We developed a machine learning-based predictive model to predict the vehicle emission score or smog rating (vehicle tailpipe CO2 emission) as defined by the Environmental Protection Agency (EPA). Essentially, the task is to perform a multi-class classification of the emission score, a discrete unitless integer quantity, varying from 1 (worst) to 10 (best). For this purpose, we collected the ‘vehicle’ and ‘emissions’ data sets from the EPA portal (EPA, 2021a, 2021b).

As of now, the ‘vehicle’ and ‘emissions’ data contain around 44000 and 42000 records each with 83 and 8 features or columns, respectively. However, these data sets must be joined based on the vehicle id column which reduces the number of available observations because of missing ids. Moreover, the emission score had incorrect entries (negative scores) which were removed. Hence, after the data merging and cleaning steps, we had around 28000 observations.

Thereafter, for categorical data transformation, we aggregated the fuelType1 column by combining midgrade and regular gasolines as 'Regular Gasoline' and 'Electricity' and 'Natural Gas' as 'Green Fuel'. Next, we performed correlation and distribution analysis, using which we were able to answer one of the sub research questions. We found that Standard Pickup Trucks and Vans (both cargo and passenger types) had the lowest emission scores in general. In addition, it was confirmed from boxplot analysis that, diesel vehicles received the lowest scores.

An interesting observation from our boxplot outlier analysis was that all scores except those in the interval [4, 7] were marked as outliers. Hence, our initial choice of not performing outlier removal as part of the second milestone was justifiable. Similarly, we found principal component analysis (PCA) to be of not much use possibly because of strong non-linearity in the data set. We may need to apply kernel PCA-based approaches for appropriately investigating the benefits of PCA.

The final component of this project was the predictive modeling where we compared seven classification models. These are Multinomial Logistic Regression (MLR), Support Vector Machine (SVM), Gradient Boosting Trees (GBT), Random Forests (RF), Boosted Random Forests (BRF), Extremely Randomized Trees (ETR), and Multi-Layer Perceptron (MLP). For the GBT and BRF algorithms, we used the LightGBM (Ke et al., 2017) package and used scikit-learn for the rest.

One issue with the given data set is extremely high class imbalance. For example, the EPA score 4 only has 6 samples (0.02%) in total while 40% of the total emission scores are marked as 5. Another potential problem was the data dimensionality (35 predictors in total including dummies from the VClass column).

Therefore, we preferred using ensemble machine learning algorithms that support class weights as parameters and provide feature selection on the fly. Moreover, we only have 28000 observations, so going for deep learning models is overkill.

In order to appropriately compare all the models, we scaled the features in the [0, 1] interval using MinMaxScaler. Next, to train the models, we initially used the scikit-learn GridSearchCV with RepeatedKFold having five-folds and a single repeat (we tried multiple repeats too, but the results were similar). Here, we could not use the stratified version because class 4 only had four observations in the training data. This was also a reason why we used 70-30 split of the training and test data. If we used 80% as training, then class 4 only had a single observation in the test data.

The model evaluation was based on the similarity of the training and validation F1 score to reduce overfitting. We then checked the F1 for the test data. For additional interpretability, we also showed balanced accuracy, precision, recall, and the Cohen’s kappa score. Note that F1 was the metric that was optimized. For, F1, precision, and recall, we used the macro averaging as part of the GridSearch scoring metrics.In this context, we had NaN issues with the ROC AUC macro OVR (one-versus-rest) scoring metric through GridSearchCV. That is why we only show the ROC AUC for the test data of the final model.

While running the scikit-learn GridSearchCV with different hyperparameters of the ensemble learning algorithms and MLP on an Alienware M17 R1 (personal PC), we noticed that for many parameter values, even 8 hrs of computation was not sufficient. As a result, we switched to the Missouri S&T Foundry High-Performance Computing cluster and used Dask (Rocklin, 2015) to train our models in a distributed computing environment. This resulted in a significant reduction of computing time wherein we could tune several parameters in a very short time. For example, the GBM model training took around 6 hrs on the Alienware and took only 15 minutes using Dask.

Finally, we found that the GBM having the error metrics shown in Table 1 performs the best.

**Table 1.** Error metrics for the test data

|  |  |
| --- | --- |
| **Error Metric** | **Score (rounded to 2 decimal places)** |
| F1 | 0.51 |
| ROC AUC | 0.62 |
| Precision | 0.47 |
| Recall | 0.63 |
| Balanced Accuracy | 0.63 |
| Kappa score | 0.33 |

We think that our results and analysis are quite relevant to EPA policymakers and that such kind of data-driven approaches using machine learning offer significant benefits to address critical challenges in the sustainable transportation industry.

**References**

EPA. (2021a). *EPA Fuel Economy*. https://www.fueleconomy.gov/

EPA. (2021b). *Fuel Economy Guide*. https://www.fueleconomy.gov/feg/pdfs/guides/FEG2021.pdf

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In I. Guyon, U. V Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

Rocklin, M. (2015). Dask: Parallel Computation with Blocked algorithms and Task Scheduling. *Proc. of the 14th Python in Science Conf*. http://conference.scipy.org/proceedings/scipy2015/pdfs/matthew\_rocklin.pdf