SUMMER TRAINING/INTERNSHIP

PROJECT REPORT

(Term June - July 2025)

(Avocado Ripeness Analysis**)**

Submitted by

Name of Students: Khushi Sharma

Avichal Srivastava

Sneha Mondal

Sakshi Sinha

Registration Number: 12308323

12308644

12308247

12311346

Course Code: PETV76

Under the Guidance of

Dr. **Sandeep Kaur (UID: 23614)**

School of Computer Science and Engineering

Certificate

This is to certify Khushi Sharma**,**Avichal Srivastava,Sneha Mondal,Sakshi Sinha are students of, PETV76, Lovely Professional University has successfully completed the project titled “Electric Vehicle Range Prediction” as part of the internship/training during the period [10/6/2025] to [15/7/2025].

Acknowledgement

I would like to express my sincere gratitude to **Sandeep Kaur**, my project guide, for his/her guidance and continuous support throughout the duration of this project. I also extend my thanks to the faculty and staff of, PETV76, Lovely Professional University, and my peers for their encouragement. This internship has been a valuable experience that has enhanced my practical knowledge and confidence in the field of data analytics and business intelligence.

I would also like to thank the company for providing access to tools and technologies and for the constant encouragement and constructive feedback throughout the project.

Table of Contents

Chapter 1: Introduction

About the Project

In this project, we developed a predictive model to estimate the driving range of electric vehicles (EVs) under various real-world conditions such as battery capacity, speed, terrain, and environmental factors. We used Python and machine learning for exploratory data analysis (EDA) and model building, applying regression techniques to identify key factors influencing EV range. Power BI was leveraged for data cleaning and to create interactive visualizations that showcased range distributions, key influencing variables, and model-specific comparisons. This end-to-end approach enabled us to deliver data-driven insights that can help manufacturers optimize EV performance and assist drivers in making more informed decisions.

Overview of Training Domain

The training focused on the domain of Data Science and Business Intelligence, with a particular emphasis on understanding data characteristics, developing machine learning models, and creating data-driven dashboards. The training enabled me to bridge the gap between theoretical knowledge and real-world applications by working on a practical problem involving the prediction of electric vehicle (EV) range. Using Python for exploratory data analysis and machine learning, and Power BI for data cleaning and visualization, I gained hands-on experience in building an end-to-end solution that estimates EV range based on variables such as Model Year, Make,Model,Electric Vehicle Type, Electric Range, Base MSRP, City, State, location conditions.

Objective of the Project

The objective of this project was to estimate the driving range of electric vehicles (EVs) under various real-world conditions using measurable inputs such as battery capacity, speed, terrain, and environmental factors. The project aimed to:

- Build a regression-based predictive model to forecast EV range.
- Perform exploratory data analysis (EDA) to understand the key variables influencing range.
- Present data insights and predictions through interactive Power BI dashboards.

• Support informed decision-making for both EV manufacturers and drivers by offering model-specific comparisons and range estimations.

This project not only demonstrated technical proficiency in machine learning and data visualization but also emphasized the delivery of actionable, real-world insights to enhance EV design and user experience.

Chapter 2: Training Overview

Tools & Technologies Used

The training provided hands-on experience with the following tools and technologies:

- **Python**: Used for data analysis, preprocessing, and modeling using libraries like Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn.
- **Jupyter Notebook**: Interactive development environment for documenting the entire analysis.
- **Power BI**: Used for creating interactive dashboards and data visualization.
- **CSV Files**: Format used for importing EV dataset.

Areas Covered During Training

- Basics of data collection, exploration, and transformation.
- Understanding different types of machine learning models.
- Techniques for model evaluation and optimization.
- Building dynamic dashboards using Power BI.
- Communicating insights through visuals and reports.

The structured training helped in acquiring both technical and soft skills required for industry-level problem-solving.

Chapter 3: Project Details

Title of the Project

Electric Vehicle Range Prediction

Problem Definition

Accurately estimating the driving range of electric vehicles (EVs) is crucial for improving user experience and supporting EV adoption. Factors such as battery capacity, speed, terrain, and environmental conditions significantly impact the actual range of an EV. Traditional range estimations often lack precision and fail to consider dynamic conditions. This project focuses on applying data analytics and machine learning to predict EV driving range more reliably under real-world scenarios.

Scope and Objectives

- Predict the driving range of electric vehicles using variables like battery capacity, speed, terrain, and environmental conditions.
- Identify key factors that influence range through exploratory data analysis.
- Visualize model outputs, distributions, and variable impacts using Power BI dashboards.
- Deliver an adaptable and data-driven solution that aids both EV manufacturers and end users.

Architecture Diagram

Data Source → Data Preprocessing (Python) → Model Training → Evaluation → Visualization (Power BI)

Data Flow / UML Diagrams

A simple data flow:

1. Load dataset
2. Clean and preprocess
3. Train ML model
4. Predict ripeness
5. Visualize results in dashboard

Chapter 4: Implementation

• Tools Used

- Python
- Power BI

• Methodology

The dataset was cleaned and analyzed using Python to understand the factors affecting EV range. Regression models were applied to predict driving range based on inputs such as Model Year, Make,Model,Electric Vehicle Type, Electric Range, Base MSRP, City, State, location conditions. The results were visualized in Power BI to highlight key influencing variables and model-specific comparisons.

• Code Snippets (EDA)

```
# 🔧 Core Libraries
 import numpy as np
import pandas as pd
 # 🔧 Data Visualization Libraries
import matplotlib.pyplot as plt
import seaborn as sns
# 🔧 Excel File Handling (if needed)
import openpyxl
📥 Load the Dataset
 df = pd.read_csv('Electric_Vehicle_Population_Data.csv')
# Display the first few rows
```

```
print(df.head(5))
```

Basic Info & Overview

```
# Shape and columns
print("Shape:", df.shape)
print("\nColumns:\n", df.columns)
print("\nData Types:\n", df.dtypes)
# Check for missing values
print("\nMissing Values:\n", df.isnull().sum())
# Summary stats for numeric columns
df.describe()
```

#1.Distribution of Model Years

```
# Now use the corrected column names
plt.grid(axis='y', linestyle='--', alpha=0.5)
# Filter recent model years
recent = df[df['Model Year'] >= 2015]
# Histogram of recent model years
sns.histplot(recent['Model Year'], bins=10, color='salmon')
plt.title("Distribution of EVs (Model Year ≥ 2015)")
plt.xlabel("Model Year")
plt.ylabel("Count")
plt.show()
```

#2. Countplot of all model years

```
sns.countplot(x='Model Year', data=df, color='skyblue')
```

```
plt.title("Count of EVs by Model Year")
plt.xlabel("Model Year")
plt.ylabel("Count")
 plt.xticks(rotation=45)
plt.tight_layout()
 plt.show()
#3. EV Range vs Base Price
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='Electric Vehicle Range(Miles)', y='Base Price ($)', alpha=0.3)
plt.title("EV Range vs Base Price")
plt.xlabel("Electric Vehicle Range (Miles)")
 plt.ylabel("Base Price ($)")
plt.tight_layout()
plt.show()
#4. Top 15 Manufacturers
top_cities = df['City'].value_counts().head(15)
# Plot
plt.figure(figsize=(12, 6))
colors = plt.cm.viridis(np.linspace(0, 1, len(top_cities)))
bars = plt.barh(top_cities.index, top_cities.values, color=colors)
plt.title("Top 15 Cities with Most Electric Vehicles (Gradient Bars)")
plt.xlabel("Number of Vehicles")
plt.gca().invert_yaxis()  # Highest on top
plt.tight_layout()
plt.show()
```

5. EV Type Pie Chart

```
ev_type_counts = df['Electric Vehicle Type'].value_counts()

plt.figure(figsize=(7, 7))

plt.pie(ev_type_counts, labels=ev_type_counts.index, autopct='%1.1f%%',
colors=['#66b3ff', '#99ff99'], startangle=140)

plt.title("Electric Vehicle Type Distribution")

plt.axis('equal')

plt.tight_layout()

plt.show()
```

6. CAFV Eligibility

```
cafv_counts = df['Clean Alternative Fuel Vehicle (CAFV)
Eligibility'].value_counts()

plt.figure(figsize=(10, 5))

sns.barplot(x=cafv_counts.values, y=cafv_counts.index, palette='pastel')

plt.title("CAFV Eligibility Status")

plt.xlabel("Number of Vehicles")

plt.ylabel("CAFV Eligibility")

plt.tight_layout()

plt.show()
```

#7. 🔬 Top 15 Cities with the Most EVs

```
plt.figure(figsize=(12, 6))

plt.hlines(y=top_cities.index, xmin=0, xmax=top_cities.values,
color='lightblue')

plt.plot(top_cities.values, top_cities.index, "o", color='darkblue')

plt.title("Top 15 Cities with Most Electric Vehicles (Lollipop Chart)")

plt.xlabel("Number of Vehicles")

plt.ylabel("City")
```

```
plt.tight_layout()
plt.show()
```

8. 🚙 Average EV Range by Manufacturer (Top 15 Only)

```
# Stacked Bar Chart (Compare EV Types per City)
top_city_names = top_cities.index.tolist()
city_type_grouped = df[df['City'].isin(top_city_names)].groupby(['City', 'Electric Vehicle Type']).size().unstack().fillna(0)


city_type_grouped.plot(kind='barh', stacked=True, figsize=(12, 7), colormap='Set3')
plt.title("EV Type Distribution in Top 15 Cities (Stacked Bar)")
plt.xlabel("Number of Vehicles")
plt.ylabel("City")
plt.tight_layout()
plt.show()
```

9. 📦 Boxplot: Base Price by EV Type

```
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='Electric Vehicle Type', y='Base Price ($)', palette='Set2')
plt.title("Base Price Distribution by EV Type")
plt.xlabel("EV Type")
plt.ylabel("Base Price ($)")
plt.tight_layout()
plt.show()
```

10. 🚨 Heatmap of Correlations

```
plt.figure(figsize=(8, 5))
```

```
corr = df[['Model Year', 'Electric Vehicle Range(Miles)', 'Base Price ($)', 'Legislative District']].corr()

sns.heatmap(corr, annot=True, cmap='coolwarm', linewidths=0.5)

plt.title("Correlation Heatmap")

plt.tight_layout()

plt.show()
```

11. useing the Interquartile Range (IQR) method:

```
# Step 1: Use IQR to filter out outliers from 'Base Price ($)'

Q1 = df['Base Price ($)'].quantile(0.25)

Q3 = df['Base Price ($)'].quantile(0.75)

IQR = Q3 - Q1


# Define lower and upper bounds

lower_bound = Q1 - 1.5 * IQR

upper_bound = Q3 + 1.5 * IQR


# Step 2: Filter the DataFrame

df_no_outliers = df[(df['Base Price ($)'] >= lower_bound) & (df['Base Price ($)'] <= upper_bound)]
```

12. Replot Boxplot without outliers

```
plt.figure(figsize=(10, 6))

sns.boxplot(data=df_no_outliers, x='Electric Vehicle Type', y='Base Price ($)', palette='Set2')

plt.title("Base Price Distribution by EV Type (Outliers Removed)")

plt.xlabel("EV Type")

plt.ylabel("Base Price ($)")
```

```
plt.tight_layout()
plt.show()
```

13. Replot Heatmap without outliers

```
plt.figure(figsize=(8, 5))
corr = df_no_outliers[['Model Year', 'Electric Vehicle Range(Miles)', 'Base Price ($)', 'Legislative District']].corr()
sns.heatmap(corr, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title("Correlation Heatmap (Outliers Removed)")
plt.tight_layout()
plt.show()
```

ML

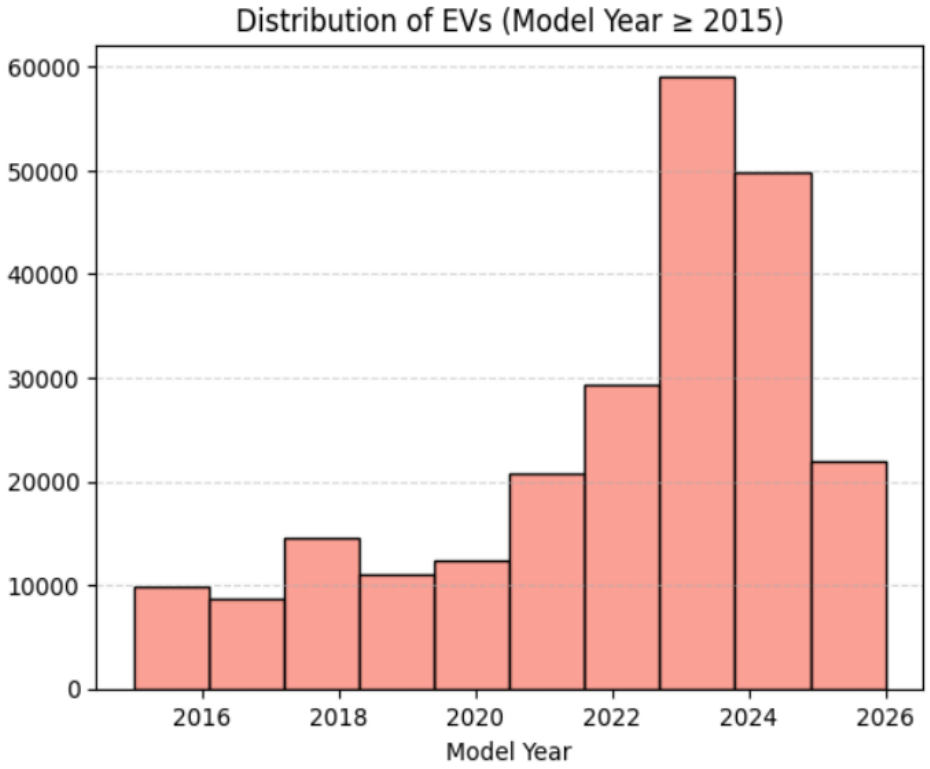
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, r2_score
from sklearn.preprocessing import LabelEncoder

# Load the dataset
df = pd.read_excel('cleanedprojectdataset.xlsx')

# Display initial data
df.head()

# Rename columns for easier access
```

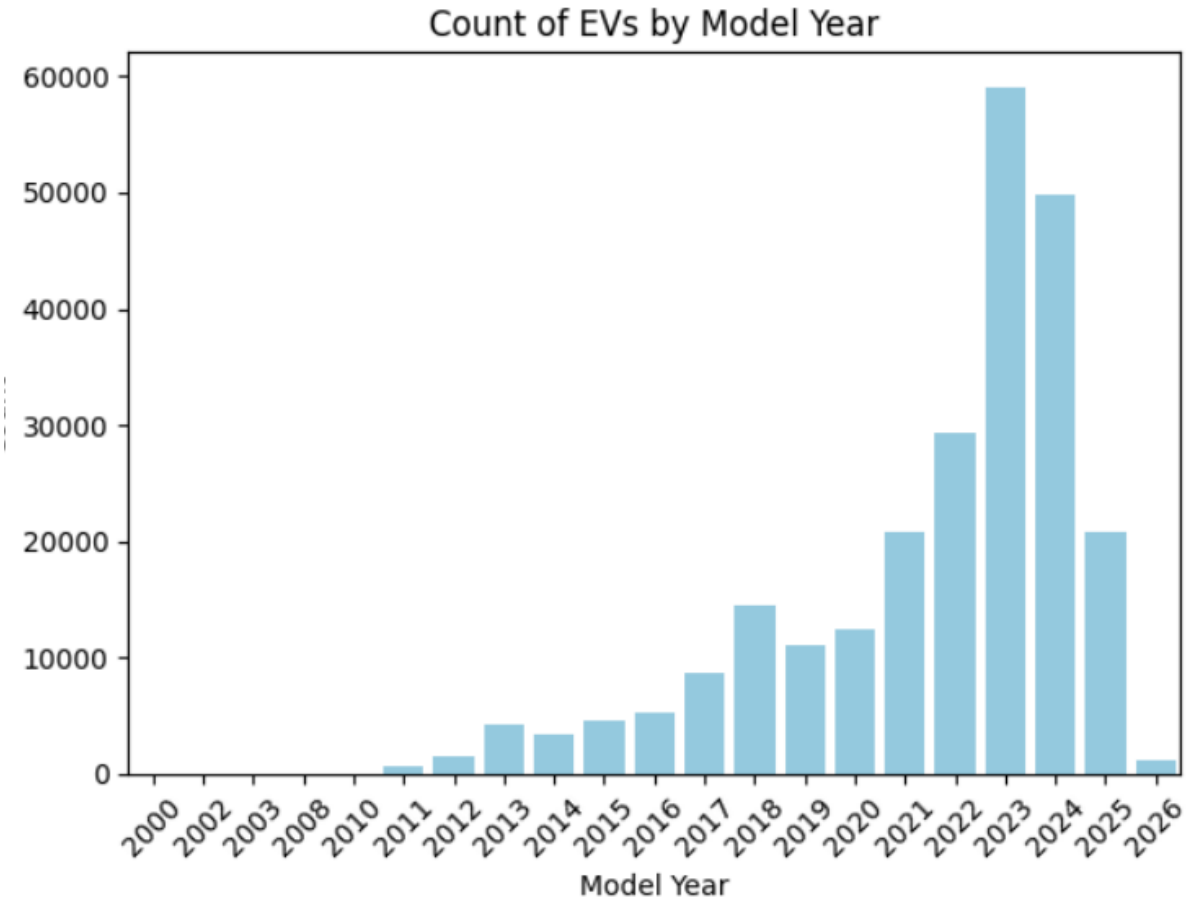
```
df.rename(columns={
    'Manufacturer': 'Make',
    'Electric Vehicle Range(Miles)': 'Electric Range',
    'Base Price ($)': 'Base MSRP',
    'Country': 'State'  # Assuming 'Country' stores state-level data
}, inplace=True)

# Create a 'location' column by combining City and State
df['location'] = df['City'].astype(str) + ', ' + df['State'].astype(str)

# Select relevant columns
df = df[['Model Year', 'Make', 'Model', 'Electric Vehicle Type',
         'Electric Range', 'Base MSRP', 'City', 'State', 'location']]

# Drop rows with missing key features
df = df.dropna(subset=['Electric Range', 'Base MSRP'])

# Extract latitude and longitude from 'Vehicle Location'
def extract_lat_lon(point):
    return pd.Series([np.nan, np.nan])  # Placeholder for real geocoding

# Step 3: Assign dummy latitude and longitude (you can replace this with actual geocoding)
df[['Latitude', 'Longitude']] = df['location'].apply(extract_lat_lon)

# Step 4: Safely drop 'Vehicle Location' only if it exists
```

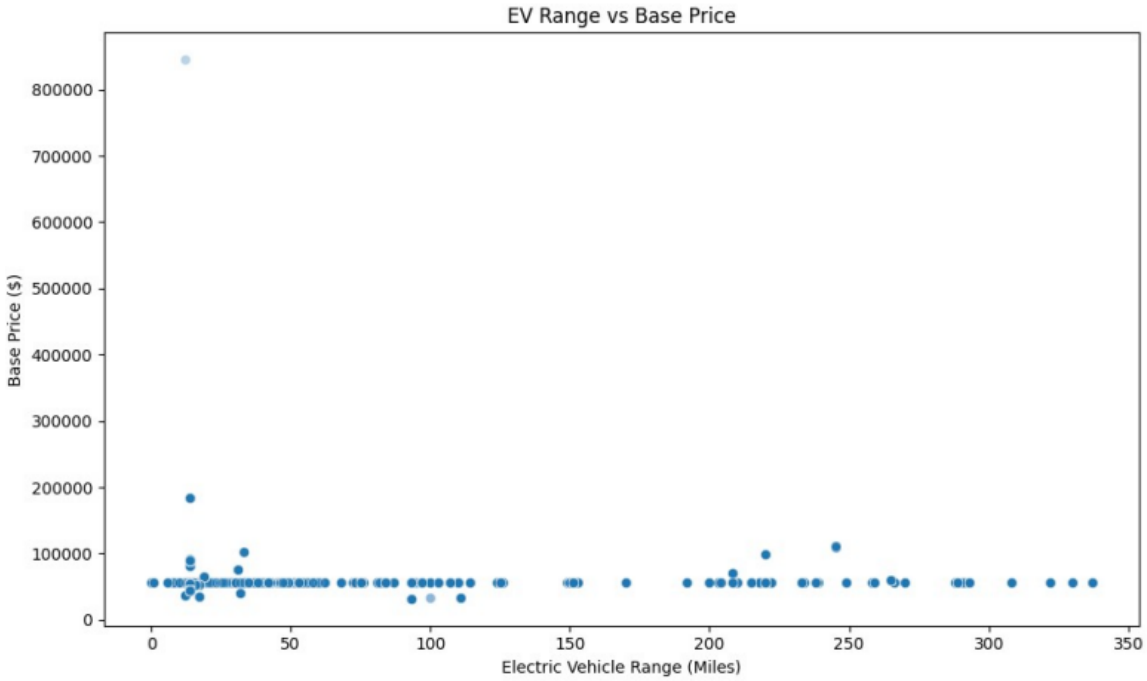
```
if 'Vehicle Location' in df.columns:
    df = df.drop(columns=['Vehicle Location'])



# Create vehicle age feature
df['Vehicle Age'] = 2025 - df['Model Year']

# Initialize label encoders
le_type = LabelEncoder()
le_make = LabelEncoder()
le_model = LabelEncoder()

# Encode categorical variables
df['EV_Type_Encoded'] = le_type.fit_transform(df['Electric Vehicle Type'].astype(str))
df['Make_Encoded']    = le_make.fit_transform(df['Make'].astype(str))
df['Model_Encoded']   = le_model.fit_transform(df['Model'].astype(str))

# Drop non-numeric columns that are no longer needed for modeling
df.drop(columns=['Electric Vehicle Type', 'Make', 'Model', 'City', 'State'], inplace=True)
X = df.drop(columns='Electric Range')
y = df['Electric Range']

# Convert categorical variables to numeric using one-hot encoding
X_encoded = pd.get_dummies(X, drop_first=True)
```

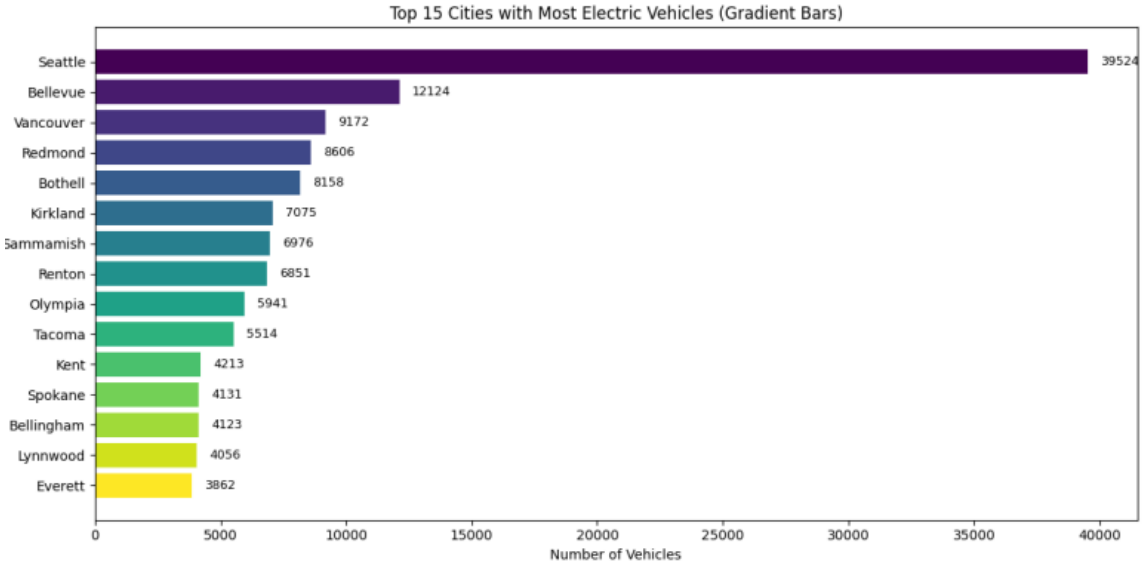
```
# Split the data
X_train, X_test, y_train, y_test = train_test_split(
    X_encoded, y, test_size=0.2, random_state=42
)

# Train Random Forest Regressor
model = RandomForestRegressor(random_state=42)
model.fit(X_train, y_train)

# Predict and evaluate
y_pred = model.predict(X_test)
print("MAE:", mean_absolute_error(y_test, y_pred))
print("R² Score:", r2_score(y_test, y_pred))

# Save predictions for Power BI
df_results = X_test.copy()
df_results['Actual Range'] = y_test
df_results['Predicted Range'] = y_pred

output_csv = "EV_Range_Predictions_For_PowerBI.csv"
df_results.to_csv(output_csv, index=False)
print(f"Results saved to {output_csv}")
```

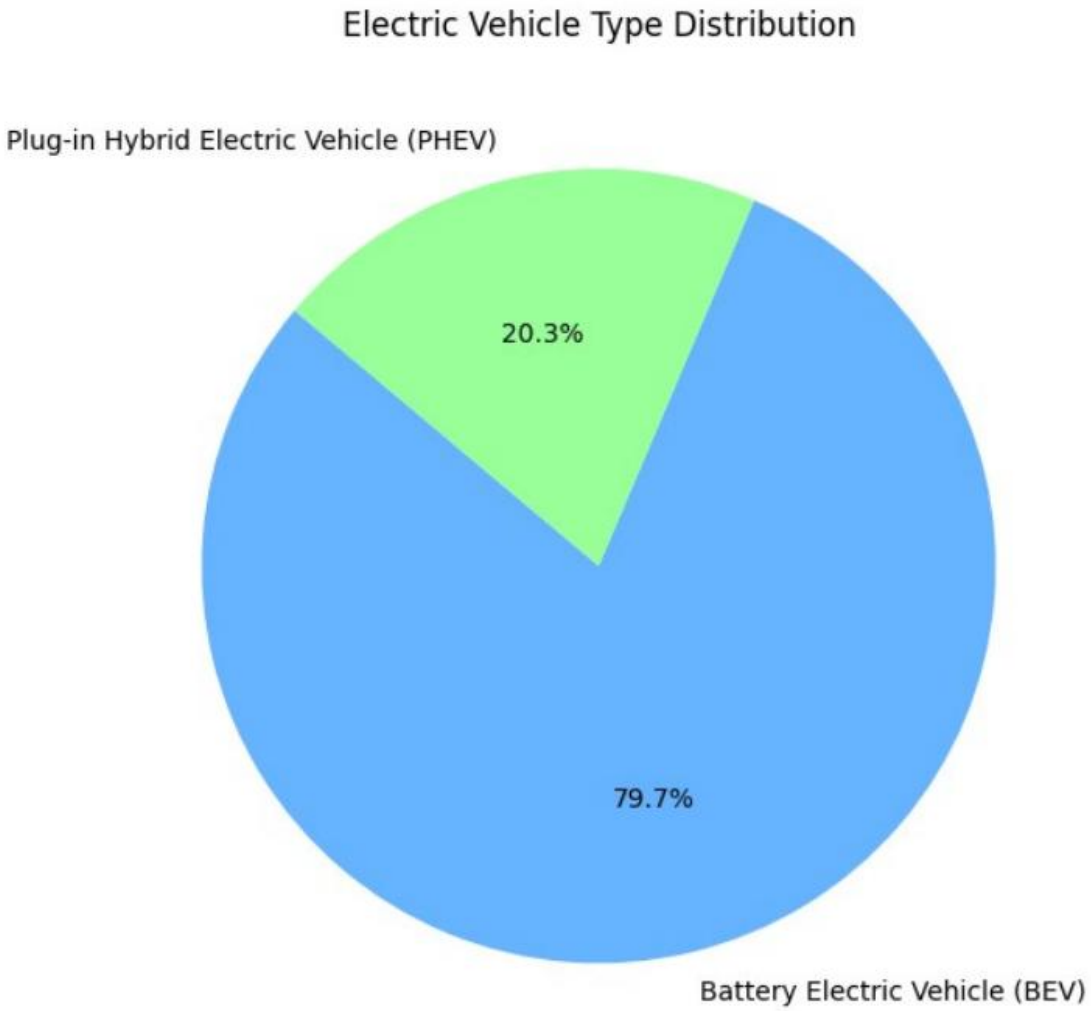
• **Modules / Screenshots**

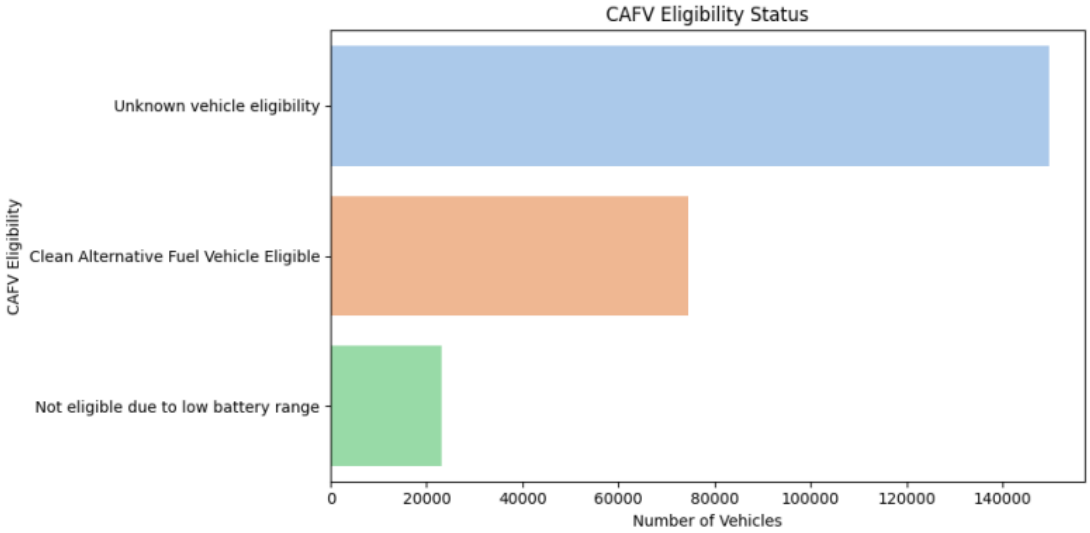


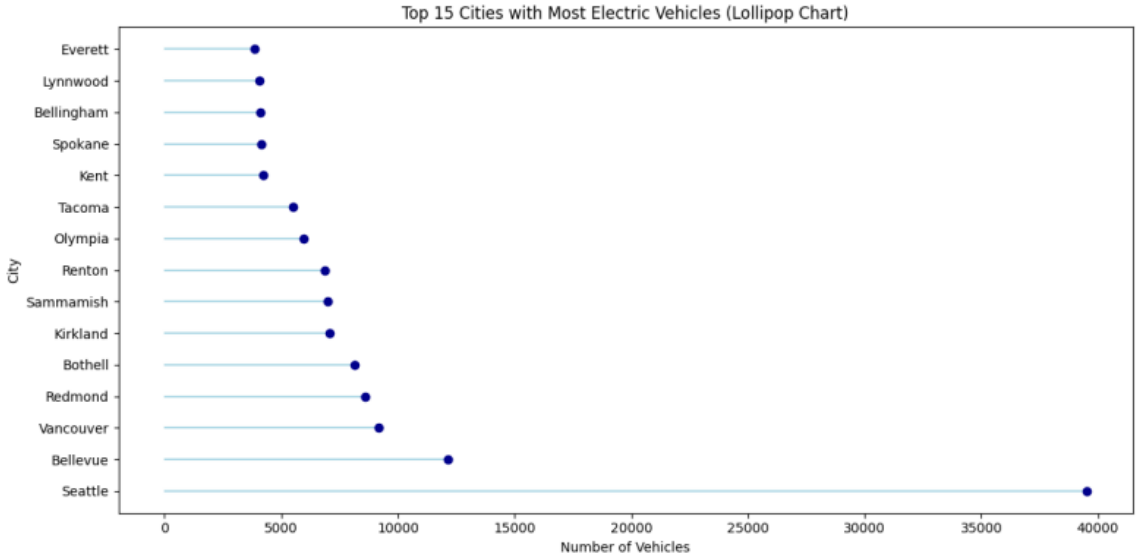

Count of EVs by Model Year
60000
50000
40000
30000
20000
10000
0
2000
2002
2003
2008
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
Model Year

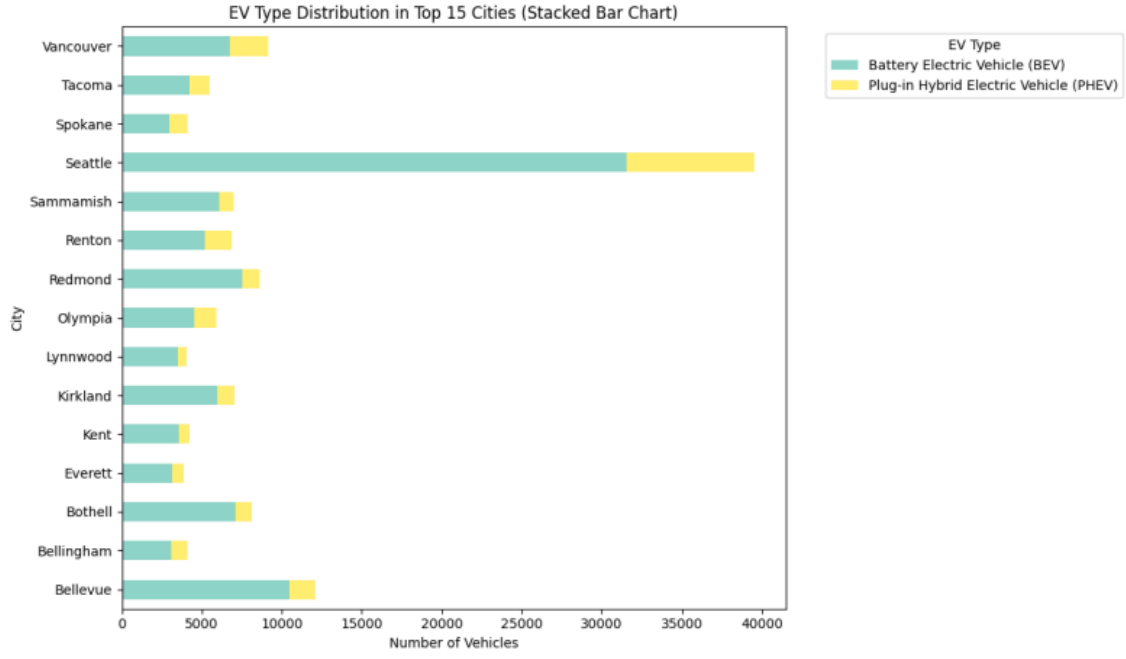

EV Range vs Base Price
800000
700000
600000
500000
400000
300000
200000
100000
0
Base Price ($)
0
50
100
150
200
250
300
350
Electric Vehicle Range (Miles)

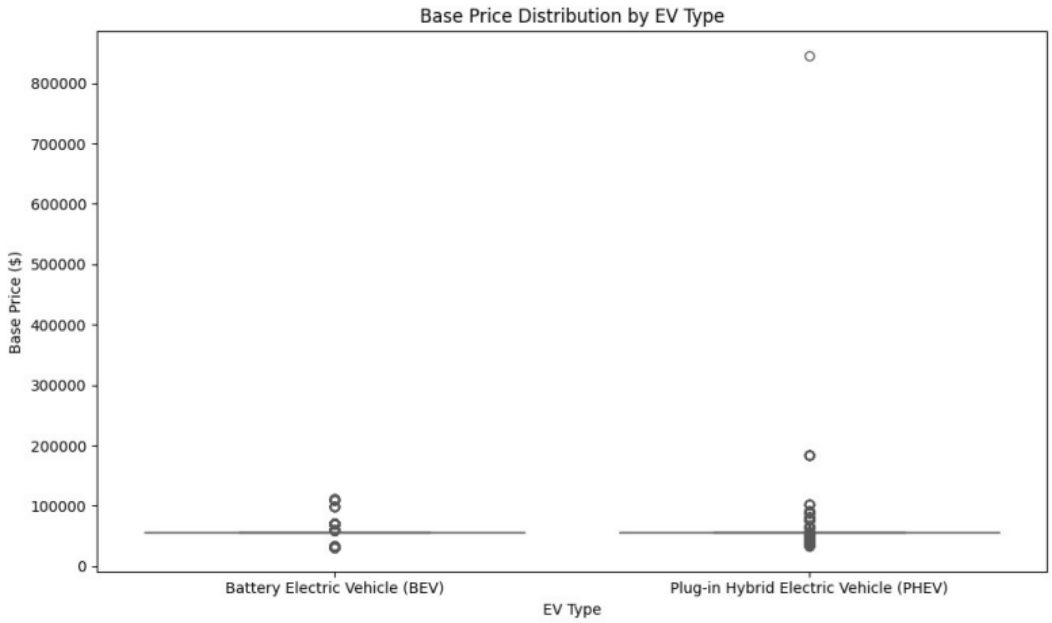

Top 15 Cities with Most Electric Vehicles (Gradient Bars)
Seattle
Bellevue
Vancouver
Redmond
Bothell
Kirkland
Sammamish
Renton
Olympia
Tacoma
Kent
Spokane
Bellingham
Lynnwood
Everett
39524
12124
9172
8606
8158
7075
6976
6851
5941
5514
4213
4131
4123
4056
3862
0
5000
10000
15000
20000
25000
30000
35000
40000
Number of Vehicles

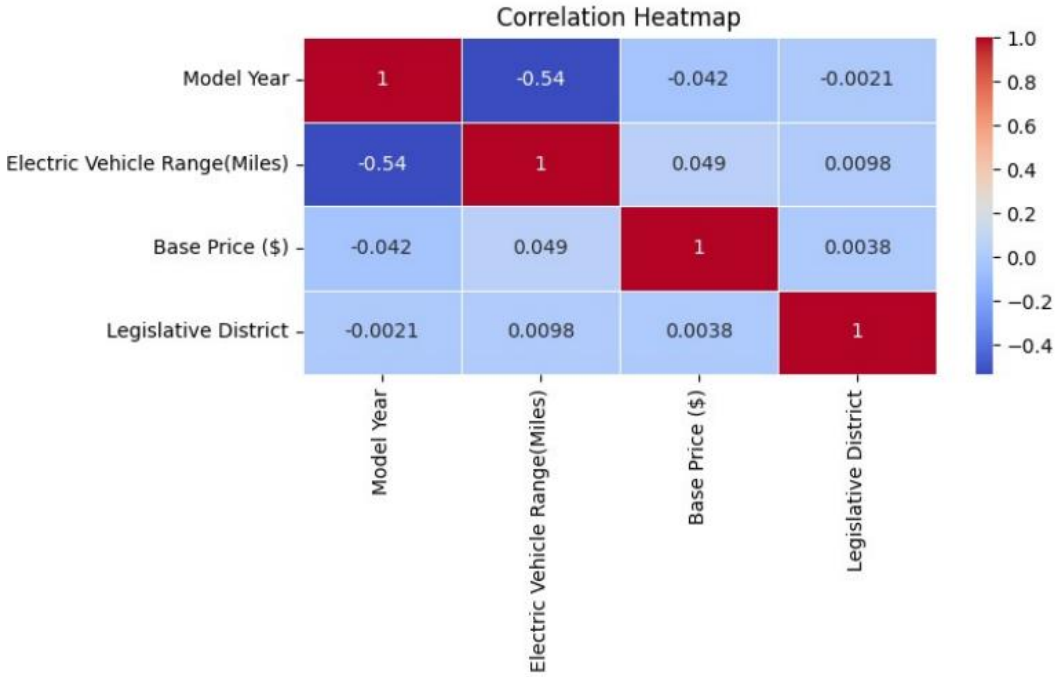

Electric Vehicle Type Distribution
Plug-in Hybrid Electric Vehicle (PHEV)
20.3%
79.7%
Battery Electric Vehicle (BEV)

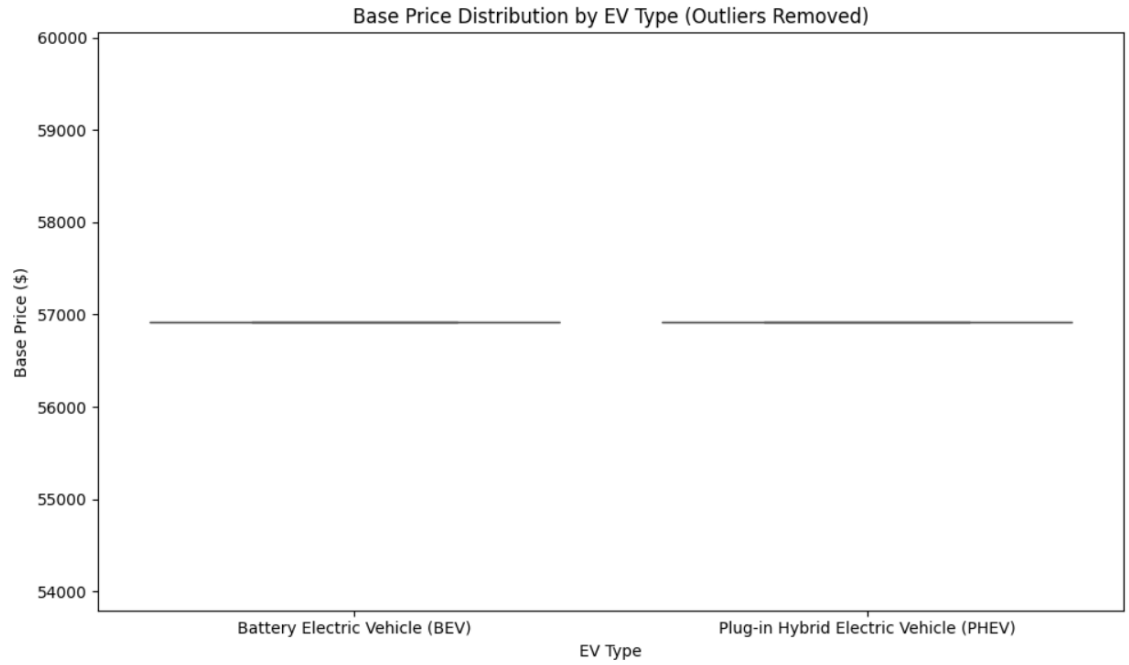

CAFV Eligibility Status
Unknown vehicle eligibility
Clean Alternative Fuel Vehicle Eligible
Not eligible due to low battery range
CAFV Eligibility
0
20000
40000
60000
80000
100000
120000
140000
Number of Vehicles

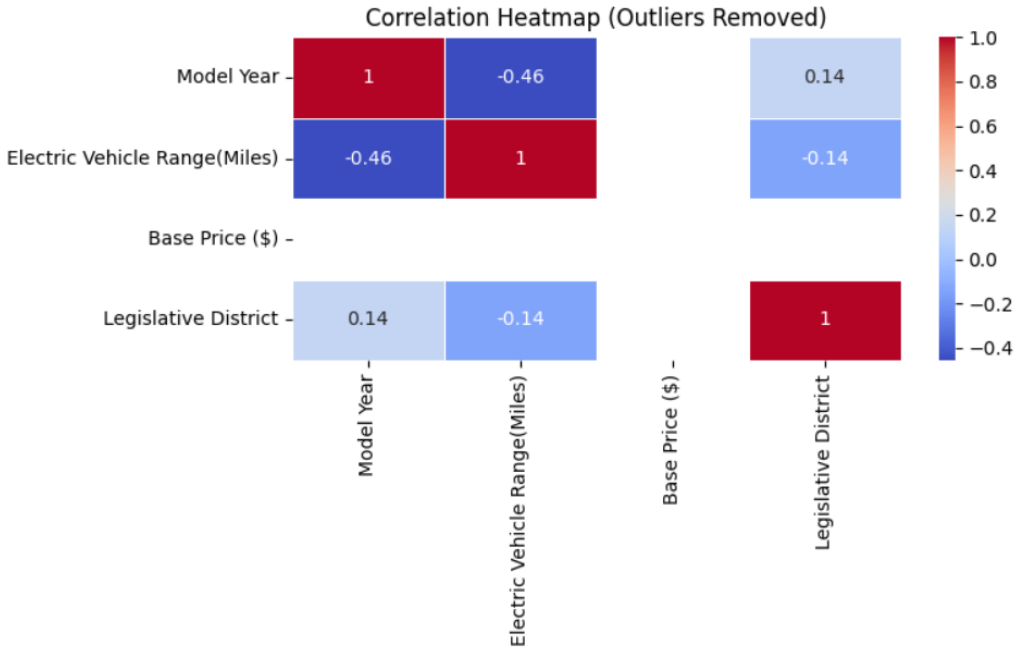

Top 15 Cities with Most Electric Vehicles (Lollipop Chart)
Everett
Lynnwood
Bellingham
Spokane
Kent
Tacoma
Olympia
Renton
Sammamish
Kirkland
Bothell
Redmond
Vancouver
Bellevue
Seattle
City
0
5000
10000
15000
20000
25000
30000
35000
40000
Number of Vehicles

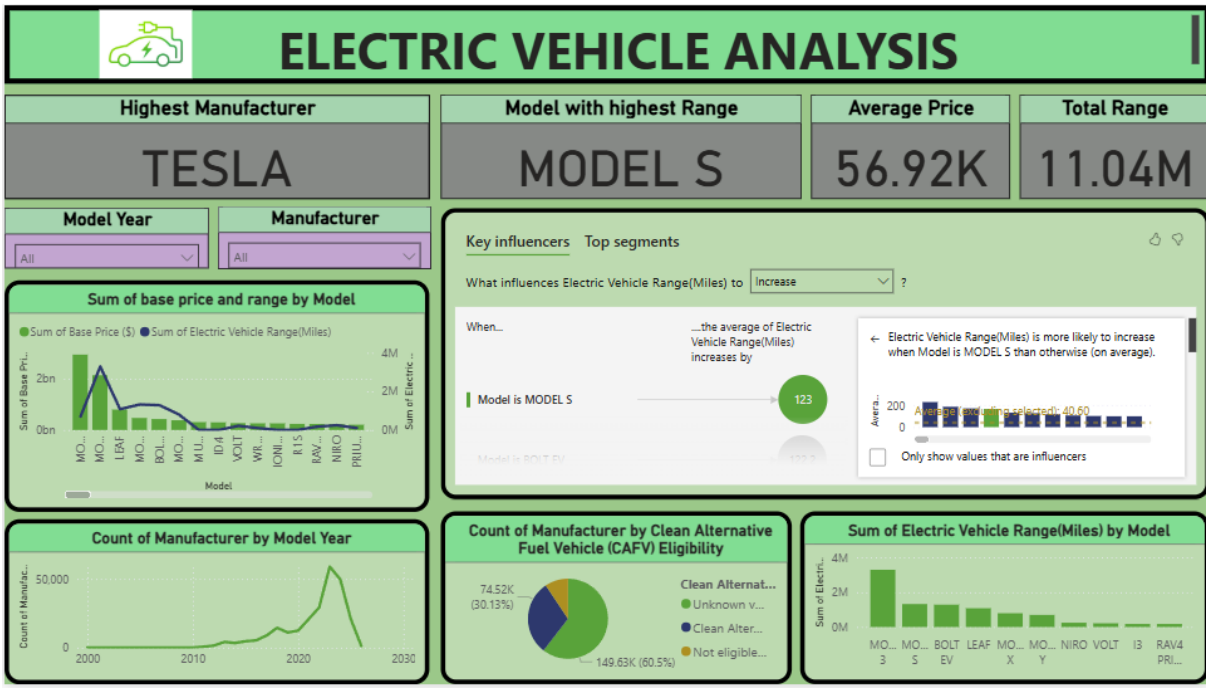

EV Type Distribution in Top 15 Cities (Stacked Bar Chart)
Vancouver
Tacoma
Spokane
Seattle
Sammamish
Renton
Redmond
Olympia
Lynnwood
Kirkland
Kent
Everett
Bothell
Bellingham
Bellevue
City
0
5000
10000
15000
20000
25000
30000
35000
40000
Number of Vehicles
EV Type
Battery Electric Vehicle (BEV)
Plug-in Hybrid Electric Vehicle (PHEV)


Base Price Distribution by EV Type
Base Price ($)
800000
700000
600000
500000
400000
300000
200000
100000
0
Battery Electric Vehicle (BEV)
Plug-in Hybrid Electric Vehicle (PHEV)
EV Type


Correlation Heatmap
Model Year
Electric Vehicle Range(Miles)
Base Price ($)
Legislative District
1
-0.54
-0.042
-0.0021
-0.54
1
0.049
0.0098
-0.042
0.049
1
0.0038
-0.0021
0.0098
0.0038
1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4


Base Price Distribution by EV Type (Outliers Removed)
Base Price ($)
60000
59000
58000
57000
56000
55000
54000
Battery Electric Vehicle (BEV)
Plug-in Hybrid Electric Vehicle (PHEV)
EV Type


Correlation Heatmap (Outliers Removed)
Model Year
Electric Vehicle Range(Miles)
Base Price ($)
Legislative District
1
-0.46
0.14
-0.46
1
-0.14
0.14
-0.14
1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
Model Year
Electric Vehicle Range(Miles)
Base Price ($)
Legislative District



Dashboard made in Power Bi

Chapter 5: Results and Discussion

Output / Report

The output of this project was twofold: a trained machine learning model capable of predicting the driving range of electric vehicles, and an interactive dashboard built in Power BI to visualize the insights. The dashboard provided clear visual representation of key variables such as Model Year, Make,Model,Electric Vehicle Type,Electric Range, Base MSRP, City, State, location conditions—all of which impact EV range. Using charts, slicers, and filters, users were able to compare models, understand influencing factors, and explore how range varies across conditions.

The machine learning model, trained on preprocessed data using algorithms like demonstrated a reliable ability to estimate EV range. Python was used for exploratory data analysis (EDA), which uncovered feature relationships, data distributions, and patterns—crucial for selecting the right variables and improving model performance.

Challenges Faced

One of the major challenges was ensuring the quality and consistency of the dataset. Missing values, unit mismatches, and noisy entries had to be carefully handled during preprocessing. Selecting the most appropriate machine learning algorithm and tuning hyperparameters was another challenge, especially since the model needed to perform well across various real-world scenarios. Additionally, maintaining consistency between the machine learning model's predictions and the Power BI dashboard insights required careful integration and validation to ensure that the end-user experience was both accurate and meaningful.

Learnings

This project offered several key learnings. Firstly, it emphasized the critical role of thorough data preparation—cleaning, standardizing, and understanding the feature distribution greatly enhanced the reliability of predictions. Secondly, I deepened my knowledge of EDA techniques, regression algorithms, and performance evaluation metrics like MAE and R^2 score. Thirdly, I learned how to bridge data science and business intelligence by integrating Python-based machine learning with Power BI dashboards, ensuring insights were not just accurate but also easy to interpret. Overall, this hands-on project strengthened

my confidence in solving real-world problems using end-to-end data workflows.

Chapter 6: Conclusion

The Electric Vehicle Range Prediction project served as a practical and insightful exploration of how data science can drive innovation in the automotive and clean energy sectors. From the outset, the project focused on addressing a real-world challenge—accurately estimating the driving range of electric vehicles using measurable factors such as battery capacity, speed, terrain, and environmental conditions. By leveraging Python for preprocessing, exploratory analysis, and machine learning, and Power BI for dashboard creation, we built a complete pipeline to both predict EV range and visualize patterns that aid informed decision-making.

A key achievement of the project was the development of a regression-based machine learning model that predicted EV driving range with strong accuracy. The early use of tools like Excel and SQL helped streamline initial data handling and exploration, laying the groundwork for more advanced modeling and analytics. This comprehensive approach demonstrated how integrating multiple tools and platforms can lead to efficient, data-driven workflows.

Through this internship and project, I gained hands-on experience with technologies like Python, Power BI, SQL, and Excel. I deepened my understanding of data preprocessing, regression modeling, feature importance analysis, and the power of interactive dashboards to communicate insights. Most importantly, the project highlighted the crucial role of data in optimizing real-world systems like electric mobility.

Overall, the project not only met its technical objectives but also contributed significantly to my growth as a data science professional. It offered a complete view of end-to-end solution development—from problem definition and data wrangling to modeling and visualization. The skills and lessons learned will be invaluable as I move forward in my academic and professional journey.