Electric Vehicle Data Analysis Project

By

G. Bani Vishwas

Video Link-
https://drive.google.com/file/d/1ZtYZSpZz6gwwxsSy6o9bWkNCyq1sypYA/view?usp=drivesdk

Introduction

Electric vehicles (EVs) represent a major innovation in the automotive industry. It is a sustainable alternative to traditional internal combustion engine (ICE) vehicles. Unlike cars that rely on gasoline or diesel, EVs are powered entirely by electricity, stored in high-capacity batteries. As concerns about climate change, air pollution, and fossil fuel dependency continue to grow, EVs have emerged as a key solution to achieving cleaner, more energy-efficient transportation.

Project Overview

This project focuses on analyzing a dataset of Electric Vehicles(EV) available in the market. The dataset includes various attributes such as manufacturer, car name, model, range, battery capacity, energy consumption, engine power, pricing, and more. Through statistical analysis and python libraries, this project aims to detect trendes, anomalies, and provide recommendations that can support EV adoption and improve customer decision-making.

Problem Statement

The key challenges addressed in this project include:

- Filtering electric vehicles based on a budget of 350000 PLN and of range 400, and average battery capacity of each manufacturer
- Detecting EVs with unusually high or low energy consumption.
- Analyzing the relationship between battery capacity and range.
- Building a recommendation system to help users in selecting the top EVs that meet their preferences.
- Testing whether there is a significant difference in the average Engine power [KM] of vehicles manufactured by two leading manufacturers (Tesla and Audi) using Two sample T-test.

Task 1: A customer has a budget of 350,000 PLN and wants an EV with a minimum range of 400 km.

a) Your task is to filter out EVs that meet these criteria

Steps:

1) Import the pandas library as p
2) Read the Excel file using p.read_excel() and load it into a variable called data
3) Filter the data to select only rows where:
    - "Minimal price (gross) [PLN]" is less than or equal to 350000
    - "Range (WLTP) [km]" is greater than or equal to 400
    - Save the result as filter_data

Code:

```
 import pandas as p
data = p.read_excel(r"C:\Users\91939\Downloads\FEV-data-Excel.xlsx")
filter_data = data[(data["Minimal price (gross) [PLN]"]<=350000) & (data["Range(WLTP) [km]"]>=400)]
```

b) Group them by the manufacturer (Make)

Steps:

1) Group the filtered data by "Make" using groupby()

Code:

```
grouped_data = filter_data.groupby("Make")
```

c) Calculate the average battery capacity for each manufacturer.

Steps:

1. Calculate the average battery capacity ("Battery capacity [kWh]") for each make using .mean()

Code:

```
avg_cap_make = grouped_data["Battery capacity [kWh]"].mean()
```

Task 1 code:

```
import pandas as p

data = p.read_excel(r"C:\Users\91939\Downloads\FEV-data-Excel.xlsx")

filter_data = data[(data["Minimal price (gross) [PLN]"]<=350000) & (data["Range(WLTP) [km]"]>=400)]

grouped_data = filter_data.groupby("Make")

avg_cap_make = grouped_data["Battery capacity [kWh]"].mean()

 print("Filtered EVs by Make and their Average Battery Capacities:")

print(avg_cap_make)
```

Result:

```
Filtered EVs by Make and their Average Battery Capacities:

 Make

 Audi      95.000000

BMW    80.000000

 Hyundai 64.000000

 Kia 64.000000

Mercedes-Benz 80.000000

 Tesla 68.000000

 Volkswagen 70.666667

 Name: Battery capacity [kWh], dtype: float64
```

Findings:

- There are about 7 EV within the given specifications

Task 2: You suspect some EVs have unusually high or low energy consumption. Find the outliers in the mean - Energy consumption [kWh/100 km] column.

Steps:

1. Import the pandas library as p
2. Read the Excel file using p.read_excel() with engine="openpyxl" into the variable data
3. Select the energy consumption column
   - energy_column = 'mean - Energy consumption [kWh/100 km]'
   - energy_data = data[energy_column]
4. Calculate First quartile Q1 = energy_data.quantile(0.25) and Third quartile Q3=energy_data.quantile(0.75)
5. Interquartile range IQR = Q3 - Q1
6. Compute outlier bounds:
   - Lower bound: lwr_bound = Q1 - 1.5 * IQR
   - Upper bound: upr_bound = Q3 + 1.5 * IQR
7. Filter outliers from the original dataset:
   - Select rows where energy consumption < lower bound or > upper bound
   - Store result in outliers
8. Print message: "Outliers in Energy Consumption:"
9. Display only the car name and energy consumption columns from the outliers DataFrame

Code:

```
import pandas as p

 data = p.read_excel("FEV-data-Excel.xlsx",engine="openpyxl")

energy_column = 'mean - Energy consumption [kWh/100 km]'

energy_data = data[energy_column]

 Q1 = energy_data.quantile(0.25)

 Q3 = energy_data.quantile(0.75)

 IQR = Q3 - Q1

lwr_bound = Q1 - 1.5 * IQR

upr_bound = Q3 + 1.5 * IQR

outliers = data[(data[energy_column] < lwr_bound) | (data[energy_column] >upr_bound)]

print("Outliers in Energy Consumption:")    print(outliers[['Car full name', energy_column]])
```

Result:

```
Outliers in Energy Consumption:

 Empty DataFrame Columns:

[Car full name, mean - Energy consumption [kWh/100 km]]

 Index: []
```

Findings:

This shows that there are no outliers in the given data.

Task 3: Your manager wants to know if there's a strong relationship between battery capacity and range.

a) Create a suitable plot to visualize.

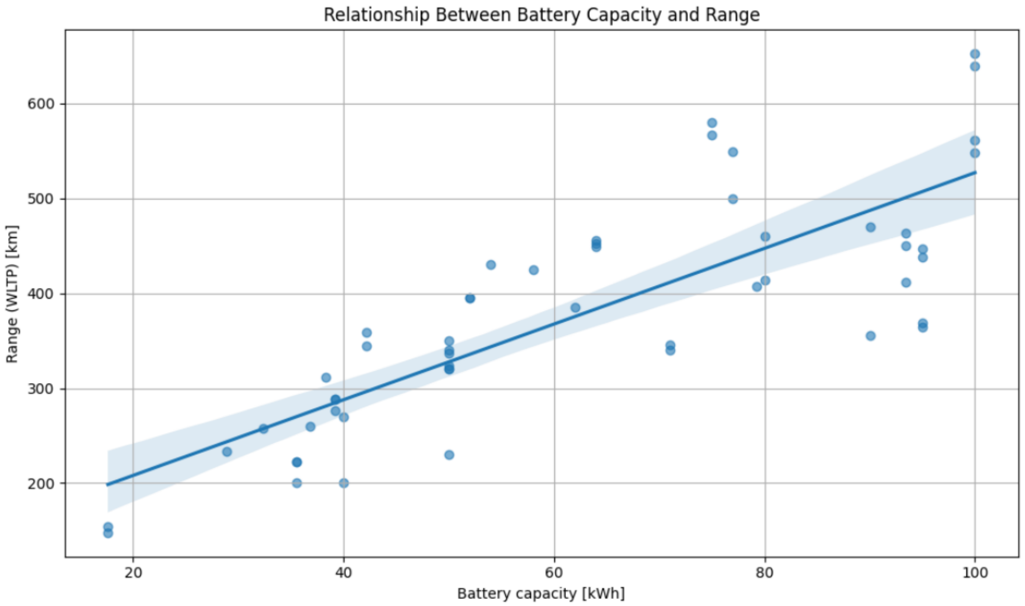
Steps:

1. Import required libraries:
   - pandas as p
   - matplotlib.pyplot as plt
   - seaborn as sns
2. Load the dataset using pd.read_excel() into variable dataset with engine="openpyxl"
3. Remove rows with missing values in either 'Battery capacity [kWh]' or 'Range (WLTP) [km]' and store in clean_data
4. Create a plot figure with a specific size using plt.figure(figsize=(10, 6))
5. Plot the data using sns.regplot():
   - Set x-axis to 'Battery capacity [kWh]'
   - Set y-axis to 'Range (WLTP) [km]'
   - Set scatter_kws={'alpha': 0.6} to make points slightly transparent
6. Set plot title as 'Relationship Between Battery Capacity and Range'
7. Label the axes:
   - x-axis: 'Battery capacity [kWh]'
   - y-axis: 'Range (WLTP) [km]'
8. Enable grid lines using plt.grid(True)
9. Optimize layout with plt.tight_layout()
10. Display the plot using plt.show()

Code:

```
import pandas as pd import

matplotlib.pyplot as plt

 import seaborn as sns

 dataset = p.read_excel('FEV-data-Excel.xlsx',engine="openpyxl")

clean_data = dataset.dropna(subset=['Battery capacity [kWh]', 'Range (WLTP) [km]'])
plt.figure(figsize=(10, 6))

sns.regplot(x='Battery capacity [kWh]', y='Range (WLTP) [km]',data=clean_data,scatter_kws={'alpha':0.6})
plt.title('Relationship Between Battery Capacity and Range')

plt.xlabel('Battery capacity [kWh]')
plt.ylabel('Range (WLTP) [km]')

plt.grid(True)

plt.tight_layout()

plt.show()
```

Result:



b) Highlight any insights

Steps:

1. Calculate the correlation using the .corr() function:
   a. correlation = df_clean['Battery capacity [kWh]'].corr(df_clean['Range (WLTP) [km]'])
2. Print the correlation result with a meaningful message: "Correlation between battery capacity and range:"

Code:

```
correlation = clean_data['Battery capacity [kWh]'].corr(clean_data['Range (WLTP) [km]'])

print("Correlation between battery capacity and range:", correlation)
```

Result:

```
Correlation between battery capacity and range: 0.8104385771936846
```

Findings:

1. The scatter plot displays a positive correlation between battery capacity and range in kilometers.
2. As battery capacity increases, the vehicle range also tends to increase.
3. The correlation between battery capacity and range is 0.81 which shows the positive correlation.

Task 4: Build an EV recommendation class. The class should allow users to input their budget, desired range, and battery capacity. The class should then return the top three EVs matching their criteria.

Steps:

1. Import the pandas library as p
2. Load the Excel file using p.read_excel() into a DataFrame named data
3. Create a class EVrecommendation with:
    - i. __init__(self, d): stores the dataset in self.d
    - ii. recommendation(self, budget, range, capacity):

    b. Filter the data:

    - i. Price ≤ budget
    - ii. Range ≥ user-defined minimum
    - iii. Battery capacity ≥ user-defined minimum

    c. Sort the filtered results by:

    - i. 'Range (WLTP) [km]' and 'Battery capacity [kWh]' in descending order

    d. Return top 3 results with selected columns using head().

4. Use try block to:
    - Get budget, range, and capacity inputs from the user and convert them to int/float
    - Create an instance of the class with the cleaned dataset: recommender = EVrecommendation(data)
    - Call the recommendation() method with user inputs
    - If results are found:
        - Print a message: “Top 3 Recommended EVs for You”
        - Print the result as a formatted table
    - Else:
        - Print: “No EVs match your criteria.”
5. Use except block to handle ValueError:
6. Print: “Please enter valid numeric inputs.”

Code:

```
import pandas as p

data = p.read_excel(r"C:\Users\91939\Downloads\FEV-data-Excel.xlsx")

class EVrecommendation:

 def __init__(self,d):

self.d=d

def recommendation(self,budget,range,capacity):

recom_data = self.d[(self.d["Minimal price (gross) [PLN]"]<=budget) & (self.d["Range (WLTP) [km]"]>=range) & (self.d["Battery capacity [kWh]"]>=capacity) ]

 top_EV = recom_data.sort_values(by=["Range (WLTP) [km]","Battery capacity [kWh]"],ascending=False).head(3)

return top_EV[['Car full name','Make','Model','Minimal price (gross) [PLN]','Range (WLTP) [km]','Battery capacity [kWh]']]

 try:

a = int(input("Enter your budget: "))

 b = int(input("Enter your minimum desired range (km): "))

c = float(input("Enter your minimum desired battery capacity (kWh): "))

recommender = EVrecommendation(data)

results = recommender.recommendation(a, b , c)

if not results.empty:

 print("\nTop 3 Recommended EVs for You:\n")

print(results.to_string(index=False))

else:

print("\nNo EVs match your criteria.")

except ValueError:

print("Please enter valid numeric inputs.")
```

Result:

```
Enter your budget: 300000

Enter your minimum desired range (km): 400

Enter your minimum desired battery capacity (kWh): 50

Top 3 Recommended EVs for You:

Car full name ,Make, Model, Minimal price (gross) [PLN], Range (WLTP) [km] ,Battery capacity [kWh]

Tesla Model 3 ,Long Range Tesla Model 3, Long Range, 235490, 580 ,75.0

Tesla Model 3, Performance Tesla Model 3, Performance, 260490, 567, 75.0

 Volkswagen ID.3, Pro S Volkswagen ID.3, Pro S, 179990, 549, 77.0
```

Task 5: Inferential Statistics – Hypothesis Testing: Test whether there is a significant difference in the average Engine power [KM] of vehicles manufactured by two leading manufacturers i.e. Tesla and Audi. What insights can you draw from the test results? Recommendations and Conclusion: Provide actionable insights based on your analysis. (Conduct a two sample t-test using ttest_ind from scipy.stats module)

Steps:

1. Import:pandas as p and ttest_ind function from scipy.stats
2. Load the Excel file using p.read_excel() into a variable named data
3. Extract engine power values for:
   - tesla: filter rows where 'Make' == 'Tesla' and select 'Engine power [KM]'
   - audi: filter rows where 'Make' == 'Audi' and select 'Engine power [KM]'
4. Perform an independent two-sample t-test using:
   - ttest_ind(tesla, audi, equal_var=False)
5. Store the resulting t-statistic and p-value in variables: t_stat, p_value
6. Round and display the t-statistic and p-value.

Code:

```
import pandas as p

 from scipy.stats import ttest_ind

data = p.read_excel(r"C:\Users\91939\Downloads\FEV-data-Excel.xlsx")

tesla = data[data['Make'] == 'Tesla']['Engine power [KM]']

audi = data[data['Make'] == 'Audi']['Engine power [KM]']

t_stat, p_value = ttest_ind(tesla, audi, equal_var=False)

print("T-statistic:", round(t_stat, 3))

print("P-value:", round(p_value, 4))
```

Result:

```
T-statistic: 1.794 P-value: 0.1068
```

Hypotheses Testing:

- H_0 (Null Hypothesis): There is no significant difference in the average engine power of Tesla and Audi vehicles.
- H_1 (Alternative Hypothesis): There is a significant difference in average engine power between the two brands.

Conclusion:

- Significance level (α) = 0.05
- T-statistic: 1.794
- P-value: 0.1068
- P value is greater than the significance level (0.1068>0.05)

This result shows that the null hypothesis is acceptable.Hence, there is no significant difference in the average engine power of Tesla and Audi vehicles.