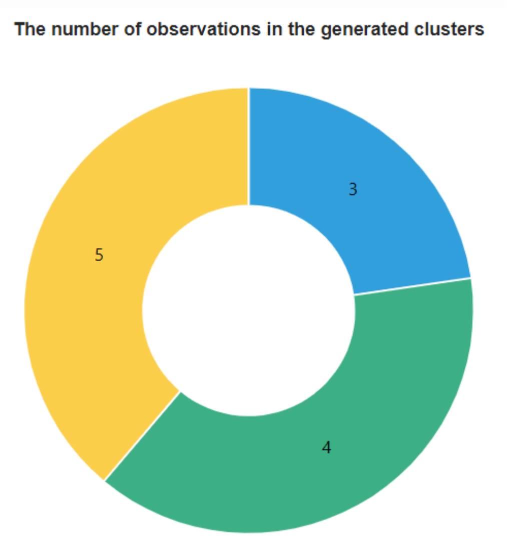


Fig. 6. Donut Chart displaying the size of each cluster

Taking this into consideration, we can say that there is a higher probability for a vehicle to be placed in the 4th cluster than in the 3rd. This is important when observing the predicted variables of the data to be tested, which shows in which cluster is more probable for an observation to fail, according to its fuel consumption and its engine size value, a prediction which can be visualized as a scatter plot.

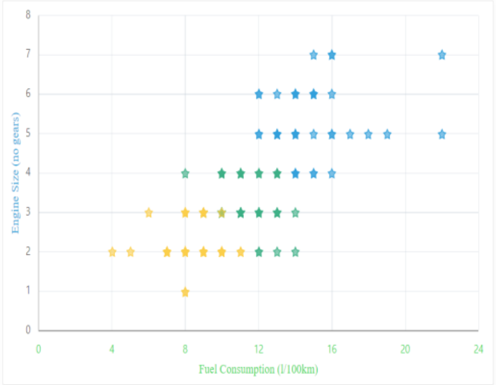


Fig. 7. Scatter Plot of the predicted clusters generated by the k-means algorithm

```
setting = {'svms_kernel_function'
:'dbms_data_mining.svms_linear',
'odms_partition_columns':'FUEL_TYPE'}
svm_mod = oml.svm("regression",
**setting)

svm_mod.fit(training_x, training_y ,
model_name =
'EMISSIONS_SVM_PARTITIONED_REGRESSION_
MODEL')
prediction =
svm_mod.predict(testing_data,
testing_data)
```

2.3. Regression – Support Vector Machine Model

In the previous regression analysis, the model was considered in its entirety, as it built a linear equation that showed the dependence of Emissions on the vehicle’s attributes. Oracle’s Machine Learning API offers, however, another class to solve a regression model, the Support Vector Machine. This model can be used not only for regression, but also for classification, based on decision planes, and for anomaly detection, which can be used as a security tool. However, SVM regression can be used to better predict an outcome or the value of a dependent variable, taking into account different categories of objects, thus providing a more accurate prediction. For example, our model can be partitioned according to the number of cylinders, the make of the vehicle or even the vehicle class, but for this analysis we used the fuel type.

We considered the observations to be X – regular gasoline, Z – premium gasoline, D – diesel, E – ethanol and N – natural gas., therefore generating five different partitions. According to this, we again divided the model into 80% for the training and 20% for the testing and we created a separate series with the values of the CO2_EMISSIONS column, which we dropped from the training data frame. The settings that the oml.svm class takes into consideration are the kernel function to be applied, which by default is linear, and the column on which the partition should be done, which in our case is the FUEL_TYPE column. As in the previous algorithms, we built the model, based on the training data, using the fit() function, and we generated a prediction for the testing data with the predict() function.

Besides showing a comparison between the actual value and the predicted value of emissions for each observation, the model provides the ability to generate global statistics for each partition, such as whether the data are converged or how many observations fit in each partition. We can remark consequently that there is no vehicle using natural gas and that most of the vehicles observe use either regular or premium gasoline, with ethanol being the least used fuel.

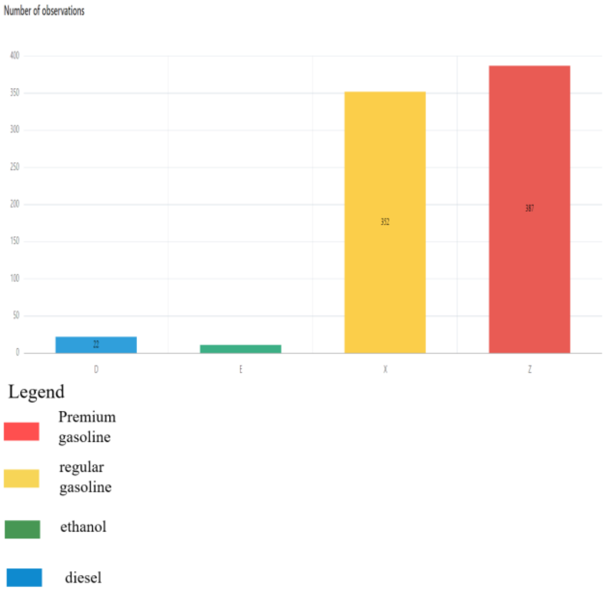


Fig. 8. Number of observations for each fuel type generated by the SVM Model

Furthermore, using the topN_attrs parameter, we can generate prediction details for each vehicle, and understand which variable is the most important in the final result and which has the least involvement. This is the classification capability of the Support Vector Machine model, and we can therefore observe that the fuel consumption is the most important attribute of a vehicle, regarding the volume of emissions, while the transmission is the least important, having the weight 5 on most of the observations. According to this classification, we can predict that, when buying a vehicle, it is more important to consider its fuel consumption than its transmission or its engine size.

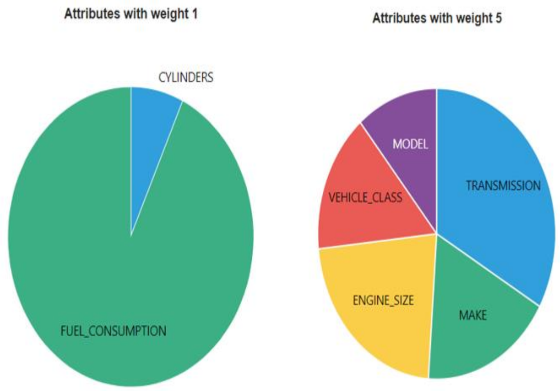


Fig. 9. Number of observations for each fuel type generated by the SVM Model

The pie charts that were built in APEX are a graphical proof that the most important aspect of a vehicle, when it comes to CO2 Emissions is its fuel consumption, which might be intuitive to some degree.

3 Conclusions and comparison between the used algorithms

After applying all these algorithms to our data, we can analyze the main similarities and dissimilarities between them and decide which one is the best to be used in the situation presented by our data.

Firstly, the most obvious comparison that can be remarked is the one between the two regression methods, the one applied on the Generalized Linear Model, and the one applied on the Support Vector Machine Model. While the former is more effective on a set of simple data, as it does not provide information about the importance of the coefficients, the SVM gives more appropriate results, classified by attribute importance, consequently being more powerful as an analysis tool. Not only that, but SVM allows the modelling of non-linear relationships, as they can discover linear separation between data. However, GLM is less memory and time consuming and may be more suited for big masses of data, while SVM heavily uses the computer’s assets and it is usually considered more of a classification algorithm, rather than a regression one.

Second, the k-means algorithm stands out from the other two, as it creates a cluster analysis, rather than a regression analysis. As we saw in the algorithm walkthrough, using the k-means algorithms requires building a model against two dependent variables, rather than only one, as in the case of regression. Moreover, considering that the Cluster Model did not predict exact values for the CO2 emissions of each vehicle, but rather placed said vehicles in certain clusters and groups, it can be considered the least reliable for our analysis. Furthermore, the clustering algorithm considers that an analysis of attribute importance was already conducted and that the clusters are formed against the two most important attributes. Consequently, this algorithm requires on more step than the other two, so it also becomes again more memory and time consuming. Not only this, but it has been observed that the k-means algorithm generates significantly different results, depending on how many groups have been used, which further proves that this algorithm is the least reliable. Even if for our data, the clustering algorithm was not the most suited, when it comes to grouping data, or discovering more general patterns of behavior in data, this algorithm comes in handier.

That being so, in the particular case of our data, we can observe that the best machine learning technique would be the Regression and Classification by Support Vector Model, as it is not applied to a big volume of data, and it provides more information about the fitness of the model and the final prediction. Besides that, we can remark that the predicted values and the actual values are similar for both regression techniques, so we can safely assume that both regression algorithms worked as expected. As a final decision, however, we can safely say that the first algorithm employed, the regression on the linear model, is the best for our data and our desired results. This is because our initial purpose was not classification of data, but rather prediction of actual results, on one hand, and because we already knew that the relationship between actual data and predicted data was linear, so we did not need a multidimensional analysis, as support-vector model is.

To conclude, after conducting this machine learning analysis on the carbon dioxide emissions of different vehicles, using the OML4Py software, we can confidently say that there is a strong relationship between certain characteristics of light-duty cars and their environmental impact. By separating datasets into training and testing data and using machine learning algorithms like regression and k-means on various models, from Generalized Linear Model to Support Vector Machine or Clustered Model, we were successful in predicting the future values for CO2 Emission, based on vehicle attributes as fuel type, consumption, or engine size. Furthermore, we were able to predict in which cluster, or group, a vehicle could be placed, based on how eco-friendly that certain vehicle is. This makes us hopeful that there is a plethora of opportunities for further work in the field, and that machine learning offers the possibility of designing, manufacturing, and buying the most environmentally friendly vehicle and, consequently, decelerating the climate change process and making a better world for the future inhabitants of Earth.

References

[1] European Parliament , "CO2 emissions from cars: facts and figures," 2016.

[2] IEA, "Transport sector CO2 emissions by mode in the Sustainable Development Scenario, 2000-2030," Jan 2022. [Online]. Available: https://www.iea.org/data-and-statistics/charts/transport-sector-co2-emissions-by-mode-in-the-sustainable-development-scenario-2000-2030. [Accessed April 2022].

[3] Government of Canada, "Natural Resources Canada," 2022. [Online]. Available: https://open.canada.ca/data/en/dataset/98f1a129-f628-4ce4-b24d-6f16bf24dd64. [Accessed April 2022].

[4] Oracle, "What is an Autonomous Database," [Online]. Available: https://www.oracle.com/database/what-is-autonomous-database/ . [Accessed April 2022].

[5] D. F. G. V. Igor Milovanović, "Python Data Visualization Cookbook," Birmingham, Packt Publishing Birmingham, 2015, p. 34.

[6] M. Hazarika, "Oracle Cloud. Using Oracle Machine Learning on Autonomous Database," Oracle, 2017.

[7] P. Flach, Machine Learning. The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2012.

[8] J. Fox, *Applied Regression Analysis and Generalized Linear Models*, California: SAGE Publications, 2016.

Miruna Teleașă is a recent graduate of Bucharest Academy of Economic Studies, having a Bachelor’s degree in Economic Informatics. She has also finished a Bachelor of International Relations at the University of Bucharest and she is currently studying for a Master’s in International Security at Sciences Po University in Paris. She is passionate about ethical technologies and how they can improve and embelish human life and, furthermore, about how Artificial Intelligence can evolve and provide a more safe and secure environment for humans. Consequently, she has researched and developed an APEX application for her Bachelor's thesis, using mostly web and database technologies, with the purpose of facilitating communication between humanitarian organizations and people in need.

Bardici Alexandra Teodora has studied Communication and Emerging Media at SNSPA and at Informatics Economics at Bucharest Academy of Economic Studies. She finished one bachelor thesis this year, at SNSPA, which was called "The Use of Internet Marketing in Small Businesses". This fall she will attend the master program Digital Communication and Innovation from SNSPA. When it comes to IT, she is drawn to the web design aspects, which is why her bachelor thesis will also be headed in this direction. For her, the web technologies blend perfectly with the communication strategies that she has studied, which is why her interests gravitate towards it.