**Case Study: Predicting CO2 Emissions of Vehicles in Canada using Machine Learning**

**Date:** July 18, 2025

**Prepared By:** Soham Kamble

Ved Jaiswal

Hrushikesh Narale

**1. Executive Summary**

This case study details the development of a predictive machine learning model to estimate CO2 emissions from various vehicle models in Canada. Leveraging a dataset containing diverse vehicle attributes, we explored traditional machine learning algorithms such as Decision Trees, Random Forests, and Polynomial Regression, alongside a more sophisticated Neural Network approach. The primary objective was to accurately predict CO2 emissions (in g/km) based on vehicle specifications, thereby providing a tool to understand emission drivers and support related policy or design decisions. Our analysis indicates that the Neural Network model achieved the most robust performance, demonstrating strong generalization capabilities with a Mean Absolute Error (MAE) of approximately 10.37 g/km on unseen data, making it a promising tool for this prediction task.

**2. Introduction**

Climate change driven by greenhouse gas emissions, particularly carbon dioxide (CO2), is a pressing global concern.1 The transportation sector is a significant contributor to these emissions.2 Understanding and accurately predicting CO2 emissions from vehicles is crucial for several reasons:

* **Environmental Policy:** Informing regulatory standards and incentive programs for cleaner vehicles.
* **Automotive Design:** Guiding manufacturers in designing more fuel-efficient and lower-emission vehicles.
* **Consumer Choice:** Empowering consumers with information to make environmentally conscious purchasing decisions.
* **Research & Development:** Identifying key vehicle characteristics that influence emissions, thus directing future research.

This case study presents the methodology and results of building predictive models for CO2 emissions using a Canadian vehicle dataset. We will compare various machine learning approaches, highlighting their strengths and weaknesses, and ultimately propose the most effective solution.

**3. Problem Statement**

The core problem addressed is: **Given a set of vehicle specifications (e.g., engine size, cylinders, fuel type, transmission), can we accurately predict its CO2 emissions (g/km)?**

This is a regression problem, as the target variable (CO2 emissions) is a continuous numerical value. The challenge lies in identifying the most influential features and building a model that can capture the complex, non-linear relationships between vehicle attributes and their CO2 output.

**4. Data Description**

The project utilizes the "CO2 Emissions\_Canada.csv" dataset. This dataset contains comprehensive information about various vehicle models, including attributes that are known to influence CO2 emissions. Key features include:

* **Make**: Vehicle manufacturer (e.g., Ford, Honda)3
* **Model**: Specific vehicle model (e.g., F-150, Civic)
* **Vehicle Class**: Type of vehicle (e.g., SUV - SMALL, COMPACT)
* **Engine Size(L)**: Engine displacement in liters
* **Cylinders**: Number of engine cylinders
* **Transmission**: Type of transmission (e.g., A6, M5)
* **Fuel Type**: Type of fuel used (e.g., Z - Premium, X - Regular, D - Diesel, E - Ethanol)
* **Fuel Consumption City (L/100 km)**: Fuel consumption in city driving
* **Fuel Consumption Hwy (L/100 km)**: Fuel consumption in highway driving
* **Fuel Consumption Comb (L/100 km)**: Combined fuel consumption
* **Fuel Consumption Comb (mpg)**: Combined fuel consumption in miles per gallon4
* **CO2 Emissions(g/km)**: The target variable, CO2 emissions in grams per kilometer.5

The dataset, as observed, is relatively clean with few missing values, simplifying the preprocessing phase. Categorical features are prominent and require careful handling for numerical model input.

**Page 3: Methodology - Data Preprocessing**

**5. Methodology**

The predictive modeling process involved several standard machine learning steps:

**5.1. Data Preprocessing**

Data preprocessing is a critical phase to prepare the raw data for machine learning algorithms.6

* Handling Missing Values:

The dataset was inspected for missing values. Given its relatively clean nature, rows containing any null values were directly removed using the df.dropna(inplace=True) command.7 This ensured that all model inputs were complete.

* Categorical Feature Encoding:

The dataset contained several categorical features (Make, Model, Vehicle Class, Transmission, Fuel Type).8 Machine learning models operate on numerical data. Therefore, these categorical features were converted into numerical representations using LabelEncoder from sklearn.preprocessing. Each unique category within a feature was assigned a unique integer label.9 While One-Hot Encoding is an alternative, Label Encoding was chosen for simplicity and due to the potentially high cardinality of 'Make' and 'Model', which would lead to a very wide dataset with One-Hot Encoding.

* Feature-Target Split:

The dataset was divided into features (X), which are the input variables, and the target variable (Y), which is 'CO2 Emissions(g/km)'.

* Train-Test Split:

The dataset was further partitioned into training and testing sets. An 80/20 split was used, meaning 80% of the data was allocated for training the models (x\_train, y\_train), and 20% was reserved for evaluating their performance on unseen data (x\_test, y\_test).10 A random\_state was set to ensure reproducibility of the splits.

* Feature Scaling:

Numerical features often have different scales. For algorithms sensitive to feature magnitudes, such as Neural Networks (and Linear/Polynomial Regression), scaling is crucial. StandardScaler was applied, which transforms features to have a mean of 0 and a standard deviation of 1. This process was performed by fitting the scaler on the training data (x\_train) and then using that fitted scaler to transform both the training and testing data (x\_test) to prevent data leakage.

**5.2. Model Selection and Training**

We experimented with several machine learning models suitable for regression tasks to identify the most effective approach for CO2 emissions prediction.

* **Traditional Machine Learning Models (for comparison/exploration):**
  + **Decision Tree Regressor:** A non-linear model that partitions the data based on feature values, useful for capturing complex relationships.11
  + **Random Forest Regressor:** An ensemble method that builds multiple decision trees and averages their predictions, generally offering higher accuracy and robustness than a single decision tree.12
  + **Polynomial Regression:** An extension of linear regression that models the relationship between the independent variable and the dependent variable as an 13n-th degree polynomial, allowing it to fit non-linear patterns.14 This involved creating polynomial features before applying a standard LinearRegression model.
  + *(Note: Logistic Regression was explored for classification tasks in earlier discussions, but the primary focus for this case study is the regression of CO2 emissions.)*
  + *(Note: K-Means Clustering was explored for unsupervised learning/grouping tasks, distinct from the predictive regression goal here.)*
* Neural Network (Deep Learning Approach):

Given the potential for complex, non-linear relationships in emissions data, a Neural Network (NN) was constructed using TensorFlow's Keras API.

* + **Architecture:**
    - **Input Layer:** Implicitly defined by the input shape of the first Dense layer, matching the number of features in our processed dataset.
    - **Hidden Layer 1:** Dense(64, activation='relu') - A fully connected layer with 64 neurons and a Rectified Linear Unit (ReLU) activation function.15 ReLU helps in introducing non-linearity and speeds up training.16
    - **Hidden Layer 2:** Dense(32, activation='relu') - Another fully connected layer with 32 neurons and ReLU activation, further processing the learned features.17
    - **Output Layer:** Dense(1) - A single neuron with a linear activation function (default) suitable for predicting a continuous numerical value (CO2 emissions).
  + **Compilation:**
    - **Optimizer:** adam - A widely used and effective optimization algorithm that adapts learning rates for each parameter.18
    - **Loss Function:** mse (Mean Squared Error) - Measures the average squared difference between actual and predicted values, aiming to minimize large errors.19
    - **Metrics:** mae (Mean Absolute Error) - Provides an easily interpretable average absolute difference between predictions and actual values.20
  + Training:

The model was trained for 50 epochs with a batch\_size of 32. A validation\_split of 0.2 was used, meaning 20% of the training data was set aside as a validation set to monitor the model's performance on unseen data during training, helping to detect overfitting.

**6. Results**

The models were evaluated using standard regression metrics. For the Neural Network, the primary evaluation was performed on the dedicated test set.

**6.1. Neural Network Performance Evaluation**

After training the Neural Network for 50 epochs, its performance was assessed on the x\_test and y\_test sets, which represent data the model had never encountered during training.

* **Test Mean Absolute Error (MAE):** Approximately 10.37 g/km
* **Test Mean Squared Error (MSE):** Approximately 152.02 (g/km)2

The MAE of 10.37 g/km indicates that, on average, the model's predictions for CO2 emissions are off by about 10.37 g/km from the actual values. This is a robust performance given the range and variability of CO2 emissions in the dataset.

**6.2. Training Performance Visualization (Neural Network)**

The plot of training and validation loss over epochs provides critical insights into the learning process:

**[Insert the "Model Training Performance" plot here, similar to the image you provided]**

* **Training Loss (Blue Line):** This curve consistently decreased throughout the training, indicating that the model was effectively learning from the training data and minimizing its errors.
* **Validation Loss (Orange Line):** Crucially, the validation loss also showed a consistent decrease and tracked closely with the training loss. This is an excellent sign, demonstrating that the model was not merely memorizing the training data but was generalizing well to unseen data.
* **No Significant Overfitting:** The gap between the training and validation loss lines remained relatively small and did not significantly diverge towards the end of training. This suggests that the model is not suffering from severe overfitting.
* **Convergence:** Both loss curves showed signs of flattening out towards the later epochs, suggesting that the model had largely converged and further training might provide only marginal improvements.

**7. Conclusion**

This case study successfully demonstrates the application of machine learning, particularly a Neural Network, to predict CO2 emissions of vehicles based on their specifications. The chosen Neural Network architecture, combined with appropriate data preprocessing (categorical encoding and feature scaling), proved to be highly effective. The consistent decrease in both training and validation loss, along with a reasonable Mean Absolute Error on unseen test data, indicates a robust and generalizable predictive model.

The ability to accurately predict CO2 emissions can serve as a valuable tool for various stakeholders, from informing environmental policies and regulations to aiding automotive manufacturers in designing more eco-friendly vehicles.

**8. Future Work and Recommendations**

To further enhance this project and the model's capabilities, the following areas could be explored:

* **Hyperparameter Tuning:** Systematically optimize the Neural Network's architecture (number of layers, neurons per layer, activation functions) and training parameters (learning rate, batch size, number of epochs, optimizers) using techniques like GridSearchCV or RandomizedSearchCV, or more advanced methods like KerasTuner.21
* **Feature Engineering:** Explore creating new features from existing ones (e.g., power-to-weight ratio, engine efficiency metrics) that might capture more complex relationships with CO2 emissions.
* **Alternative Architectures:** Investigate more complex Neural Network architectures if needed, though the current simple dense network performed well.
* **Regularization:** Implement regularization techniques (e.g., Dropout, L1/L2 regularization) in the Neural Network to further guard against potential overfitting, especially with larger models or datasets.22
* **Ensemble Modeling:** Combine predictions from multiple models (e.g., the Neural Network with a Random Forest) to potentially achieve even higher accuracy.
* **Model Interpretability:** Explore techniques (e.g., SHAP, LIME) to understand which specific vehicle features contribute most to the predicted CO2 emissions, providing more actionable insights.23
* **Data Collection:** If available, incorporating more diverse datasets or temporal data could further improve model robustness and applicability.