Name: Parwinder Singh
Batch Code: LISUM21
Submission Date: July 22, 2023
Submitted to: GitHub

Step 1: Selected EV data and performed data cleaning process

In [5]:
```
print(clean_ev_data)
```

```
            Brand                              Model  AccelSec  TopSpeed_KmH  \
0           Tesla     Model 3 Long Range Dual Motor       4.6           233
1      Volkswagen                            ID.3 Pure      10.0           160
2        Polestar                                  2       4.7           210
3             BMW                               iX3        6.8           180
4           Honda                                 e        9.5           145
..            ...                               ...       ...           ...
98         Nissan                         Ariya 63kWh       7.5           160
99           Audi     e-tron S Sportback 55 quattro       4.5           210
100        Nissan               Ariya e-4ORCE 63kWh       5.9           200
101        Nissan   Ariya e-4ORCE 87kWh Performance       5.1           200
102         Byton                  M-Byte 95 kWh 2WD       7.5           190

     Range_Km  Efficiency_WhKm FastCharge_KmH RapidCharge PowerTrain  \
0         450              161            940         Yes        AWD
1         270              167            250         Yes        RWD
2         400              181            620         Yes        AWD
3         360              206            560         Yes        RWD
4         170              168            190         Yes        RWD
```

In [6]:
```
print(norm_ev_data)
```

```
            Brand                              Model     Accel  TopSpeed   Range  \
0           Tesla     Model 3 Long Range Dual Motor   4.6 sec  233 km/h  450 km
1      Volkswagen                            ID.3 Pure  10.0 sec  160 km/h  270 km
2        Polestar                                  2    4.7 sec  210 km/h  400 km
3             BMW                               iX3     6.8 sec  180 km/h  360 km
4           Honda                                 e     9.5 sec  145 km/h  170 km
..            ...                               ...        ...       ...     ...
98         Nissan                         Ariya 63kWh   7.5 sec  160 km/h  330 km
99           Audi     e-tron S Sportback 55 quattro    4.5 sec  210 km/h  335 km
100        Nissan               Ariya e-4ORCE 63kWh    5.9 sec  200 km/h  325 km
101        Nissan   Ariya e-4ORCE 87kWh Performance    5.1 sec  200 km/h  375 km
102         Byton                  M-Byte 95 kWh 2WD   7.5 sec  190 km/h  400 km

    Efficiency FastCharge              RapidCharge          PowerTrain  \
0    161 Wh/km   940 km/h  Rapid charging possible     All Wheel Drive
1    167 Wh/km   250 km/h  Rapid charging possible    Rear Wheel Drive
2    181 Wh/km   620 km/h  Rapid charging possible     All Wheel Drive
3    206 Wh/km   560 km/h  Rapid charging possible    Rear Wheel Drive
4    168 Wh/km   190 km/h  Rapid charging possible    Rear Wheel Drive
..         ...        ...                       ...                ...
98   191 Wh/km   440 km/h  Rapid charging possible  Front Wheel Drive
99   258 Wh/km   540 km/h  Rapid charging possible     All Wheel Drive
100  194 Wh/km   440 km/h  Rapid charging possible     All Wheel Drive
101  232 Wh/km   450 km/h  Rapid charging possible     All Wheel Drive
102  238 Wh/km   480 km/h  Rapid charging possible     All Wheel Drive

       PlugType  BodyStyle Segment  Seats  PriceEuro
0    Type 2 CCS      Sedan       D      5      55480
1    Type 2 CCS  Hatchback       C      5      30000
2    Type 2 CCS   Liftback       D      5      56440
3    Type 2 CCS        SUV       D      5      68040
4    Type 2 CCS  Hatchback       B      4      32997
..          ...        ...     ...    ...        ...
98   Type 2 CCS  Hatchback       C      5      45000
99   Type 2 CCS        SUV       E      5      96050
100  Type 2 CCS  Hatchback       C      5      50000
101  Type 2 CCS  Hatchback       C      5      65000
102  Type 2 CCS        SUV       E      5      62000

[103 rows x 14 columns]
```

Step 2: Saved the Gradient Boosting Regression Model

Gradient Boosting Regression Model

In [11]:
```
# Selected features and target variables to build model around
X = norm_ev_data[['TopSpeed', 'Range']]
y = norm_ev_data['PriceEuro']
```

In [12]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.5, random_state = 42)
```

In [13]:
```
# Created and trained the Gradient Boosting Regression model
gbr_model = GradientBoostingRegressor()

# Trained the model using the training data
gbr_model.fit(X_train, y_train)
```

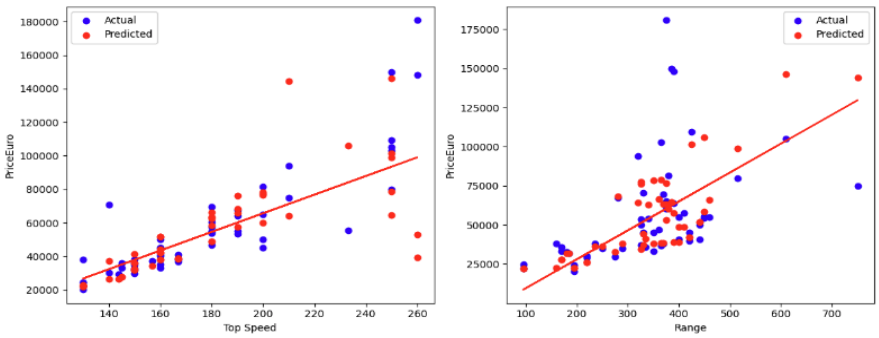
Out[13]:
```
GradientBoostingRegressor
GradientBoostingRegressor()
```

In [14]:
```
# Saved the trained model to a file for Flask deployment
with open('gbr_model.pkl', 'wb') as file:
    pickle.dump(gbr_model, file)
```

In [15]:
```
# Made predictions on the test set
y_pred = gbr_model.predict(X_test)

# Calculated the mean squared error and R-squared score
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Print the evaluation metrics
print('Mean Squared Error:', mse)
print('R-squared Score:', r2)
```

```
Mean Squared Error: 998166209.4394575
R-squared Score: 0.09532714133017783
```

In [16]:
```
plt.figure(figsize=(12, 5))

# Scatter plot for TopSpeed vs PriceEuro
plt.subplot(1, 2, 1)
plt.scatter(X_test['TopSpeed'], y_test, color='blue', label='Actual')
plt.scatter(X_test['TopSpeed'], y_pred, color='red', label='Predicted')
plt.xlabel('Top Speed')
plt.ylabel('PriceEuro')
plt.legend()

# Added the trend line for TopSpeed vs PriceEuro
trend_line_top_speed = np.polyfit(X_test['TopSpeed'], y_pred, 1)
plt.plot(X_test['TopSpeed'], np.polyval(trend_line_top_speed, X_test['TopSpeed']), color='red')

# Scatter plot for Range vs PriceEuro
plt.subplot(1, 2, 2)
plt.scatter(X_test['Range'], y_test, color='blue', label='Actual')
plt.scatter(X_test['Range'], y_pred, color='red', label='Predicted')
plt.xlabel('Range')
plt.ylabel('PriceEuro')
plt.legend()

# Added the trend line for Range vs PriceEuro
trend_line_range = np.polyfit(X_test['Range'], y_pred, 1)
plt.plot(X_test['Range'], np.polyval(trend_line_range, X_test['Range']), color='red')

plt.tight_layout()
plt.show()
```



Started process to deploy model on Mongo DB Cloud

Mongo DB Cloud Deployment

In [29]:
```
pip install pymongo
```

```
Collecting pymongo
  Downloading pymongo-4.4.1-cp310-cp310-macosx_10_9_universal2.whl (466 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 466.9/466.9 kB 5.0 MB/s eta 0:00:0000:0100:01
Collecting dnspython<3.0.0,>=1.16.0
  Downloading dnspython-2.4.0-py3-none-any.whl (300 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 300.0/300.0 kB 9.2 MB/s eta 0:00:00
Collecting httpcore>=0.17.3
  Downloading httpcore-0.17.3-py3-none-any.whl (74 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 74.5/74.5 kB 2.4 MB/s eta 0:00:00
Requirement already satisfied: sniffio<2.0,>=1.1 in /Users/parwindersingh/anaconda3/lib/pytho
n3.10/site-packages (from dnspython<3.0.0,>=1.16.0->pymongo) (1.2.0)
Collecting h11<0.15,>=0.13
  Downloading h11-0.14.0-py3-none-any.whl (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.3/58.3 kB 2.1 MB/s eta 0:00:00
Requirement already satisfied: anyio<5.0,>=3.0 in /Users/parwindersingh/anaconda3/lib/python
3.10/site-packages (from httpcore>=0.17.3->dnspython<3.0.0,>=1.16.0->pymongo) (3.5.0)
Requirement already satisfied: certifi in /Users/parwindersingh/anaconda3/lib/python3.10/site
-packages (from httpcore>=0.17.3->dnspython<3.0.0,>=1.16.0->pymongo) (2023.5.7)
Requirement already satisfied: idna>=2.8 in /Users/parwindersingh/anaconda3/lib/python3.10/si
te-packages (from anyio<5.0,>=3.0->httpcore>=0.17.3->dnspython<3.0.0,>=1.16.0->pymongo) (3.4)
Installing collected packages: h11, httpcore, dnspython, pymongo
Successfully installed dnspython-2.4.0 h11-0.14.0 httpcore-0.17.3 pymongo-4.4.1
Note: you may need to restart the kernel to use updated packages.
```

In [31]:
```
pip install --upgrade typing-extensions
```

```
Requirement already satisfied: typing-extensions in /Users/parwindersingh/anaconda3/lib/pytho
n3.10/site-packages (4.0.0)
Collecting typing-extensions
  Using cached typing_extensions-4.7.1-py3-none-any.whl (33 kB)
Installing collected packages: typing-extensions
  Attempting uninstall: typing-extensions
    Found existing installation: typing_extensions 4.0.0
    Uninstalling typing_extensions-4.0.0:
      Successfully uninstalled typing_extensions-4.0.0
Successfully installed typing-extensions-4.7.1
Note: you may need to restart the kernel to use updated packages.
```

In [33]:
```
pip install fastapi==0.68.0
```

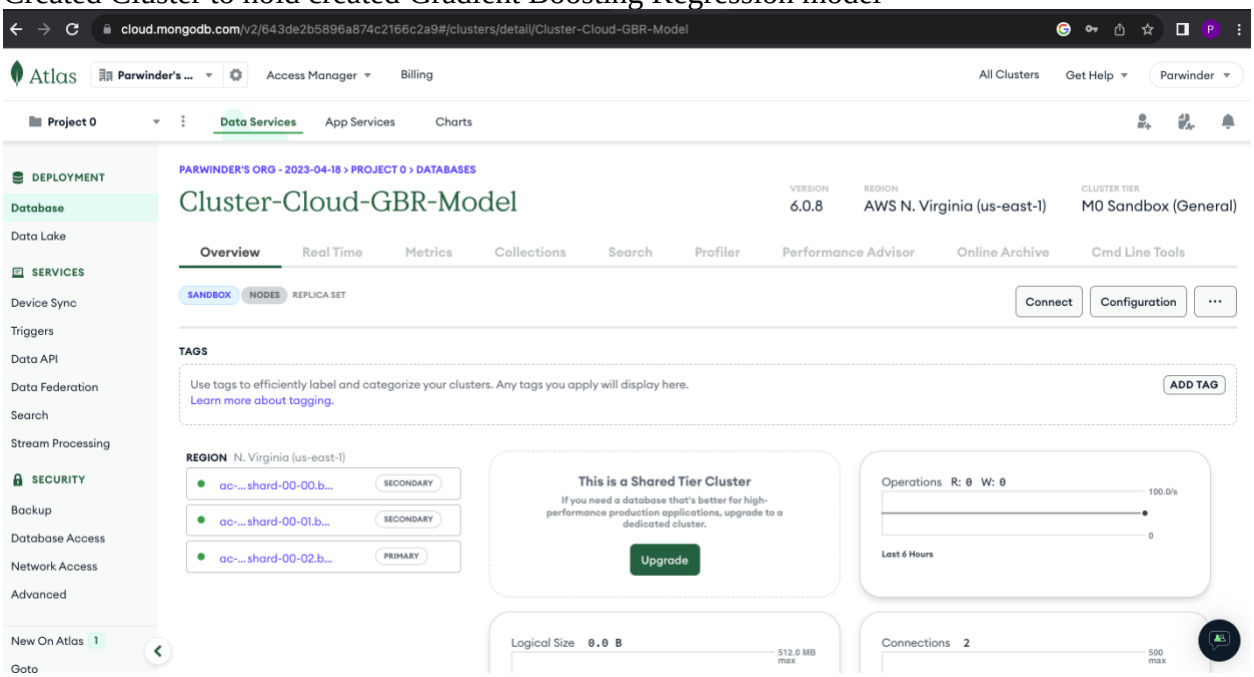
```
Collecting fastapi==0.68.0
  Downloading fastapi-0.68.0-py3-none-any.whl (52 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 52.0/52.0 kB 829.2 kB/s eta 0:00:000:01
Collecting starlette==0.14.2
  Downloading starlette-0.14.2-py3-none-any.whl (60 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.6/60.6 kB 1.7 MB/s eta 0:00:00
Collecting pydantic!=1.7,!=1.7.1,!=1.7.2,!=1.7.3,!=1.8,!=1.8.1,<2.0.0,>=1.6.2
  Downloading pydantic-1.10.11-cp310-cp310-macosx_10_9_x86_64.whl (2.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.9/2.9 MB 21.2 MB/s eta 0:00:0000:0100:01
Requirement already satisfied: typing-extensions>=4.2.0 in /Users/parwindersingh/anaconda3/li
b/python3.10/site-packages (from pydantic!=1.7,!=1.7.1,!=1.7.2,!=1.7.3,!=1.8,!=1.8.1,<2.0.0,>
=1.6.2->fastapi==0.68.0) (4.7.1)
Installing collected packages: starlette, pydantic, fastapi
  Attempting uninstall: starlette
    Found existing installation: starlette 0.27.0
    Uninstalling starlette-0.27.0:
      Successfully uninstalled starlette-0.27.0
  Attempting uninstall: pydantic
    Found existing installation: pydantic 2.0.3
    Uninstalling pydantic-2.0.3:
      Successfully uninstalled pydantic-2.0.3
  Attempting uninstall: fastapi
    Found existing installation: fastapi 0.100.0
    Uninstalling fastapi-0.100.0:
      Successfully uninstalled fastapi-0.100.0
Successfully installed fastapi-0.68.0 pydantic-1.10.11 starlette-0.14.2
Note: you may need to restart the kernel to use updated packages.
```

In [36]:
```
pip install fastapi==0.68.0 pydantic==1.8.2
```

```
Requirement already satisfied: fastapi==0.68.0 in /Users/parwindersingh/anaconda3/lib/python
3.10/site-packages (0.68.0)
Collecting pydantic==1.8.2
  Using cached pydantic-1.8.2-py3-none-any.whl (126 kB)
Requirement already satisfied: starlette==0.14.2 in /Users/parwindersingh/anaconda3/lib/pytho
n3.10/site-packages (from fastapi==0.68.0) (0.14.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /Users/parwindersingh/anaconda3/
lib/python3.10/site-packages (from pydantic==1.8.2) (4.7.1)
Installing collected packages: pydantic
  Attempting uninstall: pydantic
    Found existing installation: pydantic 1.10.11
    Uninstalling pydantic-1.10.11:
      Successfully uninstalled pydantic-1.10.11
Successfully installed pydantic-1.8.2
Note: you may need to restart the kernel to use updated packages.
```

In [38]:
```
import os
import pymongo
from fastapi import FastAPI
import numpy as np
import joblib
```

Created Cluster to hold created Gradient Boosting Regression model



Initialized API code to connect to Mongo

```
In [45]: # Set the path to model.pkl file
model_path = "/Users/parwindersingh/Desktop/Professional/Data Glacier/Week 4/function/model_pre

# Load the trained model
gbr_model = joblib.load(model_path)

# MongoDB Atlas configuration
mongo_uri = "mongodb+srv://<adminpar1>:<aaaaaaaaaaaaa>@cluster-cloud-gbr-model.bek2x90.mongo
client = pymongo.MongoClient(mongo_uri)
```

```
In [46]: app = FastAPI()
```

```
In [47]: @app.get("/model_predict")
async def predict(TopSpeed: int, Range: float):
    X_new = np.array([[TopSpeed, Range]])
    result = gbr_model.predict(X_new)
    return {"PriceEuro": round(result[0], 2)}
```

```
http://127.0.0.1:8000
```

API Model Link: https://127.0.0.1:8000

Linked API for Mongo DB and Model through Apple terminal