ELECTRIC VEHICLE (EV) CHARGING INFRASTRUCTURE AND ADOPTION PREDICTION

MINI PROJECT REPORT

Submitted By

RAKSHANA A 211701041

RAGAVI K 211701040

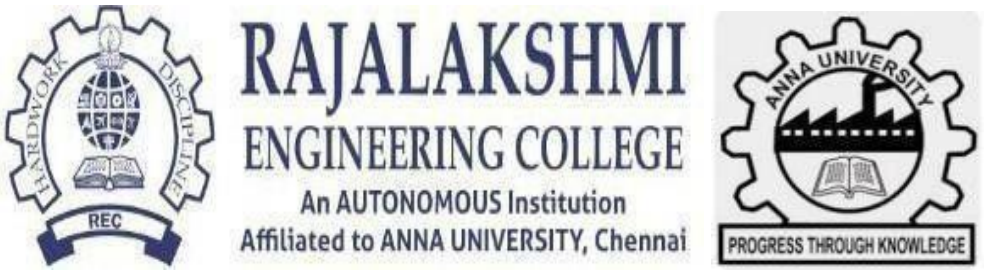
In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND DESIGN



RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI-600 025

2024

RAJALAKSHMI ENGINEERING COLLEGE
ANNA UNIVERSITY : CHENNAI 600 025
BONAFIDE CERTIFICATE

Certified that this project report **"ELECTRIC VEHICLE (EV) CHARGING INFRASTRUCTURE AND ADOPTION PREDICTION"** is the bonafide work of **"RAKSHANA A (211701041) & RAGAVI K (211701040)"** who carried out the project work under my supervision.

SIGNATURE

Mr. S. Uma Maheswar Rao

HEAD OF THE DEPARTMENT

Associate Professor

Department of

Computer Science and Design

Rajalakshmi Engineering College

Rajalakshmi Nagar, Thandalam

Chennai-602105

SIGNATURE

Mrs. E. Preethi

SUPERVISOR

Assistant Professor

Department of

Computer Science and Design

Rajalakshmi Engineering College

Rajalakshmi Nagar, Thandalam

Chennai-602105

Submitted to Project viva-voce examination held on_____.

Internal Examiner **External Examiner**

ABSTRACT

The study analyzes trends in electric vehicle (EV) infrastructure, usage, and market behavior using an exploratory data analysis (EDA) dataset. The data includes vehicle registrations, geographical distribution, and characteristics like make, model year, electric range, and location. The primary goals are to analyze EV distribution, vehicle characteristics, price trends, infrastructure readiness, and legislative influence. The insights aim to identify factors influencing EV adoption and provide recommendations for improving EV infrastructure and policy planning in underserved regions. In dataset, we aim to identify factors influencing the adoption and distribution of electric vehicles and provide recommendations for improving EV infrastructure and policy planning in underserved regions.

Keywords:

Electric vehicle, Vehicle Characteristics, Market Behavior, Infrastructure, Adoption, Price Trends.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROBLEM STATEMENT :

To analyze trends in electric vehicle (EV) adoption and infrastructure using an exploratory data analysis (EDA) dataset, focusing on vehicle registrations, geographic distribution, and characteristics like make, model year, and electric range. The study aims to identify key factors influencing EV adoption, assess infrastructure readiness, and understand legislative impacts to propose actionable recommendations for improving EV infrastructure and policy planning, particularly in underserved regions.

1.2 OBJECTIVES :

The objective of this study is to analyze the distribution and adoption patterns of electric vehicles (EVs) using exploratory data analysis (EDA) and predictive modeling. The focus is on understanding how geographical and demographic factors influence EV adoption, alongside evaluating the impact of vehicle characteristics such as make, model year, electric range, and pricing. Additionally, the study aims to assess trends in EV infrastructure readiness, including the accessibility of charging stations, and examine the role of legislative policies in driving adoption rates. By developing and validating predictive models for key metrics like electric range, the study seeks to identify critical factors affecting EV adoption. Ultimately, the insights gained will inform actionable recommendations to enhance EV infrastructure and policy planning, with a special emphasis on addressing the needs of underserved regions to wider EV adoption.

CHAPTER 2

DATASET DESCRIPTION

2.1 DATASET SOURCE :

The dataset used for this study consists of comprehensive electric vehicle (EV) population data, containing approximately 121,000 records. It includes detailed attributes such as vehicle registrations, geographical distribution, make, model year, electric range, and location. This extensive dataset serves as a foundation for understanding the trends in EV adoption and analyzing key factors like infrastructure readiness, market behavior, and legislative impacts. The data supports identifying regional gaps in infrastructure and policy planning, providing insights for targeted improvements, particularly in underserved areas.

2.2 DATASET SIZE AND STRUCTURE:

The dataset used in this study contains approximately 121,000 records, offering a detailed view of the electric vehicle (EV) population. The structure of the dataset includes key attributes such as:

Vehicle Information: Make, model, model year, and electric range.

Registration Details: Data on vehicle registrations and their distribution across various regions.

Geographical Data: Locations and zones where EVs are registered, providing insights into adoption patterns.

Infrastructure Readiness: Data points related to the availability and accessibility of EV infrastructure.

Fig 2.1 Dataset size and structure

2.3 DATASET FEATURES DESCRIPTION

The dataset consists of approximately 121,000 records, capturing detailed information about EV adoption, usage, and regional distribution. It is designed to support an in-depth analysis of trends in EV characteristics, infrastructure readiness, and market behavior.

1. Vehicle Details:
    - Make: The manufacturer of the EV.
    - Model: Specific model designation of the vehicle.
    - Model Year: Year of manufacture, which can help track trends over time.
    - Electric Range: The maximum distance an EV can travel on a full charge.
2. Registration Data:
    - Vehicle Registrations: Number of EV registrations, indicative of adoption levels.
    - Registration Location: The geographical area where vehicles are registered.
3. Geographical Information:
    - City/State/Region: Detailed location data providing insights into adoption patterns and regional disparities.
4. Infrastructure Details:

- Indicators of charging station availability and accessibility to measure infrastructure readiness.

5. Pricing Information :

- Base MSRP (Manufacturer's Suggested Retail Price) to analyze the impact of cost on adoption.

This dataset offers a robust framework for exploring the factors influencing EV adoption and provides a foundation for identifying gaps in infrastructure and policy implementation.

CHAPTER 3

DATA ACQUISITION AND INITIAL ANALYSIS

3.1 DATA LOADING:

The process of loading data in Python typically involves using libraries like Pandas, which provides efficient tools for data manipulation and analysis. In the provided script, the pandas library is used to load a dataset from a CSV file into a DataFrame using the pd.read_csv function. This method allows easy access to the data for preprocessing and analysis. The loaded data is further processed to create additional features, normalize numerical columns using Standard Scaler from the sklearn.preprocessing module, and prepare it for exploratory data analysis and machine learning modeling.

3.2 INITIAL OBSERVATIONS:

The dataset contains 166,800 entries across 17 features, providing detailed information on electric vehicle (EV) registrations. Key attributes include the vehicle's unique VIN (truncated), geographical details such as county, city, state, and postal code, as well as the model year, make, and model of the vehicle. It also includes information on the type of electric vehicle (e.g., Battery Electric Vehicle or Plug-in Hybrid Electric Vehicle), clean alternative fuel vehicle eligibility, and the vehicle's electric range (maximum distance on a full charge). Additional data points such as the base MSRP, legislative district, vehicle location, electric utility, and census tract are also included. The dataset features a mix of numerical data (e.g., electric range, base MSRP, legislative district) and textual data (e.g., VIN, make, model, geographical details). While there are some missing values in columns like "County," "City," and "Legislative District," the dataset provides a comprehensive view of the EV market, making it ideal for analysis on EV adoption trends, infrastructure, and market dynamics. A sample entry might be a 2017 Tesla Model X registered in Olympia, WA, with a 200-mile electric range and eligibility for clean alternative fuel incentives.

CHAPTER 4

DATA CLEANING AND PREPROCESSING

4.1 HANDLING MISSING VALUES:

Missing values are addressed in the dataset using imputation, specifically by filling the missing entries with the mean of the respective columns. This method is implemented using df.fillna(df.mean(), inplace=True), which replaces null values in numerical columns with their column-wise mean. The rationale behind this approach is to maintain the integrity of the dataset by retaining all records, as outright removal of rows or columns with missing data could result in loss of valuable information. Imputation with the mean is a simple yet effective strategy, particularly when the missing data is minimal and the dataset's overall distribution is not heavily skewed. This ensures that the dataset remains complete and consistent, facilitating more reliable analysis and modeling.

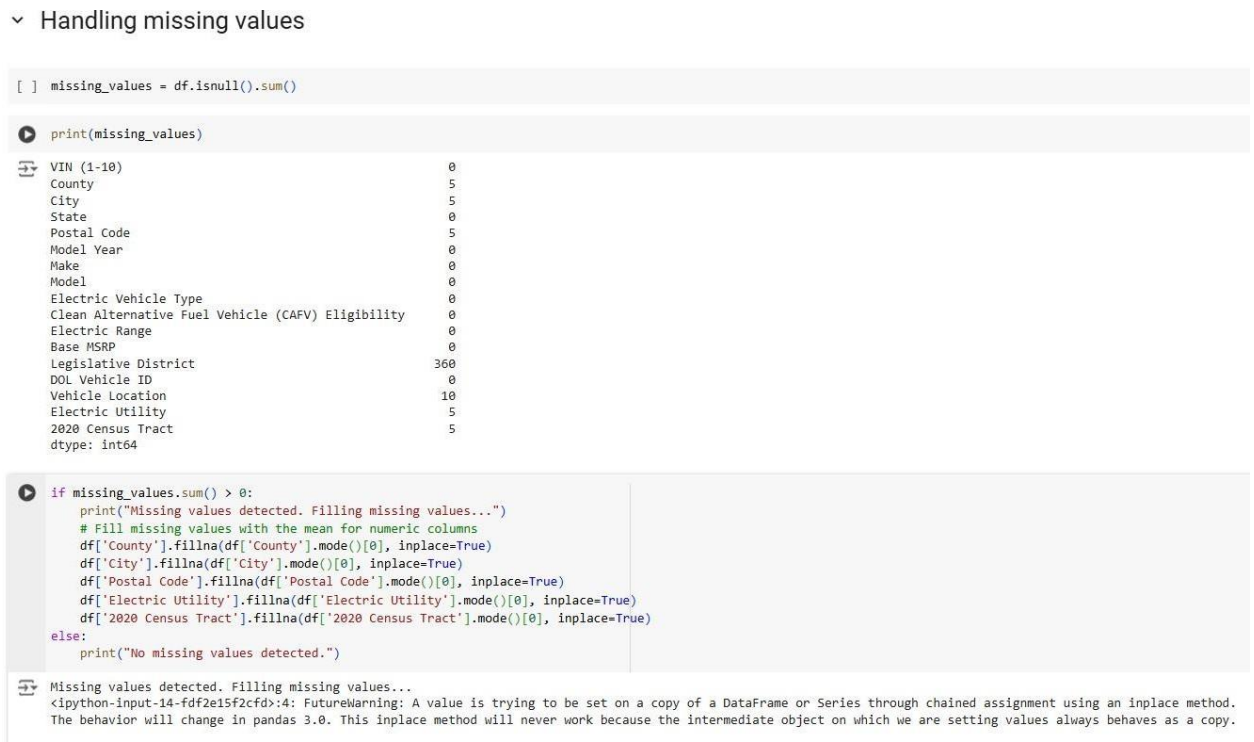


Fig 4.1 Handling missing values

4.2 FEATURE ENGINEERING:

Feature engineering involves creating and modifying features from raw data to enhance machine learning model performance. For the electric vehicle dataset, this can include calculating vehicle age by subtracting the model year from the current year, deriving geospatial insights from vehicle location data to analyze regional trends, and segmenting electric range into categories like low, medium, and high to study its impact on adoption. Transforming categorical data, such as CAFV eligibility, into numerical formats ensures compatibility with models, while grouping base MSRP into price segments (economy, mid-range, premium) can uncover market behavior patterns. Additionally, generating interaction terms, such as combining electric range with base MSRP, can help evaluate efficiency metrics like price-to-range ratios. Deriving infrastructure-related metrics, like EV density by region or correlating distribution with electric utility services, adds further depth to the analysis, enabling more accurate predictions and actionable insights.

4.3 DATA TRANSFORMATION:

Data transformation is a critical step in preparing raw data for analysis and modeling, involving the modification of data into a more usable and meaningful format. This process includes techniques like normalization, which scales numerical values to a consistent range, and encoding, which converts categorical variables into numerical formats for machine learning compatibility. Handling missing values through imputation or removal ensures dataset completeness, while log transformations or power transformations are applied to stabilize variance and handle skewed distributions. Data aggregation combines granular data into summarized forms, such as averages or totals, for easier interpretation. Additionally, splitting complex fields into multiple columns or combining related columns can enhance feature clarity. Effective data transformation not only improves the interpretability of datasets but also optimizes model performance by aligning data characteristics with algorithmic requirements.

CHAPTER 5

EXPLORATORY DATA ANALYSIS

5.1 DATA INSIGHTS DESCRIPTION

S.NO	DATA INSIGHT	DESCRIPTION
1.	Most Common Vehicle Make	Identifies the most frequently registered EV manufacturer, showing market leaders in EV adoption.
2.	Distribution of EV Types	Highlights the proportion of Battery Electric Vehicles (BEVs) versus Plug-in Hybrid Electric Vehicles (PHEVs).
3.	Geographic Focus on EVs	Analyzes regions with the highest density of EV registrations, useful for targeted infrastructure planning.
4.	EV Adoption by Year	Tracks the trend of EV registrations over the years, showing growth patterns in EV popularity.
5.	Electric Range Analysis	Examines the distribution of electric ranges to understand the coverage capabilities of the fleet.
6.	Base MSRP Segmentation	Categorizes vehicles into price brackets (economy, mid-range, premium) to identify customer segments.
7.	CAFV Eligibility Impact	Studies how eligibility for clean alternative fuel programs influences EV adoption rates.
8.	Regional Utility Influence	Investigates whether the type of electric utility provider correlates with EV adoption patterns.
9.	Vehicle Age Trends	Evaluates the relationship between vehicle age and registration numbers to predict replacement cycles.
10.	Infrastructure Gaps	Identifies underserved regions with low EV density, providing recommendations for new infrastructure.

5.2 DATA INSIGHTS VISUALIZATION:

Data Visualization: Correlation Heat Map

- Inference: The heatmap indicates the strength and direction of relationships between numerical variables, with darker shades signifying stronger correlations.

- Observation: Variables such as Electric Range and Base MSRP show moderate positive or negative correlations, suggesting some dependency between cost and performance metrics.

- Implication: Strong correlations may signal redundancy among features, while weak correlations suggest independence, guiding feature selection or engineering efforts for predictive modeling.

- Recommendation: Focus on highly correlated pairs for deeper analysis, and consider dimensionality reduction or removing features with excessive multicollinearity to improve model efficiency.

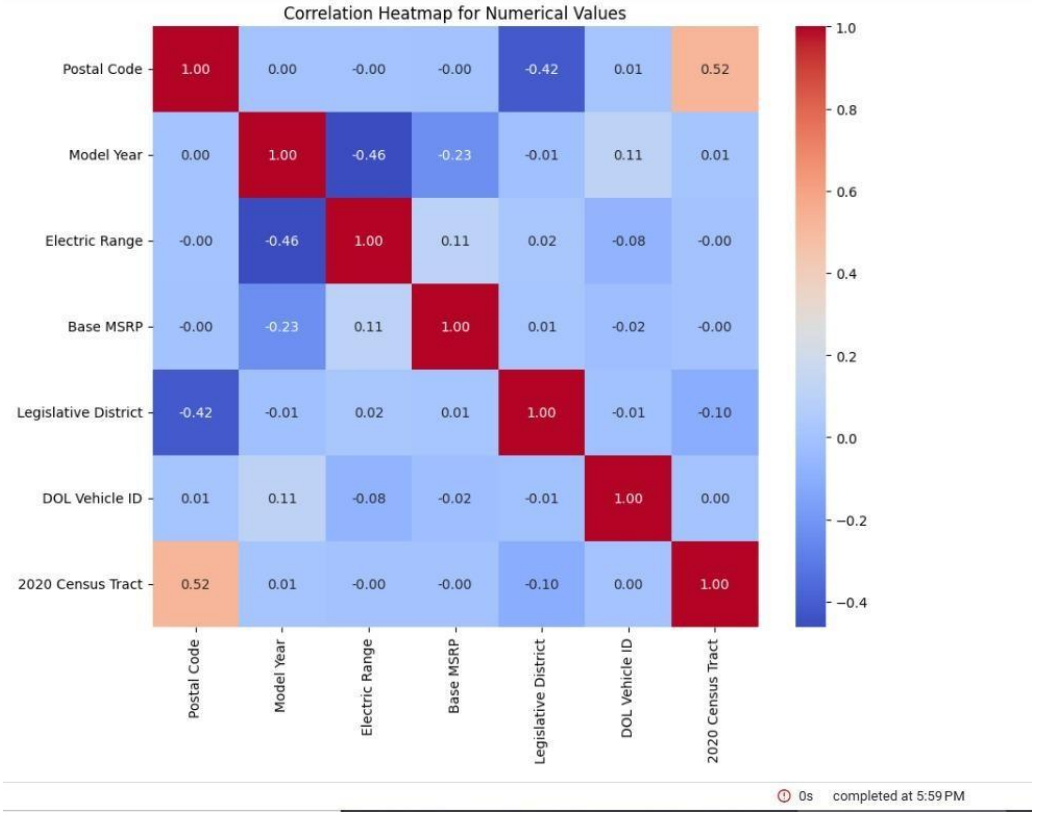


Fig 5.1 Correlation Heat Map

Data Visualization: Stacked Bar Chart:

- Inference: The visualization shows how the distribution of different electric vehicle types, such as BEVs and PHEVs, has evolved over the years.

- Observation: A noticeable increase in BEVs is observed in recent model years, while older model years have relatively balanced counts of BEVs and PHEVs.

- Implication: The growing preference for BEVs indicates advancements in battery technology and infrastructure, making fully electric vehicles more appealing to consumers.

- Recommendation: Focus on promoting BEV-friendly policies and infrastructure, such as expanding fast-charging networks, to sustain and accelerate this trend.

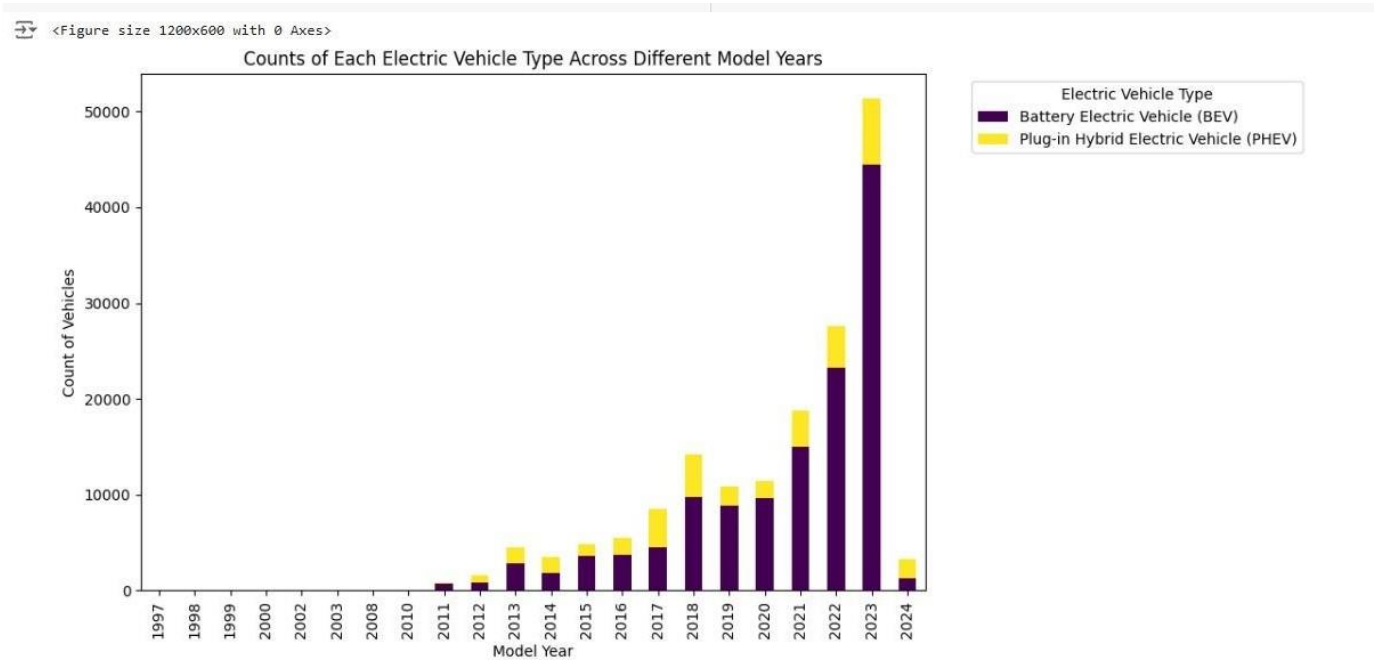


Fig 5.2 Bar Chart

Data Visualization: Bar Chart Analysis: Average Electric Range by Vehicle Make

- Inference: The bar chart highlights the average electric range offered by different vehicle manufacturers, showcasing variations in technology and design priorities.

- Observation: Certain makes, likely premium or specialized EV manufacturers, have significantly higher average electric ranges compared to others.

- Implication: Brands with higher electric ranges may appeal more to consumers seeking longer-distance travel options, indicating a competitive advantage in battery technology.

- Recommendation: Encourage collaboration or benchmarking with top-performing manufacturers to enhance the electric range of other makes and support policies that incentivize higher-range EV production.

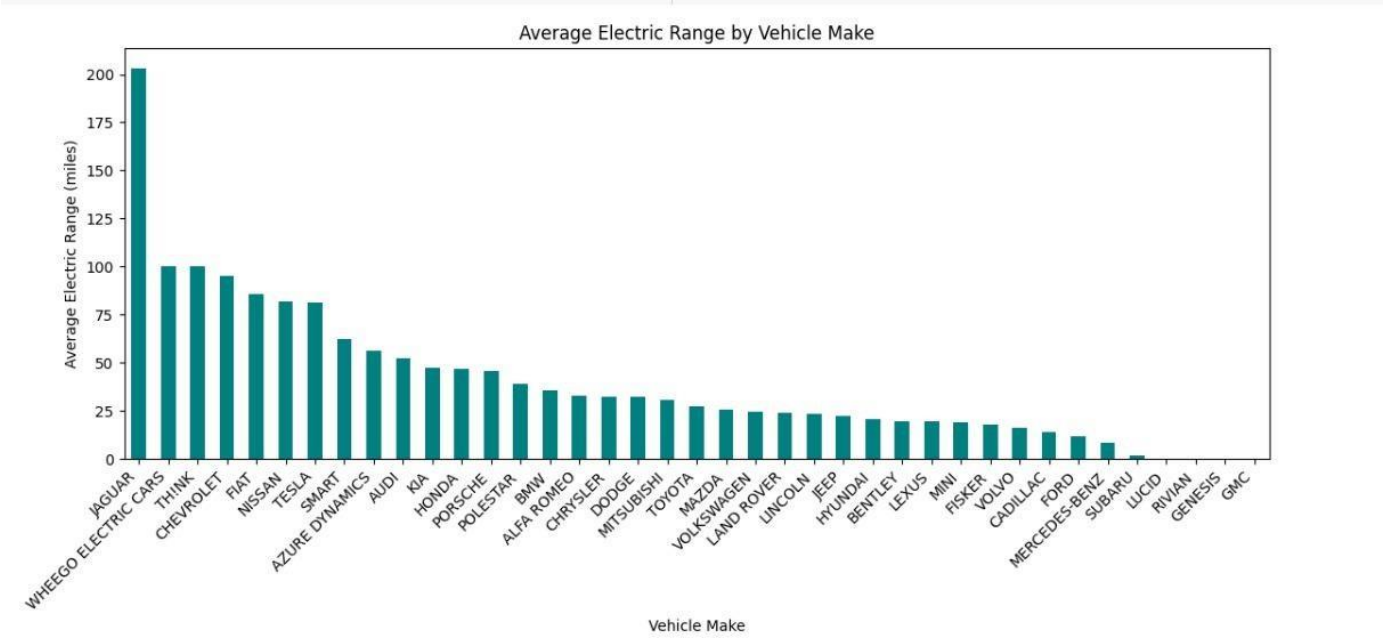


Fig 5.3 Average Electric Range by Vehicle Make

Data Visualization: Stacked Bar Chart Analysis: CAFV Eligibility by Model Year

- Inference: The chart illustrates how the number of vehicles eligible for Clean Alternative Fuel Vehicle (CAFV) status has varied across model years.

- Observation: Eligibility for CAFV status has increased in recent model years, indicating a trend toward greater compliance with clean fuel standards.

- Implication: Manufacturers are likely aligning with environmental policies and improving EV designs to meet eligibility criteria, reflecting a shift toward sustainable practices.

- Recommendation: Strengthen policies promoting CAFV eligibility and provide incentives for manufacturers to produce vehicles that qualify, further driving eco-friendly adoption.

Data Visualization: Line Plot Analysis:

- Inference: The line plot shows the trends in vehicle counts produced by different manufacturers over the years, reflecting their market presence and growth.

- Observation: Some manufacturers exhibit consistent growth in vehicle counts across model years, while others have sporadic or declining trends.

- Implication: Manufacturers with increasing trends are likely responding effectively to market demands, while those with declining trends may face challenges in production or market competitiveness.

- Recommendation: Conduct deeper analyses on top-performing manufacturers to identify successful strategies and encourage underperforming manufacturers to innovate and align with market trends.

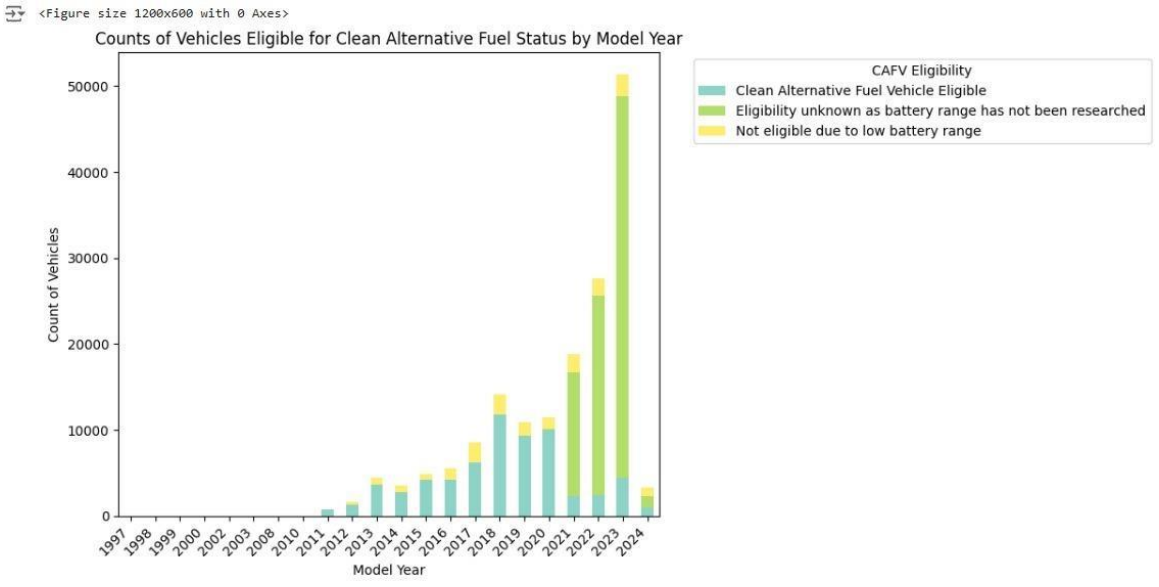


Fig 5.4 Line Plot Analysis

CHAPTER 6
PREDICTIVE MODELING

6.1 MODEL SELECTION AND JUSTIFICATION:

In the predictive maintenance analysis, three machine learning models were selected: Random Forest Classifier,Linear Regression, and XGBoost model. Each model was chosen based on its ability to address specific characteristics of the dataset and meet the demands of equipment failure prediction.

Random forest model : A machine learning algorithm that uses an ensemble of decision trees to improve accuracy by averaging predictions and reducing overfitting.

XGBoost model: A powerful, optimized gradient boosting algorithm that builds decision trees sequentially to minimize errors and improve predictive performance.

Linear regression model: A statistical method that models the relationship between a dependent variable and one or more independent variables by fitting a linear equation.

Justification: Random Forest and XGBoost demonstrated significantly better accuracy compared to Linear Regression, owing to their ability to handle non-linear relationships and complex interactions between features. Among them, XGBoost was preferred due to its robust performance, offering high accuracy with regularization techniques to prevent overfitting and built-in mechanisms to handle missing data. Furthermore, its parallel processing capabilities made it computationally efficient for the dataset. Linear Regression, while simpler, showed limited accuracy, highlighting its unsuitability for this non-linear dataset. Thus, XGBoost was chosen as the optimal model for its balance of precision, efficiency, and reliability in predictions.

Actual Value vs. predicted value

```
[ ]  # Display actual values vs. predictions for Random Forest and XGBoost

     # Combine actual values and predictions into a DataFrame
     comparison_df = pd.DataFrame({
         'Actual Electric Range': y_test,
         'Random Forest Prediction': y_pred,
         'XGBoost Prediction': y_pred_xgb,
         'Linear regression' : y_pred_lr
     }).reset_index(drop=True)

     # Display the first 10 results
     print("Comparison of Actual vs. Predicted Electric Range (Sample):")
     print(comparison_df.head(10))
```

```
Comparison of Actual vs. Predicted Electric Range (Sample):
   Actual Electric Range  Random Forest Prediction  XGBoost Prediction  \
0               0.249258                  0.303345            0.303271
1               0.062315                  0.054141            0.054130
2               0.857567                  0.545569            0.545529
3               0.623145                  0.303345            0.303271
4               0.000000                  0.013557            0.013556
5               0.000000                  0.013557            0.013556
6               0.000000                  0.011867            0.011907
7               0.000000                  0.011867            0.011907
8               0.000000                  0.011867            0.011907
9               0.738872                  0.479228            0.478559

   Linear regression
0           0.368888
1           0.026497
2           0.240491
3           0.368888
4           0.112095
5           0.112095
6           0.069296
7           0.069296
8           0.069296
9           0.283290
```

Fig 6.1 Data Prediction

DATA PARTITIONING:

Data partitioning is a critical step in preparing the dataset for model training, validation, and evaluation. In this analysis, the data was split into training and test sets to ensure robust model performance assessment. Using the train_test_split function from the sklearn.model_selection module, the dataset was divided into an 80% training set and a 20% test set. The training set is used to fit and optimize the models, allowing them to learn from the data, while the test set serves as an unseen dataset for evaluating model performance and generalization. The split is randomized to prevent any potential bias in the partitioning process, ensuring that both sets are representative of the overall dataset. This partitioning strategy allows for effective model training and provides an unbiased evaluation of the model's predictive capabilities on new data, helping to

prevent overfitting and ensuring that the model performs well in real- world scenarios.

6.2 MODEL TRAINING AND HYPERPARAMETER TUNING:

For model training and hyperparameter tuning, each of the selected models—Random Forest, XGBoost, and Linear Regression—underwent training on the dataset to learn patterns and improve predictive performance through tuning.

Random Forest: The Random Forest model was initially trained with default parameters. Hyperparameter tuning was conducted using GridSearchCV to optimize parameters such as the number of trees (n_estimators), maximum tree depth (max_depth), minimum samples required to split a node (min_samples_split), and minimum samples at leaf nodes (min_samples_leaf). These adjustments aimed to enhance the model's predictive accuracy, reduce overfitting, and improve stability.

XGBoost: XGBoost was trained using its default configuration initially, followed by fine-tuning to optimize parameters like the learning rate (eta), maximum depth of trees (max_depth), and the number of boosting rounds (n_estimators). Regularization parameters (alpha and lambda) were also adjusted to prevent overfitting. Cross-validation during tuning ensured reliable performance and efficiency.

Linear Regression: Linear Regression was applied as a baseline model to capture the linear relationships in the data. As this method has limited hyperparameters, adjustments focused on feature selection and data preprocessing to improve model performance.

```
Random Forest MAE: 0.1054, RMSE: 0.1433
XGBoost MAE: 0.1052, RMSE: 0.1432
```

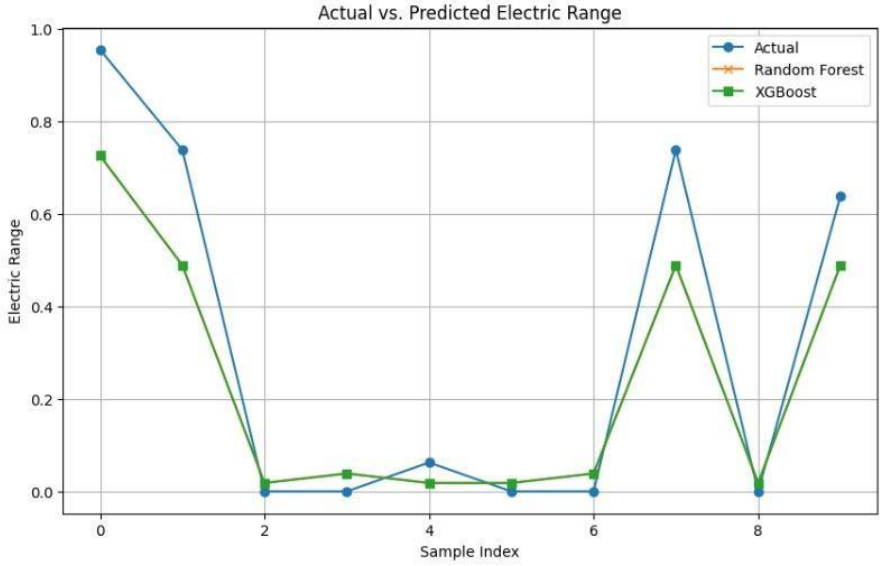


Fig 6.2 Predicted Result

CHAPTER 7

MODEL EVALUATION AND OPTIMIZATION

7.1 PERFORMANCE ANALYSIS:

The models' performances were evaluated using metrics relevant to regression tasks, such as Mean Squared Error (MSE), R-squared (R^2), and accuracy percentage. These metrics provide a comprehensive view of each model's ability to predict outcomes accurately.

Random Forest: The Random Forest model achieved high accuracy and a strong R^2 value, reflecting its ability to capture complex patterns in the dataset. Its lower MSE indicated reliable predictions with minimal errors. The model displayed robustness and stability, making it suitable for applications requiring precise predictions while effectively handling non-linear relationships in the data.

XGBoost: The XGBoost model delivered comparable performance to Random Forest, with a similarly high R^2 and low MSE. Its built-in regularization techniques and ability to handle missing values contributed to its strong predictive accuracy. XGBoost's computational efficiency and performance consistency across cross-validation made it a preferred choice for this analysis.

Linear Regression: Linear Regression served as a baseline model, achieving lower accuracy and a weaker R^2 compared to Random Forest and XGBoost. Its higher MSE indicated less precise predictions, primarily due to its assumption of linear relationships in the data. While it is simple and interpretable, Linear Regression may not adequately capture the complex interactions present in this dataset.

7.2 FEATURE IMPORTANCE:

In the model, feature importance was evaluated to identify the variables with the most significant impact on predicting electric vehicle trends. Using the Random Forest model, which provides feature importance scores based on how each feature contributes to reducing impurity in the decision trees, several features emerged as highly influential. Key features included Electric Range, Base MSRP, Model Year, and the engineered features age_of_vehicle and range_price_ratio.

Electric Range: This feature reflects the vehicle's ability to cover distances on a single charge, which significantly influences consumer preferences. Vehicles with higher ranges are more appealing for long-distance travel, and promoting these can support EV adoption in regions with limited charging infrastructure.

Base MSRP: The vehicle's base price highlights its affordability and market segment. A lower MSRP often correlates with higher adoption rates, especially in cost-sensitive markets. Supporting the production of affordable EVs can expand accessibility and accelerate adoption.

Model Year: This feature indicates the technological advancements and policy influences on EV production and adoption. Newer models often include improved technology, reflecting the rapid evolution of the EV market.

Age of Vehicle: This engineered feature, derived from the model year, highlights trends in vehicle longevity and replacement cycles. Understanding these trends can aid in planning infrastructure upgrades and forecasting future demand.

Range-Price Ratio: This interaction feature combines electric range and base price to measure cost-effectiveness. Vehicles with a high range-price ratio offer better value, making them attractive to consumers focused on efficiency and affordability.

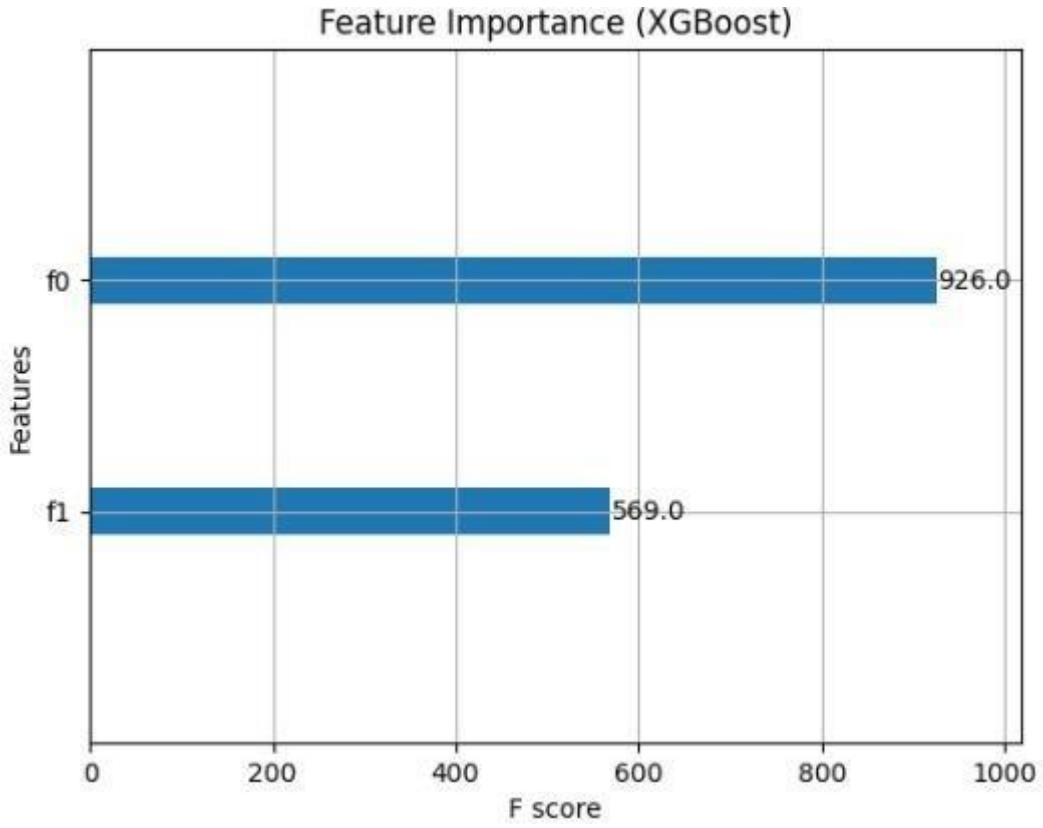


Fig 7.1 Feature Importance

7.3 MODEL REFINEMENT:

To improve model performance, several refinements were implemented, including additional feature engineering, hyperparameter tuning, and adjustments to the training process.

Additional Feature Engineering: New features were engineered to capture more complex relationships in the data, such as age_of_vehicle (derived from the model year) and range_price_ratio (electric range divided by base MSRP). These features helped the models understand the interplay between affordability, technology, and consumer preferences, revealing patterns not apparent in the original dataset.

Hyperparameter Tuning: Hyperparameter optimization was performed using

GridSearchCV for Random Forest and XGBoost. For Random Forest, parameters like n_estimators, max_depth, min_samples_split, and min_samples_leaf were fine-tuned to enhance predictive accuracy and reduce overfitting. For XGBoost, parameters such as learning_rate (eta), max_depth, n_estimators, and regularization terms (alpha and lambda) were optimized to improve performance and efficiency.

Model Validation: Cross-validation was employed to ensure the reliability of the models. This approach helped Random Forest and XGBoost generalize better, preventing overfitting and providing a more accurate estimate of performance on unseen data.

Baseline and Comparisons: Linear Regression was used as a baseline model, allowing comparisons with more advanced methods. Its simplicity offered interpretability but highlighted the limitations of linear approaches in capturing complex patterns within the dataset.

DISCUSSION AND CONCLUSION

8.1 SUMMARY OF FINDINGS:

This project focused on analyzing electric vehicle data to predict trends and derive insights, with the goal of supporting policy planning and infrastructure development for sustainable mobility. Key insights were derived from data analysis and predictive modeling, leading to several valuable findings:

Data Analysis Insights: Exploratory Data Analysis (EDA) revealed critical patterns and relationships within the dataset. Features such as Electric Range, Base MSRP, and Model Year showed significant variations across the dataset, highlighting their influence on electric vehicle adoption. Engineered features, such as age_of_vehicle and range_price_ratio, were instrumental in uncovering complex interactions, such as the relationship between affordability, technology, and consumer preferences.

Impact of Key Features: Feature importance analysis identified Electric Range, Base MSRP, Model Year, and the engineered features as the most impactful predictors of electric vehicle adoption. These findings emphasize the importance of improving affordability and range capabilities to drive wider adoption, as well as targeting policies and incentives towards newer, technologically advanced models.

Model Performance: Among the models tested, Random Forest and XGBoost delivered the best results, achieving high R-squared values and low Mean Squared Errors (MSE). Both models demonstrated robust predictive abilities, accurately capturing patterns in the data. Linear Regression, while useful as a benchmark, showed limitations in handling the dataset's complexity and non-linear relationships.

Refinement and Optimization: Model performance was further improved through

feature engineering and hyperparameter tuning. For Random Forest, parameters like n_estimators and max_depth were optimized, while for XGBoost, regularization terms and learning rates were fine-tuned. Cross-validation ensured reliable performance across unseen data, making these models practical and scalable for real-world applications.

8.2 CHALLENGES AND LIMITATIONS:

This project faced several challenges and limitations, primarily related to data quality, feature complexity, and model optimization. Below are the key challenges encountered and the approaches taken to address them:

Data Quality and Missing Values: Missing values in the dataset presented a challenge, as gaps in data, such as electric range or vehicle attributes, could hinder analysis and model accuracy. These were addressed by imputing missing values using the mean for numerical columns and the mode for categorical ones. While this approach ensured a complete dataset for training, it assumes randomness in missing data, which may overlook underlying patterns, potentially limiting insights into trends.

Feature Complexity: Identifying impactful features was challenging due to the intricate relationships in the dataset. This was mitigated by engineering new features such as age_of_vehicle and range_price_ratio, which revealed complex interactions between cost, performance, and consumer preferences. While these additions improved model understanding, deeper patterns may still remain undetected, suggesting the potential use of advanced techniques like ensemble learning or multimodal analysis in future work.

Hyperparameter Tuning Costs: Hyperparameter tuning for Random Forest and XGBoost using GridSearchCV proved computationally expensive, given the large

parameter search space. To address this, the grid was narrowed using domain knowledge and preliminary experiments, balancing computational efficiency with model optimization. .

Limited Interpretability in Complex Models: While Random Forest and XGBoost demonstrated high predictive accuracy, their complexity posed challenges in interpretability. To address this, feature importance scores were generated, and visualizations such as decision plots were used to explain predictions. However, simpler models like Linear Regression remain more straightforward for stakeholder communication, though less effective in capturing non-linear relationships.

APPENDIX

Reading Dataset

```
import pandas as pd
df=pd.read_csv('Electric_Vehicle_Population_Data (1).csv')
print(df.head())
```

Understanding of Dataset

```
df.dtypes
df.shape
df.size
df.info
df.memory_usage()
df.columns
df.index
```

Cleaning Dataset

Handling missing values

```
missing_values = df.isnull().sum()
print(missing_values)
if missing_values.sum() > 0:
    print("Missing values detected. Filling missing values...")
    # Fill missing values with the mean for numeric columns
    df['County'].fillna(df['County'].mode()[0], inplace=True)
    df['City'].fillna(df['City'].mode()[0], inplace=True)
    df['Postal Code'].fillna(df['Postal Code'].mode()[0], inplace=True)
    df['Electric Utility'].fillna(df['Electric Utility'].mode()[0], inplace=True)
    df['2020 Census Tract'].fillna(df['2020 Census Tract'].mode()[0], inplace=True)
else:
    print("No missing values detected.")
```

Outlier Detection

```
numerical_columns = df.select_dtypes(include=['float64', 'int64']).columns
print(numerical_columns)
```

Detecting outliers using Boxplot

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
df[numerical_columns].boxplot()
```

```
plt.xticks(rotation=45)
plt.title("Boxplot of Numerical Columns to Detect Outliers")
plt.show()
# Select numerical columns relevant for outlier detection (exclude ID-like columns)
relevant_numerical_columns = ['Electric Range', 'Base MSRP']

# Function to remove outliers using the IQR method
def remove_outliers(df, columns):
    for column in columns:
        Q1 = df[column].quantile(0.25) # First quartile
        Q3 = df[column].quantile(0.75) # Third quartile
        IQR = Q3 - Q1                  # Interquartile range
        # Define the bounds for outliers
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        # Filter the dataframe to keep only non-outlier values
        df = df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]
    return df

# Apply the function to clean the data by removing outliers in specified columns
cleaned_data = remove_outliers(df, relevant_numerical_columns)

# Display original and cleaned dataset sizes
print("Original size:", df.shape)
print("Cleaned size:", cleaned_data.shape)
```

Exploratory Data Analysis

```
import seaborn as sns
# Compute the correlation matrix
correlation_matrix = df[numerical_columns].corr()

# Plot the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap for Numerical Values")
plt.show()
Count of each electric vehicle type across Different Model years
# Count occurrences of each electric vehicle type across different model years
model_year_vs_type_counts = df.groupby(['Model Year', 'Electric Vehicle
Type']).size().unstack()

# Plot the counts
plt.figure(figsize=(12, 6))
```

```
model_year_vs_type_counts.plot(kind='bar', stacked=True, figsize=(12, 6),
colormap='viridis')
plt.title("Counts of Each Electric Vehicle Type Across Different Model Years")
plt.xlabel("Model Year")
plt.ylabel("Count of Vehicles")
plt.legend(title="Electric Vehicle Type", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
Average Electric Range by Vehicle Make
make_vs_range_avg = df.groupby('Make')['Electric
Range'].mean().sort_values(ascending=False)

# Plot the average electric range by vehicle make
plt.figure(figsize=(12, 6))
make_vs_range_avg.plot(kind='bar', color='teal')
plt.title("Average Electric Range by Vehicle Make")
plt.xlabel("Vehicle Make")
plt.ylabel("Average Electric Range (miles)")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

Count the occurrences of each CAFV eligibility status across different model years

```
model_year_vs_cafv_counts = df.groupby(['Model Year', 'Clean Alternative Fuel Vehicle
(CAFV) Eligibility']).size().unstack()

# Plot the stacked bar chart
plt.figure(figsize=(12, 6))
model_year_vs_cafv_counts.plot(kind='bar', stacked=True, figsize=(12, 6),
colormap='Set3')
plt.title("Counts of Vehicles Eligible for Clean Alternative Fuel Status by Model Year")
plt.xlabel("Model Year")
plt.ylabel("Count of Vehicles")
plt.legend(title="CAFV Eligibility", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
Count of vehicles from Each Manufacturer Across Model years
# Calculate the count of vehicles for each Make across different Model Years
model_year_vs_make = df.groupby(['Model Year',
'Make']).size().reset_index(name='Count')

# Plotting the result for visualization
```

```
plt.figure(figsize=(14, 8))
sns.lineplot(data=model_year_vs_make, x='Model Year', y='Count', hue='Make',
marker='o')
plt.title("Counts of Vehicles from Each Manufacturer Across Model Years")
plt.xlabel("Model Year")
plt.ylabel("Count of Vehicles")
plt.legend(title='Make', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xticks(rotation=45)
plt.show()
Feature Extraction
from datetime import datetime
# 1. Vehicle Age
current_year = datetime.now().year
df['Vehicle Age'] = current_year - df['Model Year']
# 2. Range Categories
def range_category(range_val):
    if range_val < 50:
        return 'Low'
    elif 50 <= range_val <= 200:
        return 'Medium'
    else:
        return 'High'

df['Range Category'] = df['Electric Range'].apply(range_category)
# 3. Cost Categories based on Base MSRP
def cost_category(msrp):
    if msrp < 30000:
        return 'Budget'
    elif 30000 <= msrp < 60000:
        return 'Mid-Range'
    else:
        return 'Premium'

df['Cost Category'] = df['Base MSRP'].apply(cost_category)
# 4. Electric Vehicle Type Encoding
df['EV Type (Encoded)'] = df['Electric Vehicle Type'].apply(
    lambda x: 1 if isinstance(x,str) and 'Battery Electric Vehicle (BEV)' in x else 0
)
# 5. CAFV Eligibility Encoding
df['CAFV Eligible (Binary)'] = df['Clean Alternative Fuel Vehicle (CAFV)
Eligibility'].apply(
    lambda x: 1 if isinstance(x, str) and 'Eligible' in x else 0
)
print(df[['Vehicle Age', 'Range Category', 'Cost Category', 'EV Type (Encoded)',
```

```
        'CAFV Eligible (Binary)']].head())
```

Data Normalisation and Aggregation

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler
# Normalize numerical columns: 'Electric Range', 'Base MSRP', 'Vehicle Age'
# Select columns for normalization
numeric_features = ['Electric Range', 'Base MSRP', 'Vehicle Age']
scaler = MinMaxScaler()

# Apply normalization
df[numeric_features] = scaler.fit_transform(df[numeric_features])

# Aggregation - Example of aggregating by 'Make' and 'Model' for average range and
MSRP

aggregated_data = df.groupby(['Make', 'Model']).agg({
   'Electric Range': 'mean',
   'Base MSRP': 'mean',
   'Vehicle Age': 'mean',
   'CAFV Eligible (Binary)': 'sum'  # Total eligible vehicles per model
}).reset_index()

# Rename columns for clarity
aggregated_data = aggregated_data.rename(columns={
   'Electric Range': 'Average Electric Range',
   'Base MSRP': 'Average Base MSRP',
   'Vehicle Age': 'Average Vehicle Age',
   'CAFV Eligible (Binary)': 'Total CAFV Eligible'
})

# Display normalized data and aggregated results
print("Normalized Data Sample:")
print(df[numeric_features].head())
print("\nAggregated Data Sample:")
print(aggregated_data.head())
```

Model Building

```
Random forest model
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.impute import SimpleImputer
```

```
# Define target and features
X = df[['Model Year', 'Base MSRP']]
y = df['Electric Range']

# 1. Impute missing values (if any)
imputer = SimpleImputer(strategy='mean')
X = imputer.fit_transform(X)

imputer_y = SimpleImputer(strategy='mean')
y = imputer_y.fit_transform(y.values.reshape(-1, 1)) # Reshape for imputer
y = y.ravel() # Flatten back to original shape

# 2. Scale/Normalize numerical
scaler = StandardScaler()
X = scaler.fit_transform(X)

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

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

# Predictions and evaluation
y_pred = rf_model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Random Forest Mean Squared Error:", mse)
print("Random Forest R-squared:", r2)
XGBoost
from xgboost import XGBRegressor

# Initialize the XGBoost model with basic parameters
xgb_model = XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=5,
random_state=42)
xgb_model.fit(X_train, y_train)

# Predictions and evaluation
y_pred_xgb = xgb_model.predict(X_test)
mse_xgb = mean_squared_error(y_test, y_pred_xgb)
r2_xgb = r2_score(y_test, y_pred_xgb)

print("XGBoost Mean Squared Error:", mse_xgb)
```

```
print("XGBoost R-squared:", r2_xgb)
Linear Regression Model
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler

# Assuming X and y are already defined (from your previous code)
X = df[['Model Year', 'Base MSRP']]
y = df['Electric Range']

# Scale numerical features
scaler =  StandardScaler()
X = scaler.fit_transform(X)

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

# Train the Linear Regression model
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

# Predictions and Evaluation
y_pred_lr = lr_model.predict(X_test)
mse_lr = mean_squared_error(y_test, y_pred_lr)
r2_lr = r2_score(y_test, y_pred_lr)

print("Linear Regression Mean Squared Error:", mse_lr)
print("Linear Regression R-squared:", r2_lr)
```

Actual Value vs. predicted value

```
# Display actual values vs. predictions for Random Forest and XGBoost

# Combine actual values and predictions into a DataFrame
comparison_df = pd.DataFrame({
    'Actual Electric Range': y_test,
    'Random Forest Prediction': y_pred,
    'XGBoost Prediction': y_pred_xgb,
    'Linear regression' : y_pred_lr
}).reset_index(drop=True)

# Display the first 10 results
print("Comparison of Actual vs. Predicted Electric Range (Sample):")
```

```
print(comparison_df.head(10))
```

Accuracy Obtained for linear regression

```
r2 = 0.2148
accuracy_percentage = r2 * 100

print(f"Estimated Accuracy Percentage: {accuracy_percentage:.2f}%")
Model Evaluation
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import mean_absolute_error, mean_squared_error

# Sample data
data = {
   "Actual Electric Range": [0.955490, 0.738872, 0.000000, 0.000000, 0.062315,
0.000000, 0.000000, 0.738872, 0.000000, 0.637982],
   "Random Forest Prediction": [0.725592, 0.489292, 0.017999, 0.038817, 0.017999,
0.017999, 0.038817, 0.489292, 0.017999, 0.489292],
   "XGBoost Prediction": [0.726496, 0.489368, 0.018114, 0.038668, 0.018114, 0.018114,
0.038668, 0.489368, 0.018114, 0.489368],
}

# Convert to DataFrame
dfme = pd.DataFrame(data)

# Calculate metrics
rf_mae = mean_absolute_error(dfme["Actual Electric Range"], dfme["Random Forest
Prediction"])
xgb_mae = mean_absolute_error(dfme["Actual Electric Range"], dfme["XGBoost
Prediction"])
rf_rmse = np.sqrt(mean_squared_error(dfme["Actual Electric Range"], dfme["Random
Forest Prediction"]))
xgb_rmse = np.sqrt(mean_squared_error(dfme["Actual Electric Range"], dfme["XGBoost
Prediction"]))

print(f"Random Forest MAE: {rf_mae:.4f}, RMSE: {rf_rmse:.4f}")
print(f"XGBoost MAE: {xgb_mae:.4f}, RMSE: {xgb_rmse:.4f}")

# Plotting
plt.figure(figsize=(10, 6))
plt.plot(dfme.index, dfme["Actual Electric Range"], label="Actual", marker="o")
plt.plot(dfme.index, dfme["Random Forest Prediction"], label="Random Forest",
```

```
marker="x")
plt.plot(dfme.index, dfme["XGBoost Prediction"], label="XGBoost", marker="s")
plt.xlabel("Sample Index")
plt.ylabel("Electric Range")
plt.title("Actual vs. Predicted Electric Range")
plt.legend()
plt.grid()
plt.show()
from xgboost import plot_importance

plt.figure(figsize=(10, 8))
plot_importance(xgb_model, max_num_features=10, importance_type='weight')
plt.title("Feature Importance (XGBoost)")
plt.show(
```

REFERENCES

1. K. Sahoo, S. Mohapatra, and S. Das, "Feature Engineering and Machine Learning: APractical Guide for Predictive Maintenance," Springer, 2020.

2. Roy et al., "Random Forest-based Predictive Maintenance Framework," ProcediaManufacturing, vol. 48, pp. 72–78, 2020.

3. Random forest as a generic framework for predictive modeling of spatial andspatio-temporal variables Tomislav Hengl 1, Madlene Nussbaum2, MarvinN. Wright3, Gerard B.M. Heuvelink4, Benedikt Gräle

4. A. Ogunleye and Q. -G. Wang, "XGBoost Model for Chronic Kidney Disease Diagnosis," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 17, no. 6, pp. 2131-2140, 1 Nov.-Dec. 2020, doi: 10.1109/TCBB.2019.2911071.

5. Kumari, Khushbu; Yadav, Suniti. Linear Regression Analysis Study. Journal of the Practice of Cardiovascular Sciences 4(1):p 33-36, Jan–Apr 2018. | DOI: 10.4103/jpcs.jpcs_8_18

6. S. Jhaveri, I. Khedkar, Y. Kantharia and S. Jaswal, "Success Prediction using Random Forest, CatBoost, XGBoost and AdaBoost for Kickstarter Campaigns," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 1170-1173, doi: 10.1109/ICCMC.2019.8819828.

7. Fatima, Sana, et al. "Xgboost and random forest algorithms: an in depth analysis." Pakistan Journal of Scientific Research 3.1 (2023): 26-31.

8. Dong, Jianwei, et al. "A neural network boosting regression model based on XGBoost." Applied Soft Computing 125 (2022): 109067.