SUSTAINABLE TRANSPORTATION - STUDY OF VEHICULAR CO2 EMISSION AND ELECTRIC VEHICLE IN CANADA

BY : SONAL PARMAR

1335126

12/19/2024

Motivation

- Concerns about climate change and its bizarre impacts.

- In 2023, transportation was to be the second largest contributor to Global Green House Gas.

- Insight to help and support policy makers towards tranpsortation substainability.

- Importance of Adoption of Electric Vehicles.

Related Work

- Vision Zero: to eliminate all traffic casualties and severe trauma while establishing a safe, stable, and healthy mobility.

- Ecodrive-Deep Learning Models For Accurate Prediction of Vehicle Co2 Emissions – Author- A Joshua Isaac; A Jenefa; Roshan John Renny Samuel; P John Ruben Raj; L Pratheesh Raj P; M Raghul Kanna

  publised by 2024 by ICAAIC

- Modelling of CO2 Emission Prediction for Dynamic Vehicle Travel Behavior Using Ensemble Machine Learning Technique –by Navarajan Subramaniam; Norhakim Yusof

  published by 2021 IEEE 19th Student Conference on Research and Development (SCOReD)

- Design and Development of Exploratory Model in AI for Addressing CO2 Emission for a Sustainable Future- by Parminder Singh; Saurabh Dhyani; Harish Dutt Sharma; Sanjay Mishra; Yogesh Juyal; Amarjeet Rawat

- by 2023 4th (ICCAKM)

TABLE I
DATASET DESCRIPTION

Attribute	Description
Make	Brand or manufacturer.
Model	Specific model.
Vehicle Class	Classification based on size/purpose.
Engine Size (L)	Engine size in liters.
Cylinder Count	Quantity of cylinders.
Transmission Type	Mode of power transmission.
Fuel Category	Fuel variety.
Urban Fuel Economy (L/100 km)	Consumption in city driving (L/100 km).
Rural Fuel Economy (L/100 km)	Consumption in highway driving (L/100 km).
Overall Fuel Economy (L/100 km)	Total fuel efficiency (L/100 km).
Overall Fuel Economy (mpg)	Total fuel efficiency (mpg).
CO2 Output (g/km)	Carbon dioxide output per km.

Eco Drive

- Uses Deep Learing Model
- Comparison of DCNN and KNN
- Comparison DCNN and LSTM

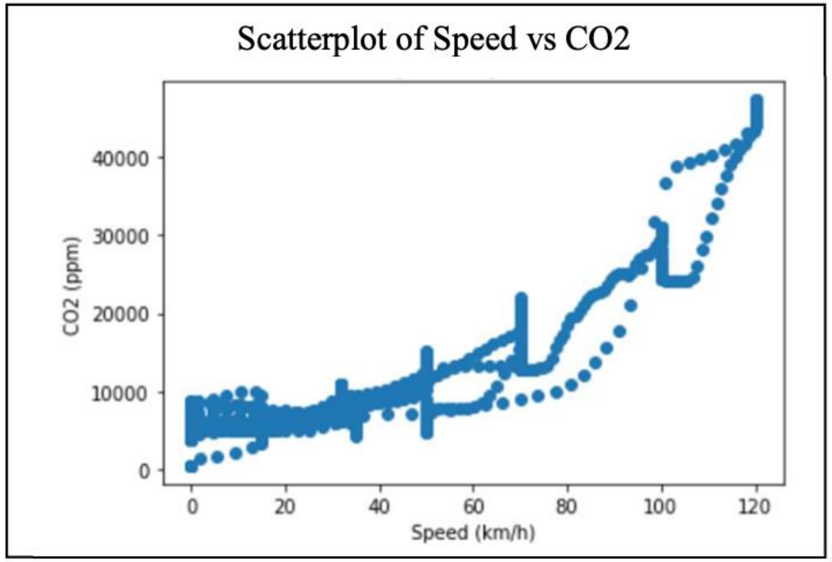


Fig. 3. Relationship between speed and CO2 emission

Speed and CO2 Emission

Uses Gradient Boosting Regressor

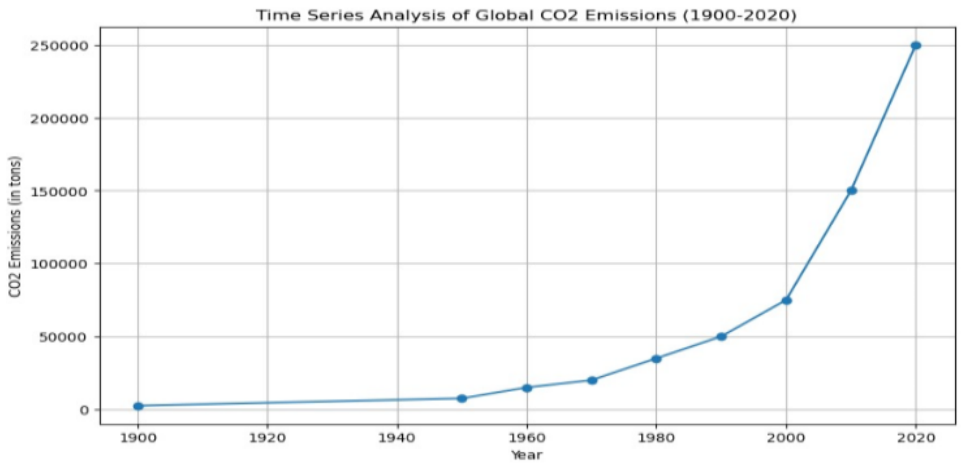


Fig – 5 : Time Series Analysis of CO2 Emissions

Time Series Analysis

- Uses Perceptron

Selection of Dataset

```
 0   Incentive Request Date
 1   Month and Year
 2   Government of Canada Fiscal Year (FY)
 3   Calendar Year
 4   Dealership Province / Territory
 5   Dealership Postal Code
 6   Purchase or Lease
 7   Vehicle Year
 8   Vehicle Make
 9   Vehicle Model
 10  Vehicle Make & Model
 11  Battery-Electric Vehicle (BEV), Plug-in Hybrid Electric Vehicle (PHEV) or Fuel Cell Elect
 12  BEV/PHEV/FCEV - Battery equal to or greater than 15 kWh or
Electric range equal to or greater than 50 km
 13  BEV, PHEV  ? 15 kWh or PHEV < 15 kWh (until April 24, 2022)
and
PHEV ?  50 km or PHEV < 50 km and  FCEVs ? 50 km or FCEVs < 50 km
(April 25, 2022 onward)  202206 non-null  object
 14  Eligible Incentive Amount
 15  Individual or Organization
(Recipient)
 16  Recipient Province / Territory
 17  Country
```

SOURCE- Co2 Emission Dataset from Kaggle (Government of Canada)

- Consist of important features of Electric vehicle for the Analysis

- Regions

- Vehicle year

- Puchase or Lease

Selection of
CO_2 Emission Dataset

SOURCE- GOVERNMENT OF CANADA

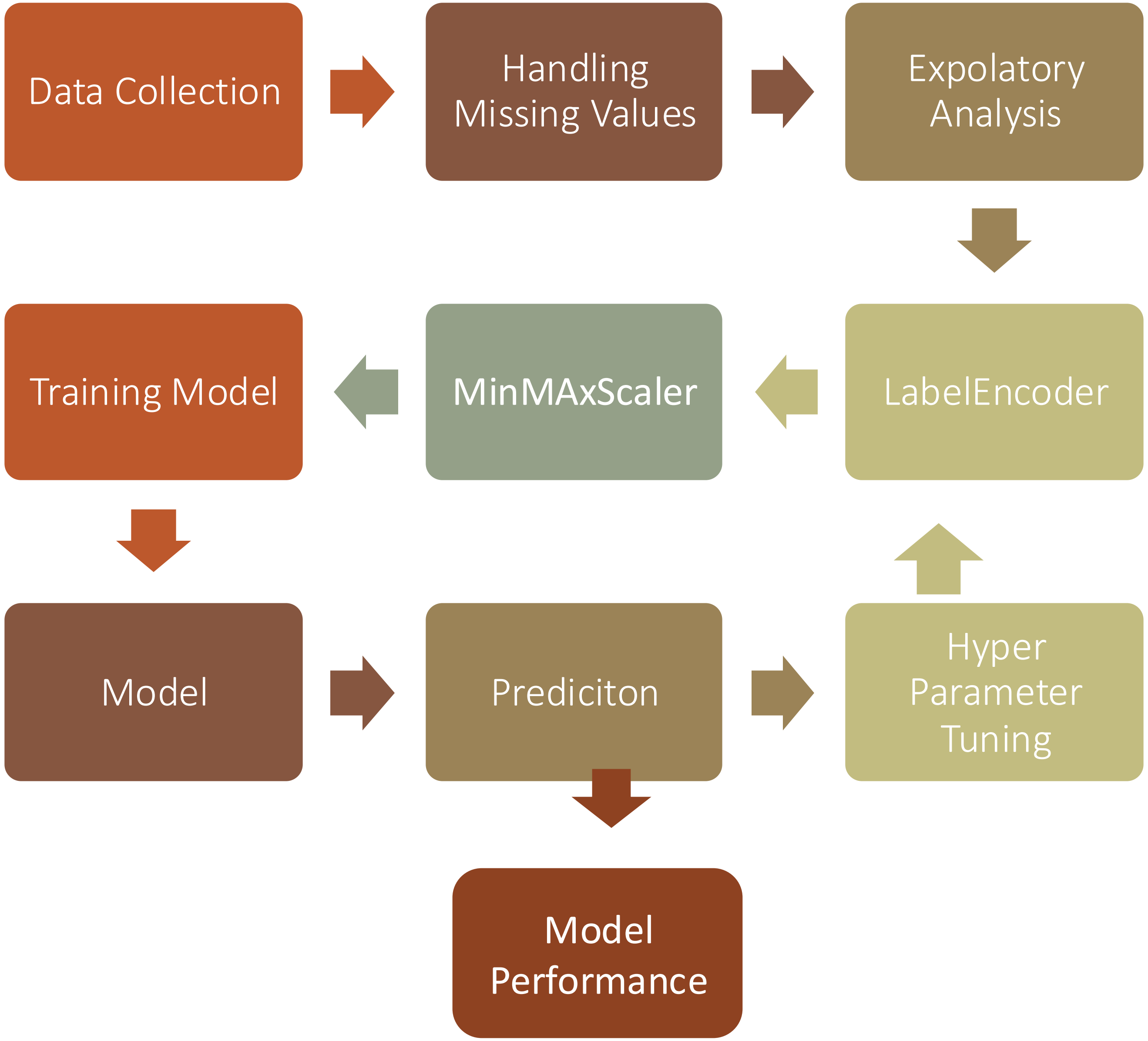
-CO Emission from Fuel Based Vehicle

-CO_2 Emission from Battery Based Vehicle

-Co_2 Emission plug in battery vehicle

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7385 entries, 0 to 7384
Data columns (total 12 columns):
 #   Column                            Non-Null Count  Dtype  
---  ------                            --------------  -----  
 0   Make                              7385 non-null   object 
 1   Model                             7385 non-null   object 
 2   Vehicle Class                     7385 non-null   object 
 3   Engine Size(L)                    7385 non-null   float64
 4   Cylinders                         7385 non-null   int64  
 5   Transmission                      7385 non-null   object 
 6   Fuel Type                         7385 non-null   object 
 7   Fuel Consumption City (L/100 km)  7385 non-null   float64
 8   Fuel Consumption Hwy (L/100 km)   7385 non-null   float64
 9   Fuel Consumption Comb (L/100 km)  7385 non-null   float64
 10  Fuel Consumption Comb (mpg)       7385 non-null   int64  
 11  CO2 Emissions(g/km)               7385 non-null   int64  
dtypes: float64(4), int64(3), object(5)
memory usage: 692.5+ KB
```

Data Analysis Pipeline



Data pre-processing

- Import that datasets

- Drop the null features

```
[8] df=df[df['Recipient Province / Territory '] != 'Nunavut']
```

```
df=df.drop("Unnamed: 18", axis=1)
```

- Handling missing values

```
for column in df.columns:
    mode_value = df[column].mode()[0]  # Calculate mode
    df[column].fillna(mode_value, inplace=True)  # Replace r

print("\nDataFrame After Replacing Missing Values:")
print(df)
```

```
202203                                              BEV
202204                                              BEV
202205                                              BEV

      BEV/PHEV/FCEV - Battery equal to or greater than 15 k
0                                                    YES
1                                                    YES
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 202206 entries, 0 to 202205
Data columns (total 19 columns):
 #   Column
---  ------
 0   Incentive Request Date
 1   Month and Year
 2   Government of Canada Fiscal Year (FY)
 3   Calendar Year
 4   Dealership Province / Territory
 5   Dealership Postal Code
 6   Purchase or Lease
 7   Vehicle Year
 8   Vehicle Make
 9   Vehicle Model
 10  Vehicle Make & Model
 11  Battery-Electric Vehicle (BEV), Plug-in Hybrid Electric Vehicle (PHEV)
 12  BEV/PHEV/FCEV - Battery equal to or greater than 15 kWh or
Electric range equal to or greater than 50 km
 13  BEV, PHEV  ? 15 kWh or PHEV < 15 kWh (until April 24, 2022)
and
PHEV ?  50 km or PHEV < 50 km and  FCEVs ? 50 km or FCEVs < 50 km
(April 25, 2022 onward)  202206 non-null  object
 14  Eligible Incentive Amount
 15  Individual or Organization
(Recipient)
 16  Recipient Province / Territory
 17  Country
```

```python
categorical_cols = df.select_dtypes(include=['object']).columns
encoder = LabelEncoder()
for col in categorical_cols:
    df[col] = encoder.fit_transform(df[col])
df.head()
```

	Incentive Request Date	Month and Year	Government of Canada Fiscal Year (FY)	Calendar Year	Dealership Province / Territory	Dealership Postal Code
0	0	31	0	2019	1	1772
1	1	31	0	2019	9	210
2	1	31	0	2019	9	210
3	1	31	0	2019	9	210
4	1	31	0	2019	9	710

Label Encoding

```
Make                               7385 non-null   object
Model                              7385 non-null   object
Vehicle Class                      7385 non-null   object
Engine Size(L)                     7385 non-null   float64
Cylinders                          7385 non-null   int64
Transmission                       7385 non-null   object
Fuel Type                          7385 non-null   object
Fuel Consumption City (L/100 km)   7385 non-null   float64
Fuel Consumption Hwy (L/100 km)    7385 non-null   float64
Fuel Consumption Comb (L/100 km)   7385 non-null   float64
Fuel Consumption Comb (mpg)        7385 non-null   int64
CO2 Emissions(g/km)                7385 non-null   int64
```

```python
# Extract Columns to Encode
columns_to_encode = ['Make', 'Model','Vehicle Class','Transmission','Fuel Type']

#categorical_cols = df.select_dtypes(include=['object']).columns
encoder = LabelEncoder()
for col in columns_to_encode:
    df1[col] = encoder.fit_transform(df1[col])
df1.head()
```

	Make	Model	Vehicle Class	Engine Size(L)	Cylinders	Transmission	Fuel Type	Fue
0	0	1057	0	2.0	4	14	4	
1	0	1057	0	2.4	4	25	4	
2	0	1058	0	1.5	4	22	4	
3	0	1233	11	3.5	6	15	4	
4	0	1499	11	3.5	6	15	4	

Label Encoding

```python
# Initialize Min-Max Scaler
scaler = MinMaxScaler()

# Scale Only Numerical Columns
numerical_cols = ['Engine Size(L)','Cylinders','Fuel Consumption City (L/100 km)','Fuel Consumption Hwy (L/100 km)',
                  'Fuel Consumption Comb (L/100 km)','Fuel Consumption Comb (mpg)','CO2 Emissions(g/km)']
df1[numerical_cols] = scaler.fit_transform(df1[numerical_cols])

print(df1)
```

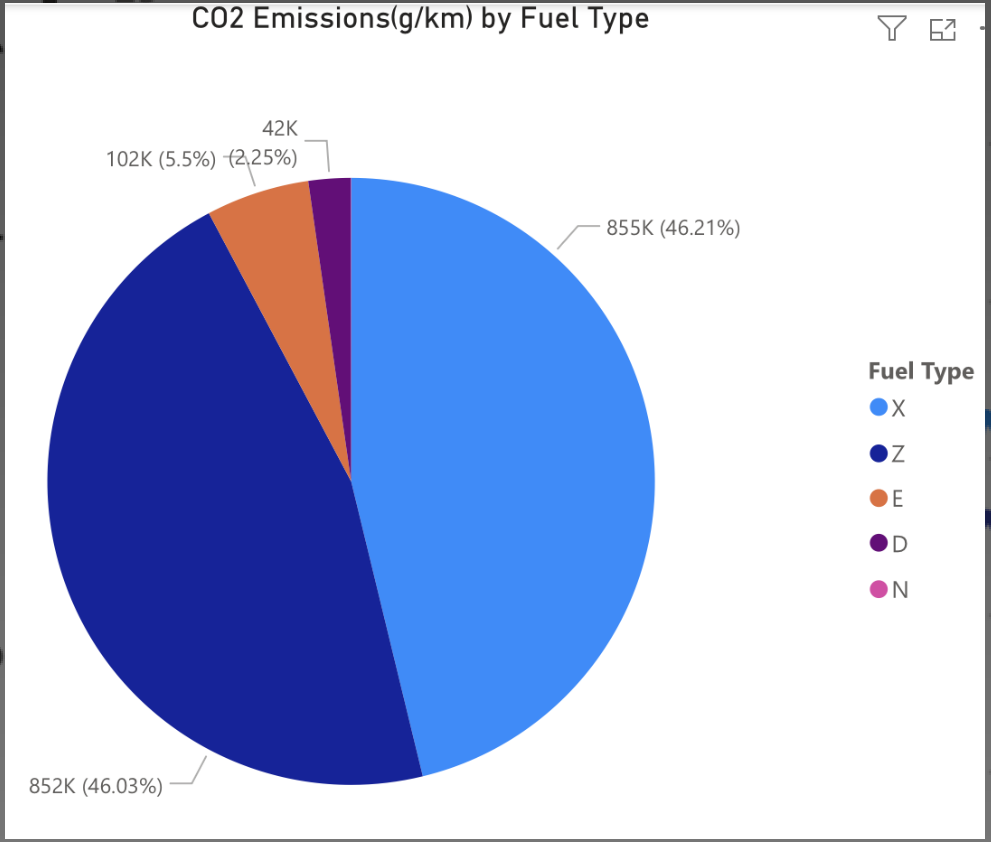
```
      Make  Model  Vehicle Class  Engine Size(L)  Cylinders  Transmission  \
0        0   1057              0        0.146667   0.076923            14
1        0   1057              0        0.200000   0.076923            25
2        0   1058              0        0.080000   0.076923            22
3        0   1233             11        0.346667   0.230769            15
4        0   1499             11        0.346667   0.230769            15
...    ...    ...            ...             ...        ...           ...
7380    41   1951             11        0.146667   0.076923            17
7381    41   1957             11        0.146667   0.076923            17
7382    41   1960             11        0.146667   0.076923            17
7383    41   1968             12        0.146667   0.076923            17
7384    41   1969             12        0.146667   0.076923            17

      Fuel Type  Fuel Consumption City (L/100 km)  \
0             4                          0.215909
1             4                          0.265152
2             4                          0.068182
3             4                          0.321970
4             4                          0.299242
...         ...                               ...
7380          4                          0.246212
7381          4                          0.265152
```

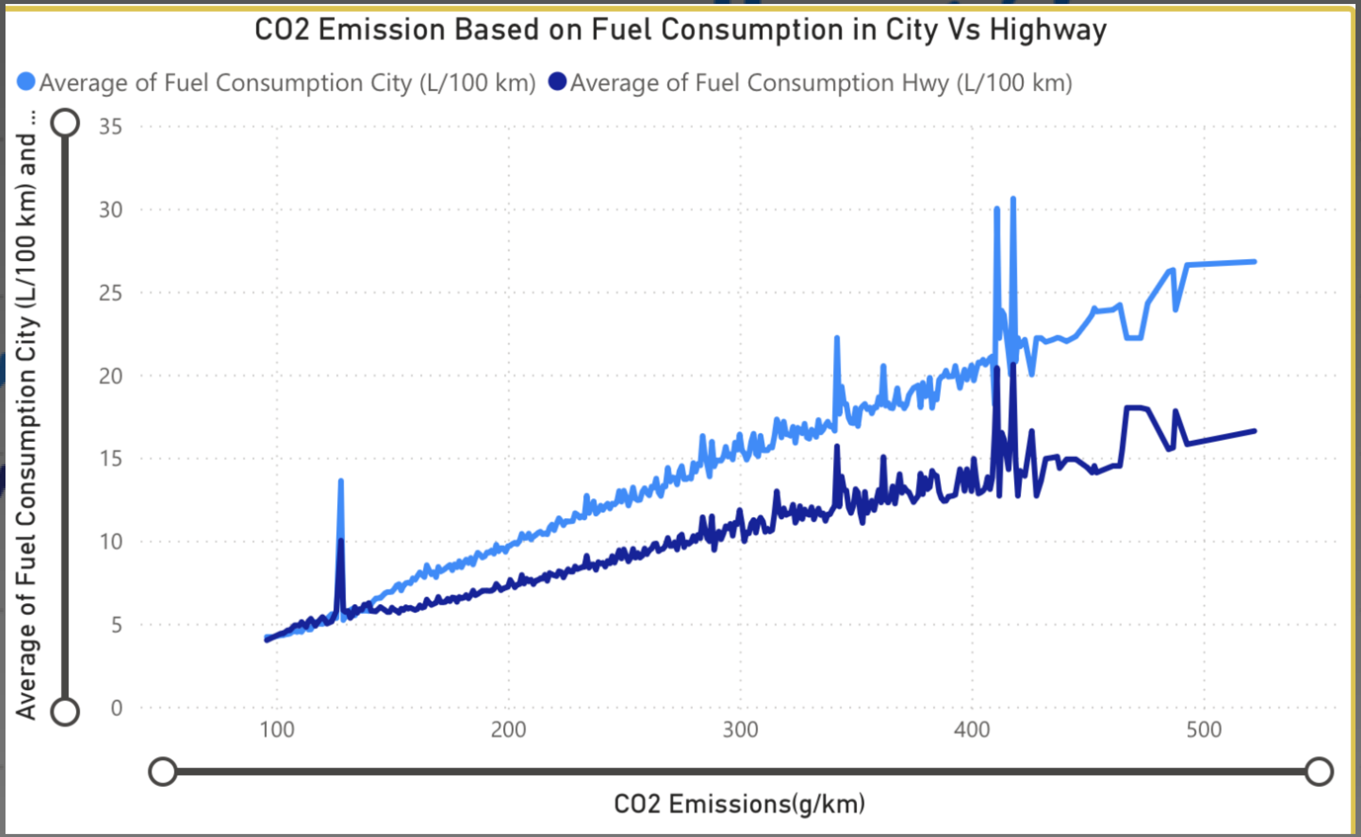
```python
scaler = MinMaxScaler()

# Scale Only Numerical Columns
numerical_cols = ['Calendar Year','Vehicle Year']
df[numerical_cols] = scaler.fit_transform(df[numerical_cols])

print(df)
```

```
        Incentive Request Date  Month and Year  \
0                            0              31
1                            1              31
2                            1              31
3                            1              31
4                            1              31
...                        ...             ...
202201                    1396              30
202202                    1396              30
202203                    1396              30
202204                    1396              30
202205                    1396              30

        Government of Canada Fiscal Year (FY)  Calendar Year  \
0                                           0            0.0
1                                           0            0.0
```

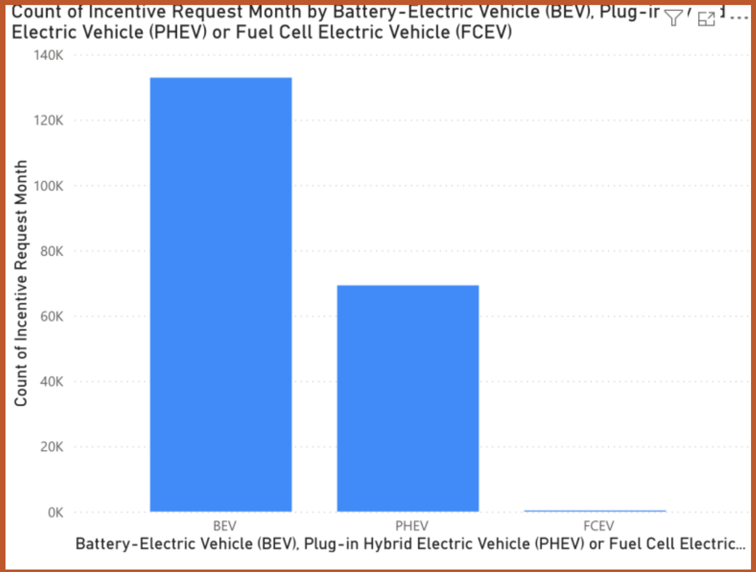
Feature Scaling

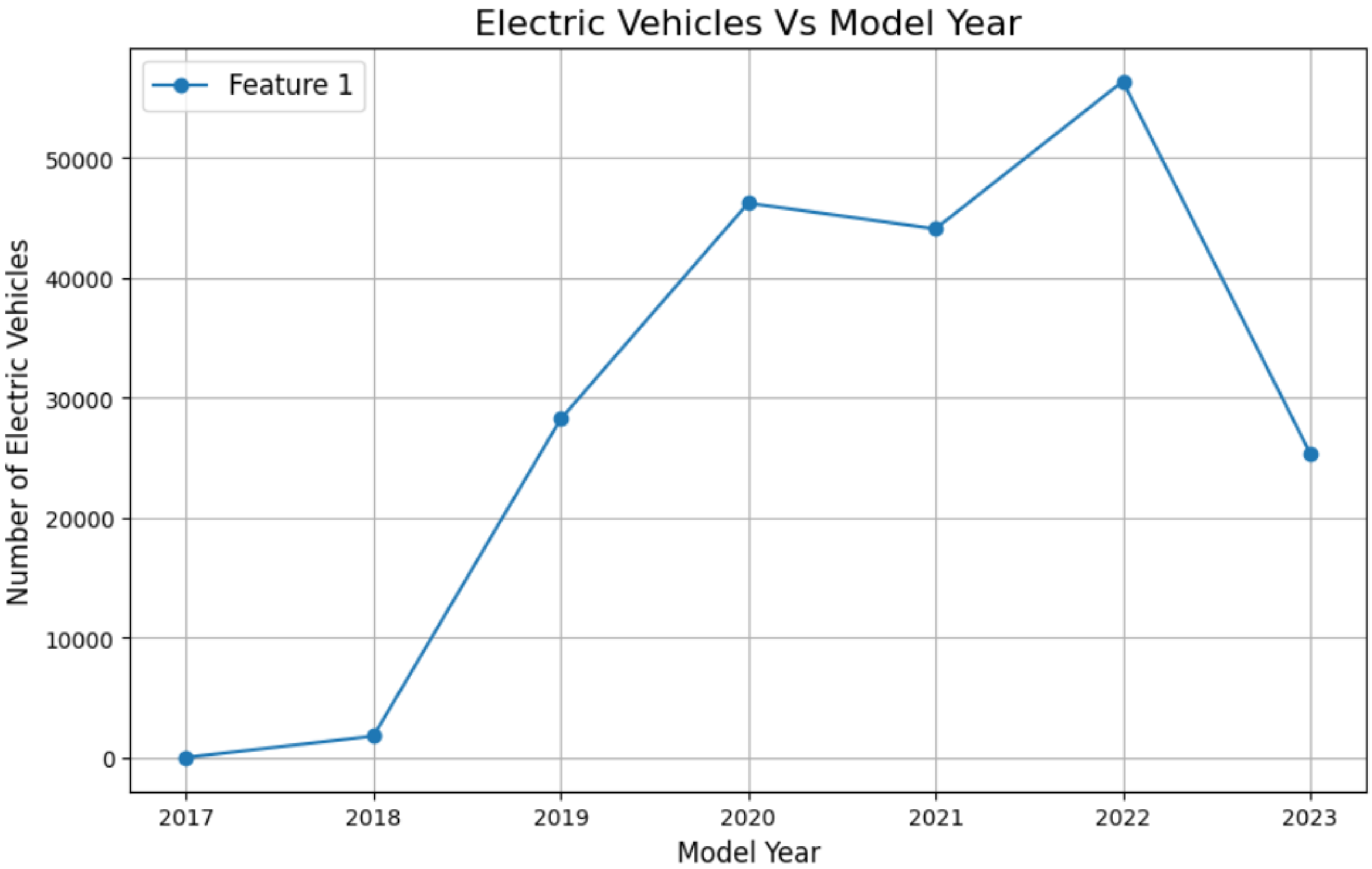
Analysis of Co2 Emission based of Fuel type



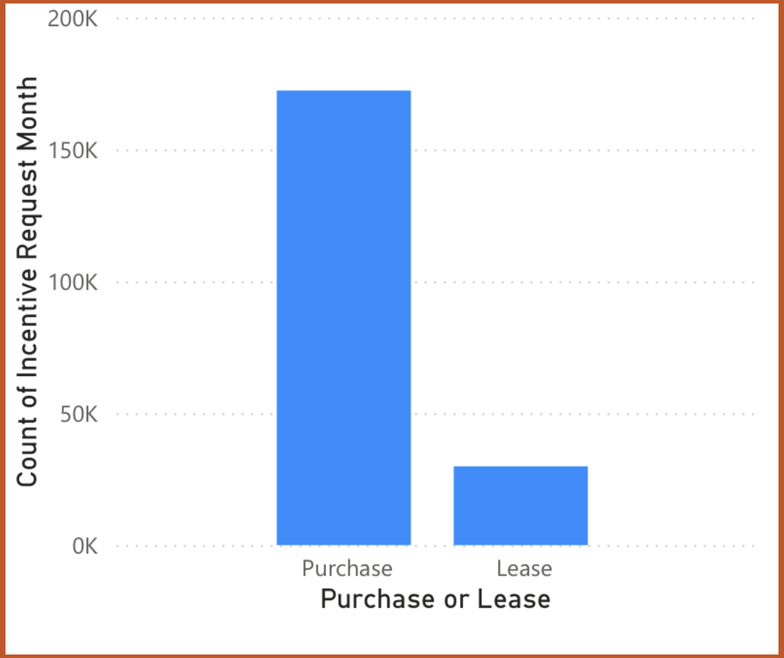


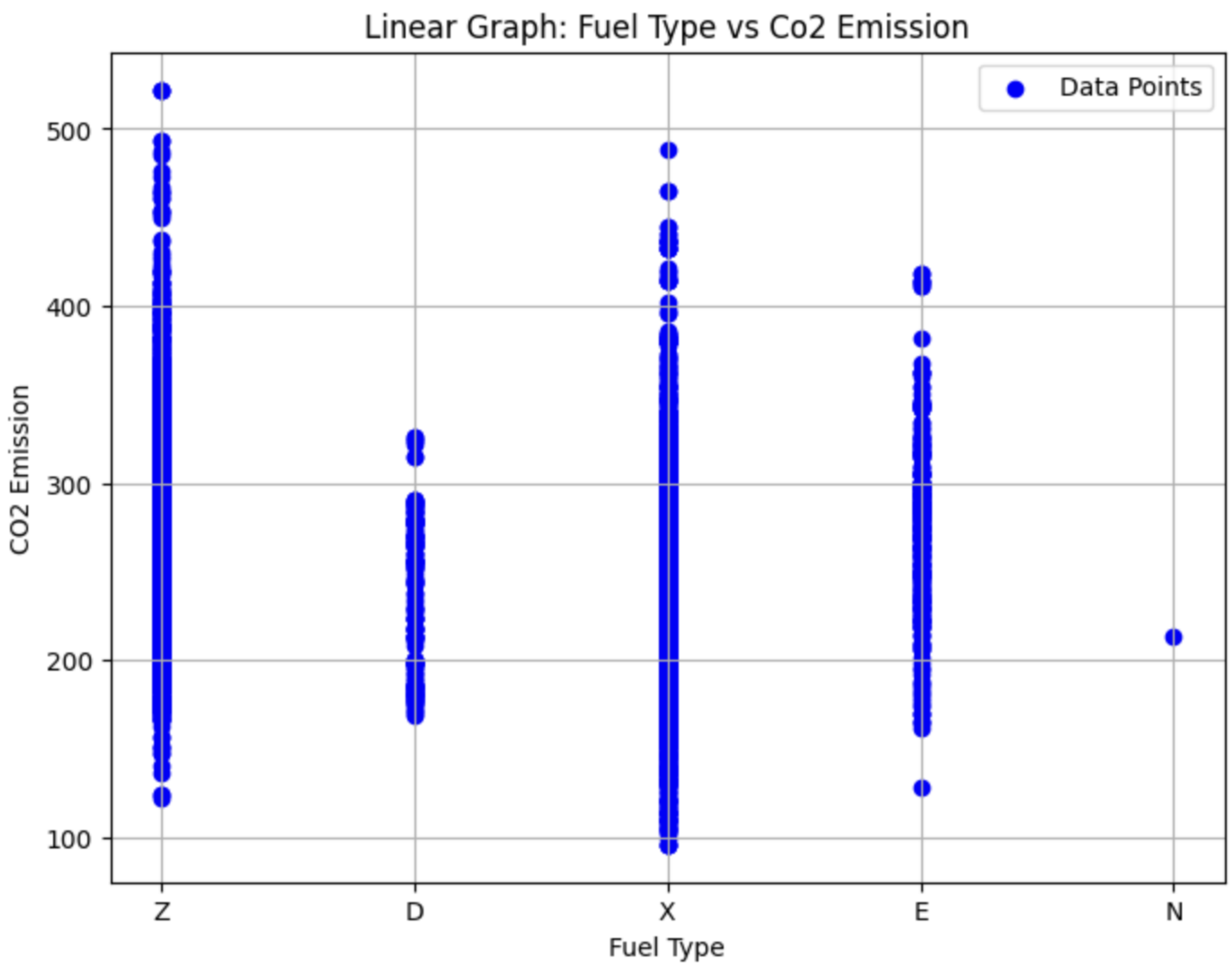
Electric Vehicles over years





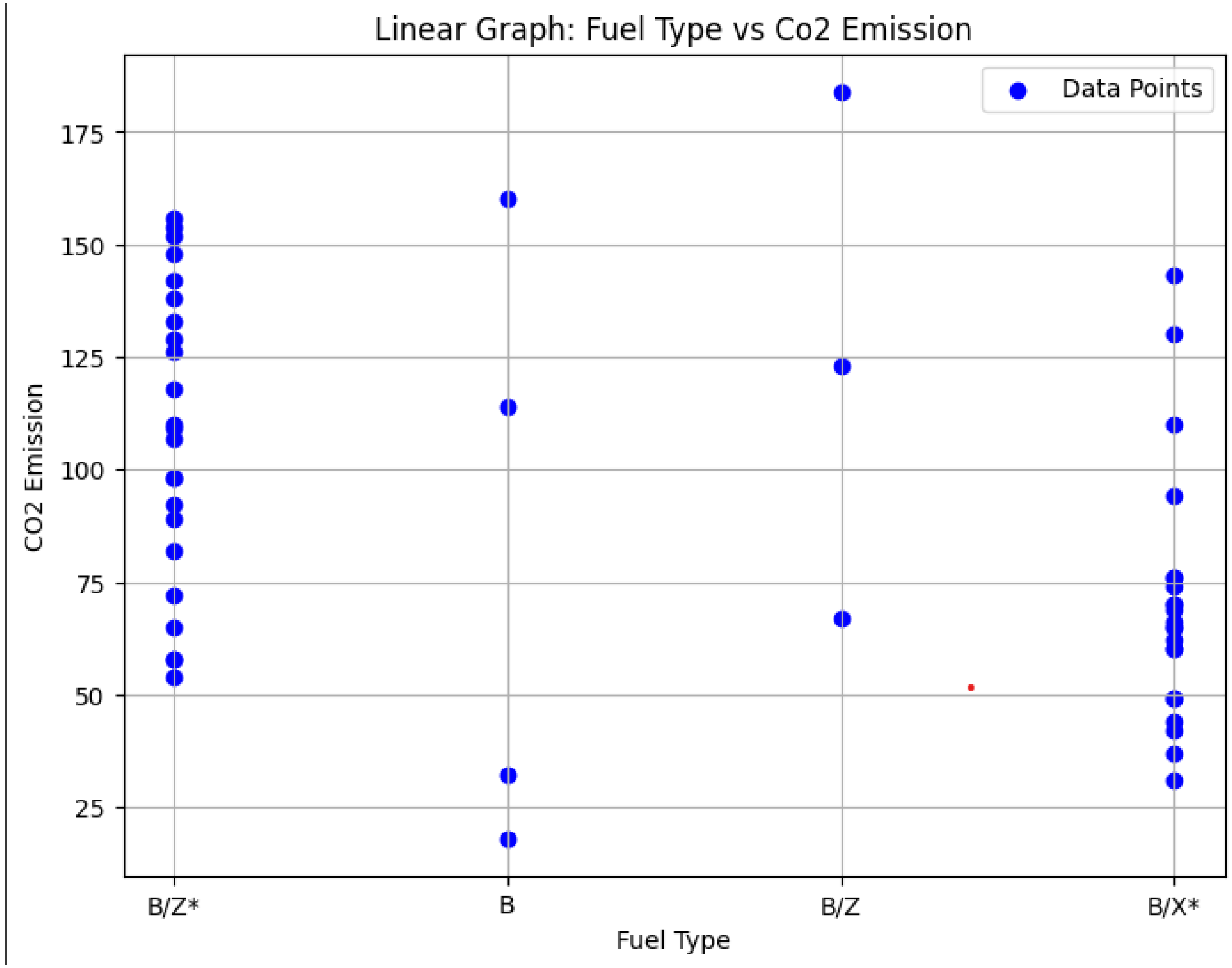
Sales of Electric Vehicles in different regions of Canada





CO2 Emission by Vehicle

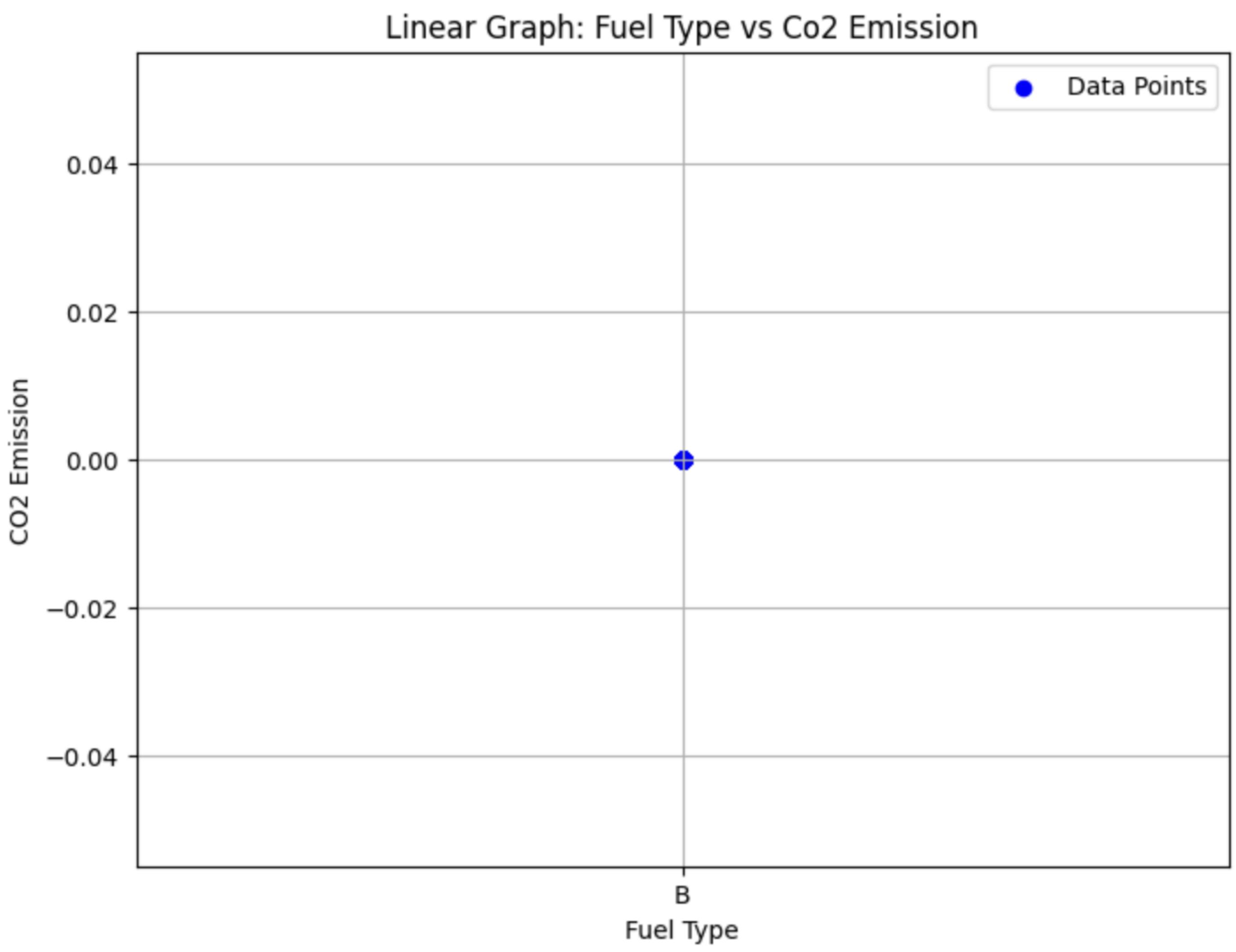
Z= Premium gasolina
D= Diesel
X= Regular gasoline
E= E85
N= Natural Gas



CO2 Emission by Plug-In Battery Vehicle

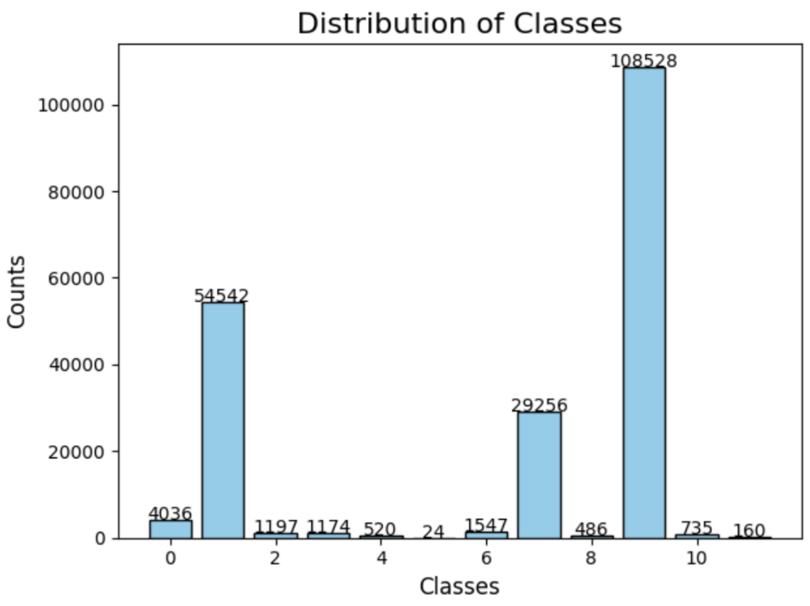
Z= Premium gasolina
D= Diesel
X= Regular gasoline
B= Battery

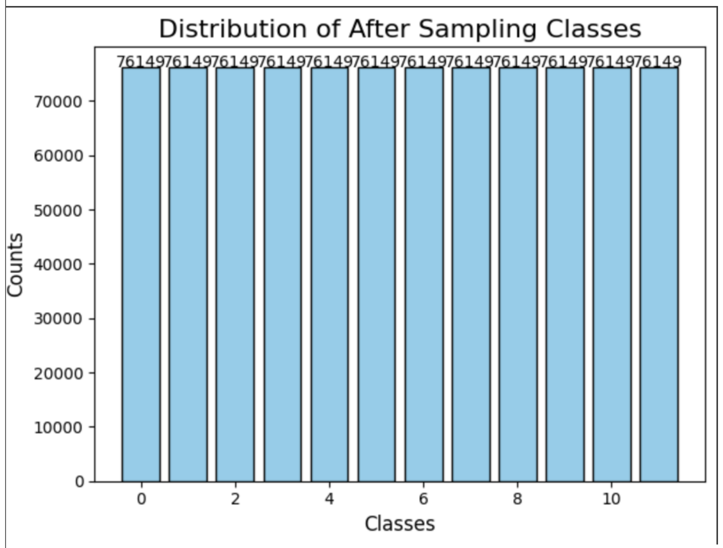
*Indicates that in testing, gasoline may have been used for following a full change.



CO2 Emission by Battery Vehicles

B= Battery



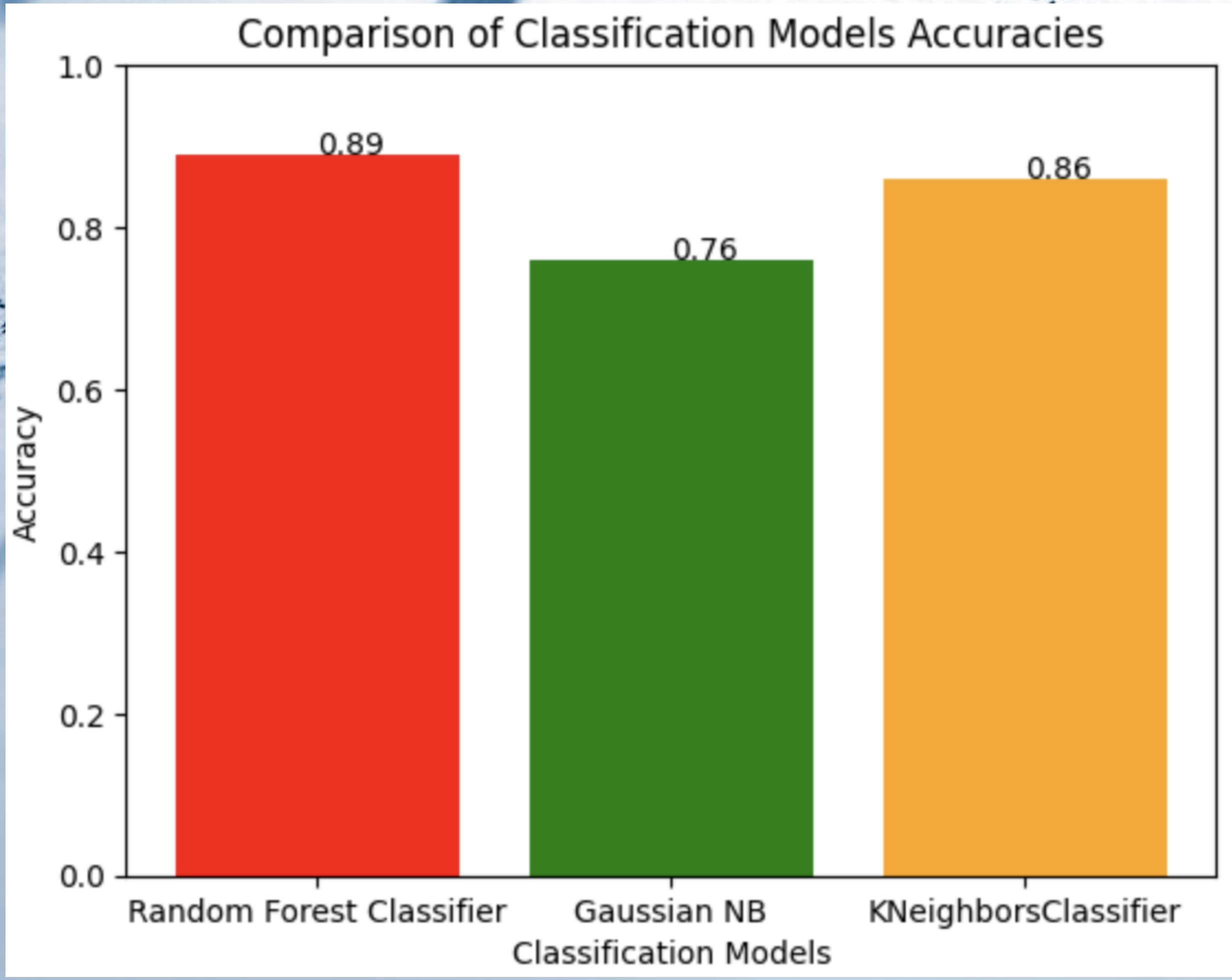



```python
from imblearn.over_sampling import SMOTE
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_
smote = SMOTE(random_state=42)
X_train_oversampled, y_train_oversampled = smote.fit_resample(X_train, y_train)
from sklearn.ensemble import RandomForestClassifier
```

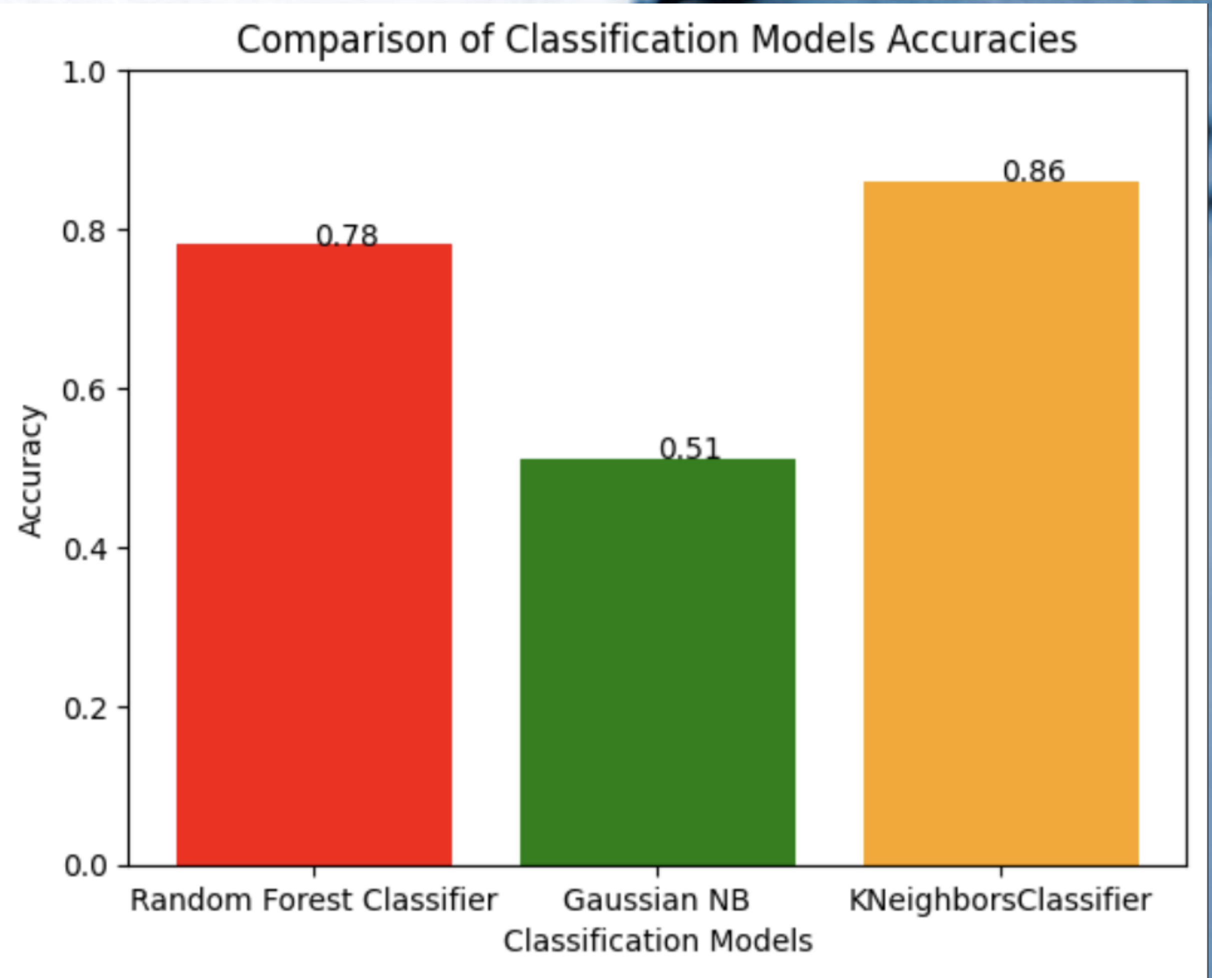
Classes distribution

CLASSIFICATION MODEL PERFORMANCE

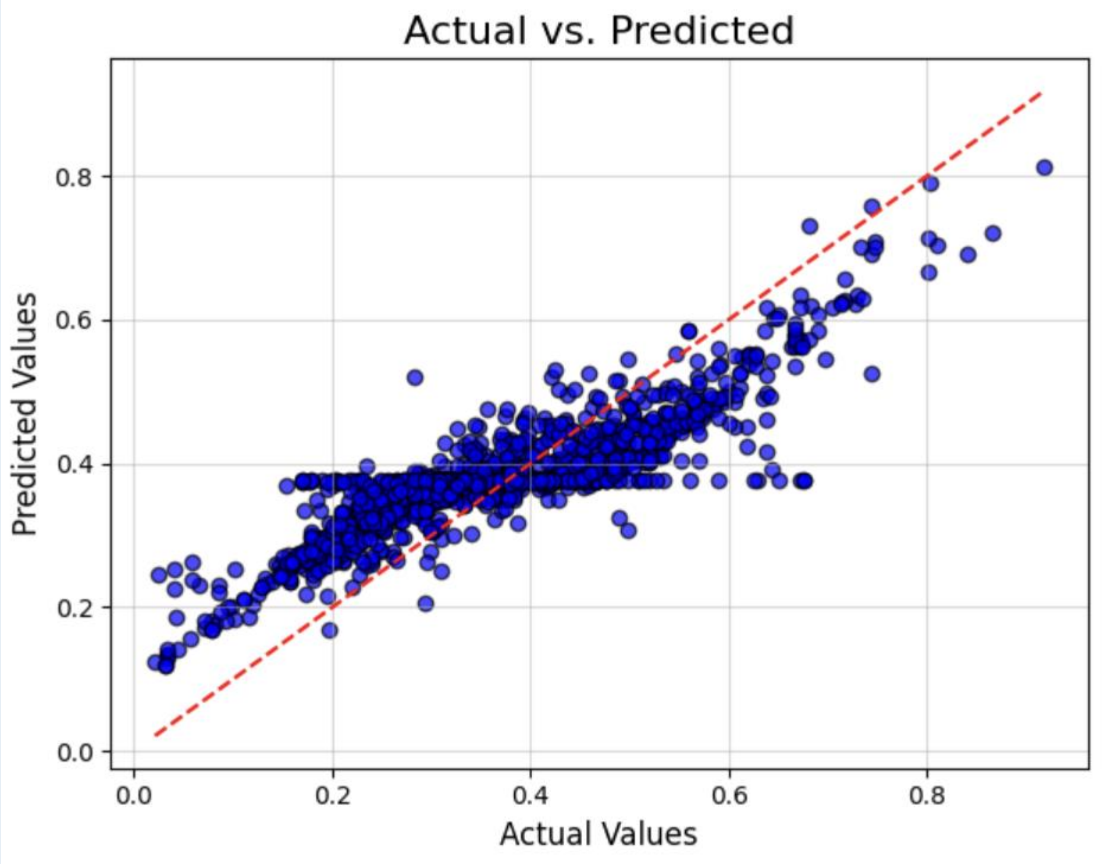
CLASSIFICATION WITHOUT SAMPLING

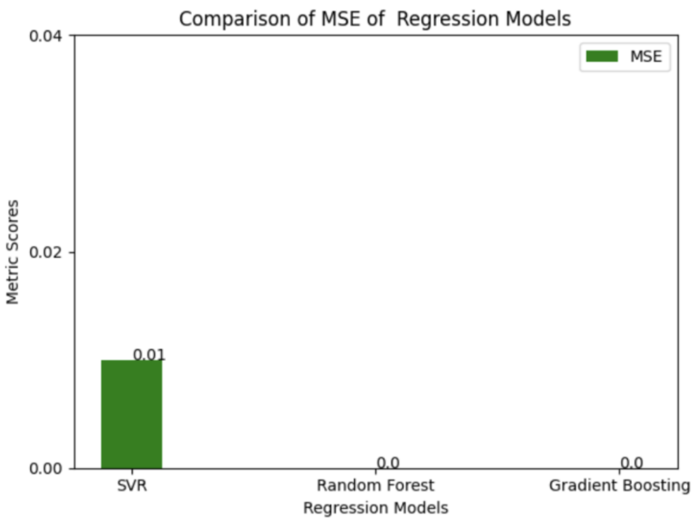


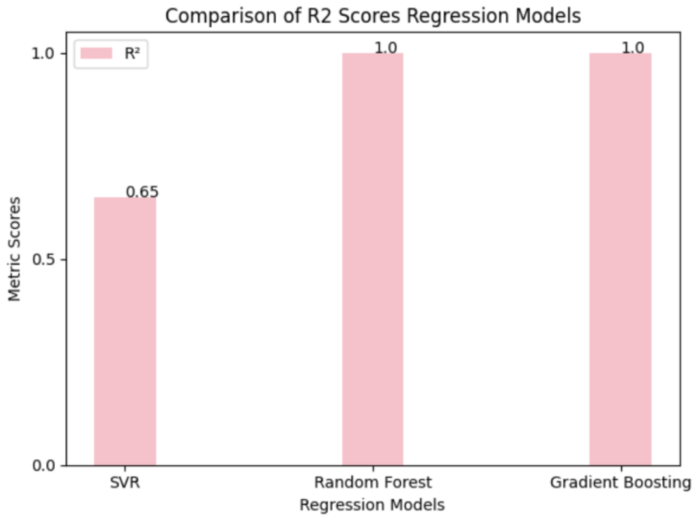
CLASSIFICATION WITH OVERSAMPLING

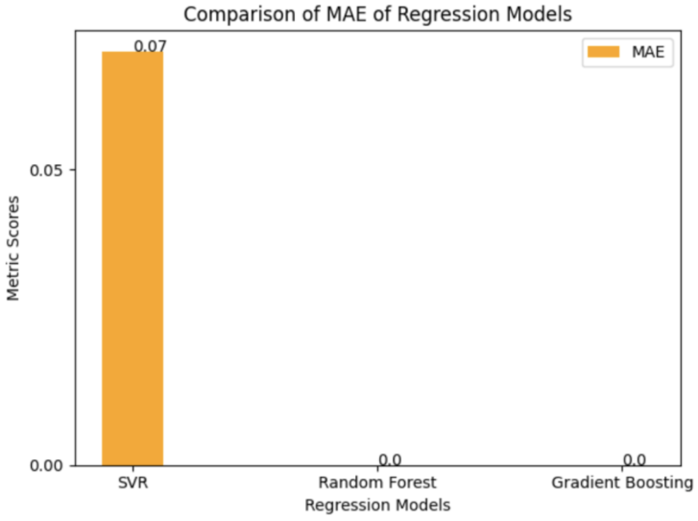


Actual Vs Predicted Plot









REGRESSOR MODEL PERFORMANCE

Limitation

Data Availability:

- The models relied on limited datasets. Incorporating larger and more diverse datasets would improve generalizability.

Feature Engineering:

- While key predictors were identified, additional features such as consumer behavior, government incentives, and geographic data could further enhance model performance.

Class Imbalance:

- Addressing class imbalance was critical for EV adoption classification. Techniques like SMOTE improved results but may introduce synthetic noise in some cases.

Future Work

- **Incorporating Additional Features**:
  - Integrating **government policies**, **consumer surveys**,
  - and **real-time sensor data** for more comprehensive predictions.
- **Advanced Machine Learning Techniques**:
- Exploring **deep learning models** such as **Neural Networks** and
- **Time-Series Analysis** to capture temporal and complex relationships.

Data Analysis using Power BI

https://app.powerbi.com/onedrive/open?pbi_source=ODSPViewer&driveId=b!zQgZvnW1Bk2OOVz4IrQ4ZMfLEgAV-ExGlRQeMCCk4vkQEbRbhdDkRqLqxu8oshTW&itemId=01YWVXE4KRXDZIEQ5BANDKAD5S5G3FGOFA

https://app.powerbi.com/onedrive/open?pbi_source=ODSPViewer&driveId=b!zQgZvnW1Bk2OOVz4IrQ4ZMfLEgAV-ExGlRQeMCCk4vkQEbRbhdDkRqLqxu8oshTW&itemId=01YWVXE4PH6YF47LUKHJD37NAW5XCBFVII

Dataset for EV Vehicle in Canada

Source: Government of Canada

Link for Dataset:

https://open.canada.ca/data/en/dataset/42986a95-be23-436e-af15-7c6bf292a2e1/resource/bba4c959-53ca-4d23-9cde-da3ce771bba2

Dataset for Co2 emission from Vehicle Canada

Source: Kaggle

Link for Dataset:

https://www.kaggle.com/datasets/debajyotipodder/co2-emission-by-vehicles

Thank you!