European Car Fleet and CO2 emissions

(Prepared by Shamil Murzin, GyungYoon Park, Marco Gagliano. February 28, 2023)

1. Introduction

Background

Since the 1800s, human activities are affecting climate change, primarily due to burning fossil fuels like coal, oil and gas. Through the last decade humanity realized the importance of it and started to react to the levels of greenhouse emissions. Passenger cars and vans ('light commercial vehicles') are respectively responsible for around 12% and 2.5% of total EU emissions of carbon dioxide (CO2), which is the main greenhouse gas.

The scope of study

This project was chosen to understand the changes in the automotive market in the EU countries from 2010 to 2021, considering the 2015 Paris Agreement. By using the EU car registration dataset, we can analyze the evolution of the vehicles fleet in Europe, study the relationship between the car features and its CO2 emissions, understand whether certain countries/manufacturers produce less carbon-intensive cars and see which countries have started to switch to lower CO2 emission vehicles, by looking at the historical trend over the past 10 years.

2. Related work

There are some available similar projects on the open web that use mostly supervised methods to solve a similar problem of car CO2 emission prediction.

CO2 Emission EDA & Visualization & Machine Learnin | Kaggle:
This study captures the details of how CO2 emissions by a vehicle can vary with the different features. The dataset has been taken from Canada Government official open data website. The author conducted ML algorithms to predict CO2 emission based on fuel type (Ethanol (E85), Regular gasoline, Premium gasoline), engine size, cylinders and fuel consumption. Supervised methods applied: linear regression, Ridge regression, Lasso regression, KNN, SVR, Random Forest. The amount of features and records, which was used is much less, compared to current study and more different algorithms (unsupervised and supervised) were applied in current study.
Predicting CO2 emissions of a vehicle using Regression| Machine Learning python: Study analyzes correlation between CO2 emission with two features: cylinders and engine size. We provide supervised analyses with more features and unsupervised methods to cluster the data.
CO2 emission of cars data analyses: By using Linear regression models the author predicts the amount of CO2 emitted by cars per unit volume and weight. In our study we use more features and various methods tested to predict CO2 emission of cars.

3. Data source

The EU car dataset can be downloaded from: http://co2cars.apps.eea.europa.eu/ (in total 12 csv files). The dataset contains 58 000 000 records of the registered cars during 2010-2021. There are several limitations of this dataset:
1. Not all characteristics are recorded for each vehicle, some are missing.
2. The dataset does not indicate the condition of the car at the present time (for example, whether it is still in use). We only have data when the car was bought.
3. The size of data ~ 58.000.000 records in total.

Features used in this study:
1. Country - in which country the new car was registered (categorical feature)
2. Mk - Manufacture name, which produced the car (categorical feature)
3. Mass, kg - mass of the car (numeric feature)
4. ENEDC, g/km - emission of CO2 in g/km based on test for emission certification (numeric feature)
5. Engine capacity, cm3 - (numeric feature)

6. Ft - fuel type (categorical feature)
7. Fm - Fuel mode (categorical feature)
8. Cn - commercial name of the vehicle model (categorical)

The second set of data which was used in this study is countries boundary geometry for visualization purposes (https://github.com/leakyMirror/map-of-europe/blob/master/TopoJSON/europe.topojson)

4. Feature engineering

All the data was loaded, formatted and analyzed in Python using mainly Numpy, Pandas, Sklearn, Statsmodels and Altair libraries.
Key preprocessing steps:
1) Data availability
2) Feature representativeness in each dataset

Data availability analysis results showed that the available EU vehicles dataset is only complete for 2018, 2019, 2020 and 2021 years (~10-16 mln records per year). For all other datasets we have only samples of data (~250 000 - 600 000 records per year). With regards to the feature representativeness, almost all the datasets are complete, however for the 2021 year ENEDC feature is filled only for 25% rows (Figure-1), more detailed analyses show these records are mostly for low CO2 emission vehicles. Countrywise, Great Britain is not represented in the 2021 dataset, Norway, Iceland and Hungary were not represented in the first 9 datasets. The 2017 dataset differs significantly from the others, 50% of the cars are related to France (Figure-2). We chose to ignore the difference in country representativity in the data.

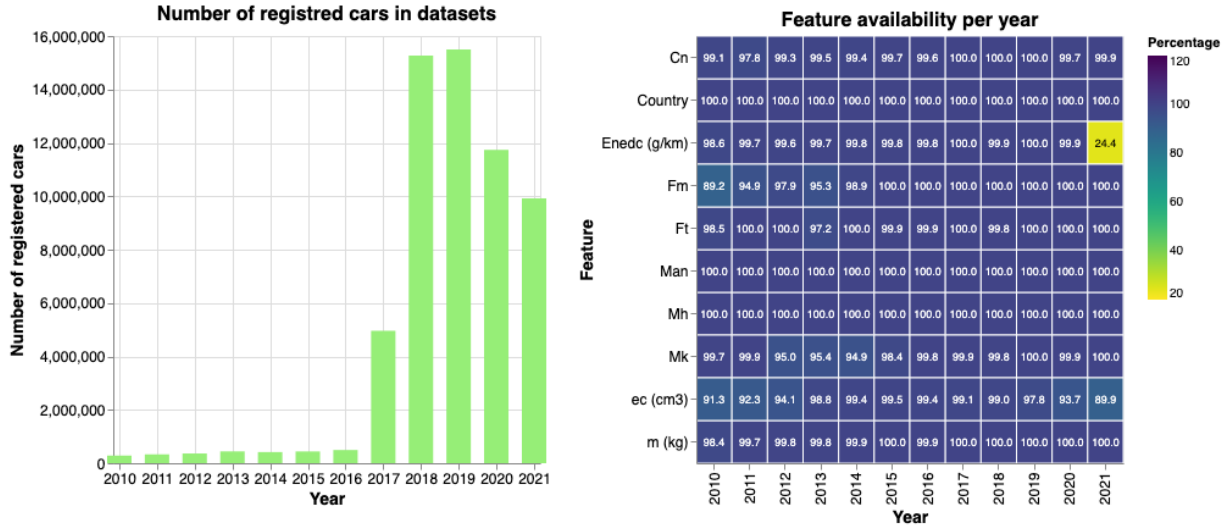


Figure 1. EU light vehicles dataset availability results.

For supervised modeling, our data contains a mix of features, some numerical like the mass, engine capacity, and the year, others categorical. For this latter, to avoid including too many features, which could slow down the training on a large dataset like ours, we decided to use the fuel type, transmission type, and carmaker brand in our analysis. This allowed us to build a model which can capture the main characteristics of a car. For the supervised part, our need was to have unique cars for every row, so we dropped all the duplicates, which allowed us to reduce the data to about 1.8 million rows, from the initial dataset which contained more than 50 million rows. The categorical features labels were first cleaned by removing duplicates and made case-consistent, and then one-hot encoded.

For unsupervised modeling: Vehicle manufacture categorical feature first was cleaned (took time) and then labeled. For the 2021 dataset ENEDC feature we used the results from the supervised learning. To equally represent each year, each year's dataset was sampled by 100 000 samples and then combined to the final dataset with 1 200 000 samples in total. Features used for clustering: vehicle mass (numeric),

engine capacity (numeric), ENEDC (numeric) and vehicle manufacture (categorical -transformed to label). All features were scaled.

Overall, our analyses might be split into four different stages:
EDA analysis: At this stage, we will explore (1) Overall trend of average Co2 emission of a car, and (2) Country comparison of average emissions.
Supervised analysis: At this stage, we intended to fill the missing data (ENEDC feature for 2021 dataset). Specific question: Can we fill missing data based on available records?
Unsupervised analysis: After filling missing ENEDC features in the 2021 dataset, we clustered the car dataset trying to separate the low CO2 emission vehicles from other cars. Specific question: Which country is moving faster towards less emission cars?
Country-focus analysis: At this stage, we focused on four largest EU countries (Germany, France, Italy and Spain) and analyzed their low CO2 emissions vehicles share change from unsupervised learning results to see if they can achieve their goal zero (zero-emission by 2035).

5. EDA analysis:

5-1. Overall trend of average Co2 emission

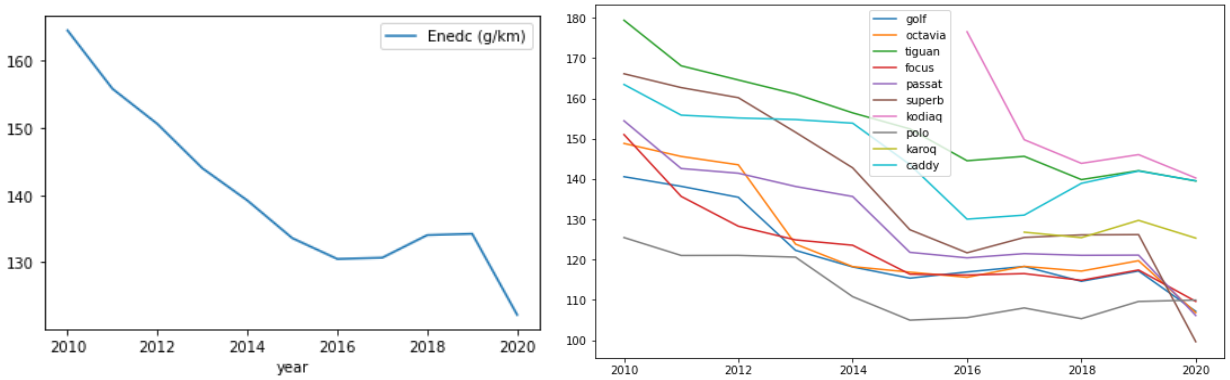


Figure 3. Average Co2 emission of a car and top 10 models

On the left is the average Co2 emission of total cars registered during the 2010~2020 period. On the right is the average emission of the top 10 most registered vehicles. the overall trend of average Co2 Emission can be summarized to 3 stages

(1) 2010 ~ 2015 : rapid decrease

Euro 6 standard was introduced in 2015 september for most new registrations (2014 september for new approvals). In order to meet the regulation, the car manufacturers continued to improve the Co2 emission of each model. The first phase shows the decrease of average Co2 emission of each model due to the effort

(2) 2016 ~ 2018 : stagnation or slight increase

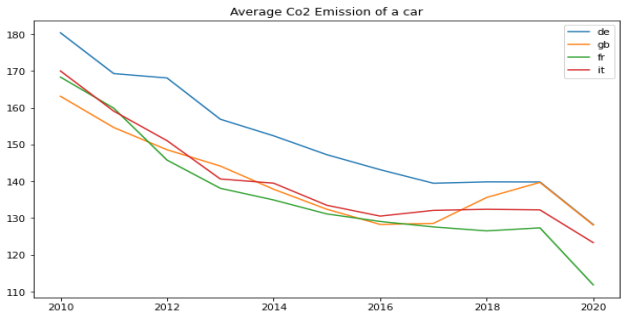
We see stagnation or slight increase of average Co2 Emission of cars. This is mainly due to the change in the ratio of diesel and petrol cars. When we look at the data below, the average Co2 emission of a car continues to decrease over time regardless of its fuel type (there is a slight increase in 2019 but it's negligible). However, there is an increase of the ratio of petrol cars and decrease of diesel cars. Since in this data, petrol cars have higher Co2 emission, the increase of petrol cars causes the increase of average Co2 emission of total cars.
So, why did diesel cars decrease during this period? There may be many reasons such as regulations, but I would like to point out (2-1) the notorious "Dieselgate" happened in late 2015 so many people avoided diesel cars (2-2) gasoline prices were relatively low during this period so people were less burdened to drive petrol cars(petrol is more expensive than diesel in general)

(3) 2019 ~ 2020 : decrease again

The third phase's slight decrease can be attributed to mostly 2 factors.
(3-1) electric cars : by 2020, electric and petrol-electric cars are added to the data. Since these types has 0 or very low Co2 emission, the average Co2 emission of a car decreases.

(3-2) improvement in technology : Aside from the electric cars, existing diesel and petrol models also improve in terms of Co2 emission. For example, the model “Corsa” launched a new version that greatly reduced Co2 emission in late 2019, and other models followed a similar pattern.

5-2. Country Comparison Analysis



de	gb	fr	it
vw group pc	vw group pc	vw group pc	vw group pc
vw-saic	bmw group	psa-opel	bmw group
daimler ag	ford-werke gmbh	bmw group	vw-saic
bmw group	daimler ag	daimler ag	daimler ag
psa-opel	vw-saic	vw-saic	psa-opel
mercedes-benz	tata motors jaguar land rover	bmw	bmw
bmw	general motors	renault	fca-tesla
ford-werke gmbh	bmw	ford-werke gmbh	ford-werke gmbh
ford-volvo	psa-opel	mercedes-benz	mercedes-benz
general motors	ford-volvo	ford-volvo	fca

Figure 4. Average Co2 Emissions of a car, and most popular brands in each country

In this section, we compare and analyze the trends of 4 countries; Germany, Great Britain, France and Italy. (4 countries with most data)

On the left is the average Co2 emission of a car in each country. There are 2 noticeable patterns. First, Germany has higher average Co2 emission compared to other countries. Second, there is a spike in the average Co2 emission in Great Britain around 2018. Both patterns can be explained with one factor, car size. We use engine capacity(‘ec (cm3)’ column) as a proxy for car size. (The correlation between the 2 factors is 0.64) When we compare the average car size of Germany to other countries, Germany’s average car size is consistently larger than the others. We can conclude that because Germans prefer bigger cars, average Co2 emissions are generally higher compared to other countries. Great Britain’s sudden increase in the average Co2 emission can be attributed to the same factor. During this time, Britain’s average car size suddenly increases. This results in the sudden spike in the average Co2 emission of the cars in Britain.

On the right are the top 10 popular brands in each country. While there is overall popularity of the German brands such as VW, daimler and bmw, we also observe “country effect” in each country. For example, psa and renault are popular in France, and fca is popular in Italy.

Another interesting aspect of vehicles are the fuel types. Diesel dominance was reversed in 2019, and petrol is the most dominant fuel in 2020. Also, petrol-electric and electric cars are observed more in the later years. When we look at individual countries, the picture is a little different. Petrol became the dominant fuel type in 2019, and is being used as twice as much as diesel in 2020. In Germany and France, in spite of the gap narrowing over the decade, diesel maintains its slight dominance. On the other hand, diesel is still the dominant fuel type in Italy is used twice as much as petrol in 2020(this is due to the low tax on diesel).

6. Supervised learning analyses

Motivation

Our project aims at developing a model to predict a car CO2 emission given various features and characteristics of the car itself. The need arose during the data exploration phase, where we discovered that many cars in the most recent year of the dataset did not have the Enedc(Co2 Emission) value available. Car makers in Europe are required to certify new cars according to stringent CO2 emissions limits, and a model which is able to predict them based on the car characteristics could be useful in the design phase of a vehicle.

Data Source

The dataset is the same as specified previously, with data cleaned and features engineered as explained in Section 4.

Methods and Evaluation

We built the machine learning algorithm by exploring different model families to select the one that could give the best results. To proceed,the models were fitted using an ML pipeline consisting of a Min-Max Scaler, to scale the features so that all the values are contained in the interval [0, 1], and the model of choice. To avoid bias and get a better evaluation of the models, we conducted a nested K-fold cross validation. The inner fold was used to tune different flavors of the models by changing key parameters. The outer fold allows us to estimate the models performance and compare them. We used a value of 5 for both the outer and inner folds. This means that our training dataset first gets split into a training and test dataset, then the train dataset is split into a train and validation dataset. The model is trained on the five inner training folds and performance tested on the five validation folds to choose the best parameters. Then, the best model is trained on the whole five outer training datasets and tested on the five test datasets to evaluate the performance. We trained all the models on a subset of our data to allow for faster performance, given the number of models to compute both in the inner and outer folds. The models that we used are:

1. Logistic regression: we used the Scikit implementation, and this forms our baseline. We needed a simple model to evaluate the improvement of more complex ones.
2. Linear Regressor: we used the Scikit learn implementation without hyperparameter tuning. This also serves a base model, and as a comparison for Lasso and Ridge regression.
3. Lasso Regressor and Ridge Regressor: we used the Scikit learn implementation. With L1 regularization for lasso and L2 for ridge, these partially solve the overfitting problem that normal linear regression shows.
4. KNN regressor: we used the Scikit implementation, with the K parameter indicating the number of clusters to consider trained in the inner fold. The KNN regressor is easy to understand and can take advantage of the clustering nature of our data.
5. Support Vector Regressor: we used the Scikit implementation, with different kernels evaluated in the inner fold. We used this model as there could be a higher feature space where car characteristics are separable, which could lead to a successful regression prediction. It is also robust to outliers which can help in our case.
6. Passive-Aggressive regressor: we used the Scikit implementation, with different values of C and intercept trained in the inner fold. The passive-aggressive algorithm could be useful with complex data and achieve high performance on big datasets.
7. SGD regressor: we used the Scikit implementation, with different values of penalty, intercept, and alpha trained in the inner fold. The gradient descent is a simple yet effective algorithm that could perform well also with complex and big data.
8. RandomForest Regressor : we used the Scikit implementation. It's a tree based ensemble model that greatly improves the results.
9. MLP regressor: we used the Scikit implementation, with different values for hidden layers and activation functions trained in the inner fold. MLP is a neural network algorithm that could be useful to learn complex features of our data for a good prediction outcome.

For our evaluation metrics, we considered the main ones for a regression analysis, mainly R2, MSE, RMSE, and MAE to compare the models. MAE provides an easy interpretation of the error, while R2 gives a general feeling of how well the model fits the data.

Table 1

	KNN	SVR	PA	SGD	MLP	Simple	Logistic	Lasso	Ridge	RF
R2	0.77	0.75	0.78	0.79	0.82	0.40	0.40	0.44	0.77	0.94
MSE	375.13	409.15	362.01	344.28	298.67	993.83	993.83	841.53	345.57	88.55
RMSE	19.34	20.20	19.01	18.53	17.25	31.50	31.50	29.01	18.59	9.41
MAE	12.83	13.56	13.16	13.18	11.87	21.01	21.01	21.43	13.07	5.05

Table 2

	KNN	SVR	PA	SGD	MLP	Simple	Logistic	Lasso	Ridge	RF
R2	0.019	0.010	0.013	0.010	0.013	0.033	0.033	0.002	0.003	0.001
MSE	44.006	42.448	32.291	32.530	36.446	84.892	84.892	7.281	5.235	2.257
RMSE	1.106	1.063	0.863	0.876	1.052	1.328	1.328	0.125	0.141	0.120
MAE	0.452	0.541	0.537	0.472	0.542	0.705	0.705	0.060	0.037	0.014

Table 1 shows the mean evaluation metrics for all our regressors over the 5 outer folds, while table 2 contains the standard deviations. We can see the bad performance of the simple, lasso, and logistic regression models. The others models all seem to perform pretty well, with the Random Forest being the best one. However, we encountered a lot of issues training the RF model, mostly due to computational capacity. Therefore, the subsequent analysis will be done on the MLP model which showed good performance and faster training time.

Features selection

Unfortunately, the MLP regression model does not have a method for quickly selecting the best features in Scikit. We decided to build our own evaluation method for the feature selection by trying to find the features that lead to the highest improvement in MAE. We trained different models on a combination of features, starting from the simple ones and then ending with the full model. For each of these models, the MAE was calculated. We then grouped the MAE for all the models that contained the feature and plotted it on the box-plot below:

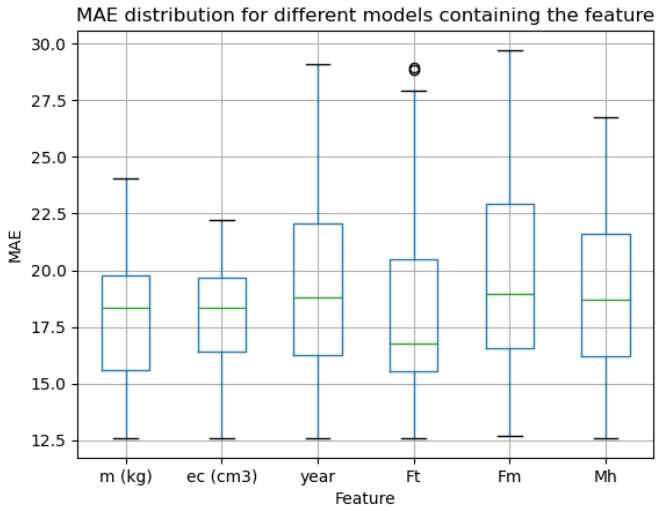
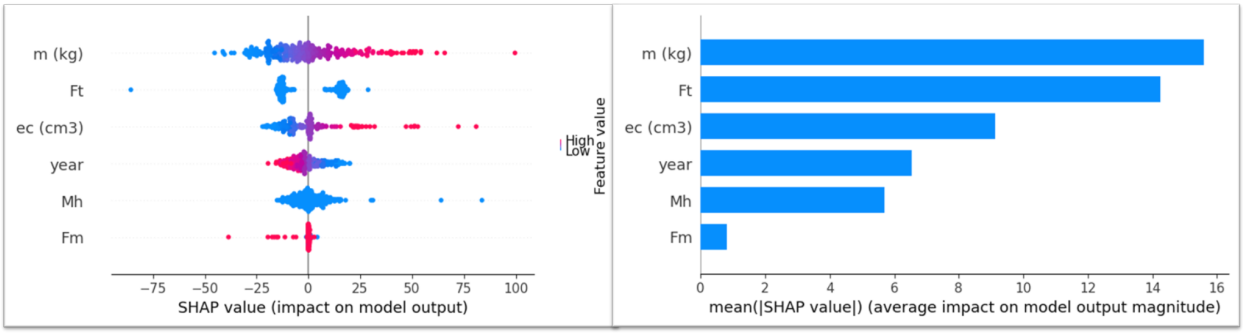


Figure 5

We can see from figure 5 the MAE distribution for each feature (lower is better). We note that on average models containing the fuel type categorical variable (Ft) performed better than the others. This makes sense as the fuel of a car is certainly highly correlated with the CO2 emissions. The other features are very similar in terms of performance. It is worth to note that the engine capacity feature (ec) has the lowest distribution of MAE, which could mean that it is the best one when used alone in the model. Overall, all the features seem to have similar importance, with the year and transmission type (Fm) being the worst performers. The caveat is that some features have a very spread distribution over MAE values, which could indicate poor performance when used alone in the model.

Figure 6



Interestingly, SHAP does agree with these findings. We used it to calculate the feature importance, with the caveat that we summed up all the categorical features variables in order to obtain a single score for each of them. As we can see from figure 6, according to SHAP the most important features of the model are the car mass and fuel type. If mass is high, it has a great impact on the model predictions, pushing them higher. All the other features seem to have a consistent importance for the model, although the transmission type (Fm) feature is the least important one.

Sensitivity analysis

As discussed previously, we performed hyperparameter tuning in our inner fold when training the model. For the MLP, we trained on different values of hidden layers and activation functions.

Figure 7

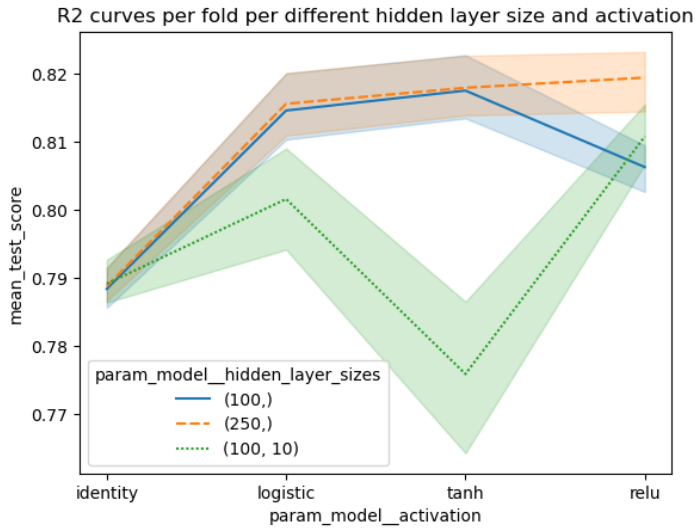
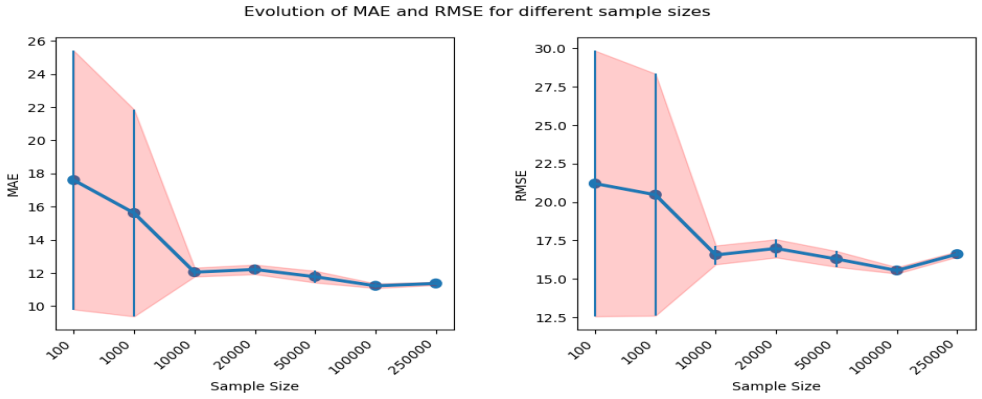


Figure 7 shows the mean R2 values. We can see how the logistic activation function seems to perform better in general, though tanh achieves a higher R2 but is heavily penalized when the number of hidden layers increases. Overall, for simple models with low hidden layers, tanh and logistics perform similarly. ReLu instead seems to perform better with more complex models, as it achieves a low performance with only 100 hidden layers.

Learning curve analysis

The training of our models, specifically the MLP, has been challenging and slow, considering the dimensions of our dataset. In particular, we used a high number of iterations to allow the model to converge, but used the early stopping technique in order to avoid overfitting. As we trained the models on a smaller subset (10,000 observations), it is useful then to perform a learning curve analysis to check if higher sample size could allow for a boost in performance. We can see from figure 8 how the sample size has a big impact on RMSE and MAE values. Small sample sizes like 100 or 1000 create higher error rates, with big variability. At 10,000 observations, the mean RMSE and MAE and the standard deviation drastically change, and they remain similar for higher sample sizes. It is then possible to assume that training the model on 10,000 observations is the best trade off between prediction performance and training time.

Figure 8



Error analysis

It is important to understand what goes wrong with our model when predictions are incorrect. First of all, we visualize the distribution of our errors to check that they are centered at zero and do not have many outliers. This is done in figure 9, and we can see how the errors are consistently concentrated around zero (though slightly skewed toward the positive territory), and no outliers. We need to say, however, that the distribution does not follow a standard normal one, as errors are much more spread out.

Figure 9

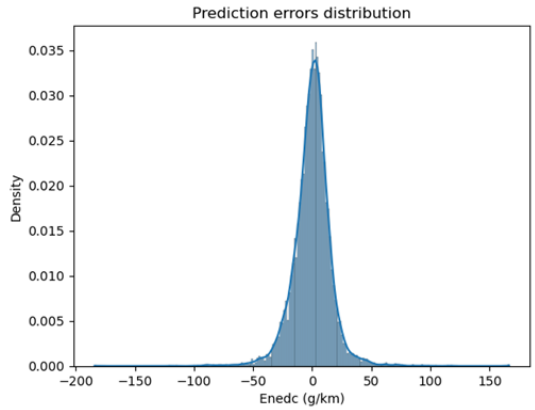
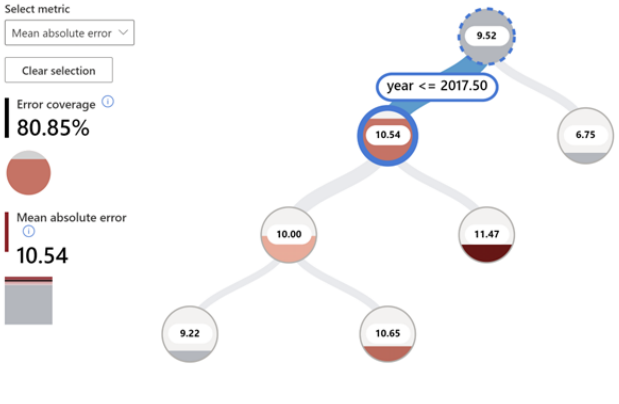


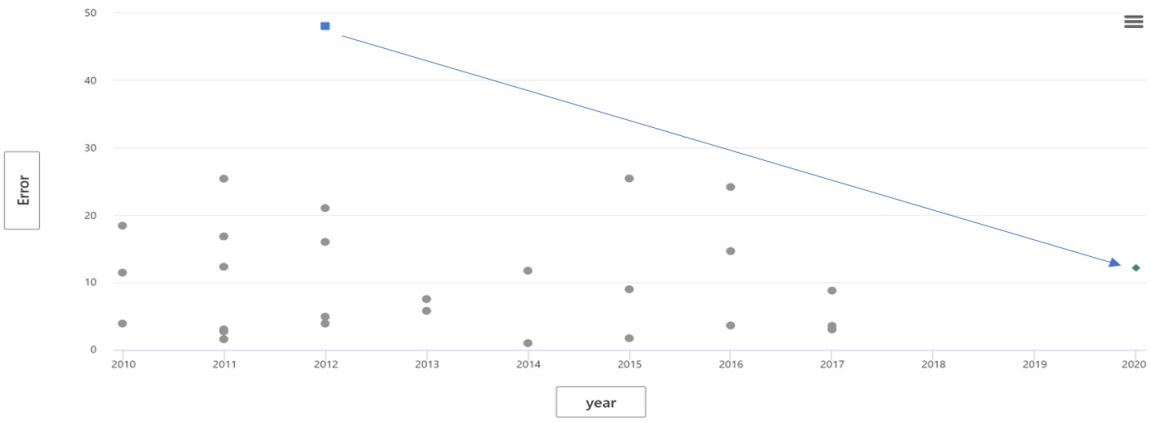
Figure 10



For a more thorough error analysis, we employ the Responsible AI dashboard by Microsoft. This dashboard offers great insights on the predictions made by the model. As a start, we can visualize the main factors impacting errors in our model. Figure 10 shows how the majority of the errors happen for data points where the year is lower than 2017. This makes sense, as our dataset has the majority of the data from 2018 onward. This split allows us to separate already more than 80% of our errors. Other important factors are the engine capacity being higher than 1970 cm3, and the weight of the car being higher than 1400kg. We can then say that our model fails mostly for outliers, or data points not sufficiently represented in our dataset. To solve this, we might look at running the model with a stratified K-fold. This strategy was evaluated at first, but abandoned as our dataset contains many unique values, which would skew our results. Possible solutions to this could be data augmentation, which we did not pursue in this case.

The dashboard also allows us to calculate counterfactuals for data points with big error rates, to see what would happen if the features change. For example, consider the point in figure 11: the prediction error is around 48. The year for the original data point is 2012. By changing it to 2020, we see how the prediction of the model drastically changes, and the error reduces to 12. This can be seen for a lot of our points with big error rates, which confirms our initial hypothesis of year being a sensible issue for our model. We can perform the analysis for other points. One of these has an error of 48 too. We try to reduce the engine capacity feature of it and see that the prediction improves, but not by much (figure 22 in appendix). To drastically reduce it, also in this case we should increase the year.

Figure 11



We can also look at yet another type of error. There is one point with a high mass, but relatively low prediction error (24). If we try to change the fuel type from diesel to petrol, the prediction error spikes to 46 (figure 23 in appendix). This allows us to see how also our categorical features actually have a big impact on our predictions.
To conclude, our model is very sensible to some features, and minor changes can impact the predicted value. Year seems to be one of the main issues, and it could be improved by gathering more data for past years, in order to overcome the prominence in the dataset of the more recent ones.

7. Unsupervised learning analyses

For the clustering, we focused on three methods: K-means, DB-scan and Spectral Clustering. Due to the non-hierarchical nature of the dataset (each manufacturer produces several line models) we did not apply the agglomerative-hierarchical method. DB-scan was chosen based on the assumption that low CO2 vehicles data points will be close to each other on the ENEDC feature (CO2 emission g/km). K-means approach was applied due to the popularity and efficiency. The Spectral Clustering approach was tested with the aim to learn a new method. Below are brief descriptions and results.

1. K-means clustering

The K-means clustering method is an unsupervised machine learning technique used to identify clusters of data objects in a dataset. It attempts to group similar items into the clusters, assigning each point to one specific cluster (a non-soft version was used). Each cluster is associated with one centroid or center point. Each point is assigned to the cluster with the closest centroid based on the distance in the space (in our case we used Euclidean distance). The K parameter is the number of clusters, which needs to be specified in advance. Then the algorithm assigns initial centroids, which can be chosen completely at random and assign with cluster label all points to the nearest center of the cluster in space. After this step, the algorithm will recalculate the centroid of each computed cluster. It repeats these steps, recalculating the centroids until it converges to a stable solution. The parameter which is responsible for the number of initiations needs to be specified in advance. There might be a case, when the number of initiations is not enough to converge to a stable clustering solution.
One of the most difficult tasks in this clustering algorithm is choosing the right k values. There are several methods, which help us to choose the number of clusters.

Sum of the squared error (SSE): The SSE is defined as the sum of the squared Euclidean distances of each point to its closest centroid. This is a measure of error, the objective of k-means is to minimize it.
Silhouette score: measures cluster cohesion (similarity of data point to its own cluster) and separation (to other clusters). It quantifies how well a data point fits into its assigned cluster based on two factors: 1) How close the data point is to other points in the cluster; 2) How far away the data point is from points in other clusters. The Value range is between -1 and 1. Larger numbers indicate that samples are closer to their clusters than to other clusters.
Calinski-Harabasz score (Variance ratio): is a ratio of the sum of the inter-cluster sum of squared distances and the sum of the intra-cluster sum of squared distances for all clusters. A high score means better clustering since observations in each cluster are closer together (denser), while clusters themselves are further away from each other (well separated).
Davies-Bouldin score: is the similarity between clusters. The lower Davies-Bouldin index is (lower similarity between clusters), the better the clusters are separated.

As the dataset contains 1.2 mln rows we took a sample of 50000 records to test for choosing the right parameters. Figure 12 is shown test results for cluster values ranging from 2 to 15. Silhouette and Davies-Bouldin plots indicated that 8 is the optimal number for clustering. Due to the large dataset, we assigned the number of initiations to 500, as the low number gave poor results (not enough to converge to a stable solution). We tested K-means outcome for 1200000 samples 30 times, and each test gave very similar (same) results. We used these 30 outcomes for the further forecasting exercise (see below). K-means cluster results are shown on Figures 13-14. Overall, northern Europe is moving ahead compared to other parts. Norway had some ~26% low CO2 emission cars (EV and hybrid vehicles) in 2020, the largest number in Europe. 2021 numbers prediction is less accurate due to the supervised

method outcome (correlation seems needs to be done for each country individually). Best-Selling Low CO2 Car Manufacturers in Europe: BMW, Mercedes, Volvo, Volkswagen, Audi.

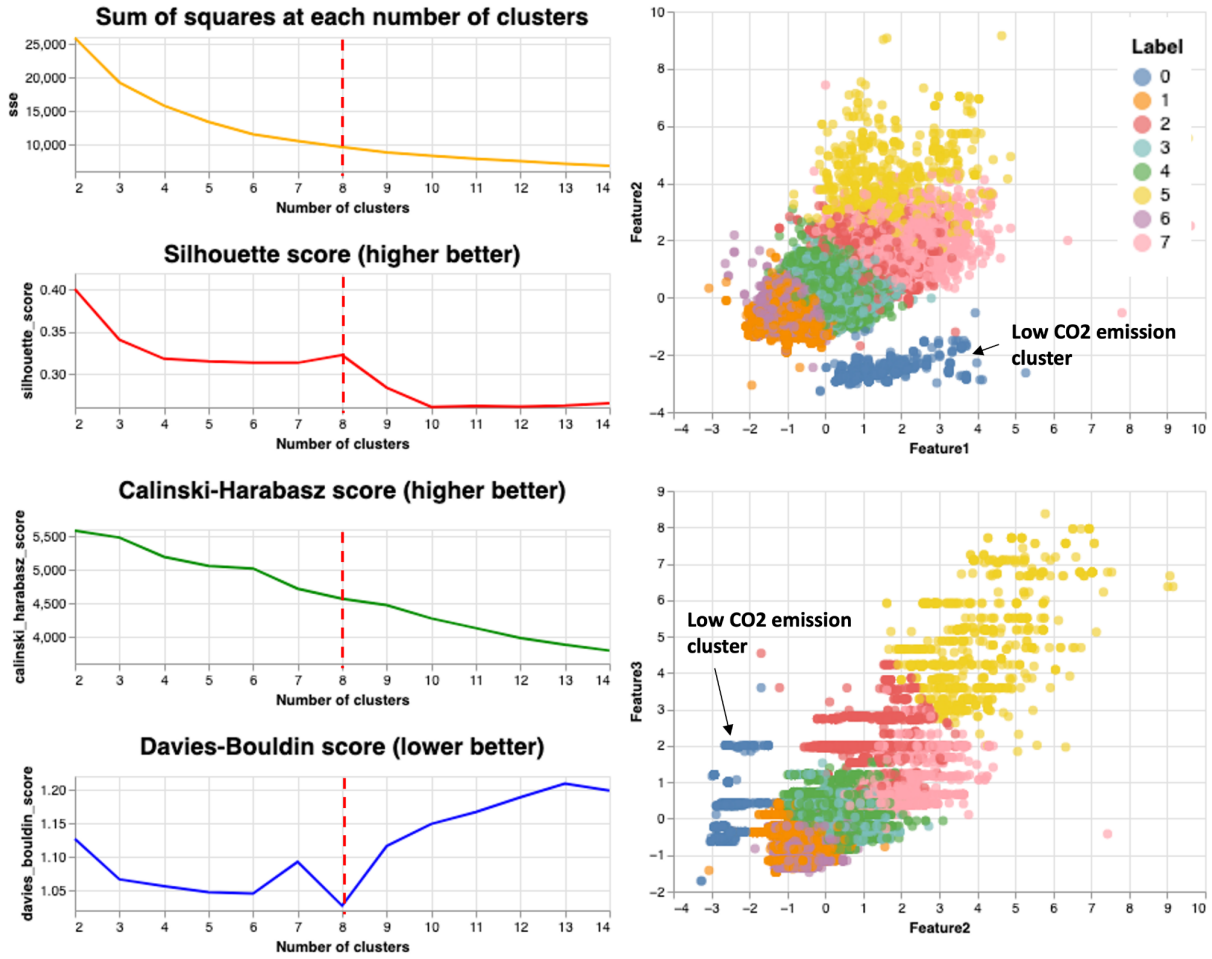


Figure 12. K-means clustering results

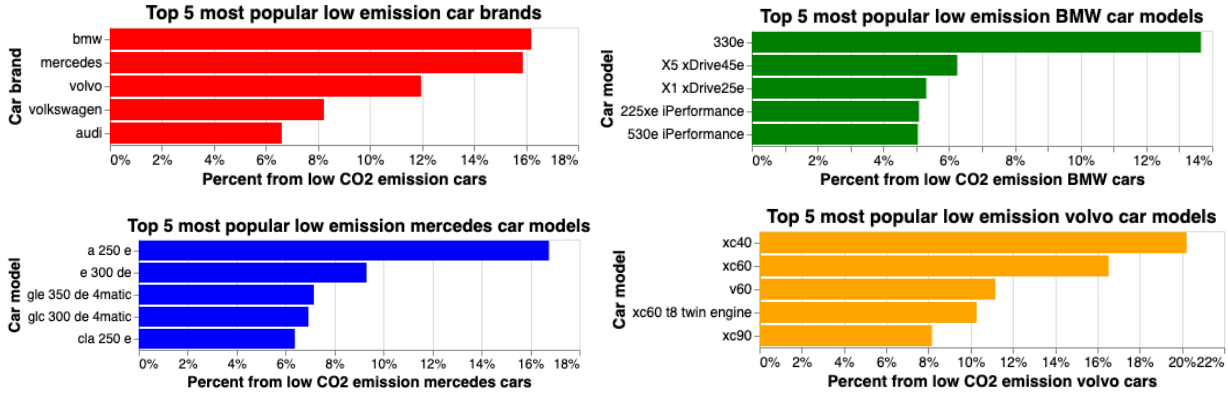


Figure 13 Most popular low CO2 emission vehicles brands and models in EU 2018-2021.

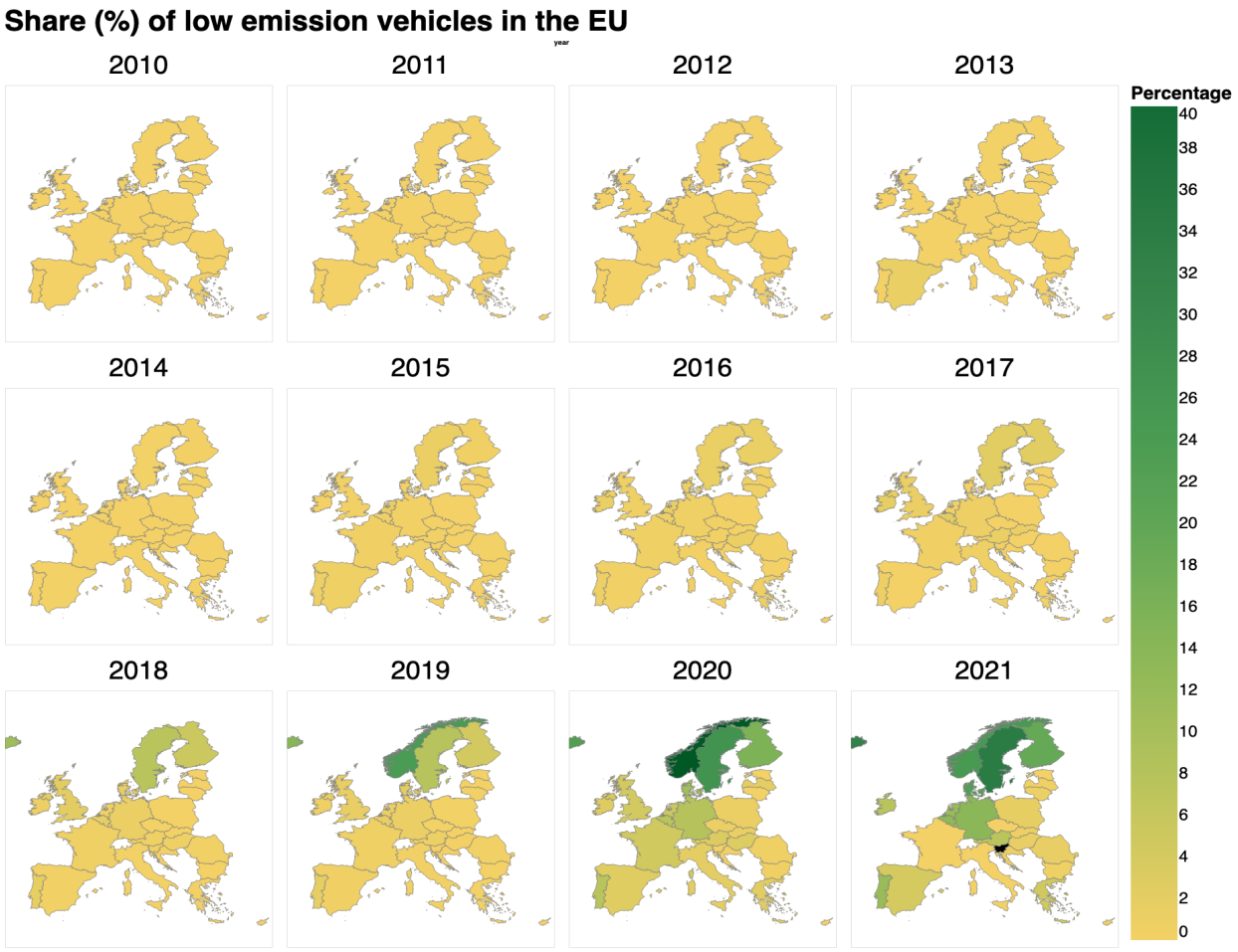


Figure 14. Low CO2 emission vehicles share change in EU during 2010-2021.

2. DB-scan clustering

The basic idea behind DBSCAN is that clusters are regions in data space that are more densely populated with data points, and separated by regions that are empty, or much less densely populated. There are three main steps to the DBSCAN algorithm. Step1: All points that lie in a more dense region are called core samples and controlled by two main parameters: minimum samples and epsilon. For a data point, if the number of other data points that lie within a distance of epsilon from it, is more than minimum samples, then that data point is labeled as a core sample. All samples that are considered closely connected neighbors for the core points are put into the same cluster. Step2: Points that are within a distance of epsilon from the core points, but don't have enough close neighbors (less than minimum samples) to be considered core points themselves, are called boundary points. Step3: Any non-core points that are not close enough to any cluster are labeled as outliers. Two main parameters need to be specified in advance to DB-scan clustering: epsilon and minimum samples, no requirement to specify the number of clusters. We tried to implement DB-scan on our data (computationally expensive - no errors only on data with less than 0.2 mln data records on a local machine), however with no good results as K-means and spectral clustering. We applied 2 available techniques to choose epsilon and minimum samples (on 200 000 sample, as changing the sample of your data impacts the density of it):

1. Calculating the distance to nearest k (we tried the range 1-100) neighbor for all points, then to find the median of it and multiplied to (we tried the range 1.5 - 3.5) and use this value as epsilon. And the number of minimum samples is the number k to the nearest neighbor
2. Calculating the distance to nearest k (same as above) neighbor for all points, then plot the sorted distances and choose as epsilon the value close to elbow (Sefidian, A. M.)

Both methods could not combine two clouds of low-CO2 emission vehicles into one cluster, as the distance between both of them is larger compared to the distance to the other cars (Figure 25 in Appendix 1).

3. Spectral clustering

The Spectral clustering can be reformulated using the similarity graph: we want to find a partition of the graph such that the edges between different groups have very low weights (which means that points in different clusters are dissimilar from each other) and the edges within a group have high weights (which means that points within the same cluster are similar to each other). The general approach is to construct an affinity matrix. An affinity matrix is like an adjacency matrix, except the value for a pair of points expresses how similar those points are to each other. If pairs of points are very dissimilar then the affinity is 0. If the points are identical, then the affinity might be 1. Then the Graph Laplacian is created by subtracting the affinity matrix from the degree matrix (degree value on diagonal). This matrix has the degree for each point on diagonal and negative similarity with other points, it has special properties (eigenvalues and eigenvectors), which helps to understand the number of clusters (from eigenvalues) and cluster labels (using eigenvectors). Input parameters: Number of clusters (default is 8), Number of the k-means algorithm initiations with different centroid seeds (default is 10). Affinity: how to construct the affinity matrix, default is using a radial basis function kernel. We applied spectral clustering on three features: 'mass', 'Enedc' and 'engine capacity'. We used default parameters, except the number of clusters = 7. Clustering results are shown on figures 15 and 26 in Appendix 1.

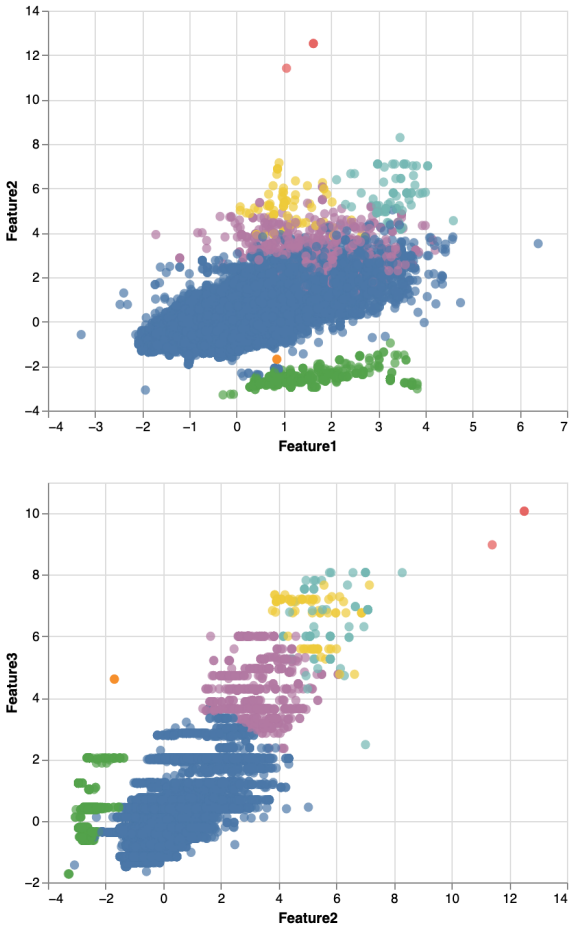


Figure 15 Spectral clustering results.

Results from K-means and Spectral clustering look similar. For Spectral clustering we tested the outcome on 12 different samples. First 10 were very stable and showed clear separation. The last 2 results showed some noise. This might be related to the sample size (50000).

8. Country-focus analysis

Time series object forecasting was conducted using the Auto Regressive Integrated Moving Average (ARIMA) model, one of the most popular and widely used statistical methods for time-series forecasting. Auto Regressive (AR) regression model is built on top of the autocorrelation concept, where the dependent variable depends on the past (p) values of itself. The I part converts the non-stationarity time-series data to a stationary one. Moving average (MA) part regresses with residuals of the past observations, the order of the error lag is denoted as q. One of the requirements for AR and MA that we need to use as an input stationary time series object (constant mean, variance, no covariance).

We applied forecasting on average values from 30 K-means clustering realizations for four countries: Germany, France, Italy and Spain (Step1). With advice from instructor, time series object was upscaled from yearly to monthly resolution (Step2) by incorporating monthly seasonal sales share from recent 2021 EU market analyses (https://www.statista.com/statistics/1104622/monthly-car-registrations-europe/). To make timeseries object stationary we used log return values (Step 3 & 4). The monthly sales share was assumed the same for previous years. With this step we incorporated strong seasonal correlation (12-24-36 months). You can see this seasonal effect on ACF and PACF plots (Step5). To take this into

account SARIMA (seasonal ARIMA) method was applied for country-level analyses (Step6). Overall the workflow key steps are shown on Figure 16. Despite the 2021 values of low CO2 emission vehicles predictions from previous analyses were not very accurate, we used them for model calibration.
To obtain the best p,q for non-seasonal and P,Q for seasonal parts we used the 'auto arima' function (input is a range for the values of p,q, P,Q). The output of the function is the model with the least AIC (less is better) (Introduction to Time Series Forecasting, 2021). With these parameters we forecast low CO2 vehicles monthly share per country and recalculated it back to a year basis. These analyses were conducted for four largest countries in the EU (Germany, France, Italy and Spain) separately. There are different methods to evaluate the performance of the ARIMA model results: RMSE, AIC and model diagnostic diagrams. If the errors are normally distributed and are uncorrelated to each other, then the model is good (Figure 17)(Duke university). In our case, the residuals are normally distributed, but there is a large residual at 2020, which may explain the overestimation and the sharp forecasted increase. As a next step we calculated the total sum of the monthly values for each year to see when all new sales will be the low CO2. Based on results, the largest countries will switch to low CO2 vehicles sales by 2030, which makes it possible to reach the 2035 target of zero emission car sales (Figure 18).

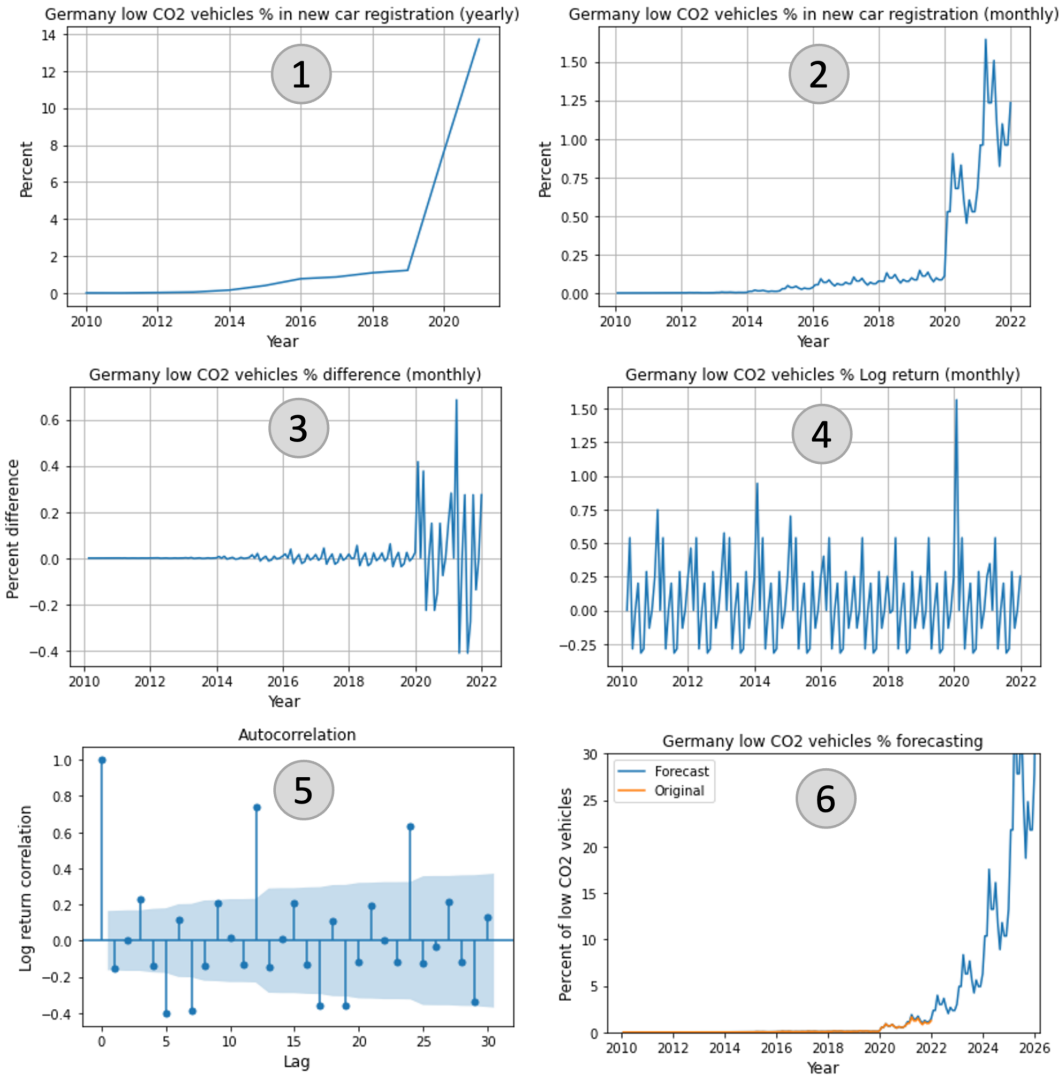


Figure 16 Low CO2 vehicles share in the new car sales forecasting workflow.

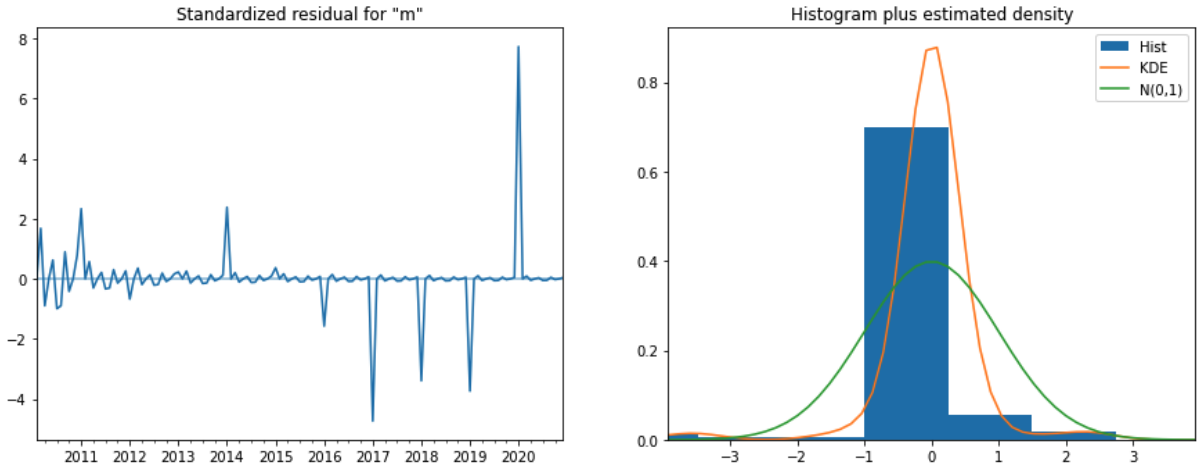


Figure 17 Model diagnostic diagrams for Germany forecasting low CO2 emission vehicles share in new car registrations.

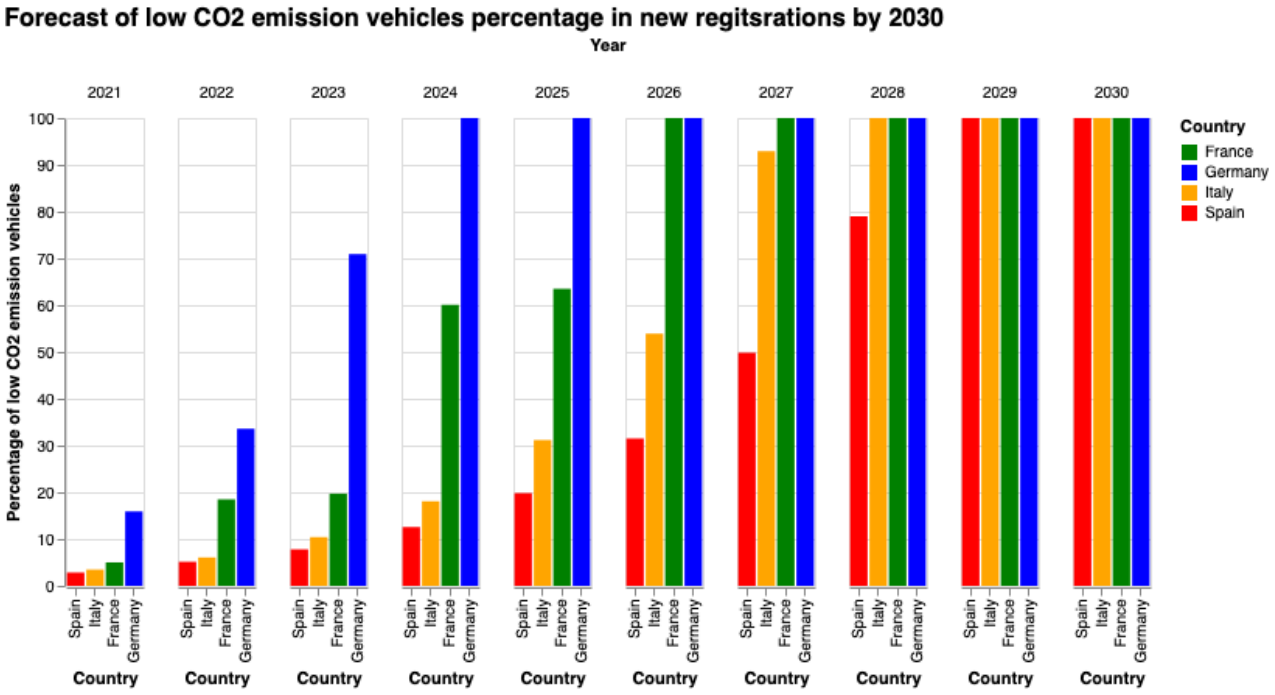


Figure 18 Forecast of low CO2 emission vehicles share change in new registrations for Germany, France, Italy and Spain by 2030

9. Discussion

In our supervised learning analysis, we learnt how the CO2 emissions are related to many features of a car. We were surprised by how relevant all the features were, and it goes to show how complex the CO2 emission problem could be. The main challenges were the training times for each model, which took away a long chunk of our project. An improvement would be to rely more on cloud computing and optimizing the sample size even more, to reduce training time. The other challenge was the skewness of data availability toward later years. This created issues with having a good representation of all the years in our dataset.

A possible solution could be to sample data from each year in equal amounts, although it would mean throwing away many observations from more recent years. Another possibility that we explored is

stratification when performing K-Folds, but also there are challenges due to low availability of some features. In the future, data augmentation techniques could be employed to solve this issue.
In our unsupervised part we learned three clustering methods: K-means, DB-scan and Spectral clustering. Each method has its advantages and disadvantages. As the dataset is large, computation took time. For K-means clustering 1.2 mln rows clustering was relatively smooth, computation time depends on number of initiations. Different scores might show different optimal numbers of clusters. For DB-scan, key learning was that the sample size for testing and final dataset should be the same, since when we change the sample size, we change the density of datapoints, and the optimal epsilon and number of samples from testing will not be relevant then. From a computational point of view, in our case DB-scan could handle only 0.2 mln data points on a local machine. For spectral clustering, key learning was that it uses eigenvectors and eigenvalues, which makes it computationally very expensive. The same machine could only process 50000 data points, but with good results.

10. Ethical Considerations

There are no particular ethical concerns for our supervised learning analysis. Our data is available per country and per car registration, so it would be very difficult to suspect privacy concerns or problems. One potential concern for this analysis is if it were used by a regulatory body to certify vehicles emissions without an appropriate road test. In this case, manufacturers could learn about the model's weakness and use it to declare lower emissions than actual. It is a very remote possibility though.
Some countries and manufacturers pay more attention to reducing emissions with anticipation that the ESG investment and reputation will play an increasing role in the future. Thus, small companies and countries with a lower economy begin to lose competition, because they will not be able to compete with such giants of car production or countries with strong economies, as result minimizing the competitiveness of the low CO2 emission car market. With regards to this study:

- Some countries's efforts towards low CO2 emission vehicles might not be well visualized on the results maps (unsupervised methods), due to the country size or missing/incomplete data
- Time projections of low CO2 emission vehicles share for countries can be overestimated, so a further recommendation is to use data with a wider time range.

11. Statement of Work

Data availability analyses: **Shamil Murzin**
EDA analysis: **GyungYoon Park**
Data preparation for supervised learning: **Marco Gagliano**
Supervised learning methods: **Marco Gagliano, GyungYoon Park**
Data preparation for unsupervised learning: **Shamil Murzin**
Unsupervised learning methods: **Shamil Murzin**
Time series forecasting: **Shamil Murzin**
Report preparation: **Shamil Murzin, GyungYoon Park, Marco Gagliano**

12. References

1. Sefidian, A. M. *How to determine epsilon and MinPts parameters of DBSCAN clustering*. Sefidian Academy. http://www.sefidian.com/2020/12/18/how-to-determine-epsilon-and-minpts-parameters-of-dbscan-clustering/
2. (2019, February 21). *Spectral Clustering*. Towards Data Science. https://towardsdatascience.com/spectral-clustering-aba2640c0d5b
3. (2021, July 30). *Introduction to Time Series Forecasting — Part 2 (ARIMA Models)*. https://towardsdatascience.com/introduction-to-time-series-forecasting-part-2-arima-models-9f47bf0f476b
4. Duke university. *ARIMA models for time series forecasting*. https://people.duke.edu/~rnau/411arim.htm

13. Appendix 1

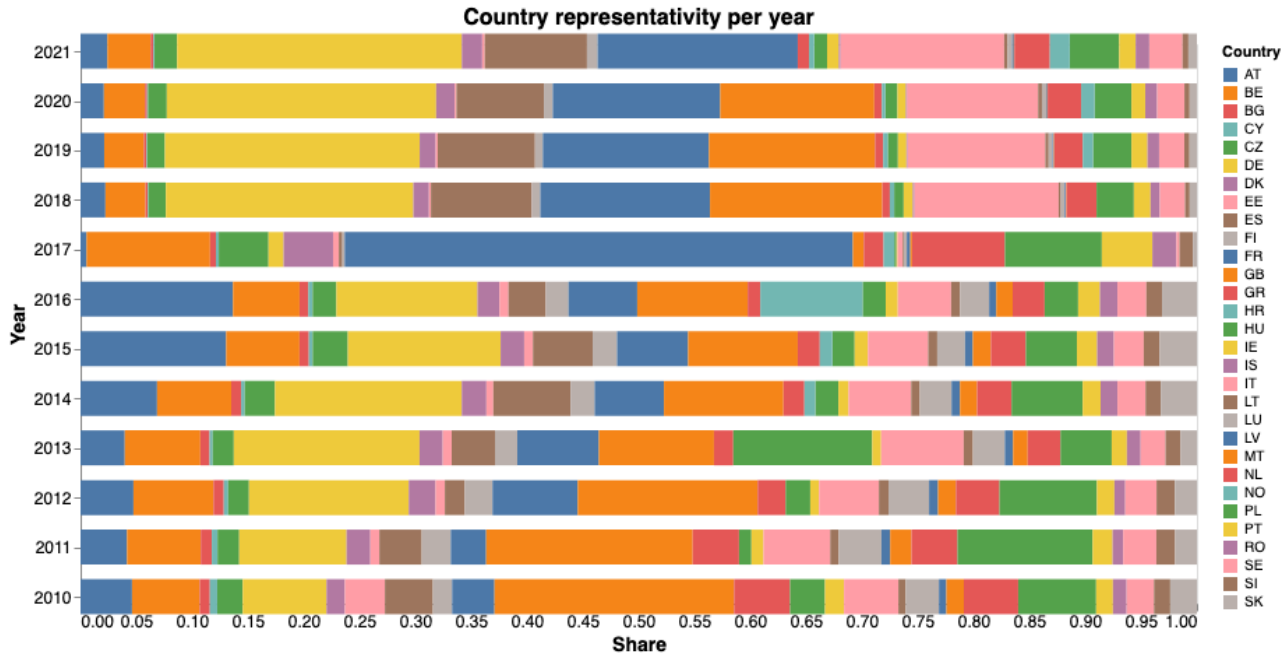


Figure 19. Country representativeness.

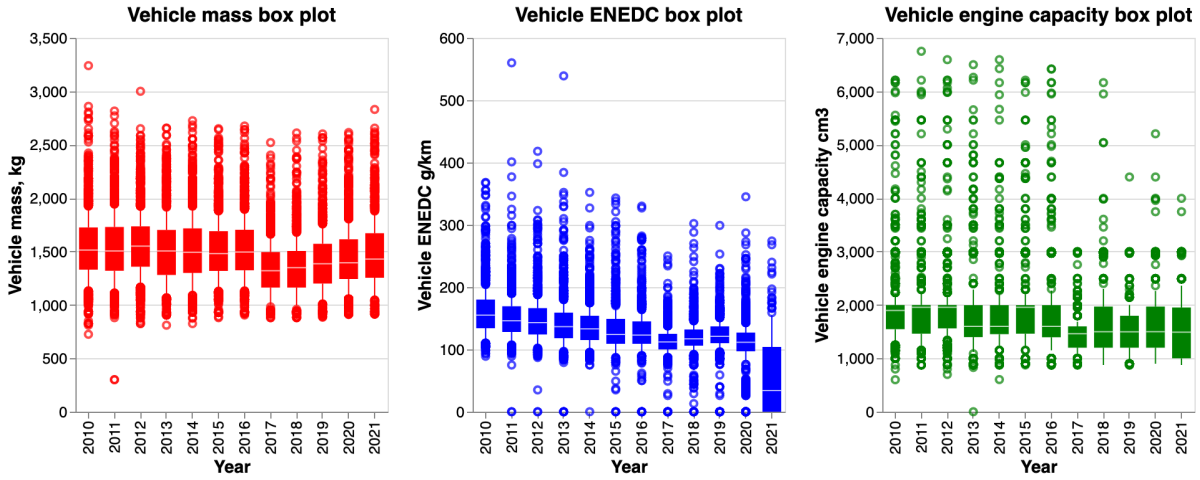
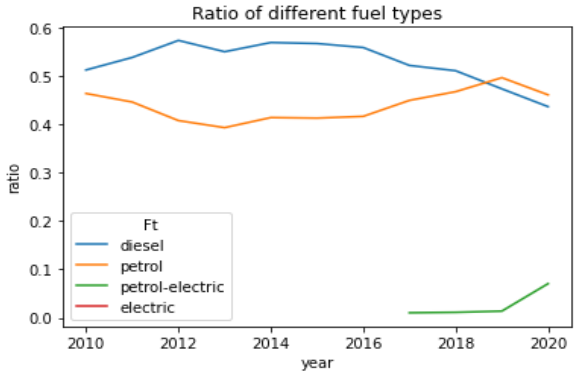


Figure 20 Numeric features statistics of initially loaded data.



	de	gb	fr	it
2010	1.1856	1.2147	1.7593	1.5839
2012	1.2016	1.4889	1.9393	1.9412
2014	1.1066	1.4717	1.7433	2.0165
2016	1.0593	1.3193	1.5012	1.971
2018	1.1079	1.0248	1.2348	2.1312
2019	1.0275	0.7511	1.0288	1.663
2020	1.0625	0.5398	1.0419	1.5563

Figure 21. Ratio of different fuel types in Europe and ratio of diesel/petrol in each country.

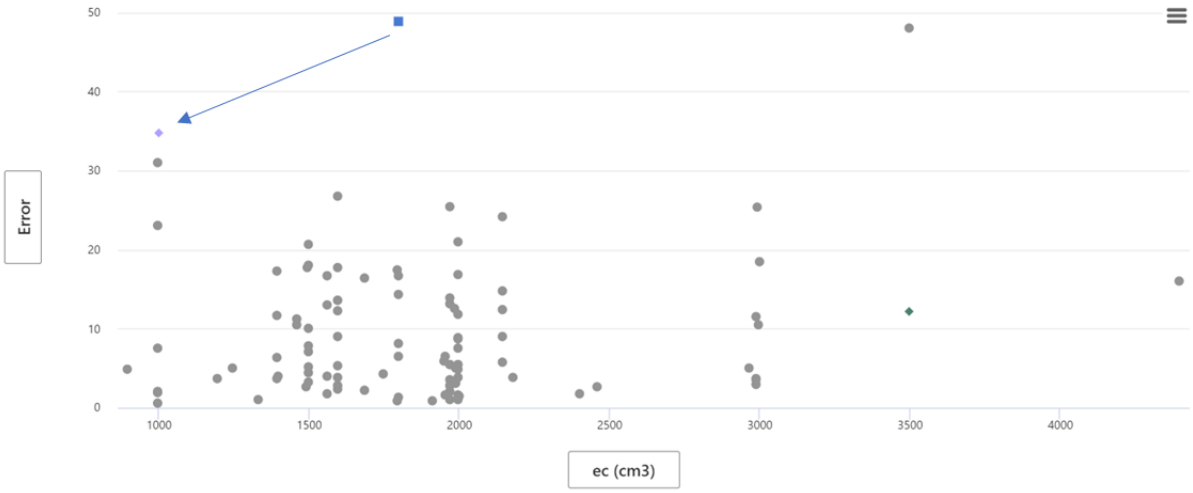


Figure 22

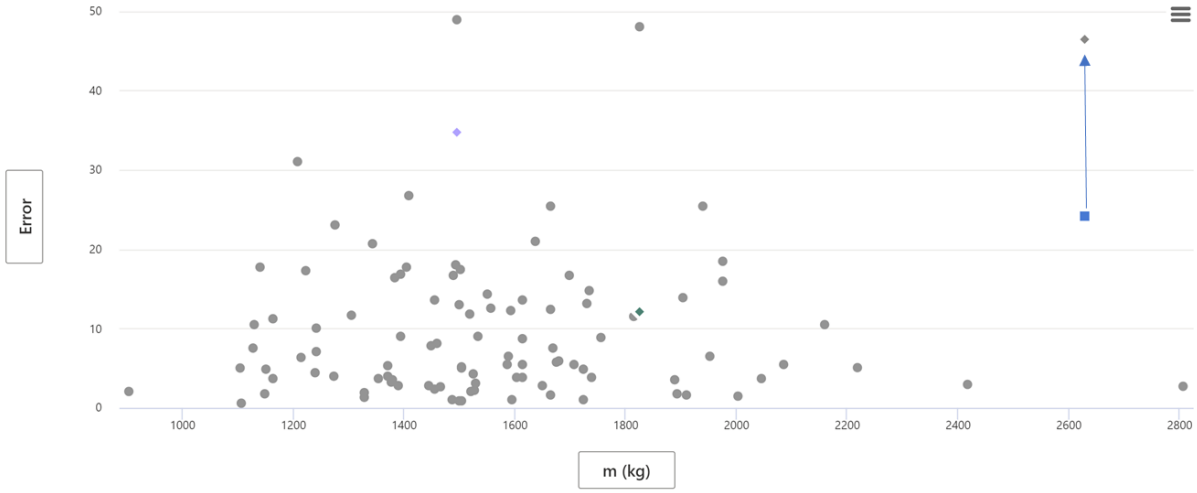


Figure 23

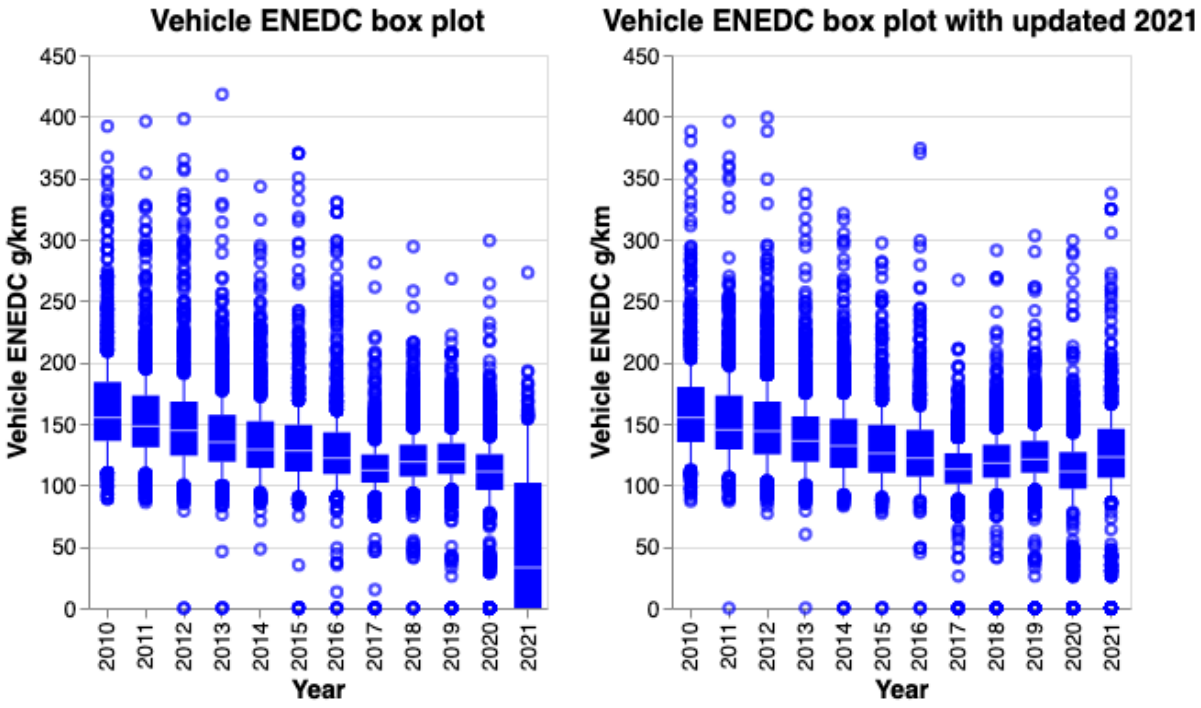


Figure 24 ENEDC feature for 2021 dataset before and after supervised learning method application.

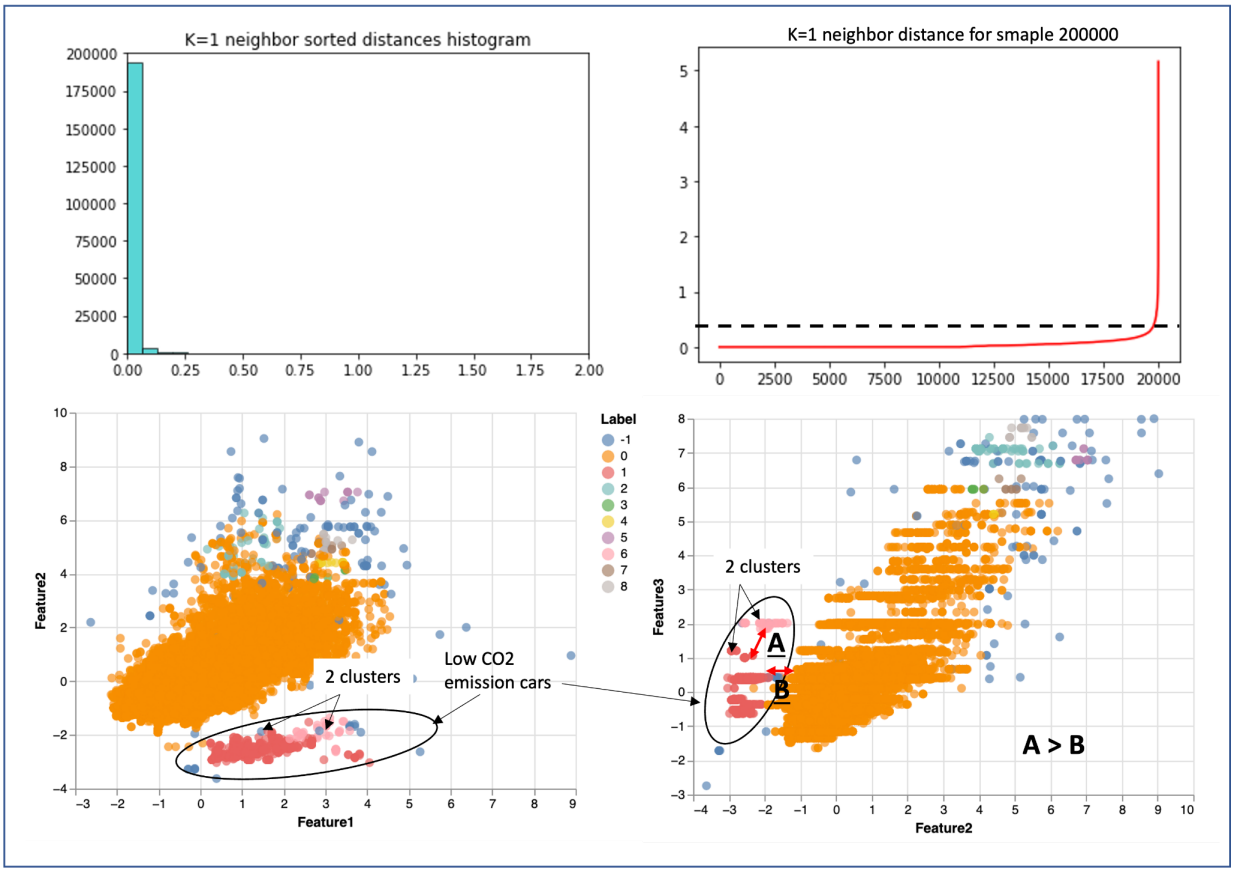


Figure 25 DB-scan clustering results.

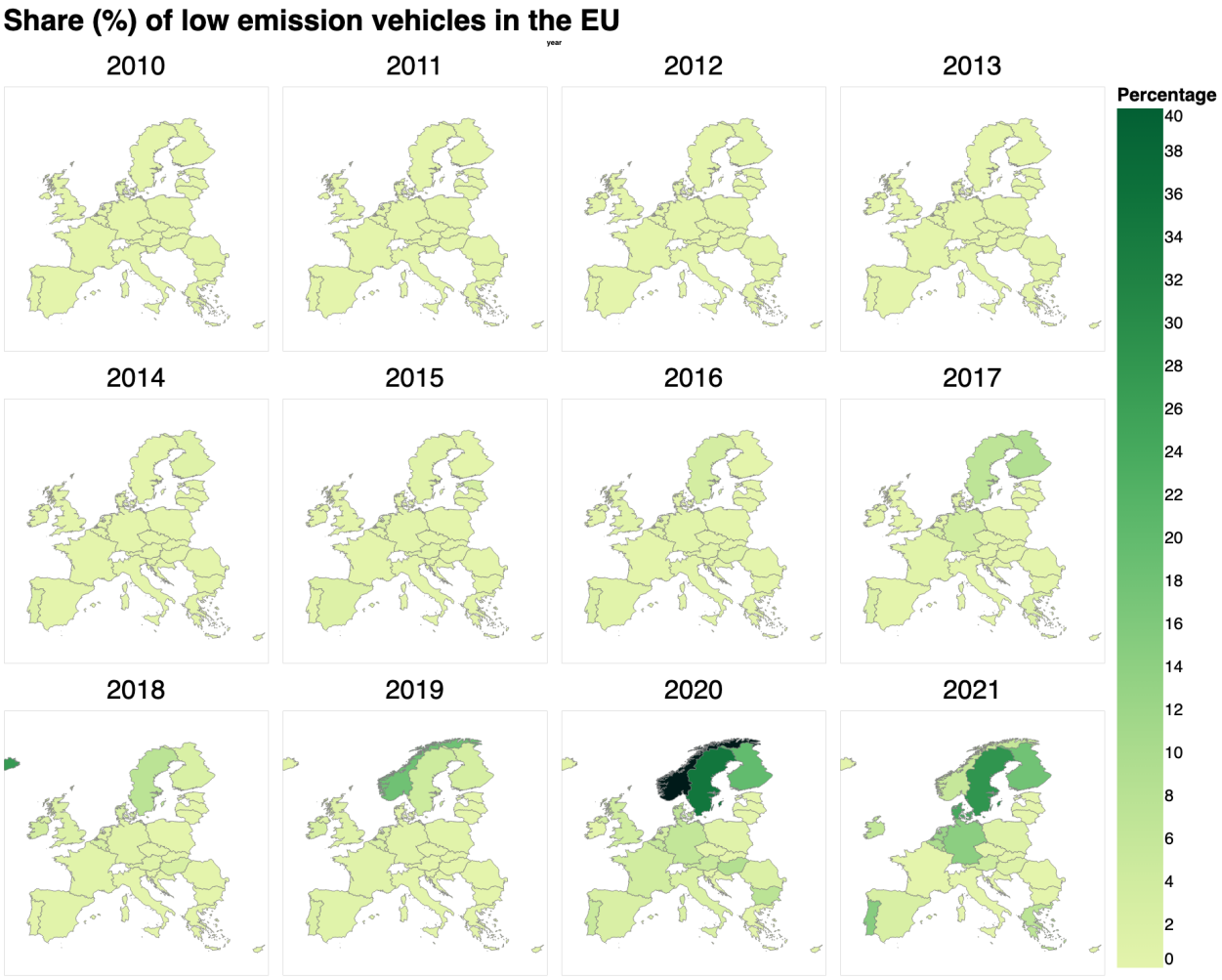


Figure 26 Spectral clustering results