Predicting Vehicle Emissions: An Analysis of Fuel Consumption Data

---

1 Introduction

The dataset we are working with is about vehicle fuel consumption from 2000 to 2022, providing model-specific fuel consumption ratings and estimated carbon dioxide emissions for new light-duty vehicles for retail sale in Canada extracted from the open Canada data, available on Kaggle and labelled as open database content. The values are approximate and were generated from the original ratings, not from vehicle testing. The columns are defined as:

1. YEAR: Model's manufacture year.
2. MAKE: Vehicle's brand.
3. Model:
    a. 4WD/4X4: Four-wheel drive.
    b. AWD: All-wheel drive.
    c. CNG: Compressed natural gas.
    d. FFV: Flexible-fuel vehicle.
    e. NGV: Natural gas vehicle.
    f. #: High-output engine that provides more power than the standard engine of the same size.
4. Transmission:
    a. A: Automatic.
    b. AM: Automated manual.
    c. AS: Automatic with select shift.
    d. AV: Continuously variable.
    e. M: Manual.
    f. 3 - 10: Number of gears.
5. Fuel Type:
    a. X: Regular gasoline.
    b. Z: Premium gasoline.
    c. D: Diesel.
    d. E: Ethanol (E85).
    e. N: Natural Gas.
6. HWY (L/100 km): Highway fuel consumption ratings shown in litres per 100 kilometres.
7. COMB (L/100 km): Combined rating (55% city, 45% highway) shown in litres per 100 kilometres.
8. COMB (mpg): combined rating (55% city, 45% highway) shown in miles per imperial gallon.
9. EMISSIONS: substances (generally gases) released into the atmosphere when fuels are burnt in grams per kilometre.

The dataset consists of 22556 instances, and we aim to find a model which predicts the emission as a regression task. Furthermore, based on an article on Ageco, emissions up to 150g/km are considered low, 160 to 255g/km emissions are considered medium and above 255g/km emissions are considered high. We will utilize this information to split our data into 3 classes and will try to predict the emission class in a classification task.

2 Methodology

This project is written using Python version 3.8.3 and the following libraries:

1. Pandas: Read, save and handle datasets.

2. NumPy: Operations related to arrays.
3. Matplotlib and Seaborn: visualizing and demonstrating.
4. Scikit-learn and XGBoost: Preprocessing the data, machine learning models.
5. TensorFlow and Keras: Neural network models and optimization techniques for them.

The dataset first examined to get an overall sense of what are the dataset columns, how many instances it contains and how each feature is distributed using histograms and other visualizations. The "TRANSMISSION" and "FUEL" which are non-numeric features have been converted into numeric ones by one hot encoding method to be used in our models. Feature selection approaches namely, linear correlation for linear models and feature importance found by random forest have been utilized to extract the most associated features. Moreover, all the input data have been normalized by the standard scalar and stratified by predefined classes, while 20% of the data is used as the test set and 20% of the training set is used as the validation set. Since our dataset is highly imbalanced in terms of emissions classes defined earlier, instead of 3 classes of "low", "medium" and "high" emission we changed it to a binary class of "acceptable" and "unacceptable" emission followed by randomly selecting a sub-sample of the bigger class equal in term of size to the smaller class.

We evaluate and sometimes demonstrate each model's performance on the validation set after being trained on the training set and the results are added to a predefined data frame. The models used in this project are:

1. Linear regression (Regression task)
2. Polynomial regression (Regression task)
3. Random Forest (Regression and classification task)
4. Decision Tree (Regression and classification task)
5. KNeighbors (Regression and classification task)
6. XGBoost (Regression and classification task)
7. Ridge (Regression and classification task)
8. Lasso (Regression task)
9. ElasticNet Regression task)
10. Stochastic gradient descent (Regression and classification task)
    a. No penalty
    b. With L1 penalty
    c. With L2 penalty
    d. With L1L2 penalty
11. Logistic Regression (Classification task)
    a. With L1 penalty
    b. With L2 penalty
12. Neural networks (Regression and classification task)
    a. Randomized search with no penalty
    b. Randomized search with L1 penalty
    c. Randomized search with L2 penalty
    d. Randomized search with dropout
    e. Hyperband optimization
    f. Bayesian optimization

For the machine learning models, we mostly defined a range of changeable parameters followed by either a grid or random search with cross-validation to find the proper parameters based on the complexity and diversity of the model's parameters. Furthermore, for the neural network models, we tried the randomized search, hyperband optimization and Bayesian optimization on our predefined architecture trained for 100 epochs monitoring the validation set for early stopping and hyperparameters (activation function, loss etc.) concerning the task. It is worth mentioning that since randomized search is fast by its nature, in addition to the base

architecture, we determined 3 more architectures respectively, with L1 regularizer, L2 regularizer and dropout added to the base model. For the regression task, RMSE (root-mean-square error) and MAE (mean absolute error) were acquired to evaluate the model and for the classification task, the F1-score were employed to have a balance between precision and recall score followed by threshold adjusting to achieve better score. For a better understanding, the progress of RMSE on the validation and training set for the regression models that was not time-consuming has been illustrated. Moreover, for the classification tasks, we demonstrate the classification report, the confusion matrix, the precision-recall curve, the recall and precision score by threshold and the ROC (receiver operating characteristic) curve. At the end of the regression and classification task, the best model has been chosen and evaluate its performance on the test set.

3 Analysis

In our initial dataset, 8 out of 12 of our feature types are numeric and the rest are objects without any NaN or null values. The distribution of our relevant numeric data is available in Table 1 which grants us an overview of the data. As the "ENGINE SIZE", "CYLINDERS", "COMB (mpg)" and "YEAR" parameters are discrete which led to an uninformative average and standard deviation, they have been avoided in the table. Moreover, in Figure 1 we can see a tail on the right side of the histograms which indicates the data is not normally distributed.

	FUEL CONSUMPTION	HWY (L/100 km)	COMB (L/100 km)	EMISSIONS
mean	12.76	8.91	11.03	250.06
std	3.50	2.27	2.91	59.35
min	3.50	3.20	3.60	83.00
max	30.60	20.90	26.10	608.00

Table 1: Data overview

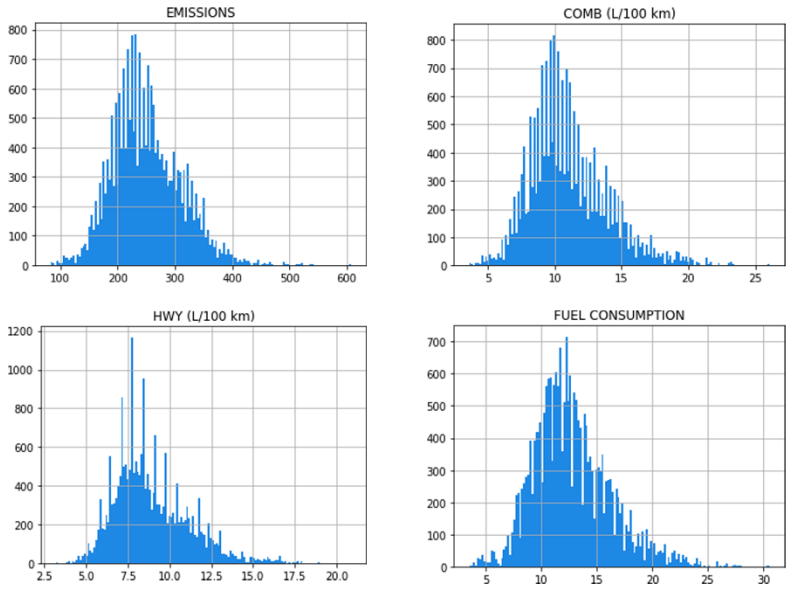

Figure 1: Features Histogram

3.1 Regression

For the regression task, we demonstrated each feature versus our label (EMISSIONS) to check for meaningful patterns, the important ones are shown in Figure 2, and we can find some linear

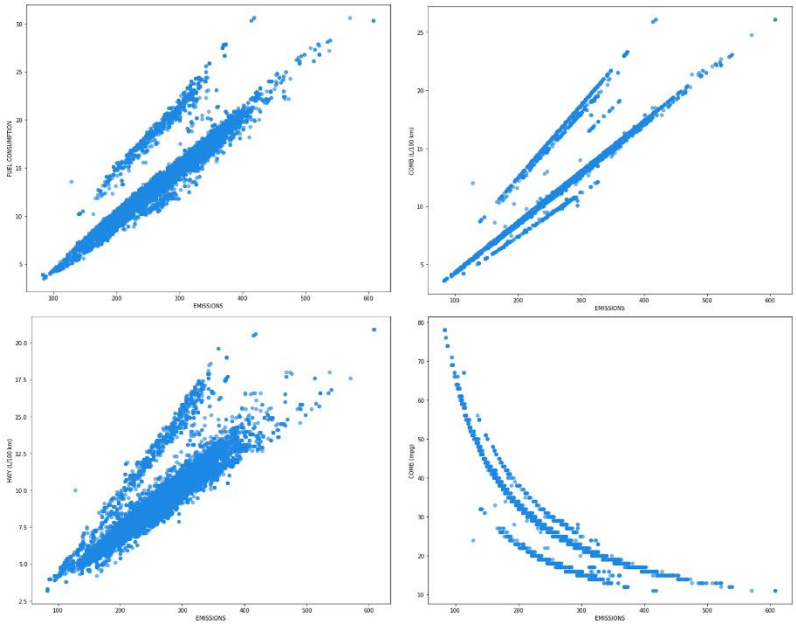

Figure 2: Features scatter plot

patterns between "FUEL CONSUMPTION" and emissions, "HWY (L/100 km)" and emissions, "COMB (L/100 km)" and emissions and somehow a parabola shape between "COMB (mpg)" and emissions.

In feature selection, we checked the correlation matrix for linearity which found "COMB (L/100 km)", "FUEL CONSUMPTION", "HWY (L/100 km)", "ENGINE SIZE", "CYLINDERS" and "COMB (mpg)" related to emissions. In addition, based on the random forest regressor we selected "COMB (L/100 km)", "COMB (mpg)", and "FUEL_E". We fed all features, linear features and random forest-selected to models defined earlier, which the models' parameters selected by grid or random search. The results in Table 2 are the models' performance on the validation data and are sorted on implementation order in the notebook.

	Linear features		Random Forest features		All features	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
Linear regression	19.41	11.85	6.83	3.66	4.58	2.41
polynomial regression	14.89	7.23	5.45	2.79	47666477	793453
Random Forest	8.33	2.31	5.46	2.80	1.97	0.33
Decision Tree	12.43	5.05	5.61	2.91	3.86	2.44
KNeighbors	8.37	2.24	6.57	3.06	3.59	1.71

XGBoost	7.40	2.56	5.38	2.81	1.48	0.55
Ridge	NA	NA	6.83	3.66	4.58	2.41
Lasso	NA	NA	6.83	3.74	4.60	2.47
ElasticNet	NA	NA	6.83	3.74	4.62	2.50
SGD (No penalty)	NA	NA	7.16	3.08	249511	13073
SGD + L1	NA	NA	8.87	5.94	2.51E+11	1.5E+10
SGD + L2	NA	NA	200.95	111.05	12.28	9.17
SGD + L1L2	NA	NA	8.87	5.94	8.36	5.89
NN + Random search (No penalty)	NA	NA	NA	5.28	NA	1.53
NN + Random search + L1	NA	NA	NA	7.04	NA	1.80
NN + Random search + L2	NA	NA	NA	7.04	NA	1.32
NN + Random search + Dropout	NA	NA	NA	20.42	NA	28.72
NN + Hyperband	NA	NA	NA	2.59	NA	1.90
NN + Bayesian	NA	NA	NA	5.28	NA	1.53

Table 2: Regression Performance

The RMSE is not defined by default in TensorFlow, therefor we checked MAE first and if the models' MAE score was similar then the RMSE would be examined. We can see that in some cases such as polynomial regression using selected features outcomes better results while in other cases such as XGBoost acquiring all features emerge with lower error. By checking the training and validation set MAE curve based on the epoch, we can conclude that using dropout led to overfitting (the validation data achieving lower MAE than the training set) in all instances. The other nominated models by the random search validation score were usually near the training set, however using L1 and L2 regularization on random forest-selected features causes a less unstable learning curve while on all features the curve has more variance. Eventually, the random forest regressor using all the features was chosen as the best model and was evaluated on the test set which led to 0.33 MAE and 2.18 RMSE.

3.2 Classification

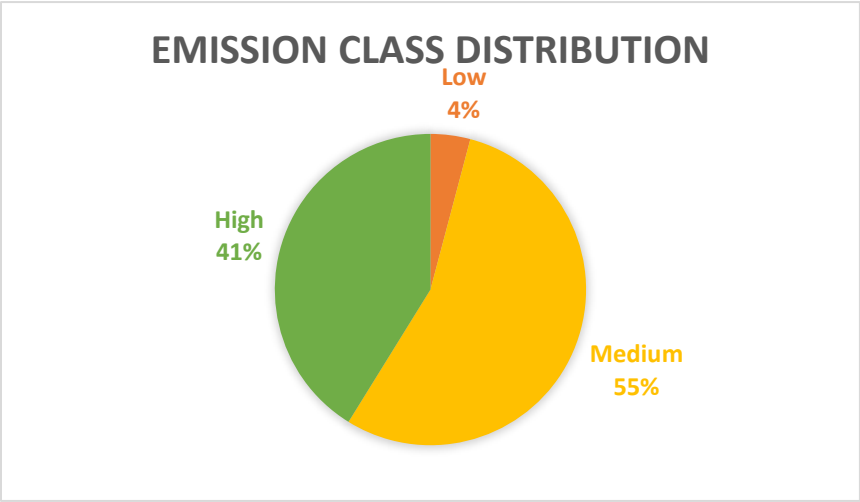

Figure 3: Emission class distribution

The emission is classified as "low", "medium" and "high" based on formerly defined principles and the distribution can be seen in Figure 3.

As demonstrated in Figure 3 the number of instances of "low" class in comparison to other classes is significantly deficient. Therefore, we combined the "low" and "medium" classes named them as "acceptable" and labelled the "high" class as "unacceptable" to achieve a binary classification. However, even after that, there was roughly a 4000 case difference, thus we randomly selected a sub-sample of the bigger class equivalent to the smaller class and labelled the acceptable class as 1 and the other as 0. The random forest classifier was used to find the relevant features and we found a significant relation between "COMB (L/100 km)" and "COMB (mpg)" with the emission class and more mildly with "FUEL CONSUMPTION" and "HWY (L/100 km)". The strong association between "COMB (L/100 km)" and "COMB (mpg)" with emission class seems reasonable as Gasoline is 90% carbon and the car mostly expels all carbon into the atmosphere therefore, fuel efficiency parameters will correlate strongly with CO2 emission. The normalized and stratified by new labelling data fed into the predefined models for training and tested on the validation set afterwards for any signs of underfitting or overfitting and the suitable threshold was discovered based on the F1-score and the ROC curve. As defined earlier the F1-score has been chosen to a achieve balance between precision and recall, the F1 scores was calculated in all ML models and the proper threshold was adjusted based on the highest F1 score, and the ROC curve to find the closest point in our curve to the top left corner (which is the ideal model of having 1 true positive rates and 0 false positive rates), implied to the model. The results are available in Table 3.

Model	Thresholds	F1 - Score
Random Forest	0.28	0.9711
	0.50	0.9720
Decision Tree	0.49	0.9721
	0.49	0.9721
KNeighbors	0.50	0.9737

	0.60	0.9755
XGBoost	0.49	0.9730
	0.51	0.9739
Ridge	-0.05	0.9520
	-0.05	0.9520
SGD (No penalty)	0.04	0.9670
	0.04	0.9670
SGD + L1	-0.04	0.9663
	-0.04	0.9663
SGD + L2	0.01	0.9649
	0.01	0.9649
SGD + L1L2	0.00	0.9654
	0.00	0.9654
Logistic Regression + L2	-0.12	0.9665
	-0.05	0.9659
Logistic Regression+ L1 or L2	-0.11	0.9661
	-0.07	0.9659
NN + Random search (No penalty)	Default threshold	0.9709
NN + Random search + L1	Default threshold	0.9672
NN + Random search + L2	Default threshold	0.9643
NN + Random search + Dropout	Default threshold	0.9683
NN + Hyperband	Default threshold	0.9701
NN + Bayesian	Default threshold	0.9672

Table 3: Classification performance

The precision-recall curve, precision and recall score based on the threshold curve and the true positive-false positive rate curve are demonstrated for each model to have a better

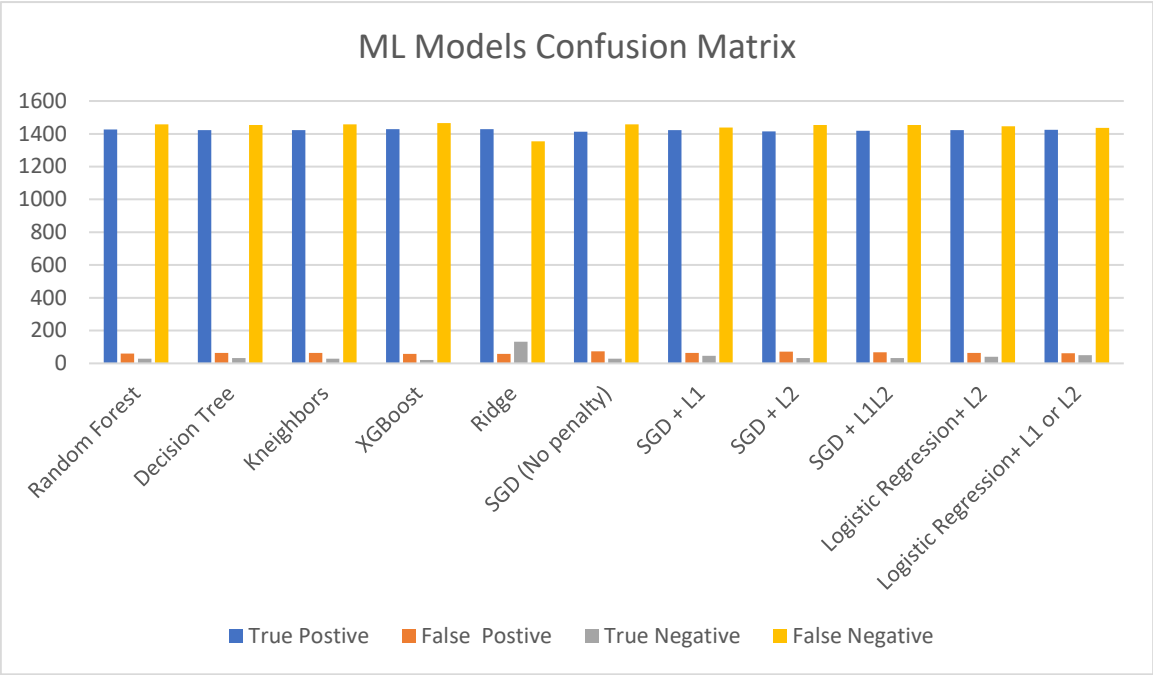

Figure 4: Confusion matrix

understanding. Moreover, the true positive, false positive, true negative and false negative of each model is demonstrated in Figure 4.

Eventually, the K-nearest neighbors were selected as the best model and evaluated on the test set with the proper threshold which was found for the validation set achieving 0.9712 F1-score.

4 Conclusion

In this project, we aimed to predict carbon dioxide emissions value as a regression task and predict whether the amount of carbon dioxide is permissible or not as a binary classification task for vehicles from 2000 to 2022. The data was first preprocessed and then the selected features were transformed into proper format before feeding into the models. Throughout the analysis, we demonstrate the distribution of features and scatter plots of the features in comparison to emissions for a better understanding of the data. For the regression task, we utilized 11 types of models some of which also have variants and 8 types of models for classification with variants. The most robust regression model (random forest regressor) reached an MAE of 0.33 and RMSE of 2.18 on the test set and the classification model (K-nearest neighbors) achieved a 0.9712 F1-score on the test set.

References

Aurelien Geron. 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd. ed.). O'Reilly Media, Inc.