EVision – Intelligent Range Prediction for Smarter EV Decisions

Project by

Ujjwal Pandit | Vedant More | Chetan Bhangare | Hari Prasath K P | Chetan Kharkar

Introduction

What is the Problem?

The problem is to identify the electric vehicle (EV) model with the highest mileage per unit of charge.This requires building a regression model to predict the mileage of each EV model based on features such as Cost per unit charge,Time taken to charge and Energy consumed etc

Why Does it Matter?

1. Sustainability
2. Market Value

Objective of the Project :

1. Use regression models to predict EV mileage based on key factors like charging cost, time, and energy consumption and analyze the relationship between these factors

Dataset Overview

User Details: Unique User ID, Vehicle Model, Vehicle Age, and User Type (e.g., Commuter, Long-Distance Traveler).

Charging Session Information: Charging Station ID, Location, Charger Type, Start/End Times, Duration, Energy Consumed, and Charging Rate.

Battery Metrics: Battery Capacity (kWh), State of Charge at Start and End (%).

Driving Patterns: Distance Driven Since Last Charge (km).

Environmental and Contextual Data: Ambient Temperature (°C), Time of Day, and Day of Week.

Financial Insights: Charging Cost (USD).

Data preprocessing

Handling Missing Values: Energy Consumed and Distance Driven (66 missing values each).

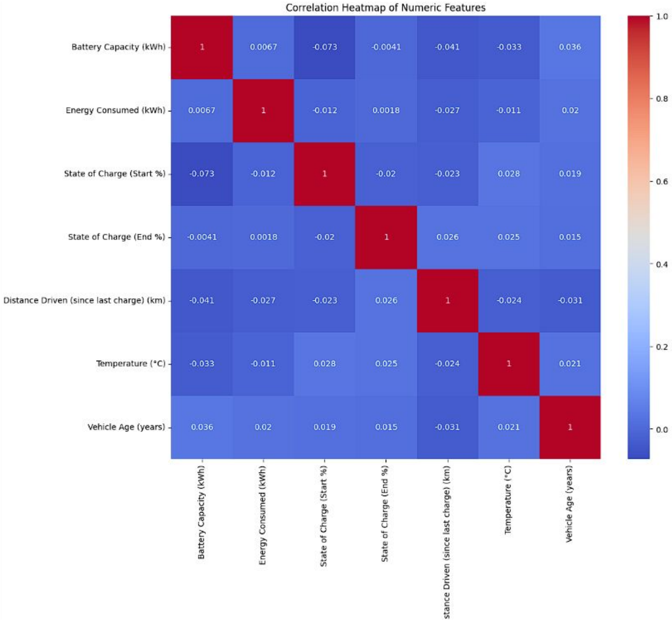
Approaches Taken: Target Label ('Distance Driven'): Rows with NaN values were removed.

Training Feature ('Energy Consumed'): Experimented with imputation methods:

Mean Imputation, Median Imputation (chosen for better performance), Removal of NaN values

Outlier Handling

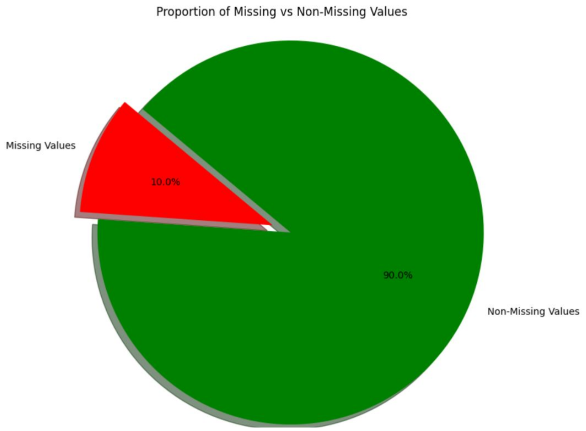
Correlation and Feature Selection




```
Index(['Battery Capacity (kWh)', 'Energy Consumed (kWh)',
       'State of Charge (Start %)', 'State of Charge (End %)',
       'Distance Driven (since last charge) (km)', 'Temperature (°C)',
       'Vehicle Age (years)', 'Vehicle Model'],
      dtype='object')
```

Handling NaN Values

1. For the target label (likely 'Distance driven'), we opted to remove the rows with NaN values.
2. For 'Energy consumed', which is a training feature, we experimented with several imputation methods:
    - Mean imputation
    - Median imputation
    - Removal of NaN values




```
*******************************************************************
Battery Capacity (kWh)                        0
Energy Consumed (kWh)                        66
State of Charge (Start %)                     0
State of Charge (End %)                       0
Distance Driven (since last charge) (km)     66
Temperature (°C)                              0
Vehicle Age (years)                           0
Vehicle Model                                 0
dtype: int64
*******************************************************************
Msiisng percentage: 10.0
msiing value:  132
*******************************************************************
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1320 entries, 0 to 1319
Data columns (total 8 columns):
 #   Column                                    Non-Null Count  Dtype
---  ------                                    --------------  -----
 0   Battery Capacity (kWh)                    1320 non-null   float64
 1   Energy Consumed (kWh)                     1254 non-null   float64
 2   State of Charge (Start %)                 1320 non-null   float64
 3   State of Charge (End %)                   1320 non-null   float64
 4   Distance Driven (since last charge) (km)  1254 non-null   float64
...
 7   Vehicle Model                             1320 non-null   object
dtypes: float64(7), object(1)
```

Outliers handling

1. Why Remove Outliers?

improve Model Accuracy: Eliminates extreme values that can skew predictions and lead to bias.

Enhance Model Generalization: Helps the model focus on typical data patterns and reduces the risk of overfitting.

Cleaner Dataset: Ensures only relevant, realistic data points are used for training.

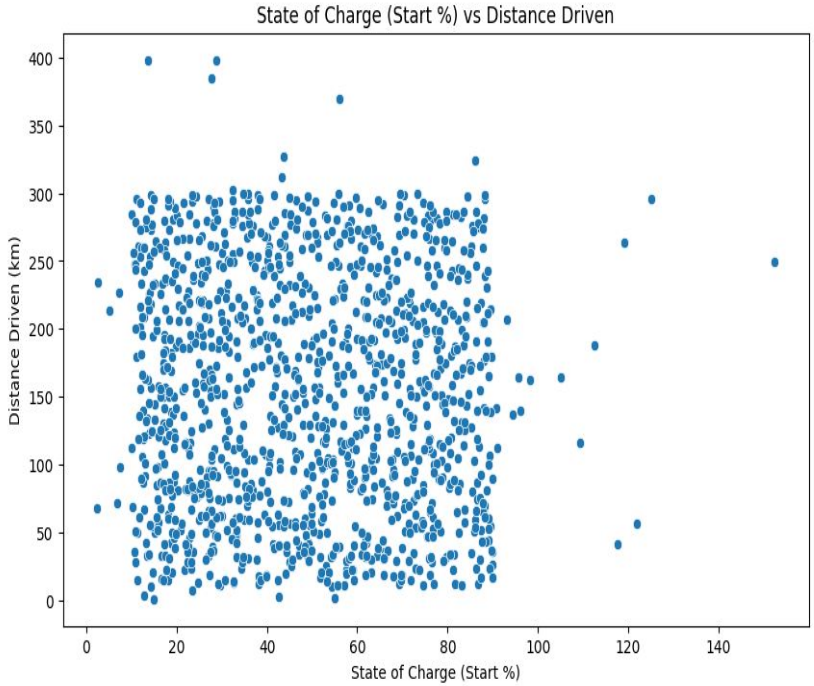
Identifying and removing outliers

1.Identified 10 rows where the target column "distance driven" exceeded 300 km.

2.Outlier Identification Method :

a.Visual Inspection

b.Thresholding Approach



Performance Comparison of Regression Models

Objective: Evaluate multiple machine learning model to predict the target variable

Models Tested:

Random Forest Regressor

Gradient Boosting Machine

Support Vector Regressor

K-Nearest Neighbors

Key Metrics Used:

R-squared value

Mean Absolute Error (MAE)

Mean Squared Error (MSE)

Root Mean Squared Error (RMSE)

Random Forest Regressor: Superior Performance

R-squared: 0.8524 (Explains 85.24% of data variance)

Error Metrics:

MAE: 27.23

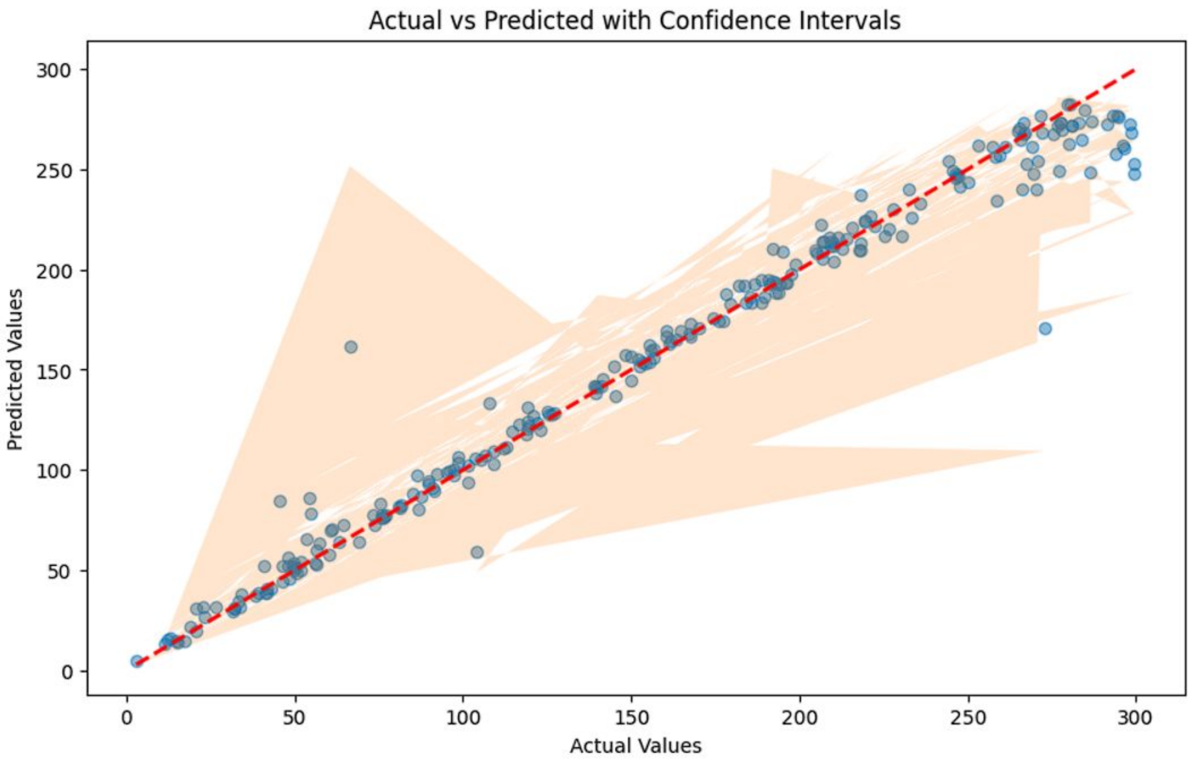
MSE: 1065.48

RMSE: 32.64

Conclusion

The Random Forest model demonstrated superior performance, achieving an R-squared value of 0.8524, indicating it explains 85.24% of the variance in the target variable. Its error metrics, including a Mean Absolute Error (MAE) of 27.23, a Mean Squared Error (MSE) of 1065.48, and a Root Mean Squared Error (RMSE) of 32.64, highlight its accuracy and reliability in making predictions. These results establish Random Forest as the most effective model for this dataset.

How well Random Forest Regressor Performs?



Future scope

Enhanced Dataset Quality
Improve prediction accuracy by acquiring comprehensive datasets, including real-world driving conditions, weather patterns, road types, and vehicle-specific parameters.

Advanced Machine Learning Techniques

- **Artificial Neural Networks (ANNs)**: Leverage ANNs to model complex relationships between features and EV mileage.

Price-Based Model Selection
Integrate price data to expand the model's functionality, enabling the prediction of the best EV model based on mileage, cost efficiency, and user preferences.

THANK YOU!

Any questions?