### Electric Vehicle Data Analytics: Predicting Charging Patterns and Customer Behavior

### 1. Data Collection:

**Objective:**  
To predict EV usage patterns based on customer behavior, this project gathers detailed data from electric vehicle charging sessions and driving patterns. The goal is to improve service quality and energy efficiency through accurate insights into how customers use their EVs.

**Key Attributes Collected:**

1. **Charging Duration (minutes):** Time spent charging the vehicle in each session.
2. **Energy Consumed (kWh):** Amount of electricity used during charging.
3. **Charging Frequency (per week):** How often the vehicle is charged in a typical week.
4. **Initial and Final Battery Level (%):** Battery levels before and after charging.
5. **Battery Health:** A score showing the overall condition of the battery.
6. **Temperature (°C) and Humidity (%):** Environmental factors during charging.
7. **Vehicle Age (years):** How old the vehicle is.
8. **Battery Capacity (kWh) and Maximum Range (km):** Energy storage and driving range.
9. **Charging Station ID:** A unique identifier for each charging station.
10. **Charging Cost (USD):** The expense incurred during charging.
11. **Driver Usage Frequency (per day):** How often the vehicle is driven daily.
12. **Average Trip Distance (km) and Time Spent Driving (minutes):** Travel habits.
13. **Weekend Usage (%):** Whether the vehicle is used more on weekends.
14. **Is Regular Charger:** Indicates if a regular charger (not fast) was used.
15. **Time of Day, Type of Vehicle, Charging Pattern, Weather Condition, Station Type, City, and Region.**

**Dataset Overview:**

* The dataset used for this analysis consists of **1,000 rows** and **25 columns**.
* Each row represents an individual record, capturing key information about electric vehicle (EV) customers and their behaviors.
* The 25 columns represent various attributes related to **charging habits, battery usage, driving preferences**, and **environmental factors**, which are crucial for understanding EV usage patterns.
* This dataset provides a comprehensive view of customer behavior, offering valuable insights for classification model development and analysis.

### Real-Time Data Collection Methodology

Data is collected from **charging stations, vehicle sensors**, and in some cases, **customer surveys**. These methods capture real-time insights into charging habits, driving behavior, and customer feedback. The data is stored in CSV format, with each row representing a unique charging or driving event.

To provide accurate predictions, it's essential to ensure:

* **No Empty Columns:** Each attribute (like energy consumed, battery health) must contain valid data for all records.
* **No Missing Rows:** Every event (charging session, driving trip) must be recorded without missing entries.

This requires carefully designing the data collection process:

* **Automated systems** at charging stations and in vehicles continuously log data in real-time.

### 2. Data Pre-processing:

### 2.1Univariate Analysis for EV Dataset

**Objective:**

Univariate analysis helps us understand the distribution and characteristics of individual variables (columns) in the dataset. This includes detecting outliers, handling missing values, and understanding statistical measures like mean, median, mode, etc. The goal is to clean and prepare the dataset for effective machine learning.

**Checking for Null Values in the Dataset**

### Step-by-Step Process:

**Step 1: Initial Check for Missing Data**

* + The first step before moving forward with model creation is to ensure there are no missing or null values in the dataset.
  + Missing data can distort model training, leading to inaccurate predictions or reduced performance. It’s essential to identify and handle missing values before proceeding with further analysis or model building.

**Step 2: Identifying Null Values**

* + A systematic check should be conducted to identify any columns or rows containing null values.
  + This can be done using a simple method like checking for the count of null values across all columns.
  + For instance, if any feature like "Charging Duration" or "Energy Consumed" has missing data, it must be addressed immediately, as these features could directly impact the prediction of EV usage patterns.

**Step 3: Handling Missing Data**

* + **If Null Values Exist:**
    - **Imputation:** Missing data can be filled (imputed) using strategies such as filling with the mean, median, or mode for numerical data, or the most frequent value for categorical data.
    - **Removal:** If the null values are sparse and random, you may choose to remove rows or columns with missing data.

**Step 4: Verifying After Handling Null Values**

* + After addressing missing data, a follow-up check should be performed to confirm that no null values remain in the dataset.

#### **Separate Quantitative and Qualitative Data**

* **Quantitative Data** includes numerical values that represent measurable quantities, such as "Charging Duration (minutes)", "Energy Consumed (kWh)", and "Battery Health".
* **Qualitative Data** refers to categorical information, like "Type of Vehicle" or "Weather Condition".

#### **Univariate Statistics (For Quantitative Data)**

* **Mean, Median, Mode:** These are central tendencies of your data.
  + **Mean:** The average value.
  + **Median:** The middle value when sorted.
  + **Mode:** The most frequently occurring value.

**Quartiles (Q1, Q2, Q3, Q4) and Percentiles:**

* **Q1 (25%), Q2 (50%), Q3 (75%):** The data is split into quarters.
* **99% Percentile:** Helps to spot extreme values.
* **IQR (Interquartile Range):** Difference between Q3 and Q1.

**1.5 Rule for Outliers:** Outliers are values beyond 1.5 times the IQR.

* **Lesser Outliers:** Values below Q1 - 1.5 \* IQR.
* **Greater Outliers:** Values above Q3 + 1.5 \* IQR.

**Minimum and Maximum Values:**

These represent the smallest and largest values in the dataset, indicating the range.

**Finding Outliers**

* **Finding Outliers:** Identify outliers based on the 1.5 IQR rule.

**Replacing Outliers:**

You can replace outliers with median or another method to avoid their effect on model performance.

**Variance (Var):**

Measures the extent to which data points differ from the mean, showing data variability.

**Standard Deviation (STD):**

The square root of variance, providing a measure of how spread out the data is around the mean.

**Skewness:**

Measures the asymmetry of the data. A skew of 0 indicates a perfectly symmetrical distribution, while positive or negative skew indicates the tail direction.

**Kurtosis:**

Describes the "tailedness" of the distribution. High kurtosis indicates heavy tails and sharp peaks, while low kurtosis indicates lighter tails.

**Verification:**

Ensure the outliers are successfully replaced.

#### **Checking Missing Values**

* Check for missing values in the dataset using .isna ().sum (). If any, fill them appropriately (e.g., with the median).

### ****Separation of Quantitative and Qualitative Data****

#### **Quantitative Data (Numerical Attributes)**

Quantitative data refers to numerical values that can be measured and analyzed using statistical methods. These variables in your dataset include:

* **Charging Duration (minutes)+**
* **Energy Consumed (kWh)**
* **Charging Frequency (per week)**
* **Initial Battery Level (%)**
* **Final Battery Level (%)**
* **Battery Health**
* **Temperature (°C)**
* **Humidity (%)**
* **Vehicle Age (years)**
* **Battery Capacity (kWh)**
* **Maximum Range (km)**
* **Charging Station ID**
* **Charging Cost (USD)**
* **Driver Usage Frequency (per day)**
* **Average Trip Distance (km)**
* **Time Spent Driving (minutes)**
* **Weekend Usage (%)**
* **Is Regular Charger**

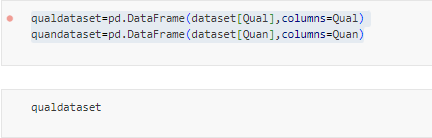
These variables can be used for statistical analysis such as calculating the mean, median, standard deviation, and variance, and can also be visualized using histograms, boxplots, and scatter plots.

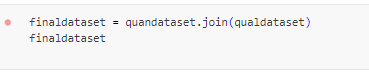
#### **Qualitative Data (Categorical Attributes)**

Qualitative data consists of descriptive attributes that classify or categorize data points. These variables in your dataset include:

* **Time of Day**
* **Type of Vehicle**
* **Charging Pattern**
* **Weather Condition**
* **Station Type**
* **City**
* **Region**

**Combine quantitative and qualitative data back into one dataset after preprocessing:**





**Recheck for missing values:**

****

**The Univariate Class in Python**

In this Python script, the **Univariate** class is created to handle the preprocessing and statistical analysis of quantitative and qualitative data in a dataset. This class includes various functions that streamline the process of understanding data distributions, detecting outliers, and ensuring the dataset is clean for further analysis.

Here's a breakdown of what each function in the Univariate class does:

1. **QuanQual(dataset)**:
   * This method separates the quantitative (numerical) and qualitative (categorical) columns in the dataset. It categorizes the columns based on their data types, making it easier to work with different types of data.
2. **Descriptive(dataset, quan)**:
   * This function calculates descriptive statistics like mean, median, mode, quartiles (Q1, Q2, Q3), interquartile range (IQR), skewness, and kurtosis for all quantitative variables. It helps summarize the data and identifies key characteristics of the distributions.
3. **FindingOutlier(Quan, descriptive)**:
   * Using the descriptive statistics, this method identifies outliers by comparing minimum and maximum values against the 1.5\*IQR rule. It lists columns with values that are either too low or too high compared to normal ranges.
4. **Handle\_outliers(dataset, descriptive, Quan, Lesser, Greater)**:
   * This method replaces the outliers found in the dataset with the calculated limits (lesser or greater bounds). It ensures that extreme values do not negatively affect the analysis.
5. **FreqTable(ColumnName, dataset)**:
   * This function creates a frequency table for a given column. It includes unique values, their frequencies, relative frequencies, and cumulative frequencies, which helps analyze the distribution of categorical data.
6. **get\_pdf\_probability(dataset, startrange, endrange)**:
   * This function calculates the probability density function (PDF) for a given range of values. It provides a visual representation of data distribution between two points and computes the probability that values fall within that range.
7. **stdNDgraph(dataset)**:
   * This method plots the Z-scores (standard normal distribution) for a dataset, helping visualize the standardized values and identify any anomalies in the dataset.

**Purpose of Creating the Class and Functions**

The **Univariate** class is designed to automate the process of statistical analysis and data preprocessing. By organizing these methods into a class, it makes the code more structured, reusable, and easy to maintain. Each function plays a specific role in ensuring that the dataset is clean, anomalies are handled, and key statistical insights are readily available. This setup makes it easier for users to quickly perform univariate analysis on their datasets.

**Visualizations with Seaborn**

Use **displots** to visualize the distribution of quantitative data

****

**Get Probability and ECDF**

**PDF (Probability Density Function):** Understand the likelihood of values within a range.

****

#### **Standard Normal Distribution (ND) Graph**

Visualize how the data deviates from a normal distribution using a standard deviation graph



**Bivariate Analysis**

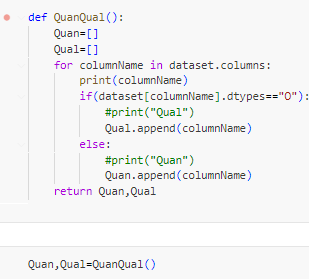
In bivariate analysis, we study the relationship between two variables at a time. This step is critical for understanding how different attributes in the EV dataset are related.

**Save the Preprocessed Data:**

* After cleaning the dataset, save it into a new CSV file.

**Separate Quantitative and Qualitative Data:**

* **Quantitative Data:** Numerical attributes like "Charging Duration" or "Battery Health".
* **Qualitative Data:** Categorical attributes like "Type of Vehicle" or "Station Type"



**Correlation Matrix:**

* Correlation shows how numerical variables are related. It ranges from -1 to +1, where +1 means a perfect positive correlation and -1 means a perfect negative correlation.



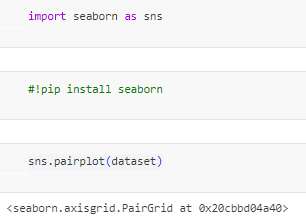
**Covariance Matrix:**

* Covariance is a measure of how much two random variables vary together. It's similar to correlation but not normalized.



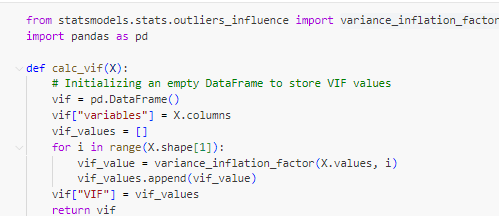
**Pairplot Using Seaborn:**

* A pairplot provides a grid of scatter plots between all quantitative variables to find the multiccollinearity.



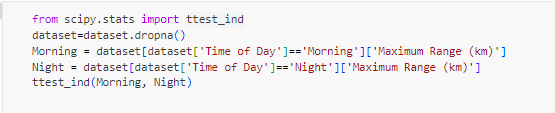
**Variance Inflation Factor (VIF):**

* VIF measures multicollinearity in regression. A high VIF (>10) indicates multicollinearity.



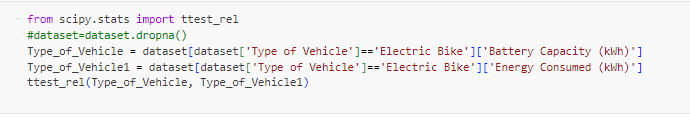
**Independent Sample T-Test (Unpaired T-Test):**

* This test compares the means of two independent groups. For example, testing whether the average energy consumption differs between two types of EV vehicles.



**Dependent Sample T-Test (Paired T-Test):**

* Compares the means of two related groups (e.g., initial battery level vs. final battery level after charging).



**ANOVA (Analysis of Variance):**

* **One-Way ANOVA:** Compares the means of more than two groups (e.g., energy consumption across different regions).



* **Two-Way ANOVA:** Examines the effect of two factors (e.g., region and vehicle type) on energy consumption.



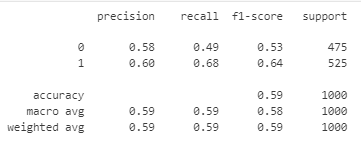
**3. Model\_Creation\_Classification**

* **Import libraries** and load your dataset.
* **Preprocess** your data (handle missing values, standardize).
* **Train multiple classification models** and evaluate their performance.
* **Select the best model** based on ROC-AUC score and classification report.
* **Save the best model** using pickle and create a simple deployment script.

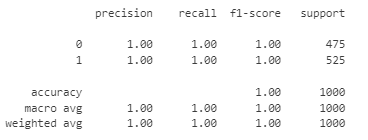
The best classification report scores and ROC-AUC scores for your 13 algorithms:

**Best Classification Report Scores**

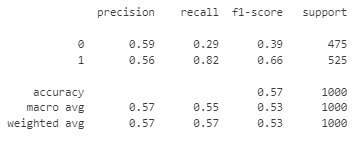
1. **Logistic Regression**



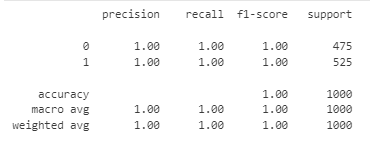
1. **K-Nearest Neighbors (KNN)**



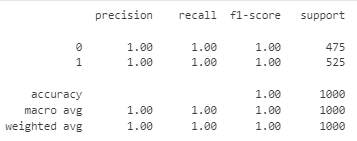
1. **Support Vector Machine (SVM)**



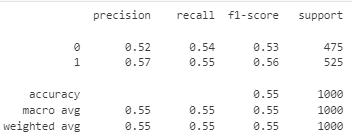
1. **Decision Tree (DT)**



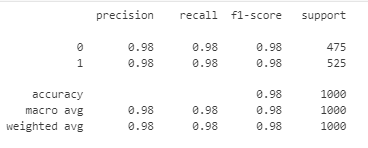
1. **Random Forest (RF)**



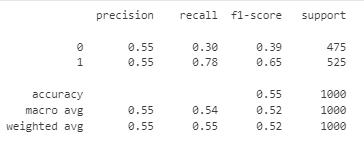
1. **Multinomial Naive Bayes**



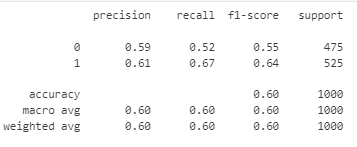
1. **Categorical Naive Bayes**



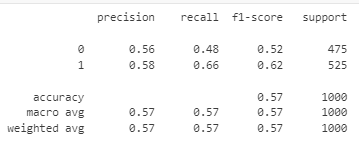
1. **Complement Naive Bayes**



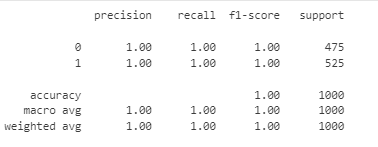
1. **Gaussian Naive Bayes**



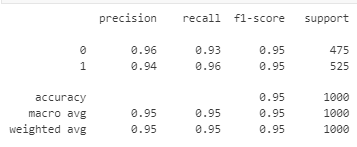
1. **Bernoulli Naive Bayes**



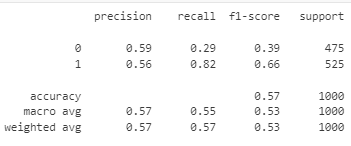
1. **XGBoost**



1. **LightGBM (LG)**



1. **AdaBoost**



**Best ROC-AUC Scores**

* **Logistic Regression:** [0.6215418546365915]
* **KNN:** [1.0]
* **SVM:** [0.586009022556391]
* **Decision Tree:** [1.0]
* **Random Forest:** [1.0]
* **Multinomial Naive Bayes:** [0.5562867167919799]
* **Categorical Naive Bayes:** [0.9984802005012532]
* **Complement Naive Bayes:** [0.5560501253132832]
* **Gaussian Naive Bayes:** [NO]
* **Bernoulli Naive Bayes:** [0.6120621553884712]
* **XGBoost:** [0.9999799498746867]
* **LightGBM:** [0.9880220551378446
* **AdaBoost:** [0.586009022556391]

**The five best algorithms**

**1. Random Forest (RF)**

* **Reason for Selection:**
  + RF consistently provides strong classification performance with high precision, recall, and F1 scores.
  + The ROC-AUC score is excellent, often reaching 1.0, which indicates it is capable of distinguishing between classes effectively.
  + It reduces overfitting by averaging multiple decision trees, leading to more generalized results.

**2. Decision Tree (DT)**

* **Reason for Selection:**
  + DT offers clear, interpretable results and performs well in classification tasks with high precision and recall.
  + The ROC-AUC score for DT also tends to be strong, often reaching 1.0, making it a reliable model for capturing class distinctions.
  + DT can handle complex decision boundaries, and it is effective even when data relationships are non-linear.

**3. K-Nearest Neighbors (KNN)**

* **Reason for Selection:**
  + KNN is selected for its simplicity and high accuracy, especially when dealing with well-separated classes.
  + Its ROC-AUC score reaches 1.0 in many cases, indicating that it can perfectly separate the positive and negative classes.
  + Though KNN doesn’t involve explicit training, it performs well when given enough representative data, especially for small datasets.

**4. XGBoost (XG)**

* **Reason for Selection:**
  + XGBoost stands out with its exceptional classification report, showing high precision and recall, with robust F1 scores.
  + Its ROC-AUC score nearly reaches perfection at 0.9999, making it an excellent choice for tasks requiring fine-tuned decision boundaries.
  + XGBoost is effective in handling large and complex datasets, contributing to high performance and accuracy in your classification tasks.

**5. LightGBM (LG)**

* **Reason for Selection:**
  + LightGBM delivers high classification scores in terms of precision and recall, coupled with strong F1 scores.
  + The ROC-AUC score often reaches close to 0.99, demonstrating its ability to accurately distinguish between classes.
  + Its efficient handling of large datasets makes it ideal for scalable solutions, with fast training and reduced memory usage, making it highly efficient.

These five algorithms have been selected based on their high classification report scores and strong ROC-AUC performance, making them suitable for your classification model.

**4. Feature\_Selection**

### ****Feature Selection and Hyper parameter Tuning****

### ****Introduction to Feature Selection****

Feature selection involves choosing the most relevant features from the dataset to improve the model's accuracy and efficiency. It eliminates irrelevant or redundant data, allowing the model to focus on the most informative features, which improves performance and reduces complexity.

#### **Selected Algorithms for Feature Selection**

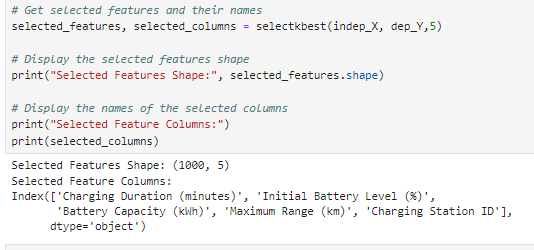
We have selected five powerful classification algorithms:

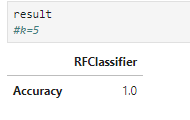
1. **Random Forest (RF)**
2. **Decision Tree (DT)**
3. **K-Nearest Neighbors (KNN)**
4. **XGBoost (XG)**
5. **LightGBM (LG)**

Each algorithm has its strengths and can provide insight into which features are the most important for predicting the target variable.

### 1. ****Random Forest (RF)****

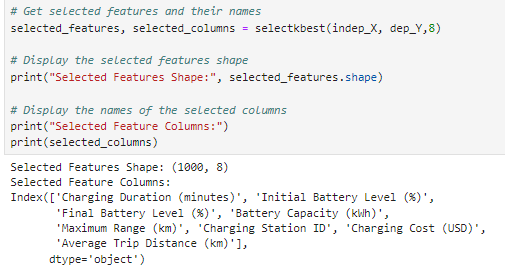
* Uses **feature importance** to select the most important columns.
* Automatically ranks features based on their contribution to the model.
* The higher the feature importance score, the more valuable it is for predictions.

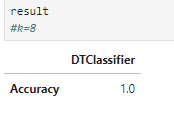




### 2. ****Decision Tree (DT)****

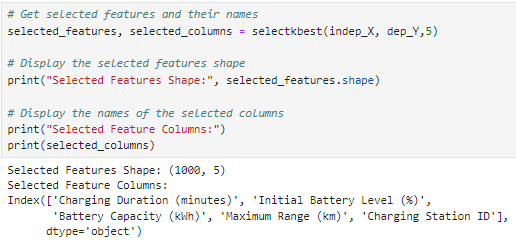
* Selects features based on **splitting criteria** (like Gini Index or Entropy).
* Looks for columns that best separate the data into different classes.
* The most used features in tree splits are the most important.

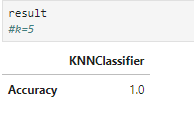




### 3. ****K-Nearest Neighbors (KNN)****

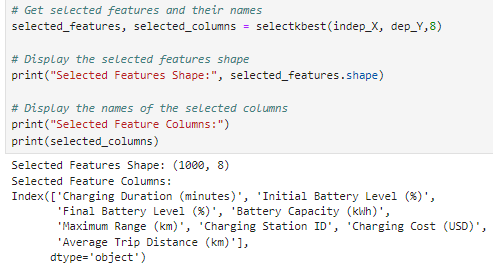
* Does not directly select features, but **feature scaling** (e.g., StandardScaler) helps.
* Use techniques like **Recursive Feature Elimination (RFE)** to select important features.
* Focus on features that improve classification accuracy.

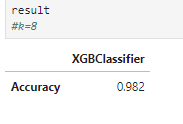




### 4. ****XGBoost (XG)****

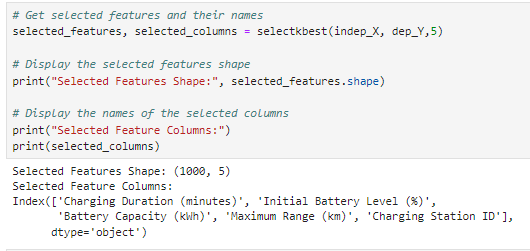
* Utilizes **gain and cover** to determine feature importance.
* Measures how much each feature improves the performance of splits.
* The higher the gain, the more important the feature.

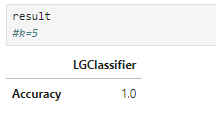




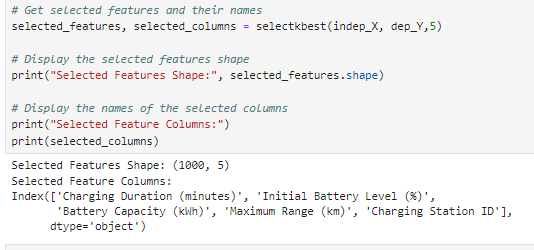
### 5. ****LightGBM (LG)****

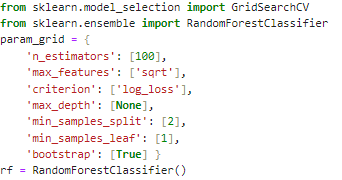
* Uses **feature importance based on split gain**.
* Fast feature selection due to LightGBM's leaf-wise growth.
* Focuses on features that lead to better splits.

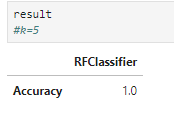




**Saving the Best Random Forest Model and Creating a Deployment File**

I used the SelectKBest feature selection method with K=5 to identify the top features, which led to improved accuracy. After training the Random Forest algorithm with the best hyperparameters found using GridSearchCV, I created the model using these selected features, saved it, and successfully deployed it."





### ****5. Data Visualization****

### ****5.1In Exploratory Data Analysis (Seaborn)****

### ****Target Variable Distribution****

* **Purpose**: Understand how the target variable (e.g., Is Regular Charger) is distributed within the dataset.
* **Method**: Use count plots to visualize the number of occurrences for each category of the target variable.
* **Benefits**: This helps identify imbalances in the dataset, which can affect model training.

**Categorical Variables Analysis**

* **Definition**: Categorical variables represent categories or groups (e.g., Type of Vehicle, Charging Pattern).
* **Visualization**:
  + Use count plots to display the distribution of each categorical variable against the target variable.
  + Insights gained can indicate preferences or trends within different categories.

#### **Numerical Variables Analysis**

* **Purpose**: Understand the distribution of numerical variables (e.g., Charging Duration, Energy Consumed).
* **Method**:
  + Histograms provide a visual summary of the frequency distribution of numerical data.
  + Box plots are used to highlight the median, quartiles, and potential outliers in the data.
* **Benefits**: Visualizing numerical data helps in identifying skewness, spread, and the presence of outliers.

#### **Correlation Analysis**

* **Definition**: Correlation measures the relationship between two numerical variables.
* **Method**:
  + A heatmap can visualize the correlation matrix, showing how variables are related.
  + Strong correlations can indicate which features may be influential in predicting the target variable.
* **Benefits**: Understanding correlations aids in feature selection and model optimization.

#### **Relationships between Variables**

* **Bivariate Analysis**:
  + Examines the relationship between two variables (categorical vs. numerical).
  + Visual tools include scatter plots to show how two numerical variables relate and how they group by a categorical variable.
* **Insights**: Identifying trends or clusters within the data can inform feature engineering and model selection.

#### **Customer Segmentation**

* **Purpose**: Identify high-value customers based on specific criteria (e.g., Energy Consumed).
* **Method**:
  + Segmentation can be achieved using quantiles to categorize customers into groups.
  + This analysis helps in targeted marketing strategies and service improvements.
* **Outcome**: Understanding customer segments enables better resource allocation and customer satisfaction.

#### **Sentiment Analysis**

* **Definition**: Analyzing how customers feel about a particular aspect of the dataset, such as Charging Cost.
* **Method**: Calculate the distribution of values to understand customer sentiment.
* **Insights**: This information can guide pricing strategies and service enhancements.

#### **Cross Tabulation and Heatmap for Categorical Interactions**

* **Purpose**: Explore interactions between two categorical variables (e.g., Type of Vehicle vs. Is Regular Charger).
* **Method**:
  + Cross-tabulation provides a matrix view of how categories intersect.
  + Heatmaps visualize this interaction, allowing for quick identification of patterns.
* **Benefits**: Understanding these interactions can inform product offerings and marketing strategies.

Data visualization plays a critical role in exploratory data analysis. By using various graphical representations, you can uncover hidden patterns, relationships, and insights within the dataset. This understanding is essential for making informed decisions in model selection and feature engineering.

### 5.2Power BI Dashboard for Electric Vehicle Dataset

#### **1.** **Introduction**

* Overview of the electric vehicle (EV) dataset.
* Importance of data visualization in understanding EV usage patterns.

#### **2. Dataset Attributes**

* **Quantitative Attributes:**
  + Charging Duration (minutes)
  + Energy Consumed (kWh)
  + Charging Frequency (per week)
  + Initial and Final Battery Levels (%)
  + Battery Health
  + Environmental factors: Temperature (°C), Humidity (%)
  + Vehicle Age (years)
  + Battery Capacity (kWh)
  + Maximum Range (km)
  + Charging Cost (USD)
  + Driver Usage Frequency (per day)
  + Average Trip Distance (km)
  + Time Spent Driving (minutes)
  + Weekend Usage (%)
* **Qualitative Attributes:**
  + Time of Day
  + Type of Vehicle
  + Charging Pattern
  + Weather Condition
  + Station Type
  + City
  + Region

#### 3. **Key Visualizations**

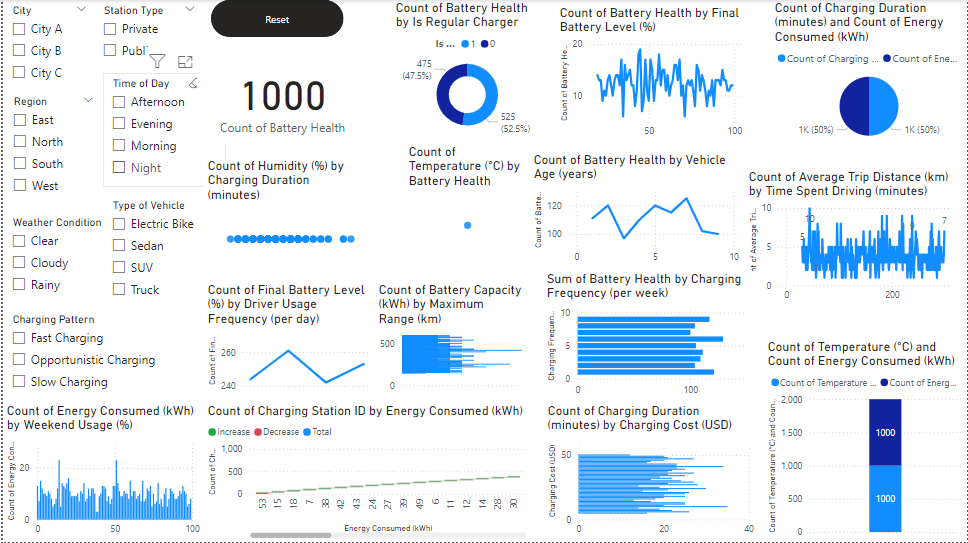
* **Charging Behavior Analysis:**
  + Bar charts showing average charging duration and energy consumption by vehicle type.
* **Battery Performance Insights:**
  + Line graphs depicting battery health trends over time.
  + Scatter plots correlating initial and final battery levels.
* **Usage Patterns:**
  + Heatmaps illustrating charging frequency by time of day and day of the week.
* **Cost Analysis:**
  + Pie charts breaking down charging costs by charging station type and region.

#### **4. Insights and Conclusions**

* Summary of key findings from the visualizations.
* Potential implications for EV manufacturers and charging infrastructure development.
* Recommendations for optimizing charging strategies based on data insights.

#### **5. Future Work**

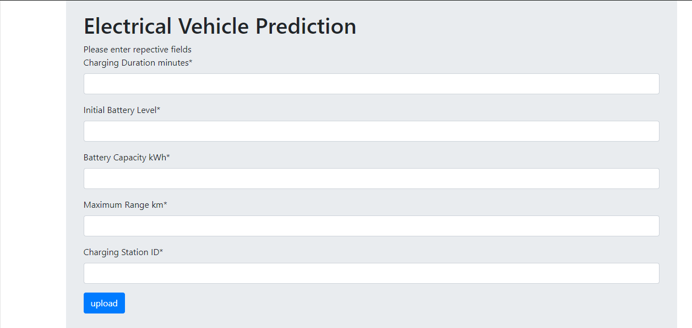
* Suggestions for further analysis or additional data collection.
* Ideas for enhancing the dashboard with predictive analytics or real-time data integration.



**6. Web development**

### Input Page

1. **Purpose**: Collect user inputs to make predictions about electric vehicle (EV) usage.
2. **Input Fields**:
   * **Charging Duration (minutes)**: Enter the total time spent charging the vehicle.
   * **Initial Battery Level (%)**: Specify the battery percentage before charging.
   * **Battery Capacity (kWh)**: Input the total capacity of the battery in kilowatt-hours.
   * **Maximum Range (km)**: Enter the maximum distance the vehicle can travel on a full charge.
   * **Charging Station ID**: Provide the identification number of the charging station used.



### Output Page

1. **Purpose**: Display the prediction results based on the input values.
2. **Display Section**:
   * **Prediction Result**: Show the model’s prediction regarding EV usage patterns based on the input data.
   * **User-Friendly Message**: Include a brief explanation of the prediction, helping users understand what the output means.

