**AI-Driven Optimal Placement of Electric Vehicles Charging Stations in Kenya**

**Business Understanding**

**Business overview**

Kenya is currently experiencing a significant transformation in its transportation and energy sectors. The adoption of electric vehicles (EVs) is steadily increasing, driven by several key factors. Rising fuel costs are encouraging individuals and businesses to seek alternative, more affordable transportation options, with electric vehicles offering long-term savings. According to Siemens Stiftung (2023), the Kenyan government has introduced various incentives to support the shift toward greener energy, including tax exemptions and rebates for EV purchases. This, coupled with a global push for sustainability, has positioned Kenya as a growing market for EVs, contributing to its broader environmental goals of reducing carbon emissions and improving air quality.

Despite the momentum towards EV adoption, the country's infrastructure development has not kept pace. One of the main challenges hindering the widespread use of EVs is the absence of a data-driven approach to charging station placement (Capital FM., 2024). Currently, the deployment of charging stations is largely arbitrary, with decisions often being made reactively or based on limited insights. As a result, charging infrastructure is confined to a few key areas, usually urban centers or major highways, leaving vast regions underserved.

This uneven distribution of charging stations leads to several problems that slow down EV adoption. First, the underutilization of existing charging stations occurs because they are not strategically located to meet the needs of drivers across the country. Second, many EV owners experience range anxiety the fear of running out of battery power without access to a nearby charging point because of the uncertainty around the availability of charging stations. To overcome these challenges, a more systematic, data-driven approach to planning and placing charging stations is crucial. Such an approach would ensure that the deployment of infrastructure is aligned with actual demand, optimizing the EV charging network, reducing range anxiety, and supporting the sustainable growth of the EV market in Kenya.

**Problem Statement**

The adoption of electric vehicles (EVs) in Kenya is gaining momentum, fueled by the rising cost of fossil fuels, government incentives, and growing environmental awareness. However, despite this progress, the lack of a well-planned and optimized EV charging infrastructure continues to be a significant barrier to achieving widespread EV adoption.

Currently, the deployment of EV charging stations in Kenya is often done without the use of data-driven insights or strategic planning. As a result, the existing charging stations are placed in locations that may not align with the actual needs of EV owners, leading to several challenges. Many charging stations are located in areas with limited traffic or low demand, resulting in underutilization. This not only wastes valuable resources but also leads to inefficient infrastructure investment.

**Project Objectives**

1. **Primary Objective:**

To develop an AI-powered platform that uses machine learning, geospatial data, and predictive analytics to identify optimal locations for electric vehicle (EV) charging stations.

1. **Secondary Objectives:**
2. To integrate machine learning algorithms for analyzing EV usage patterns and infrastructure needs.
3. To incorporate geospatial data for determining the most efficient and accessible sites for charging stations.
4. To ensure the platform is scalable and adaptable to various regions or cities.
5. To improve the overall efficiency of EV charging infrastructure deployment and usage.

**Approach Methodology**

The approach follows a structured process that includes data preprocessing, feature extraction, model selection, and evaluation. Through an iterative process of testing and refining various techniques, the aim is to accurately identify geographic clusters of charging stations by modeling data from a U.S. city with the highest concentration of charging stations. This methodology will generate valuable insights that can be applied to map optimal charging station locations in Nairobi, ensuring a more strategic deployment of infrastructure in the city.

To address the project objectives, the methodology follows a structured process:

1. Data Cleaning: Start by handling missing values, removing duplicates, standardizing data types, and managing outliers. This ensures that the dataset is accurate and ready for analysis.
2. Feature Engineering: Create new features that can improve the performance of clustering models. This step is important for enhancing the model's ability to identify meaningful patterns within the data.
3. Data Scaling and Normalization: Since clustering models like K-means are sensitive to the scale of the data, it is essential to standardize the features. Techniques like Min-Max scaling ensure that all features contribute equally to the distance calculations. Also check for correlations between features to determine if dimensionality reduction is necessary.
4. Choosing the Number of Clusters: The elbow method is used to determine the optimal number of clusters (K). This method plots the within-cluster sum of squares (WCCS) against the number of clusters and identifies the "elbow" where the rate of improvement slows down. The silhouette score is then calculated to validate the chosen K.
5. Modeling: Various clustering algorithms, including K-means, DBSCAN, and Gaussian Mixture Models (GMM), are applied. Each model is initialized with the optimal K value from the Elbow plot and then fitted to the data.
6. Model Evaluation and Results Interpretation: After fitting the model, evaluate the cluster centers, assign labels to the data points, and visualize the clusters. If necessary, refine the model by adjusting the number of clusters (K) and reassessing the data point assignments.
7. Application to Nairobi: Once the model is trained and refined, it is applied to a new dataset from Nairobi. The trained clustering model, which has been learned from another city, is used to map potential EV charging station locations in Nairobi, allowing insights from one area to inform decisions in another.

**Metrics of Success**.

To assess the effectiveness of the cluster models, the following evaluation metrics will be used:

**Prediction Accuracy for High-Demand Locations:**

The model should correctly predict at least 90% of high-demand locations, minimizing false positives (incorrectly predicting high demand) and false negatives (failing to predict high demand) when identifying optimal sites for charging stations.

**R² Score:**

The model should achieve an R² score of at least 0.85, indicating a strong correlation between the predicted charging demand and observed actual demand. This ensures the model's reliability in estimating demand.

**Proximity to Power Grid:**

At least 80% of the suggested charging station locations should be within 500 meters of a power grid connection, ensuring that the proposed sites are practical for deployment and accessible to the existing energy infrastructure.

**Adaptability to New Urban Areas:**

The model should maintain an accuracy above 85% when tested on new urban areas, demonstrating its ability to adapt as Nairobi's EV market expands and ensuring that it is not over-fitted to a specific dataset but can generalize well to new locations.

**Data Understanding**

**Obtaining the data**

The data for this project was sourced from the U.S. Department of Transportation, Bureau of Transportation Statistics, specifically focusing on Electric Vehicle Public Charging Stations in the United States as of January 2020. The dataset provides comprehensive information about the locations and characteristics of EV charging stations across the U.S., serving as the basis for clustering and predictive modeling.

You can access the dataset via the following link:

[Electric Vehicle Public Charging Stations Data](https://data-usdot.opendata.arcgis.com/datasets/alternative-fueling-stations/explore)

We read the dataset in pandas as shown below:

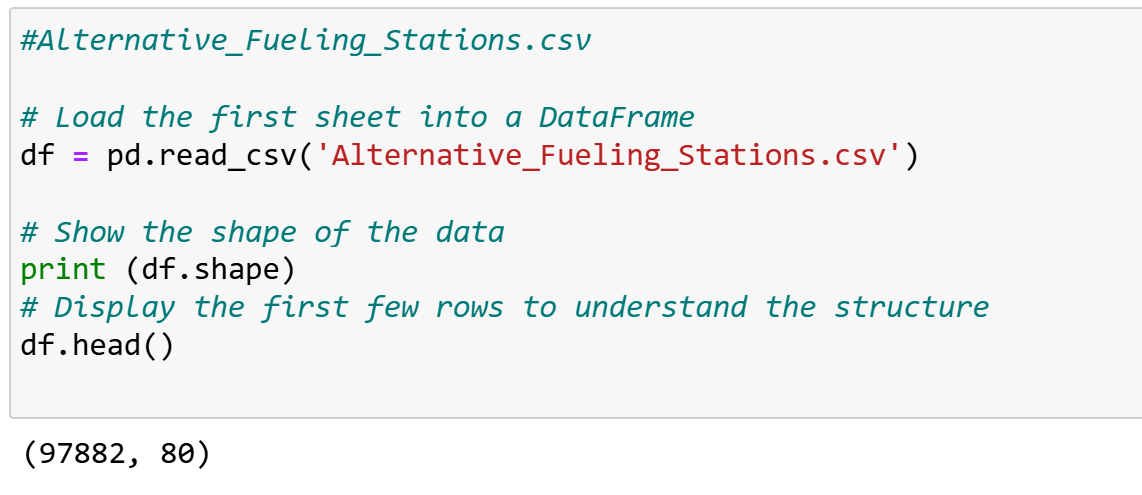


Fig.1: Loading the data.

This dataset will identify patterns and apply clustering models to predict optimal charging station locations, which can later be adapted for use in Nairobi or other cities with similar infrastructure needs.

**Data Preparation**

1. Filtering the dataset tailored for our analysis
2. Data cleaning
3. Data preprocessing: feature engineering transformation and preprocessing.

The dataset provided information about alternative fueling stations, including Electric Vehicle (EV) charging stations. It consisted of 97,882 rows and 80 columns. Key columns relevant to EV charging stations include:

* Access code: Indicates whether the station is public or private.
* Access\_days\_time: Specifies availability, such as 24-hour access.
* Fuel\_type\_code: Identifies the fuel type (e.g., "EV" for Electric Vehicles).
* Groups\_with\_access\_code: Details the access permissions.
* Ev\_pricing\_fr: Describes the pricing model for EV charging.
* Ev\_network\_ids\_station: Identifies the EV charging network.
* X, Y: Longitude and latitude coordinates for mapping the station location.
* Federal\_agency\_name: Indicates whether the station is government-owned or privately owned.

We began by filtering the dataset to include only rows with the 'ELEC' fuel\_type\_code, focusing on Electric Vehicle (EV) charging stations.

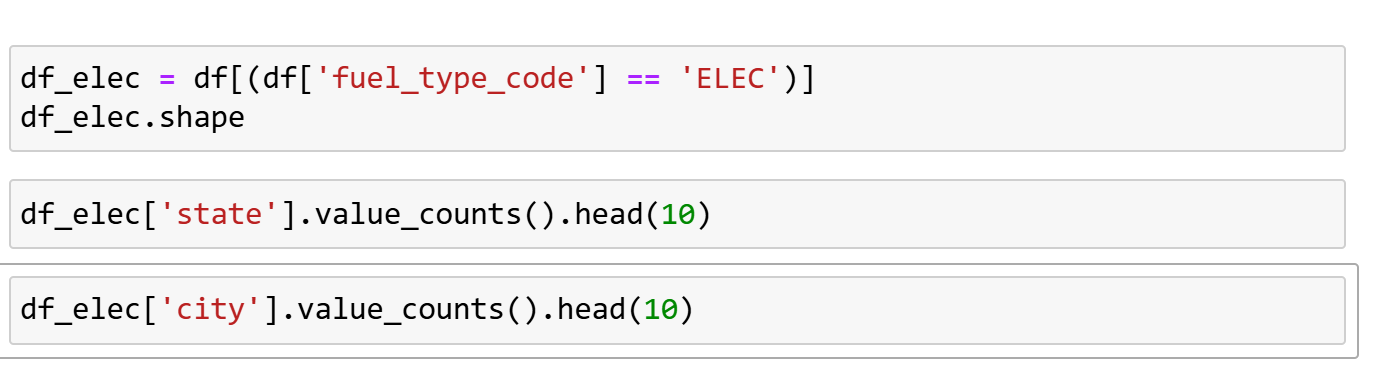


Fig.2: Filtering the Data frame.

Next, we identified the features most relevant to our analysis and filtered the data frame to retain only those columns. To ensure the precision of our analysis, we further filtered the dataset to include only the 'ELEC' fuel type and then narrowed it down to the state with the highest count of EV stations in the filtered data. California emerged as the state with the highest number of EV charging stations, and Los Angeles had the highest concentration of EV stations within California. Therefore, we filtered the data to focus on Los Angeles, ensuring we examined the most pertinent subset of the dataset for our analysis

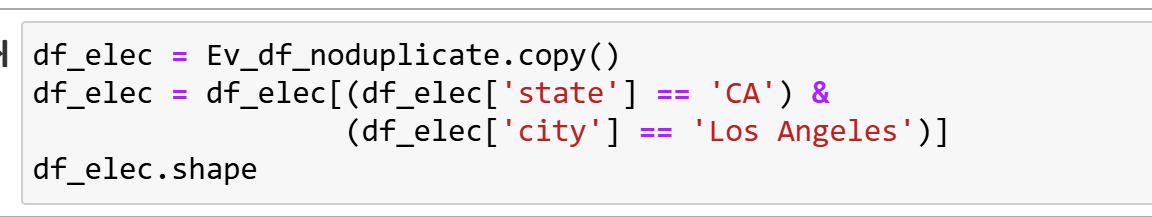


Fig.3: Filtering the Data frame.

The filtered dataset now contained only Electric Vehicle (EV) charging stations located in California, specifically in Los Angeles. While the dataset was primarily focused on EV stations, it still included columns related to other fuel types such as LPG and ethanol. To refine the dataset further, we removed these irrelevant columns, retaining only those that pertain to EV charging stations. To ensure we had the correct and relevant columns, we created a list of these columns and used the check info function from the my functions directory to examine the dataset's structure and confirm the data's accuracy and relevance. This allowed us to verify that only the necessary columns were retained for further analysis.



Fig.4: Filtering the relevant columns.

The dataset now contained 1,941 records and 23 columns, providing detailed information about electric vehicle (EV) charging stations, including station names, locations (latitude, longitude, city, state), status, access details, EV-specific features (such as connector types and charging levels), and operational attributes like pricing and workplace charging options. Although the dataset focuses on EV stations, it includes missing values in several columns, particularly in areas related to renewable sources, pricing, and specific charging capabilities. The data is predominantly of object and float64 types, and there are no duplicated rows, ensuring that each record is unique.

The filtered dataset now focuses on electric vehicle (EV) charging stations in California and Los Angeles. This refinement ensures that our analysis is more targeted and relevant, eliminating any irrelevant data that could distort the results. In this section, the focus shifts to data preparation, a critical step in the analysis process. This involves cleaning and transforming the dataset to ensure it is suitable for exploratory data analysis (EDA) and modeling. Data cleaning will address missing values, duplicates, and irrelevant columns, maintaining the integrity of the analysis. Additionally, transformations will be applied to standardize data formats, handle outliers, and engineer new features, all of which are essential for improving the performance of machine learning models.

**Data Cleaning**

1. Deal with the missing values and filtering for important columns.
2. Deal with Duplicates
3. Further cleaning and transformation; creating new columns, encoding features etc.

**Dealing with Missing Values**

We identified missing entries across the dataset, and we will use the .isna() method. This method calculates the percentage of missing values for each column. The results will then be sorted in descending order to highlight the columns with the most missing data. The final output will be displayed as a data frame, showing each column and the corresponding percentage of missing entries.

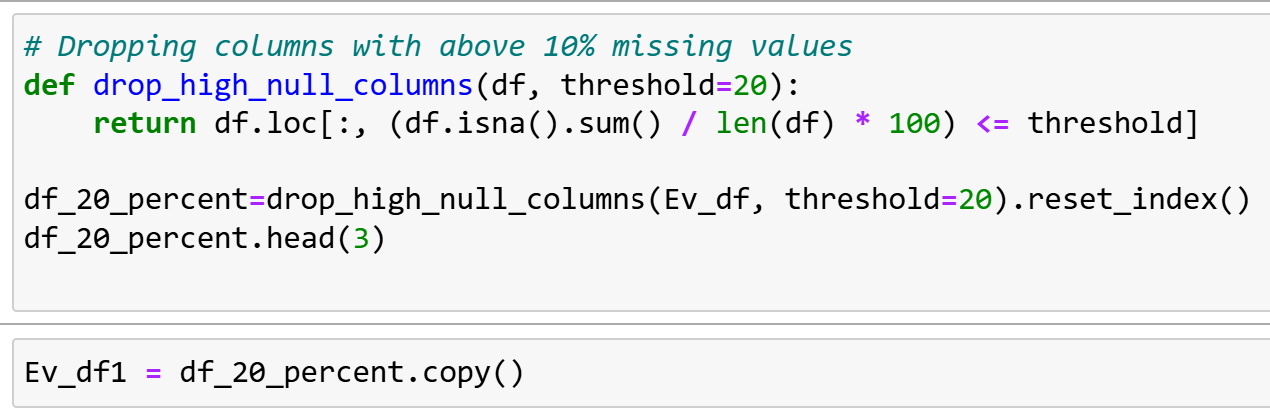


Fig.5: Dropping columns with more than 10% missing.

The filtered EV Charging Station Data set now included several key categories of information.

station\_name: The name of the EV charging station typically reflects the location or brand.

Latitude: The north-south geographic coordinate of the station (in degrees).

Longitude: The east-west geographic coordinate of the station (in degrees).

City: The city where the EV charging station is located.

State: The state or region where the station is located (if applicable).

Country: The country where the EV charging station is located.

street\_address: The full street address, including building number and street name.

status\_code: Indicates the station's current status (e.g., "Available," "Out of Service").

access\_code: A code or identifier required to access the charging station (e.g., PIN, card number).

ev\_connector\_types: The types of connectors available (e.g., Type 1, Type 2, CCS, CHAdeMO).

ev\_dc\_fast\_num: The number of DC fast chargers that provide rapid charging.

ev\_level1\_evse\_num: The number of Level 1 EVSE units offering slow charging (typically residential).

ev\_level2\_evse\_num: The number of Level 2 EVSE units offering faster charging commonly found in public networks.

Network ev\_network: The network's name operating the station (e.g., Tesla Supercharger, Charge Point).

**Imputing ev\_level2\_evse\_num with zeroes**

In this step, we addressed the missing values in the ev\_level2\_evse\_num column, which indicates the number of Level 2 electric vehicle (EV) chargers at each station. The absence of data here is not due to data collection errors but reflects that certain stations do not have Level 2 chargers. Instead of imputing these missing values with a mean or median, which could artificially inflate the data and give a false impression of charger availability, we imputed these missing values with zeroes. This approach ensures that the data remains truthful and does not suggest the presence of infrastructure where none exists, maintaining the integrity of the dataset without introducing erroneous assumptions.

**Dropping the access\_days\_time missing rows**

The access\_days\_time column denotes the hours a charging station is operational during the day. Missing values in this column represent a lack of information about the station's operating schedule. Instead of filling in the missing entries with zeroes or a median value, implying that the station operates for a full or default period, potentially distorting the data, we dropped rows with missing values. This decision ensures the accuracy of the dataset, as it avoids making unwarranted assumptions about the operational hours of the stations. By excluding these rows, we preserve the integrity of the dataset and ensure that the analysis reflects the true, available data without introducing biases.

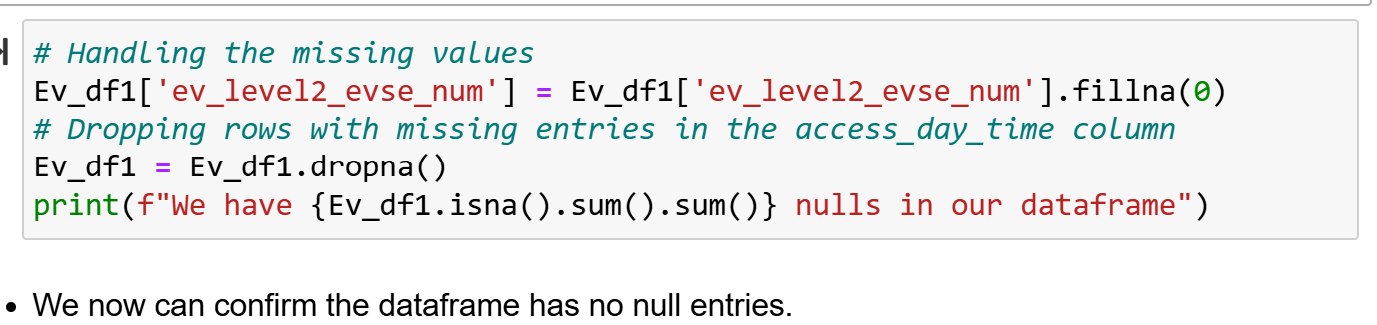


Fig.6: Imputing missing entries.

**Further Data Cleaning and Feature Engineering**

In this section, we split the dataset into categorical and numerical columns using the .select\_dtypes () method. This allowed us to create two separate data frames for each data type. This step's main focus was cleaning the categorical columns to ensure they were ready for analysis or modeling.

Cleaning the categorical columns involved addressing several key tasks, including identifying and resolving any inconsistencies within the data, handling missing values appropriately, and converting the variables into formats compatible with further analysis or modeling techniques. These steps are crucial because well-prepared categorical data ensures that subsequent models can effectively interpret and utilize the variables without errors or misinterpretations.

The emphasis on categorical data in this step is to ensure that these variables are properly processed and structured for modeling purposes. Properly prepared categorical data enhances the accuracy of any predictive models and ensures that any insights derived from the analysis are based on clean and reliable inputs.

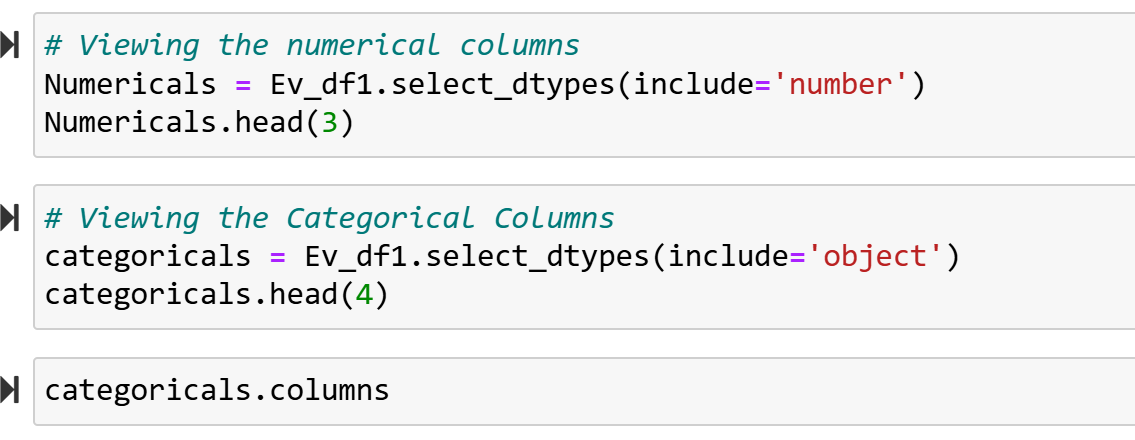


Fig.7: Splitting Data set into numerical and categorical.

Cleaning Categorical Columns and Preprocessing

The station\_name column

To begin the cleaning process for the station\_name column, we created a function named clean\_and\_split\_station\_name. This function performs a series of steps to ensure the column is properly processed for further analysis and modeling:

1. Stripping Leading and Trailing Spaces: The function removes any unnecessary spaces at the beginning or end of the station names to maintain consistency and prevent issues during analysis.
2. Splitting the String: The function splits the station\_name into two parts:
   * station\_region: This is the portion of the string before the hyphen (-).
   * station\_subregion: This is the portion of the string after the hyphen (-). If the hyphen is missing, both new columns (station\_region and station\_subregion) will be filled with the same string, representing the entire station name as the region and subregion.
3. Handling Invalid Entries: If the entry is not a valid string (e.g., a numeric value or missing data), the function will return None for both the station\_region and station\_subregion columns to ensure that these invalid values do not interfere with further processing.
4. Frequency Encoding: To prepare these newly created columns for modeling, the function applies frequency encoding to both the station region and station\_subregion. This encoding technique assigns each category a numeric value based on how frequently that category appears in the dataset, providing a way for the model to interpret the categorical data numerically.

The function was then called from the Python file containing custom functions in the project directory to apply these transformations to the station\_name column. This preprocessing ensures the data is in a suitable format for further modeling and analysis.

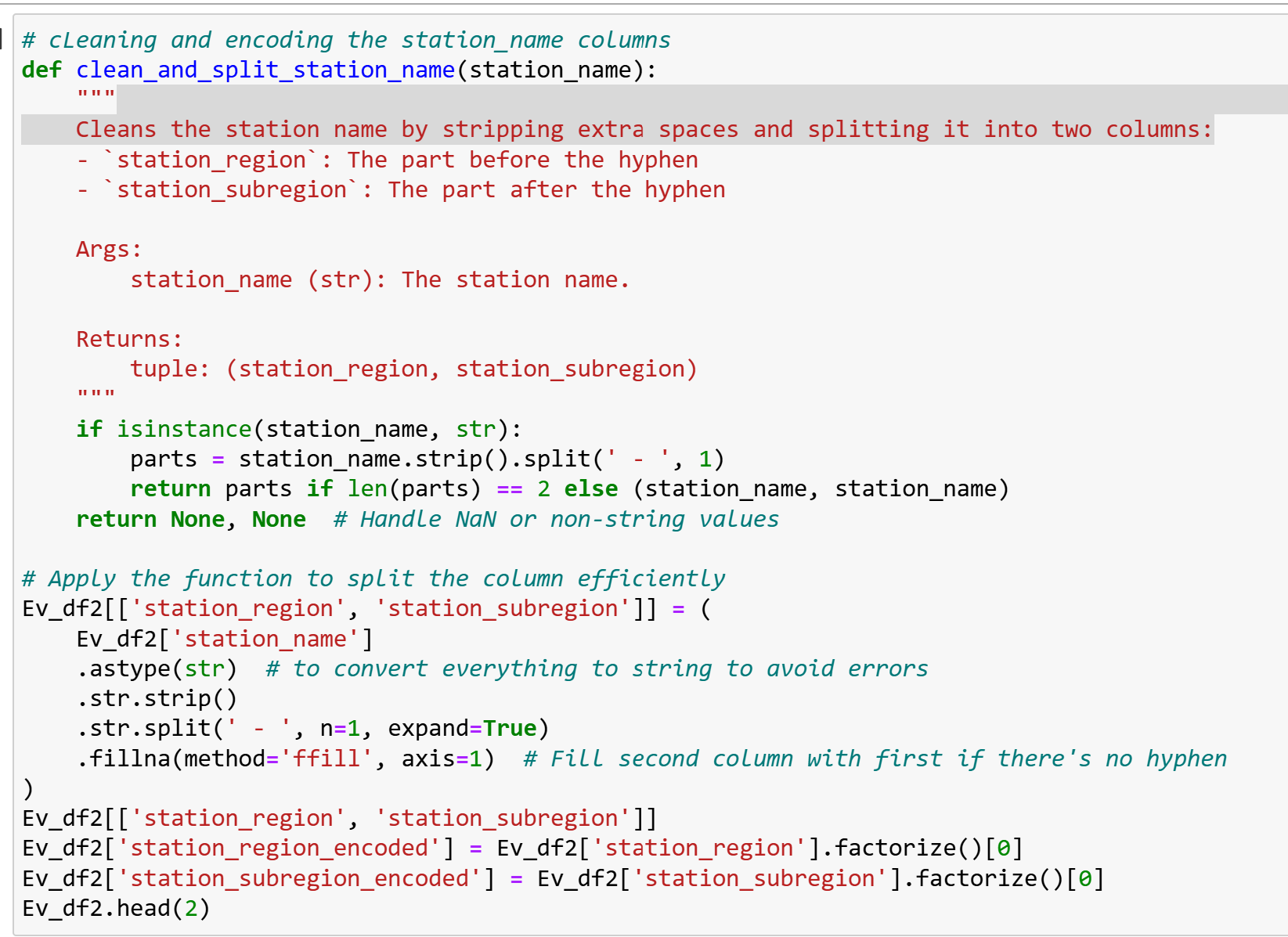


Fig.8: Function to clean the station\_name column.

**The access\_days\_time Column**

To clean and process the access\_days\_time column, we developed a function called calculate\_hours. This function calculated the number of operational hours for each entry based on the start and end times extracted from the column. The function ensured that the end time was always greater than the start time, addressing any inconsistencies in the data.

The calculate\_hours function included an internal function, extract\_time\_and\_hours, which is responsible for extracting time details from the parking lot access descriptions. This internal function handled various time formats, including:

1. "24 hours" Descriptions: For descriptions that indicated "24 hours," the function automatically assigned a value of 24 hours, representing continuous access throughout the day.
2. Day-Specific Time Ranges: The function processed entries with specific days and time ranges (e.g., "Mon 9:00 am - 5:00 pm"). It extracted the start and end times, converted them into a 24-hour format, and calculated the total number of hours between the two times.

The function also ensured that if the end time was earlier than the start time, indicating a potential error in the data, it handled the situation by adjusting the times or flagging the entry for review.

After processing the time descriptions, we used the process\_dataframe function to extract and calculate the time and hours from the access\_days\_time column. This resulted in adding a new column, access\_hours, to the dataset, which stored the calculated number of hours each parking lot was accessible.

This process standardized the access\_days\_time column, ensuring the data was consistent and reliable for analysis. It allowed us to calculate the total access hours for each parking lot, which simplified further analysis and modeling efforts.

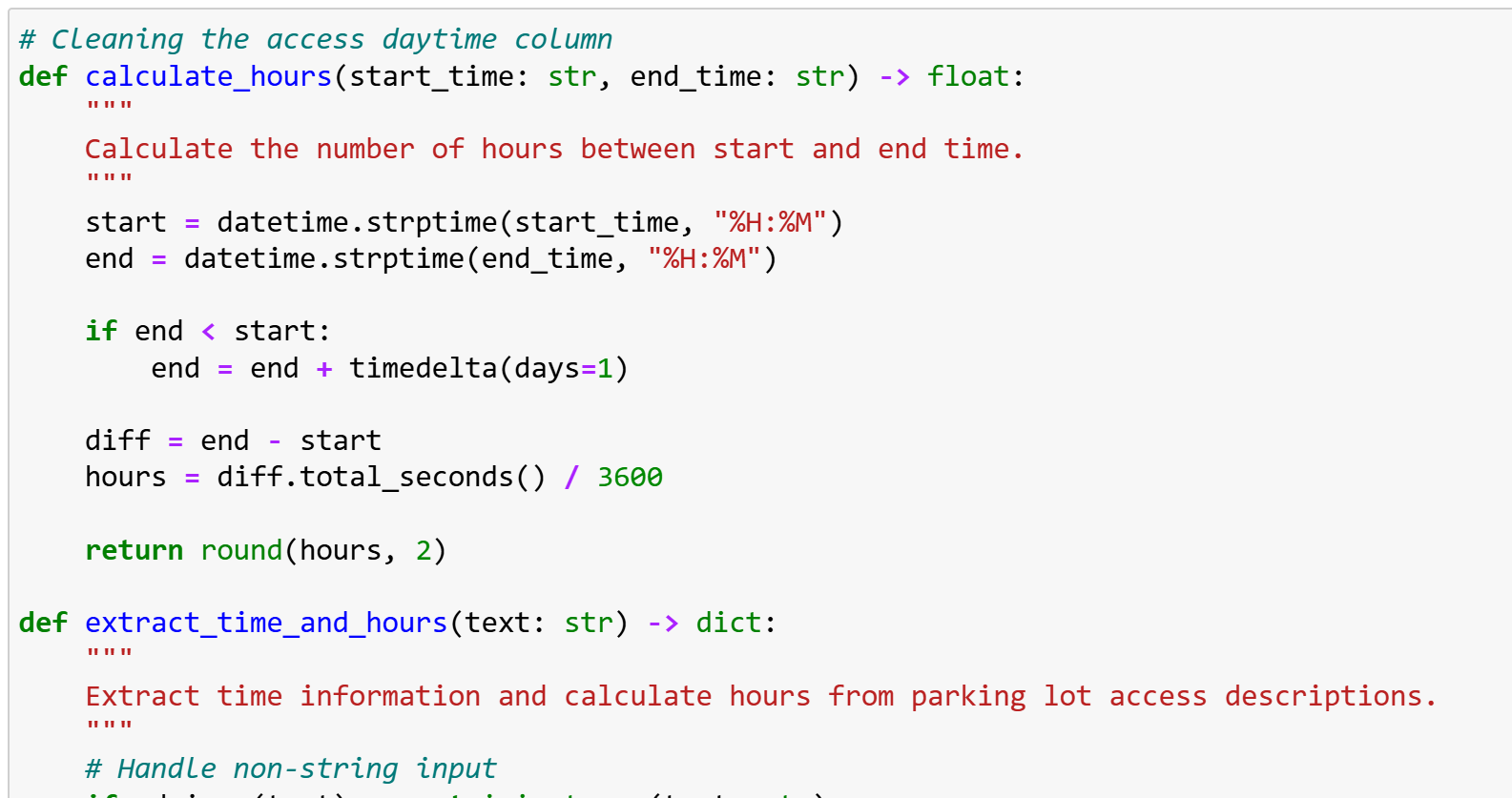


Fig.8: Function to access\_day\_time column.

**The ev\_connector\_types Column**

To clean and preprocess the ev\_connector\_types column, we first addressed the entries consisting of lists of charger types. The cleaning process involved several key steps:

1. Removing Unwanted Quotes: We replaced any double quotes ("") in the entries with single quotes ("), ensuring uniformity in the data.
2. Stripping Leading and Trailing Whitespaces: We removed any unnecessary leading or trailing whitespace from the column values to ensure formatting consistency.

Next, we created a function that converted any string representations of lists into actual Python lists using the ast.literal\_eval function. This function was applied only to string values, as other data types (e.g., already formatted lists) did not require conversion.

After applying this function to the Ev\_df2 DataFrame, the ev\_connector\_types column was properly formatted as Python lists.

We then performed one-hot encoding for the cleaned column to prepare the data for modeling. We created a function named encode\_connector\_types that carried out the following steps:

1. Checking for Lists: The function first checked that each entry in the ev\_connector\_types column was a list.
2. Handling Missing or Non-List Entries: It accounted for missing values or entries that were not lists, ensuring that such cases were appropriately managed and did not cause errors during encoding.
3. Encoding with MultiLabelBinarizer: The function used the MultiLabelBinarizer from the sklearn.preprocessing module to perform the one-hot encoding. This transformed the lists of charger types into multiple binary columns, each representing a specific EV connector type.
4. Concatenating the Encoded Columns: The newly created binary columns were then concatenated with the original data Frame.
5. Dropping the Original Column: After encoding, the original ev\_connector\_types column was dropped from the data Frame, as it was no longer necessary.

Finally, we applied the encode\_connector\_types function to the Ev\_df2 data Frame, and the function returned the resulting data frame with the newly encoded columns.

This process ensured that the ev\_connector\_types column was properly cleaned, encoded, and integrated into the data frame for further analysis and modeling.

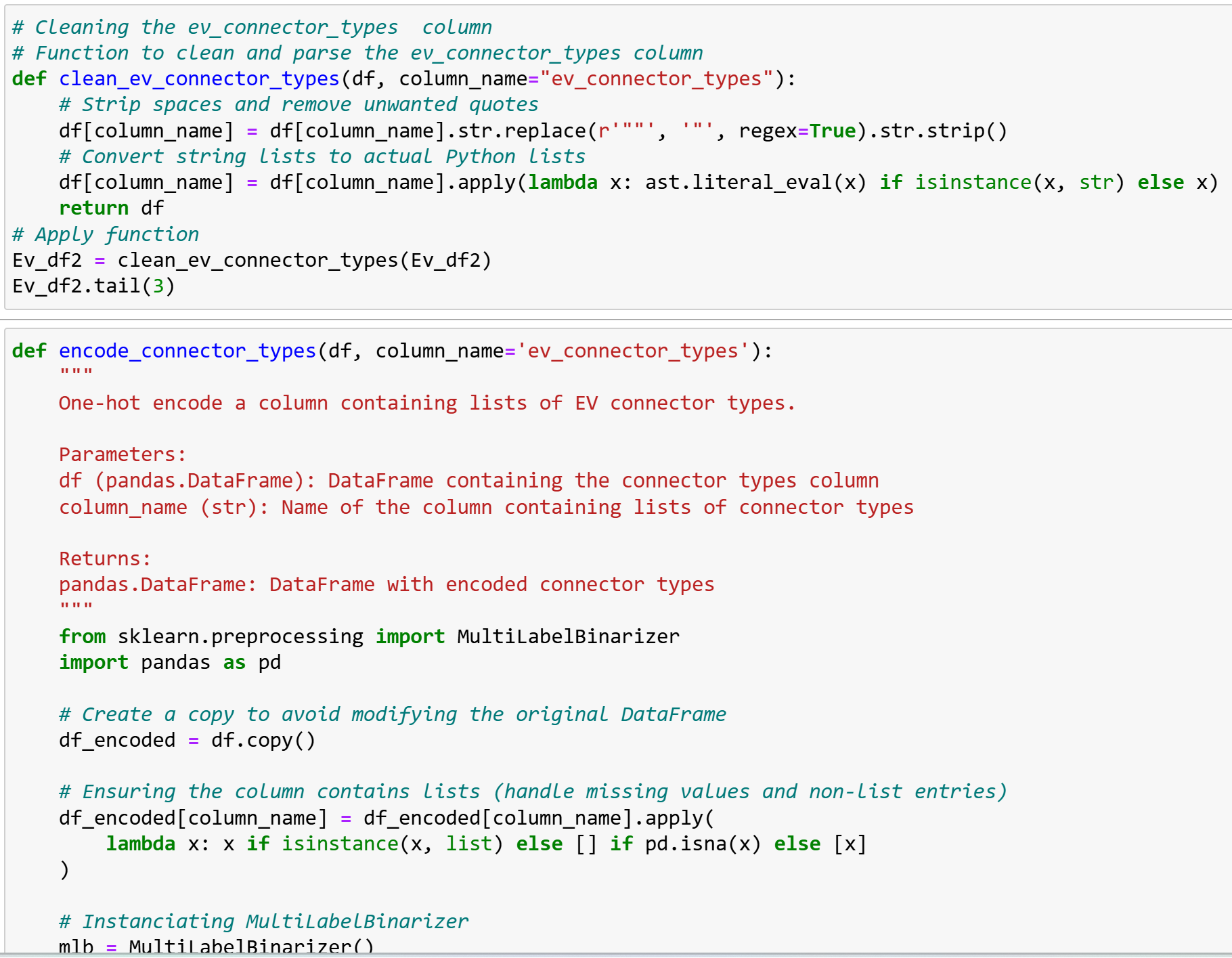


Fig.9: Function to clean the ev\_connector\_types column.

**The status\_code Column**

The status\_code column contains three entries: E, T, and P. One-hot encoding was the most appropriate method for handling this column. This technique converts the unique values in the status\_code column into separate binary columns, making it suitable for modeling.

We used the PD.get\_dummies function from the pandas library to perform this encoding. This function automatically transforms the unique values in the status\_code column into individual binary columns. Each new column represents one of the unique status codes, where:

* A 1 indicates the presence of that specific status code for the respective row.
* A 0 indicates its absence.

As a result, the original status\_code column was replaced with multiple binary columns, each corresponding to a different status code (E, T, and P). This transformation ensured that the status\_code column was properly formatted for machine learning models. Each status code is now represented as a distinct feature with binary values.

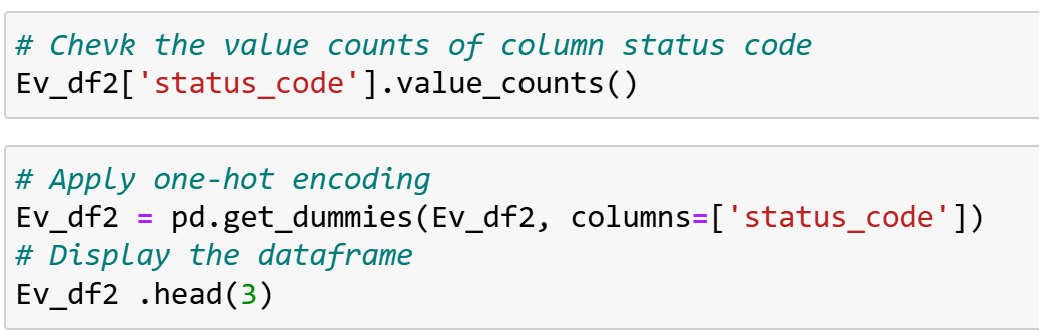


Fig.10: Encoding the status\_code.

**The access\_code Column**

To clean the access\_code column, which contains binary entries ("private" and "public"), we followed these steps:

1. Convert to Lowercase: We first converted all the values in the access\_code column to lowercase to ensure consistency and handle any discrepancies in capitalization.
2. Remove Leading and Trailing Whitespace: Any unwanted leading or trailing whitespace was removed from the values to ensure that the data did not suffer from unnecessary spaces.
3. Map to Binary Numeric Values: The cleaned values were then mapped to binary numeric values:
   * "private" was mapped to 0
   * "public" was mapped to 1
4. Store in New Column: The cleaned, mapped values were stored in a new column called access\_code\_cleaned.

This process ensured that the access\_code column was properly cleaned and converted into a numerical format more suitable for analysis and modeling.



Fig.11: Encoding the access\_code.

**The ev\_network Column**

To clean and process the ev\_network column, which represents the type of chargers or the manufacturer, we followed these steps:

1. Stripping Leading and Trailing Spaces: We removed any leading or trailing spaces from the entries in the ev\_network column to ensure that no unnecessary whitespace would interfere with further analysis.
2. Applying Frequency Encoding: We then applied frequency encoding to the column. In this step, each unique value in the ev\_network column was mapped to its corresponding frequency, representing the number of times that value appeared in the column.
3. Storing in a New Column: The calculated frequencies were stored in a new column called ev\_network\_encoded.

This approach ensured that the ev\_network column was properly cleaned and encoded, with each unique charger type or manufacturer now represented by its occurrence frequency. The new column, ev\_network\_encoded, was now suitable for analysis or modeling.

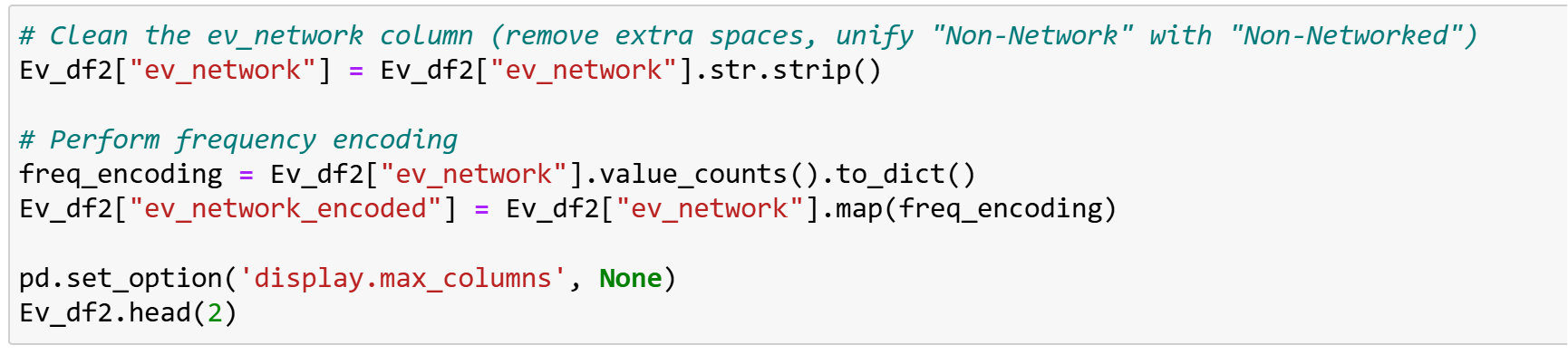


Fig.12: Frequency encoding the ev\_network.

**Exploratory Data Analysis (EDA)**

**Univariate Analysis**

In this section, we conducted a univariate analysis to examine each variable (or feature) individually, aiming to understand its distribution, spread, and central tendency. The goal was to analyze one column at a time, focusing on key statistical and visual measures to assess the nature of the data.

**For Numerical Columns:**

We started by evaluating the descriptive statistics for each numerical column, which included:

* Mean: The average value of the data, providing a central tendency measure.
* Median: The middle value is less sensitive to outliers than the mean.
* Mode: The most frequently occurring value, highlighting any repeated data points.
* Standard Deviation: A measure of the spread or dispersion of the data, indicating how much the values deviate from the mean.
* Percentiles: Specific values representing the distribution of the data, such as the 25th, 50th, and 75th percentiles, provide insight into the spread and distribution.

**Visualizations for Numerical Data:**

To better understand the numerical columns visually, we used the following plots:

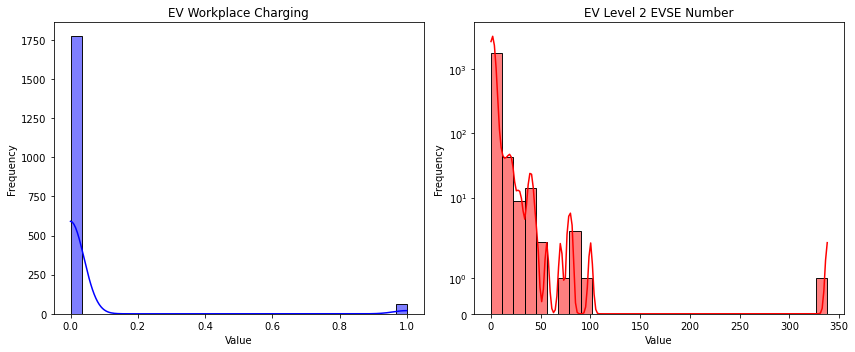
* Histograms: These were used to assess the distribution of the numerical data, showing how the data points are spread across different ranges.
* Box Plots helped us detect outliers. They showed the distribution of data based on quartiles and highlighted any extreme values that might require further examination.
* Density Plots: These visualized the probability distribution of the numerical data, providing a smooth representation of how data points are likely to be distributed.

**For Categorical Columns:**

For categorical variables, the analysis focused on summarizing the frequency of different categories within each column:

* Frequency Counts show how often each category appears in the dataset, giving an idea of the distribution of categories. Bar Charts: We used bar charts to represent the frequency distributions of categorical data, visually displaying the count of each category and helping to highlight any imbalances or trends in the data.

This univariate analysis helped us better understand the individual features, their distributions, and potential issues such as outliers, skewed distributions, or imbalances in categorical data. This information informed further steps in our data processing and modeling.



##### Fig.12: *Univariate Analysis for Numerical Features*.

Both ev\_level2\_evse\_num and ev\_workplace\_charging show positive skewness, with a longer right tail in their distributions.

1. ev\_level2\_evse\_num (Level 2 Chargers per Station): Most stations have only a few chargers, but a few stations have many, creating the right tail. This indicates that most stations are less equipped, with outliers representing larger or more advanced stations.
2. ev\_workplace\_charging (Workplace Charging Availability): The positive skew suggests that most workplaces do not offer charging facilities, with only a small number providing them.

Both columns indicate a concentration of lower values, with a few higher values stretching the distribution, highlighting infrastructure imbalances.

**Univariate Analysis for Categorical Features.**

**References**

Siemens Stiftung. (2023). Unlocking the growth potential of Kenya's e-mobility sector. Siemens Stiftung. <https://www.siemens-stiftung.org/wp-content/uploads/2030/09/studie-unlocking-the-growth-potential-of-Kenya-E-Mobility-Sector.pdf-1.pdf>

Capital FM. (2024, September). Lack of infrastructure hinders EV adoption in Kenya. Capital FM. <https://www.capitalfm.co.ke/business/2024/09/lack-of-infrastructure-hinders-ev-adoption-in-kenya/>