# Data 602 – Final Project Proposal


## By: Mubashira Qari


**Project Title: Electric Vehicles Data Analysis and Price Prediction Using Machine Learning**

**Describing Dataset:** This dataset is about electric vehicles. It has 103 row and 14 columns.

This dataset requires detailed data cleaning before performing the data manipulation steps.

The metrics for the columns in the dataset are as below:

Vehicle Battery capacity (in KWH)

Acceleration (0-100) in Seconds

Top Speed in Km/hr

Range of Vehicle in km

The efficiency of Vehicle in Wh/km

Fast charge speed in km/hr

Drive Configuration

Number of seats

Price in Germany (Euro)

Price in the UK (Pound)

## Data Source Link:

This dataset is acquired from Kaggle website, and the link is provided below:

https://www.kaggle.com/datasets/geoffnel/evs-one-electric-vehicle-dataset?select=ElectricCarData_Norm.csv

https://www.kaggle.com/datasets/kkhandekar/cheapest-electric-cars

## Justification for Dataset Selection:

There are multiple reasons for choosing this dataset.

First, I eagerly wanted to do the research about the electric vehicles because I am interested in buying one for myself. In terms of carbon emission, Electric cars are good for the environment by preventing greenhouse gas emission.

The operating and maintenance costs are lot lower for electric vehicles and that makes them more economical and viable in the long run.

Initially, one of the biggest challenges is range anxiety and uncomfortable feeling of not finding the charging stations within the battery range. These concerns are being taken care by the long-range batteries hitting the market and GPS apps, providing the location of the nearest charging station. And people are more likely to buy electric vehicles.

Now there are couple of questions related to electric vehicle that require analysis and those are my research questions.

## Research Questions & Objectives:

The research questions for the electric vehicle's dataset are the following:

1- Which car has the fastest charging battery capacity?

2- Which vehicle has the highest efficiency?

3- Does a difference in power train effect the range, top speed, efficiency?

4- What is the correlation between the variables.

5- Which vehicle has the most range?

5- Which manufacturer has the greatest number of vehicles?

6- Which factors affect the price of the vehicle?

7- Merge the two datasets.

8- Upload the dataset in the SQL database for further analysis.

9- Predict the price of electric vehicles using machine learning.

## Libraries Used for Project Implementation:

- Python Pandas
- Python NumPy
- Python Matplotlib
- SQL Database
- Python sklearn
- Python seaborn

## EDA and Summary Statistics:

Below are the images of exploratory data analysis and summary statistics:

```
csv_path = 'resources/electric_car_data_norm.csv'
cars_df = pd.read_csv(csv_path, encoding="utf-8")
cars_df.head(5)
```

:

| | Brand | Model | Accel | TopSpeed | Range | Efficiency | FastCharge | RapidCharge | PowerTrain |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Tesla | Model 3 Long Range Dual Motor | 4.6 sec | 233 km/h | 450 km | 161 Wh/km | 940 km/h | Rapid charging possible | All Wheel Drive |
| 1 | Volkswagen | ID.3 Pure | 10.0 sec | 160 km/h | 270 km | 167 Wh/km | 250 km/h | Rapid charging possible | Rear Wheel Drive |
| 2 | Polestar | 2 | 4.7 sec | 210 km/h | 400 km | 181 Wh/km | 620 km/h | Rapid charging possible | All Wheel Drive |
| 3 | BMW | iX3 | 6.8 sec | 180 km/h | 360 km | 206 Wh/km | 560 km/h | Rapid charging possible | Rear Wheel Drive |
| 4 | Honda | e | 9.5 sec | 145 km/h | 170 km | 168 Wh/km | 190 km/h | Rapid charging possible | Rear Wheel Drive |

```
cars_df.tail(5)
```

| | Brand | Model | Accel | TopSpeed | Range | Efficiency | FastCharge | RapidCharge |
|---|---|---|---|---|---|---|---|---|
| **98** | Nissan | Ariya 63kWh | 7.5 sec | 160 km/h | 330 km | 191 Wh/km | 440 km/h | Rapid charging possible |
| **99** | Audi | e-tron S Sportback 55 quattro | 4.5 sec | 210 km/h | 335 km | 258 Wh/km | 540 km/h | Rapid charging possible |
| **100** | Nissan | Ariya e-4ORCE 63kWh | 5.9 sec | 200 km/h | 325 km | 194 Wh/km | 440 km/h | Rapid charging possible |
| **101** | Nissan | Ariya e-4ORCE 87kWh Performance | 5.1 sec | 200 km/h | 375 km | 232 Wh/km | 450 km/h | Rapid charging possible |
| **102** | Byton | M-Byte 95 kWh 2WD | 7.5 sec | 190 km/h | 400 km | 238 Wh/km | 480 km/h | Rapid charging possible |

```
cars_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103 entries, 0 to 102
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Brand        103 non-null    object
 1   Model        103 non-null    object
 2   Accel        103 non-null    object
 3   TopSpeed     103 non-null    object
 4   Range        103 non-null    object
 5   Efficiency   103 non-null    object
 6   FastCharge   103 non-null    object
 7   RapidCharge  103 non-null    object
 8   PowerTrain   103 non-null    object
 9   PlugType     103 non-null    object
 10  BodyStyle    103 non-null    object
 11  Segment      103 non-null    object
 12  Seats        103 non-null    int64 
 13  PriceEuro    103 non-null    int64 
dtypes: int64(2), object(12)
memory usage: 11.4+ KB
```

```
cars_df.columns
```

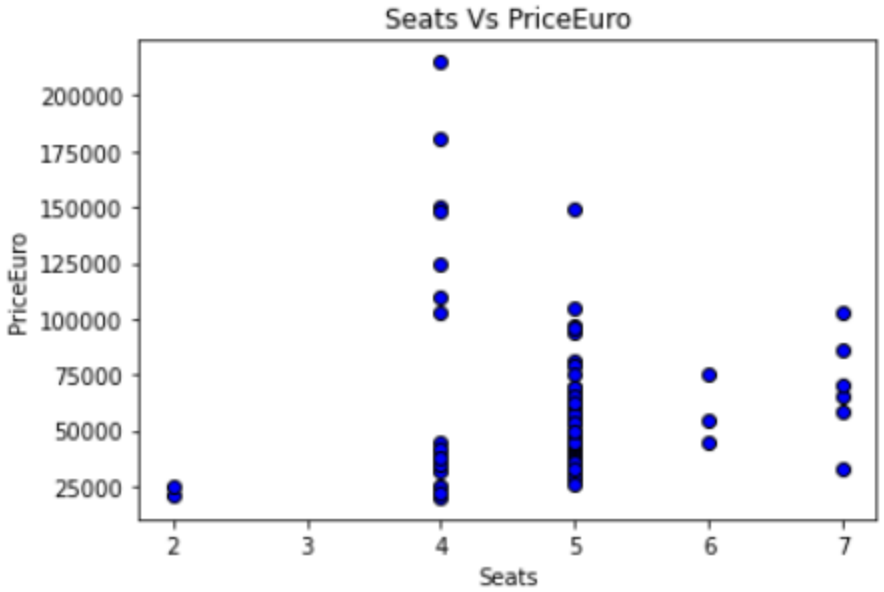
```
Index(['Brand', 'Model', 'Accel', 'TopSpeed', 'Range', 'Efficiency',
       'FastCharge', 'RapidCharge', 'PowerTrain', 'PlugType', 'BodyStyle',
       'Segment', 'Seats', 'PriceEuro'],
      dtype='object')
```

```python
# Find null values
for column in cars_df.columns:
    print(f"Column {column} has {cars_df[column].isnull().sum()} null values")
```

```
Column Brand has 0 null values
Column Model has 0 null values
Column Accel has 0 null values
Column TopSpeed has 0 null values
Column Range has 0 null values
Column Efficiency has 0 null values
Column FastCharge has 0 null values
Column RapidCharge has 0 null values
Column PowerTrain has 0 null values
Column PlugType has 0 null values
Column BodyStyle has 0 null values
Column Segment has 0 null values
Column Seats has 0 null values
Column PriceEuro has 0 null values
```

```python
cars_df.describe()
```

| | Seats | PriceEuro |
|---|---|---|
| **count** | 103.000000 | 103.000000 |
| **mean** | 4.883495 | 55811.563107 |
| **std** | 0.795834 | 34134.665280 |
| **min** | 2.000000 | 20129.000000 |
| **25%** | 5.000000 | 34429.500000 |
| **50%** | 5.000000 | 45000.000000 |
| **75%** | 5.000000 | 65000.000000 |
| **max** | 7.000000 | 215000.000000 |


Seats Vs PriceEuro
PriceEuro
200000
175000
150000
125000
100000
75000
50000
25000
2
3
4
5
6
7
Seats