LovePythonProjectInternshal

January 21, 2025

Electric Vehicle Data Analysis Project

Project Overview

w In this project, you will analyze a dataset related to electric vehicles (EVs). The dataset contains various features such as electric range, energy consumption, price, and other relevant attributes. Your goal is to conduct a thorough analysis to uncover meaningful insights, tell a compelling story, conduct hypothesis testing and provide actionable recommendations based on the data.

[1]:
```
import numpy as np
import pandas as pd
```

[2]:
```
df = pd.read_csv("FEV-data-Excel.xlsx - Auta elektryczne.csv")
```

Data Overview

Car full name: The full name or designation of the vehicle, often combining make, model, and variant. Make: The brand or manufacturer of the car. Model: The specific model or version of the car. Minimal price (gross) [PLN]: The minimum retail price of the car, in Polish złoty (PLN). Engine power [KM]: The car's engine power, measured in horsepower (KM in Polish). Maximum torque [Nm]: The peak torque the engine can produce, measured in Newton-eters (Nm). Type of brakes: The braking system used, such as disc or drum brakes. Drive type: The drivetrain configuration, like FWD (front-wheel drive), RWD (rear-whee drive), or AWD (all-wheel drive). Battery capacity [kWh]: Total energy capacity of the car's battery, measured in kilwatt-hours (kWh). Range (WLTP) [km]: Estimated driving range on a full charge under WLTP tandards, in kilometers. Wheelbase [cm]: The distance between the front and rear axles, in centimeters. Length [cm]: The overall length of the car, in centimeters. Width [cm]: The car's width, in centimeters. Height [cm]: The car's height, in centimeters. Minimal empty weight [kg]: The car's minimum weight when empty, measured in kilograms. Permissiblekg]: Maximum legally allowed weight, incl ding passengers and cargo, in kilograms. Maximum load capacity [kg]: The maximum weight the car can carry, in kilograms. Number of seats: The number of passenger seats in the car. Number of doors: The number of doors on the car. Tire size [in]: The tire size, measured in inches. Maximum speed [kph]: The top speed of the car, in kilometers per hour. Boot capacity (VDA) [l]: Trunk or cargo space capacity, measured in liters according to VDA standards. Acceleration 0-100 kph [s]: Time taken to accelerate from 0 t 100 kilometers per hour, in seconds. Maximum DC charging power [kW]: The highest charging ower supported when using a DC fast charger, in kilowatts (kW). Mean - Energy consumption [kWh/100 km]: Average energy consumption per 100 kilometers, in kilowatt-hours (kWh).

[4]:
```
df.head()
```

[4]:

```
                    Car full name  Make                         Model  \
0           Audi e-tron 55 quattro  Audi           e-tron 55 quattro
1           Audi e-tron 50 quattro  Audi           e-tron 50 quattro
2            Audi e-tron S quattro  Audi            e-tron S quattro
3  Audi e-tron Sportback 50 quattro  Audi  e-tron Sportback 50 quattro
4  Audi e-tron Sportback 55 quattro  Audi  e-tron Sportback 55 quattro

   Minimal price (gross) [PLN]  Engine power [KM]  Maximum torque [Nm]  \
0                       345700                360                  664
1                       308400                313                  540
2                       414900                503                  973
3                       319700                313                  540
4                       357000                360                  664

       Type of brakes Drive type  Battery capacity [kWh]  Range (WLTP) [km]  \
0  disc (front + rear)        4WD                    95.0                438
1  disc (front + rear)        4WD                    71.0                340
2  disc (front + rear)        4WD                    95.0                364
3  disc (front + rear)        4WD                    71.0                346
4  disc (front + rear)        4WD                    95.0                447

   …  Permissable gross weight [kg]  Maximum load capacity [kg]  \
0  …                         3130.0                       640.0
1  …                         3040.0                       670.0
2  …                         3130.0                       565.0
3  …                         3040.0                       640.0
4  …                         3130.0                       670.0

   Number of seats  Number of doors  Tire size [in]  Maximum speed [kph]  \
0                5                5              19                  200
1                5                5              19                  190
2                5                5              20                  210
3                5                5              19                  190
4                5                5              19                  200

   Boot capacity (VDA) [l]  Acceleration 0-100 kph [s]  \
0                    660.0                         5.7
1                    660.0                         6.8
2                    660.0                         4.5
3                    615.0                         6.8
4                    615.0                         5.7

   Maximum DC charging power [kW]  mean - Energy consumption [kWh/100 km]
0                             150                                   24.45
1                             150                                   23.80
2                             150                                   27.55
3                             150                                   23.30
```

```
4                              150                                     23.85

[5 rows x 25 columns]
```

The all about discrition of dataset statics

[5]:
```
df.describe()
```

[5]:
```
       Minimal price (gross) [PLN]  Engine power [KM]  Maximum torque [Nm]  \
count                    53.000000          53.000000            53.000000
mean                 246158.509434         269.773585           460.037736
std                  149187.485190         181.298589           261.647000
min                   82050.000000          82.000000           160.000000
25%                  142900.000000         136.000000           260.000000
50%                  178400.000000         204.000000           362.000000
75%                  339480.000000         372.000000           640.000000
max                  794000.000000         772.000000          1140.000000

       Battery capacity [kWh]  Range (WLTP) [km]  Wheelbase [cm]  Length [cm]  \
count               53.000000          53.000000       53.000000    53.000000
mean                62.366038         376.905660      273.581132   442.509434
std                 24.170913         118.817938       22.740518    48.863280
min                 17.600000         148.000000      187.300000   269.500000
25%                 40.000000         289.000000      258.800000   411.800000
50%                 58.000000         364.000000      270.000000   447.000000
75%                 80.000000         450.000000      290.000000   490.100000
max                100.000000         652.000000      327.500000   514.000000

       Width [cm]  Height [cm]  Minimal empty weight [kg]  \
count   53.000000    53.000000                  53.000000
mean   186.241509   155.422642                1868.452830
std     14.280641    11.275358                 470.880867
min    164.500000   137.800000                1035.000000
25%    178.800000   148.100000                1530.000000
50%    180.900000   155.600000                1685.000000
75%    193.500000   161.500000                2370.000000
max    255.800000   191.000000                2710.000000

       Permissable gross weight [kg]  Maximum load capacity [kg]  \
count                      45.000000                   45.000000
mean                     2288.844444                  520.466667
std                       557.796026                  140.682848
min                      1310.000000                  290.000000
25%                      1916.000000                  440.000000
50%                      2119.000000                  486.000000
75%                      2870.000000                  575.000000
max                      3500.000000                 1056.000000
```

```
       Number of seats  Number of doors  Tire size [in]  Maximum speed [kph]  \
count        53.000000        53.000000       53.000000            53.000000
mean          4.905660         4.849057       17.679245           178.169811
std           0.838133         0.455573        1.868500            43.056196
min           2.000000         3.000000       14.000000           123.000000
25%           5.000000         5.000000       16.000000           150.000000
50%           5.000000         5.000000       17.000000           160.000000
75%           5.000000         5.000000       19.000000           200.000000
max           8.000000         5.000000       21.000000           261.000000

       Boot capacity (VDA) [l]  Acceleration 0-100 kph [s]  \
count                52.000000                    50.00000
mean                445.096154                     7.36000
std                 180.178480                     2.78663
min                 171.000000                     2.50000
25%                 315.000000                     4.87500
50%                 425.000000                     7.70000
75%                 558.000000                     9.37500
max                 870.000000                    13.10000

       Maximum DC charging power [kW]  mean - Energy consumption [kWh/100 km]
count                       53.000000                               44.000000
mean                       113.509434                               18.994318
std                         57.166970                                4.418253
min                         22.000000                               13.100000
25%                        100.000000                               15.600000
50%                        100.000000                               17.050000
75%                        150.000000                               23.500000
max                        270.000000                               28.200000
```

[6]:
```
df.isnull().sum()
```

[6]:
```
Car full name                     0
Make                              0
Model                             0
Minimal price (gross) [PLN]       0
Engine power [KM]                 0
Maximum torque [Nm]               0
Type of brakes                    1
Drive type                        0
Battery capacity [kWh]            0
Range (WLTP) [km]                 0
Wheelbase [cm]                    0
Length [cm]                       0
Width [cm]                        0
Height [cm]                       0
Minimal empty weight [kg]         0
```

```
Permissable gross weight [kg]              8
Maximum load capacity [kg]                 8
Number of seats                            0
Number of doors                            0
Tire size [in]                             0
Maximum speed [kph]                        0
Boot capacity (VDA) [l]                    1
Acceleration 0-100 kph [s]                 3
Maximum DC charging power [kW]             0
mean - Energy consumption [kWh/100 km]     9
dtype: int64
```

here we see that column "types of break" having one null value and column permissible gross weight having and maximum load capicity having 8 null values.boot capicity having 1 null value, acclerationing 0-100kph having 3 null value , mean-energy consumption having 9 null values Type of brakes are string type column Permissable gross weight [kg] is float type column. Maximum load capacity [kg] also a float type column Boot capacity (VDA) [l] also having int type column Acceleration 0-100 kph [s] is float type column mean - Energy consumption [kWh/100 km] is float type column

string type column replace null value using mode(which frequency is higher)

[7]:
```
mode_value = df['Type of brakes'].mode()[0]
print(mode_value)
df['Type of brakes'].fillna(mode_value,inplace=True)
```

```
disc (front + rear)
```

For all type of float type column using Interpolate Missing Values which a prediction method

[8]:
```
df['Permissable gross weight [kg]'].interpolate(method='linear', inplace=True)
```

[9]:
```
df['Maximum load capacity [kg]'].interpolate(method='linear', inplace=True)
```

[10]:
```
df['Boot capacity (VDA) [l]'].interpolate(method='linear', inplace=True)
```

[11]:
```
df['Acceleration 0-100 kph [s]'].interpolate(method='linear', inplace=True)
```

[12]:
```
df['mean - Energy consumption [kWh/100 km]'].interpolate(method='linear',␣
↪inplace=True)
```

[13]:
```
df.head()
```

[13]:
```
                     Car full name  Make                      Model  \
0            Audi e-tron 55 quattro  Audi          e-tron 55 quattro
1            Audi e-tron 50 quattro  Audi          e-tron 50 quattro
2             Audi e-tron S quattro  Audi           e-tron S quattro
3  Audi e-tron Sportback 50 quattro  Audi  e-tron Sportback 50 quattro
4  Audi e-tron Sportback 55 quattro  Audi  e-tron Sportback 55 quattro
```

```
   Minimal price (gross) [PLN]  Engine power [KM]  Maximum torque [Nm]  \
0                       345700                360                  664
1                       308400                313                  540
2                       414900                503                  973
3                       319700                313                  540
4                       357000                360                  664

        Type of brakes Drive type  Battery capacity [kWh]  Range (WLTP) [km]  \
0  disc (front + rear)        4WD                    95.0                438
1  disc (front + rear)        4WD                    71.0                340
2  disc (front + rear)        4WD                    95.0                364
3  disc (front + rear)        4WD                    71.0                346
4  disc (front + rear)        4WD                    95.0                447

   …  Permissable gross weight [kg]  Maximum load capacity [kg]  \
0  …                         3130.0                       640.0
1  …                         3040.0                       670.0
2  …                         3130.0                       565.0
3  …                         3040.0                       640.0
4  …                         3130.0                       670.0

   Number of seats  Number of doors  Tire size [in]  Maximum speed [kph]  \
0                5                5              19                  200
1                5                5              19                  190
2                5                5              20                  210
3                5                5              19                  190
4                5                5              19                  200

   Boot capacity (VDA) [l]  Acceleration 0-100 kph [s]  \
0                    660.0                         5.7
1                    660.0                         6.8
2                    660.0                         4.5
3                    615.0                         6.8
4                    615.0                         5.7

   Maximum DC charging power [kW]  mean - Energy consumption [kWh/100 km]
0                             150                                   24.45
1                             150                                   23.80
2                             150                                   27.55
3                             150                                   23.30
4                             150                                   23.85

[5 rows x 25 columns]
```

[14]:

```
df.isnull().sum()
```

[14]:
```
Car full name                                 0
Make                                          0
Model                                         0
Minimal price (gross) [PLN]                   0
Engine power [KM]                             0
Maximum torque [Nm]                           0
Type of brakes                                0
Drive type                                    0
Battery capacity [kWh]                        0
Range (WLTP) [km]                             0
Wheelbase [cm]                                0
Length [cm]                                   0
Width [cm]                                    0
Height [cm]                                   0
Minimal empty weight [kg]                     0
Permissable gross weight [kg]                 0
Maximum load capacity [kg]                    0
Number of seats                               0
Number of doors                               0
Tire size [in]                                0
Maximum speed [kph]                           0
Boot capacity (VDA) [l]                       0
Acceleration 0-100 kph [s]                    0
Maximum DC charging power [kW]                0
mean - Energy consumption [kWh/100 km]        0
dtype: int64
```

Here the dataset with null values

[ ]:
```

```

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.(2 Marks) b) Group them by the manufacturer (Make).(6 mar ks) c) Calculate the average battery capacity for each manufacturer. (8 Ma rks)

[15]:
```
#filtered datset
filtered_df = df[(df['Minimal price (gross) [PLN]'] <= 350000) & (df['Range␣
↪(WLTP) [km]'] >= 400)]
```

[16]:
```
#making group using groupby() function
grouped_by_make = filtered_df.groupby('Make')
```

[33]:
```
avg_battery_capacity = grouped_by_make['Battery capacity [kWh]'].mean()
avg_battery_capacity.reset_index(name='Average Battery Capacity [kWh]')
```

[33]:

	Make	Average Battery Capacity [kWh]
0	Audi	95.000000
1	BMW	80.000000
2	Hyundai	64.000000
3	Kia	64.000000
4	Mercedes-Benz	80.000000
5	Tesla	68.000000
6	Volkswagen	70.666667

Task 2: You suspect some EVs have unusually high or low energy consumption.

<nFind the outliers in the mean - Energy consumption [kWh/100 km] column.(16 Marks)

to detect outlier mean-energy consumption we can use statical method such as interquartile range

step 1:

calculate the first and third quartile

step 2:

compute IQR IQR = Q3 - Q1

step 3:

Define Outliers as [Q1 - 1.5*IQR , Q3+1.5*IQR]

step 4:

identify outlier by conditions

[42]:
```
# Step 1: Calculate Q1 and Q3
q1 = df['mean - Energy consumption [kWh/100 km]'].quantile(0.25)
q3 = df['mean - Energy consumption [kWh/100 km]'].quantile(0.75)
print(q1,q3)
```

```
15.5 21.85
```

[43]:
```
# Step 2: Compute IQR
iqr = q3 - q1
print(iqr)
```

```
6.350000000000001
```

[45]:
```
# Step 3: Define lower and upper bounds for outliers
lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr
print(lower_bound,upper_bound)
```

```
5.974999999999998 31.375000000000004
```

[49]:
```
# Step 4: Identify outliers
outliers = df[
    (df['mean - Energy consumption [kWh/100 km]'] < lower_bound) |
```

```
    (df['mean - Energy consumption [kWh/100 km]'] > upper_bound)
]

# Display the outliers
outliers
```

[49]:

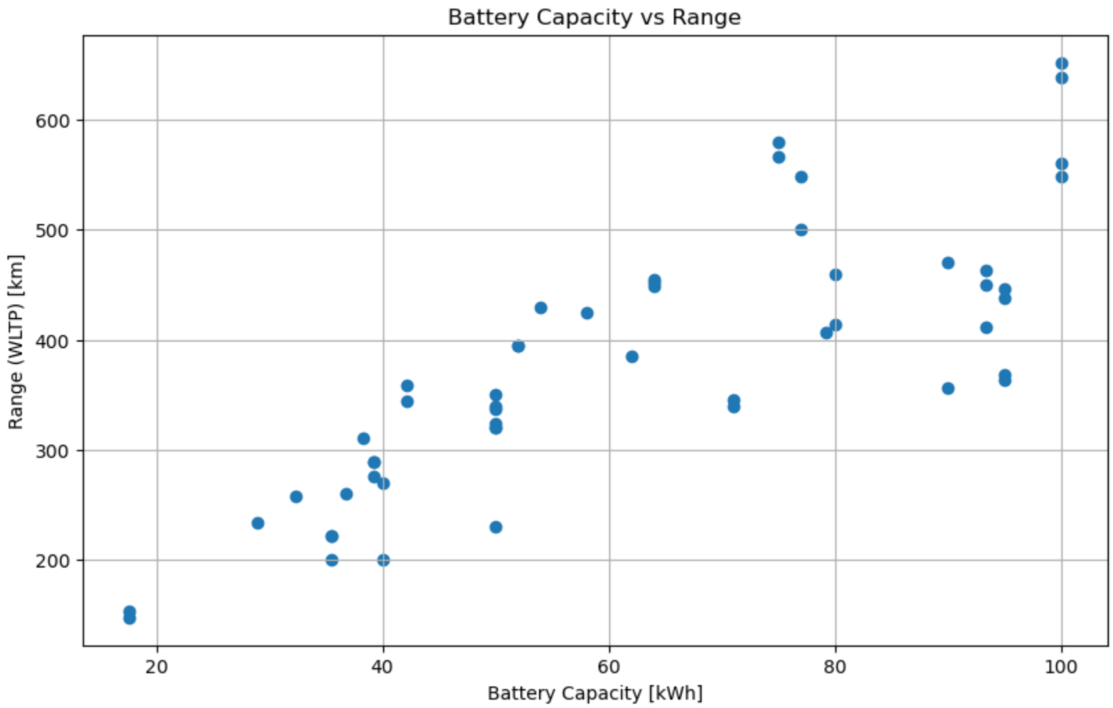
```
Empty DataFrame
Columns: [Car full name, Make, Model, Minimal price (gross) [PLN], Engine power
[KM], Maximum torque [Nm], Type of brakes, Drive type, Battery capacity [kWh],
Range (WLTP) [km], Wheelbase [cm], Length [cm], Width [cm], Height [cm], Minimal
empty weight [kg], Permissable gross weight [kg], Maximum load capacity [kg],
Number of seats, Number of doors, Tire size [in], Maximum speed [kph], Boot
capacity (VDA) [l], Acceleration 0-100 kph [s], Maximum DC charging power [kW],
mean - Energy consumption [kWh/100 km]]
Index: []

[0 rows x 25 columns]
```

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.(8 Marks)
b) Highlight any insights.(8 Marks)

[26]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.scatter(df['Battery capacity [kWh]'], df['Range (WLTP) [km]'])
plt.title('Battery Capacity vs Range')
plt.xlabel('Battery Capacity [kWh]')
plt.ylabel('Range (WLTP) [km]')
plt.grid(True)
plt.show()
```



Insights of datsets

Costumer prefer the brand and viechal which has higher Range(WLTP)[KM] AND having high Battery capicity with friendly budget

[ ]:

```

```

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. (8+8 Marks)

[52]:

```
class EVRecommendation:
    def __init__(self, dataframe):
        self.dataframe = dataframe

    def recommend(self, budget, desired_range, battery_capacity):
        recommendations = self.dataframe[
            (self.dataframe['Minimal price (gross) [PLN]'] <= budget) &
            (self.dataframe['Range (WLTP) [km]'] >= desired_range) &
            (self.dataframe['Battery capacity [kWh]'] >= battery_capacity)
        ].nsmallest(3, 'Minimal price (gross) [PLN]')
        return recommendations

# Create an instance of the recommendation class
ev_recommender = EVRecommendation(df)
```

```python
# Example usage
top_ev_recommendations = ev_recommender.recommend(350000, 400, 50)
top_ev_recommendations
```

[52]:

```
                     Car full name        Make                Model  \
47  Volkswagen ID.3 Pro Performance  Volkswagen  ID.3 Pro Performance
20                 Kia e-Soul 64kWh         Kia          e-Soul 64kWh
18                 Kia e-Niro 64kWh         Kia          e-Niro 64kWh

    Minimal price (gross) [PLN]  Engine power [KM]  Maximum torque [Nm]  \
47                       155890                204                  310
20                       160990                204                  395
18                       167990                204                  395

               Type of brakes   Drive type  Battery capacity [kWh]  \
47  disc (front) + drum (rear)   2WD (rear)                    58.0
20         disc (front + rear)  2WD (front)                    64.0
18         disc (front + rear)  2WD (front)                    64.0

    Range (WLTP) [km]  …  Permissable gross weight [kg]  \
47                425  …                         2270.0
20                452  …                         1682.0
18                455  …                         2230.0

    Maximum load capacity [kg]  Number of seats  Number of doors  \
47                       540.0                5                5
20                       498.0                5                5
18                       493.0                5                5

    Tire size [in]  Maximum speed [kph]  Boot capacity (VDA) [l]  \
47              18                  160                    385.0
20              17                  167                    315.0
18              17                  167                    451.0

    Acceleration 0-100 kph [s]  Maximum DC charging power [kW]  \
47                         7.3                             100
20                         7.9                             100
18                         7.8                             100

    mean - Energy consumption [kWh/100 km]
47                                    15.4
20                                    15.7
18                                    15.9

[3 rows x 25 columns]
```

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) (16 Marks)

Step 1:

Import the Library for hypothesis testing

[58]:
```
from scipy.stats import ttest_ind
```

Step 2:

Filter the datasets on bases tesla and audi

[59]:
```
tesla_audi_data = df[df['Make'].isin(['Tesla', 'Audi'])]
```

Step 3:

diffrinciate between manufactror as tesla and audi

[60]:
```
tesla_power = tesla_audi_data[tesla_audi_data['Make'] == 'Tesla']['Engine power␣
↪[KM]']
audi_power = tesla_audi_data[tesla_audi_data['Make'] == 'Audi']['Engine power␣
↪[KM]']
```

Step 4:

conduct T-Test for

[61]:
```
t_stat, p_value = ttest_ind(tesla_power, audi_power, equal_var=False)
t_stat, p_value
```

[61]:
```
(1.7939951827297178, 0.10684105068839565)
```

insights

t-static is 1.793 p-value is 0.1106

P Value Insights

the p avlue is gretar than 0.05 this indicates that we fail to reject null hypothesis. there is no staticlly significant diffrance in the avrage ingine power between audi and tesla viehal based on this dataset

T-static Value Insights

while the t-static is positive which tells us about that Tesla viehle may have a slightly higher avrage engine power than Audi the diffrance is not staticly significant

Task 6:

Project Video Explanation (20 Marks)

Video Explanation

```
[ ]:
```