# Experiment 5

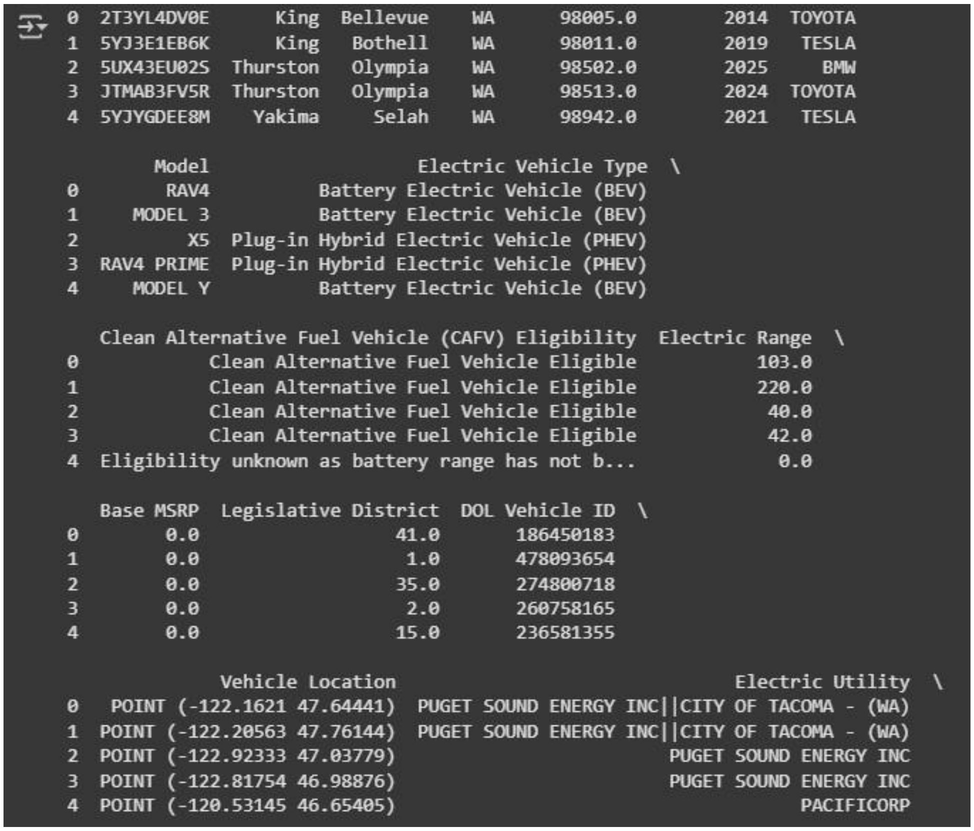
**Aim**: Perform Regression Analysis using Scipy and Sci-kit learn.

a) Perform Logistic regression to find out relation between variables

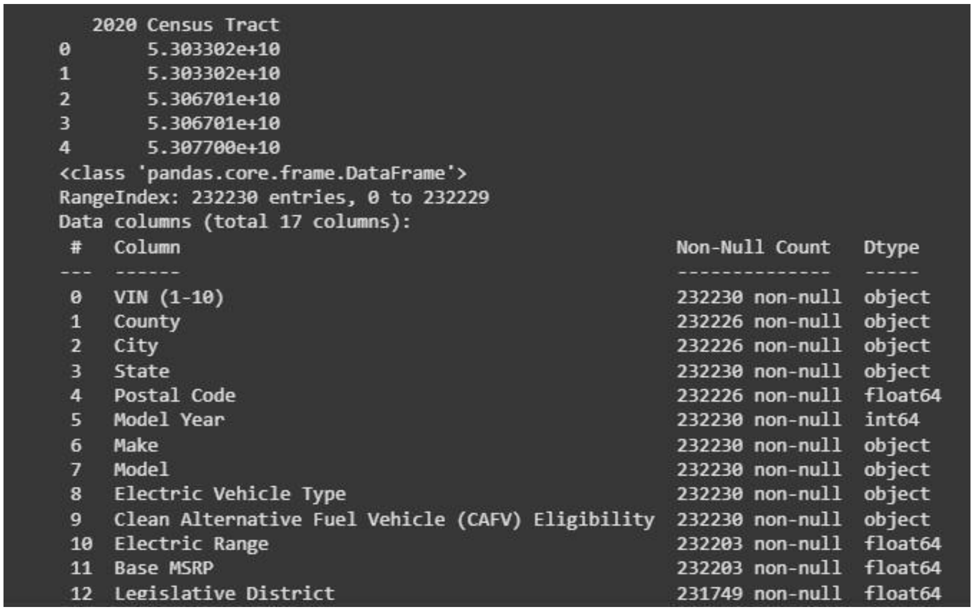
b) Apply regression model technique to predict the data on the above dataset.

**Performance:**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns from sklearn.model_selection
import train_test_split from sklearn.preprocessing
import StandardScaler from sklearn.linear_model
import LogisticRegression, LinearRegression from sklearn.metrics
import accuracy_score, classification_report, confusion_matrix,
mean_absolute_error, mean_squared_error, r2_score

df = pd.read_csv('set3.csv') print(df.head())
print(df.info())
```



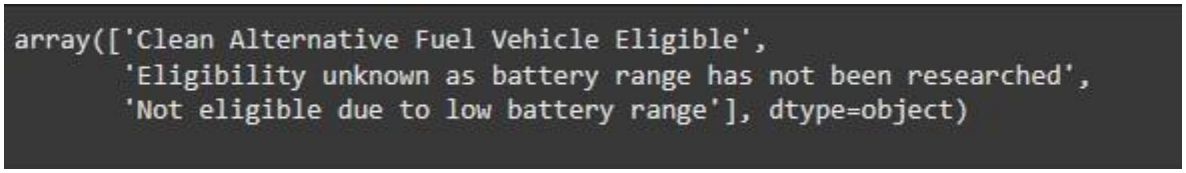
```
0  2T3YL4DV0E      King  Bellevue    WA       98005.0        2014  TOYOTA
1  5YJ3E1EB6K      King   Bothell    WA       98011.0        2019   TESLA
2  5UX43EU02S  Thurston   Olympia    WA       98502.0        2025     BMW
3  JTMAB3FV5R  Thurston   Olympia    WA       98513.0        2024  TOYOTA
4  5YJYGDEE8M    Yakima     Selah    WA       98942.0        2021   TESLA

        Model                   Electric Vehicle Type  \
0        RAV4          Battery Electric Vehicle (BEV)
1     MODEL 3          Battery Electric Vehicle (BEV)
2          X5  Plug-in Hybrid Electric Vehicle (PHEV)
3  RAV4 PRIME  Plug-in Hybrid Electric Vehicle (PHEV)
4     MODEL Y          Battery Electric Vehicle (BEV)

   Clean Alternative Fuel Vehicle (CAFV) Eligibility  Electric Range  \
0            Clean Alternative Fuel Vehicle Eligible           103.0
1            Clean Alternative Fuel Vehicle Eligible           220.0
2            Clean Alternative Fuel Vehicle Eligible            40.0
3            Clean Alternative Fuel Vehicle Eligible            42.0
4  Eligibility unknown as battery range has not b...             0.0

   Base MSRP  Legislative District  DOL Vehicle ID  \
0        0.0                  41.0       186450183
1        0.0                   1.0       478093654
2        0.0                  35.0       274800718
3        0.0                   2.0       260758165
4        0.0                  15.0       236581355

                 Vehicle Location                              Electric Utility  \
0   POINT (-122.1621 47.64441)  PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)
1  POINT (-122.20563 47.76144)  PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)
2  POINT (-122.92333 47.03779)                         PUGET SOUND ENERGY INC
3  POINT (-122.81754 46.98876)                         PUGET SOUND ENERGY INC
4  POINT (-120.53145 46.65405)                                     PACIFICORP
```



```
   2020 Census Tract
0       5.303302e+10
1       5.303302e+10
2       5.306701e+10
3       5.306701e+10
4       5.307700e+10
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 232230 entries, 0 to 232229
Data columns (total 17 columns):
 #   Column                                             Non-Null Count   Dtype
---  ------                                             --------------   -----
 0   VIN (1-10)                                         232230 non-null  object
 1   County                                             232226 non-null  object
 2   City                                               232226 non-null  object
 3   State                                              232230 non-null  object
 4   Postal Code                                        232226 non-null  float64
 5   Model Year                                         232230 non-null  int64
 6   Make                                               232230 non-null  object
 7   Model                                              232230 non-null  object
 8   Electric Vehicle Type                              232230 non-null  object
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility  232230 non-null  object
 10  Electric Range                                     232203 non-null  float64
 11  Base MSRP                                          232203 non-null  float64
 12  Legislative District                               231749 non-null  float64
```

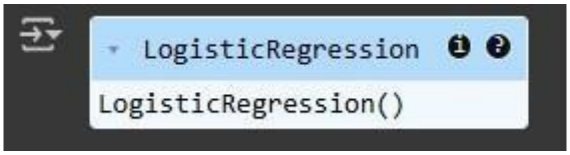
a) Perform Logistic regression to find out relation between variables:

```
df['Clean Alternative Fuel Vehicle (CAFV) Eligibility'].unique()
```



```
array(['Clean Alternative Fuel Vehicle Eligible',
       'Eligibility unknown as battery range has not been researched',
       'Not eligible due to low battery range'], dtype=object)
```

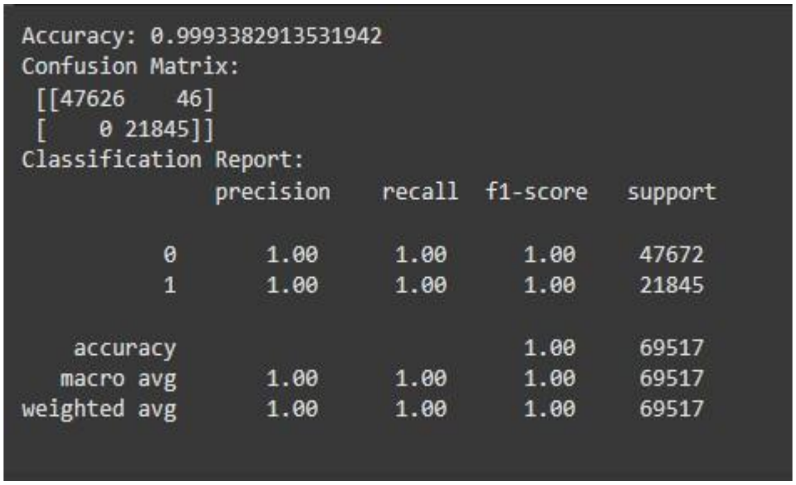
```
df_selected = df[['Model Year', 'Electric Range', 'Base MSRP', 'Legislative District']]
df_selected = df_selected.dropna() X = df_selected y = df.loc[df_selected.index,
'Eligibility_Binary']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
logreg = LogisticRegression()
logreg.fit(X_train_scaled, y_train)
```



```
LogisticRegression()
```

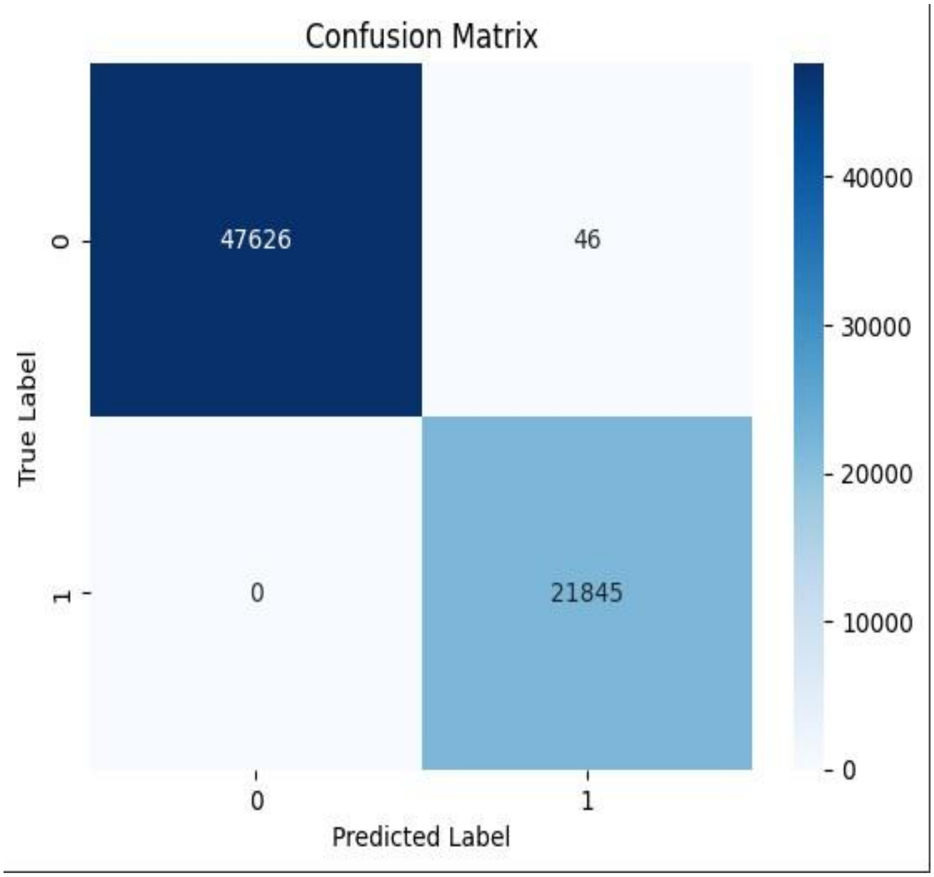
This step initializes a Logistic Regression model using LogisticRegression().

```
y_pred = logreg.predict(X_test_scaled)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```



```
Accuracy: 0.9993382913531942
Confusion Matrix:
 [[47626    46]
 [    0 21845]]
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00     47672
           1       1.00      1.00      1.00     21845

    accuracy                           1.00     69517
   macro avg       1.00      1.00      1.00     69517
weighted avg       1.00      1.00      1.00     69517
```

```
sns.heatmap(confusion_matrix(y_test, y_pred),
annot=True, fmt='d',cmap='Blues')
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.show()
```

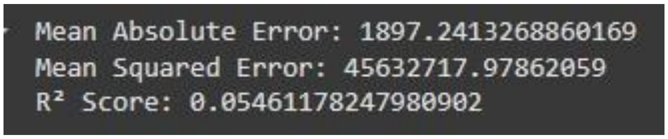


This step visualizes the confusion matrix using Seaborn's heatmap() function.

b) Apply regression model technique to predict the data on the above dataset:

```
y_reg = df_selected['Base MSRP']
X_reg = df_selected.drop(['Base MSRP'], axis=1)
X_train_reg, X_test_reg, y_train_reg, y_test_reg =
train_test_split(X_reg, y_reg, test_size=0.3,
random_state=42)
scaler_reg = StandardScaler()
X_train_reg_scaled =
scaler_reg.fit_transform(X_train_reg)
```

```
X_test_reg_scaled = scaler_reg.transform(X_test_reg)
linreg = LinearRegression()
linreg.fit(X_train_reg_scaled, y_train_reg)
 y_pred_reg = linreg.predict(X_test_reg_scaled)
print("Mean Absolute Error:",
mean_absolute_error(y_test_reg, y_pred_reg))
print("Mean Squared Error:",
mean_squared_error(y_test_reg, y_pred_reg))
print("R² Score:", r2_score(y_test_reg, y_pred_reg))
```



```
Mean Absolute Error: 1897.2413268860169
Mean Squared Error: 45632717.97862059
R² Score: 0.05461178247980902
```

This step evaluates the Linear Regression model's performance using three key metrics. Mean Absolute Error (MAE) measures the average absolute difference between actual and predicted values, while Mean Squared Error (MSE) penalizes larger errors more heavily.

## Conclusion:

The Logistic Regression model demonstrated strong classification performance with an accuracy of 99.93%, Meanwhile, the Linear Regression model for predicting base MSRP showed moderate predictive accuracy, with an $R^2$ score of 0.05, significant errors in both MAE and MSE suggest that the model could benefit from additional relevant features. In summary, the regression model exhibited limited predictive power for Base MSRP.