Machine Learning & Data Science Project: Data Analysis and Model Training

Yevgen Petronyuk Pidhaynyy

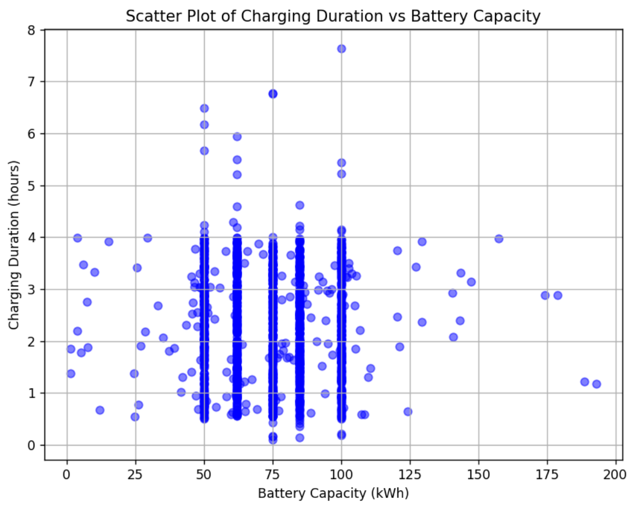
Phase 1: Data Analysis

1.1 Exploring Relationships Between Features and Target Value

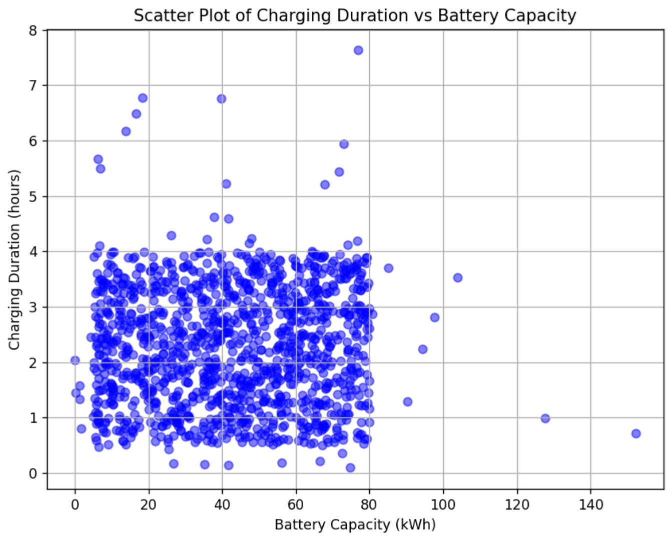
This project analyzes a CSV dataset containing electric vehicle charging patterns. The objective is to explore charging behaviors and identify key factors influencing our target variable. The chosen target variable for prediction is **Charging Duration**, which is the available driving time the vehicle has left. Machine learning techniques will be applied to develop models for accurate regression.

Some relevant features which could contain potential relation with our target variable are the following:

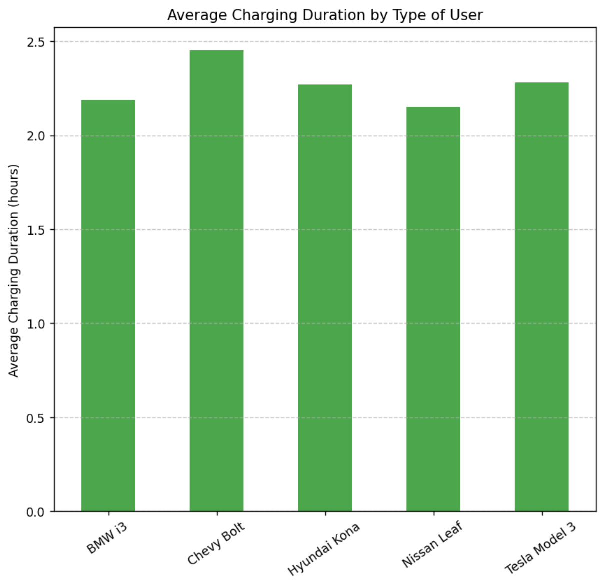
- **Battery Capacity:** Scatter plot that shows charging duration for batteries with different capacities:



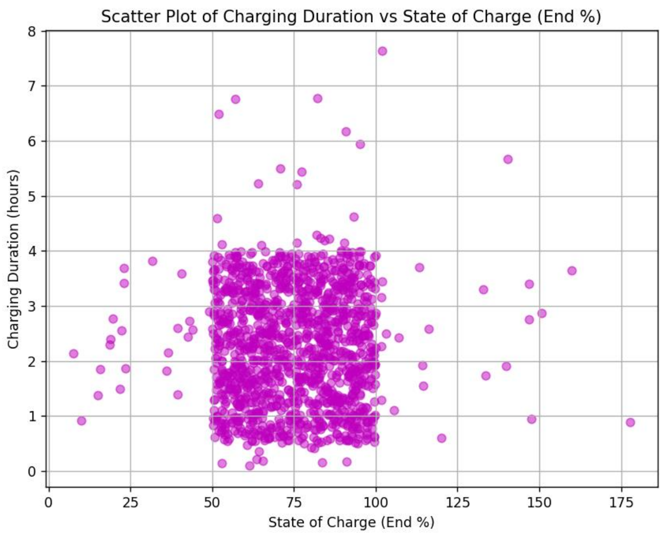
- **Energy Consumed (kWh):** Scatter plot representing data with different charging duration values and the associated consumed energy:



- **Vehicle Model:** Bar Plot showing how different vehicle models have different average charging durations:

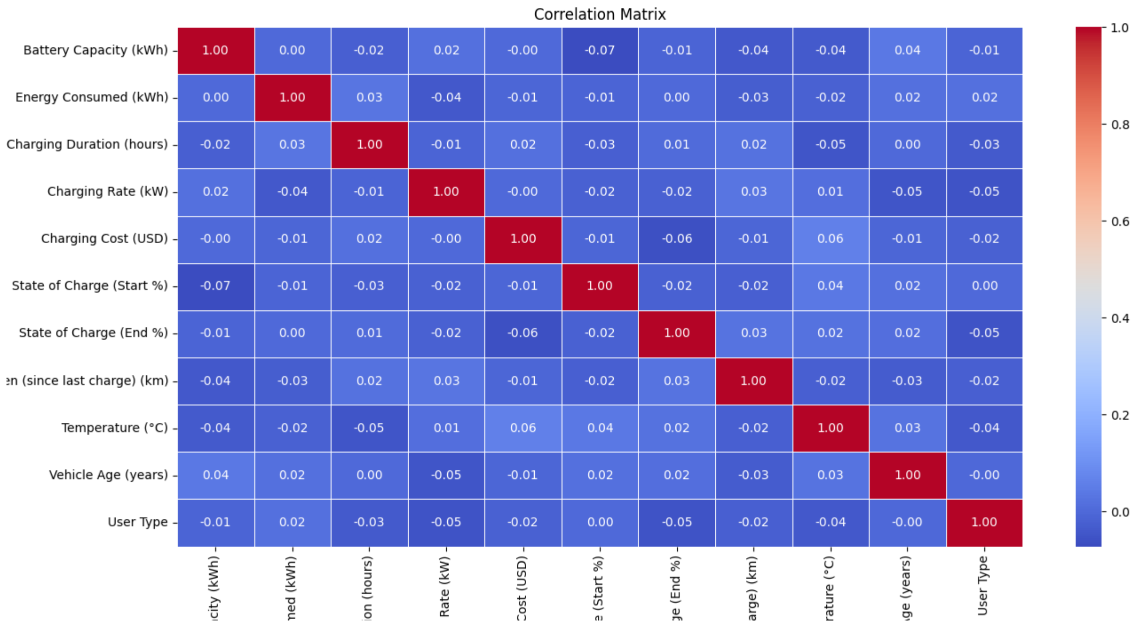


- **State of Charge (End %):** Scatter plot that shows relation between State of Charge (End %) variable and the target variable:



1.2 Measuring Correlations

Some of the variables in the dataset may have potential correlations with the target variable **Charging Duration**. However, if we observe the Correlation Matrix, there are very few correlated variables in general (not only to out target variable)



Correlation Matrix

	city (kWh)	ned (kWh)	on (hours)	Rate (kW)	Cost (USD)	e (Start %)	ge (End %)	arge) (km)	rature (°C)	ge (years)	User Type
Battery Capacity (kWh)	1.00	0.00	-0.02	0.02	-0.00	-0.07	-0.01	-0.04	-0.04	0.04	-0.01
Energy Consumed (kWh)	0.00	1.00	0.03	-0.04	-0.01	-0.01	0.00	-0.03	-0.02	0.02	0.02
Charging Duration (hours)	-0.02	0.03	1.00	-0.01	0.02	-0.03	0.01	0.02	-0.05	0.00	-0.03
Charging Rate (kW)	0.02	-0.04	-0.01	1.00	-0.00	-0.02	-0.02	0.03	0.01	-0.05	-0.05
Charging Cost (USD)	-0.00	-0.01	0.02	-0.00	1.00	-0.01	-0.06	-0.01	0.06	-0.01	-0.02
State of Charge (Start %)	-0.07	-0.01	-0.03	-0.02	-0.01	1.00	-0.02	-0.02	0.04	0.02	0.00
State of Charge (End %)	-0.01	0.00	0.01	-0.02	-0.06	-0.02	1.00	0.03	0.02	0.02	-0.05
en (since last charge) (km)	-0.04	-0.03	0.02	0.03	-0.01	-0.02	0.03	1.00	-0.02	-0.03	-0.02
Temperature (°C)	-0.04	-0.02	-0.05	0.01	0.06	0.04	0.02	-0.02	1.00	0.03	-0.04
Vehicle Age (years)	0.04	0.02	0.00	-0.05	-0.01	0.02	0.02	-0.03	0.03	1.00	-0.00
User Type	-0.01	0.02	-0.03	-0.05	-0.02	0.00	-0.05	-0.02	-0.04	-0.00	1.00

Using Pearson’s correlation, we can observe that this are the 3 most correlated variables:

```
Top 3 Correlated Variables with 'Charging Duration (hours)' (Pearson's Correlation):
Energy Consumed (kWh)                         0.028369
Distance Driven (since last charge) (km)      0.023644
Charging Cost (USD)                           0.015902
Name: Charging Duration (hours), dtype: float64
```

We can notice how they have a very insignificant correlation degree.

Here, we see correlation between different categorical variables and our target variable:

```
Top 5 Chi-Squared Test Results (p-values) for Categorical Variables:
Charging Station ID: 0.33777825518594745
Day of Week: 0.4788747699833128
Charging Station Location: 0.4818864839589379
Vehicle Model: 0.4818864839589414
Time of Day: 0.48356511080907866
```

We can see that they are more strongly correlated to **Charging Duration** than numerical variables.

1.3 Handling Missing Values

Identification of rows with missing values:

```
Rows with missing values:
       User ID  Vehicle Model  Battery Capacity (kWh) Charging Station ID  ... Temperature (°C) Vehicle Age (years)     Cha
21     User_22   Hyundai Kona               62.000000         Station_485  ...        -2.755069            5.000000
28     User_29         BMW i3               75.000000         Station_129  ...        20.378562            3.000000
30     User_31     Chevy Bolt              100.000000         Station_239  ...         8.352906            2.000000  DC Fas
44     User_45    Nissan Leaf               75.000000          Station_44  ...        32.014577            3.680053
48     User_49    Nissan Leaf               85.000000         Station_463  ...        -4.446671            1.000000
```

Different methods were applied to handle this phenomenon. For example, row deletion; imputing missing values with average for numerical columns or with the mode, for categorical columns. Here is the code fragment that computes this process:

```
# a) Impute missing values for numerical columns with the mean
numerical_columns = df.select_dtypes(include=['float64', 'int64']).columns
imputer_mean = SimpleImputer(strategy='mean')
df_imputed_mean = df.copy()
df_imputed_mean[numerical_columns] = imputer_mean.fit_transform(df[numerical_columns])
print(df_imputed_mean.head())

# b) Impute missing values for categorical columns with the mode
categorical_columns = df.select_dtypes(include=['object']).columns
imputer_mode = SimpleImputer(strategy='most_frequent')
df_imputed_mode = df.copy()
df_imputed_mode[categorical_columns] = imputer_mode.fit_transform(df[categorical_columns])
print(df_imputed_mode.head())
```

1.4 Initial Conclusions

Missing values were identified and handled by removing rows with missing data, and imputing missing values in numerical columns with the mean and in categorical columns with the mode.

Key features (mostly categorical) like **Time of Day, Charging Station Location**, and **Vehicle Model** show potential correlations with **Charging Duration**, indicating they could be useful predictors. The numerical showed weaker correlations, suggesting it may be less relevant.

Some values deviated from expected ranges, but no extreme outliers were found. Further investigation is required to determine their impact on model performance.

In conclusion, the dataset is ready for further analysis, with key features identified, missing values handled, and additional steps required for outlier management and feature refinement.

Phase 2: Model Training and Improvement

2.1 Picking Metrics for Model Evaluation

This model will be a regression model. Therefore, the metrics chosen for it are:

- **Mean Absolute Error (MAE)**
- **Mean Squared Error (MSE)**
- **R-squared**

2.2 Data Splitting

The process of creating a regression model to predict a variable includes three sub-processes. The model must be trained to learn how to predict the target variable. Also, there is a phase where the hyperparameters must be tuned to prevent overfitting. Finally, the model must be tested to address its effectivity.

Therefore, our dataset needs to be split into three sets. This is the proportion chosen for each of the sub-sets:

- **Training set**: 80% of dataset
- **Validation set**: 10% of dataset
- **Test set**: 10% of dataset

This is the code fragment that splits the dataset into three sub-sets:

```
# Splitting data (80 - 10 - 10)
train_data, temp_data = train_test_split(df, test_size=0.2, random_state=42)
val_data, test_data = train_test_split(temp_data, test_size=0.5, random_state=42)
```

2.3 Training Base Models

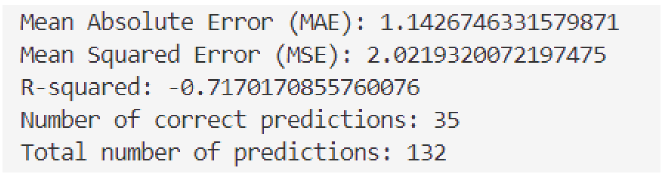
The selected regression models for the target variable prediction are the following:

- **Decision Tree Regressor**
- **Support Vector Regressor (SVR)**
- **Linear Regressor**

Decision Tree Regressor

A Decision Tree Regressor is a model used for predicting continuous values by recursively splitting the data into subsets based on feature values, forming a tree-like structure. Each internal node represents a decision rule, while the leaf nodes provide the final predictions. This model is easy to interpret and does not require feature scaling. However, if the tree grows too deep, it may overfit the data. Proper tuning of hyperparameters can enhance performance and mitigate overfitting.

Having trained the model, and seeing how it predicts our target value, we can observe it has a bad performance. It does not manage to predict our target value efficiently:

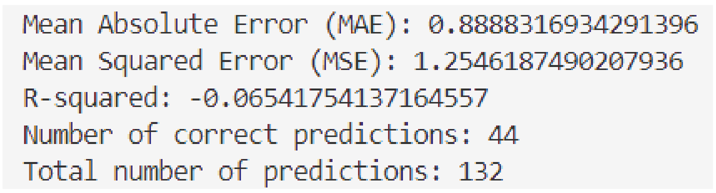


```
Mean Absolute Error (MAE): 1.1426746331579871
Mean Squared Error (MSE): 2.0219320072197475
R-squared: -0.7170170855760076
Number of correct predictions: 35
Total number of predictions: 132
```

With a **Mean Absolute Error (MAE)** of 1.14 and a **Mean Squared Error (MSE)** of 2.02, the predictions are fairly inaccurate. Additionally, the **R-squared** value of -0.72 indicates that the model performs worse than a simple mean-based prediction, suggesting it is not capturing the underlying patterns well. Only 35 out of 132 predictions were deemed "correct" based on the set threshold, further showing poor predictive performance.

Support Vector Regressor (SVR)

SV regressor is a robust technique for predicting continuous values. It works by identifying the hyperplane that best fits the data while maximizing the margin between points. SV regressor can handle both linear and non-linear relationships using kernel functions (e.g., linear, polynomial, or radial basis function). Its main strengths are good performance with high-dimensional data and resistance to outliers. Let's see how it performs with our data set.

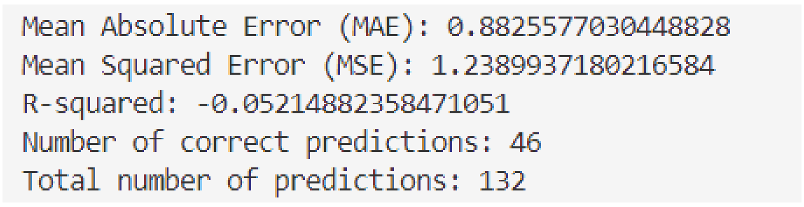


```
Mean Absolute Error (MAE): 0.8888316934291396
Mean Squared Error (MSE): 1.2546187490207936
R-squared: -0.06541754137164557
Number of correct predictions: 44
Total number of predictions: 132
```

We can see it worked a bit better, but results are still very poor. With a high MAE of 0.89 and a negative R-squared value. This suggests that the model struggles to fit the data and is not providing meaningful predictions for the target variable.

Linear Regressor

Linear regression is a statistical technique used to understand the connection between a dependent variable and one or more independent variables. It assumes a straight-line relationship, where the dependent variable is predicted as a sum of the independent variables, each multiplied by a weight. The objective is to find the line that best matches the data by minimizing the gap between the predicted and actual outcomes.



```
Mean Absolute Error (MAE): 0.8825577030448828
Mean Squared Error (MSE): 1.2389937180216584
R-squared: -0.05214882358471051
Number of correct predictions: 46
Total number of predictions: 132
```

The poor performance of the models, as indicated by the negative R-squared values and high error metrics, suggests that the dataset lacks meaningful relationships between the features and the target variable. This implies that the input variables in the dataset might not be strongly correlated with the charging duration. It's highly likely that the features do not contain enough relevant information to effectively predict the target, and the model is essentially unable to identify any useful patterns. This could be due to either irrelevant features or the absence of key variables that would provide a stronger connection to the target. Without a solid link between the features and the target, any predictive model will struggle to provide accurate predictions.

2.4 Feature Importance and Model Evaluation

Eventhough the models performed with a low effectivity, there are some features that helped determine our target value better than others. Here we can see a functionality added to the Decision Tree Regressor that displays the 5 most meaningful features that helped making the few correct predictions that were made:

Code functionality:

```
print("\n5 most influential features:")
for feature, importance in zip(top_5_features, top_5_importances):
    print(f"{feature}: {importance}")
```

Output:

```
5 most influential features:
Charging Rate (kW): 0.11974566574582998
Charging Station ID: 0.09161341107931711
State of Charge (End %): 0.09091885005136048
Energy Consumed (kWh): 0.08391363264450814
Temperature (°C): 0.07681506866995753
```

As we can see, their influence was fairly low.

Evaluation:

The performance of the models Linear Regression, SVR, and Decision Tree was generally disappointing when predicting Charging Duration (hours). While Linear Regression yielded the best results among the three, all models demonstrated significant shortcomings, with poor accuracy and predictive power.

The most influential features, such as Charging Rate (kW) and Charging Station ID, had relatively low importance values, further highlighting the challenge. Despite some correlation between these features and the target, the lack of strong relationships within the dataset likely limited the models' effectiveness.

The results suggest that the features available in the data are not strongly correlated with the target, making it difficult for the models to identify meaningful patterns. This issue appears to stem from the nature of the data itself rather than flaws in the models.

2.5 Model Improvement

Hyperparameter Tuning:

Grid Search was used for the Support Vector Regressor in order to find optimal parameters. Performance slightly improved as we can see here:

```
Best Hyperparameters: {'C': np.float64(66.35222843539819), 'epsilon': np.float64(0.16585553804470549), 'gamma': 'auto', 'kernel': 'rbf'}
Mean Absolute Error (MAE): 0.77425479001039
Mean Squared Error (MSE): 0.8848721787856347
R-squared: -0.0014134843685362775
Number of correct predictions: 12
Total number of predictions: 33
```

Accuracy has moved from 33% to 36%, but it did not show important improvements. The results indicate that the model struggles to find a meaningful pattern in the data. The **R^2 score (-0.0014)** suggests that the predictions are no better than simply using the mean of the target variable. The **MAE (0.77 hours)** and **MSE (0.88)** show that the errors are relatively high, and with only **12 out of 33 correct predictions**, the model lacks predictive power. This likely reflects a lack of correlation in the data rather than an issue with the model itself.

Scaling and Normalizing

Numerical features have been pre-scaled using StandardScaler, which ensures that they have a mean of 0 and a standard deviation of 1 before being fed into the SVR model. This have been done with all 3 models. Only SVR has increased by 3 it's correct predictions:

```
scaler = StandardScaler()
df_features = scaler.fit_transform(df_features)
```

```
Mean Absolute Error (MAE): 0.9116274109749789
Mean Squared Error (MSE): 1.31630653312746
R-squared: -0.11780257652824999
Number of correct predictions: 47
Total number of predictions: 132
```

Also, MinMaxScaler() has been used for normalization, obtaining the same results (47 correct predictions):

```
scaler = MinMaxScaler()
df_features = scaler.fit_transform(df_features)
```

Feature Selection

Feature selection techniques have been applied to the Decision Tree regressor in order to potentially improve performance. However, performance was not improved, but all the way around. Here we can see the results for each of the feature selection "improvements":

```
Variance Thresholding Results:
MAE: 1.2014248308954536, MSE: 2.178683962667359, R2: -0.850130258887452
Correct predictions: 36/132

SelectKBest Results:
MAE: 1.2014248308954536, MSE: 2.178683962667359, R2: -0.850130258887452
Correct predictions: 36/132

R_regression Results:
MAE: 1.279337497558936, MSE: 2.4902896785754107, R2: -1.114744665438621
Correct predictions: 25/132

SelectFromModel Results:
MAE: 1.2963020274699475, MSE: 2.5009361802251626, R2: -1.1237856347535953
Correct predictions: 30/132
```

The feature selection techniques applied (Variance Thresholding, SelectKBest, R_regression, and SelectFromModel) did not improve the model's performance. The results showed that all methods produced similar metrics, with negative R2 values indicating poor model fit. This suggests that the features in the dataset are not strongly related to the target variable. Feature selection typically improves performance when the features are informative, but in this case, the data seems to lack relevant relationships with the target, limiting any potential improvement. Therefore, no improvements can be made through feature selection.

Adding New Features

For the Linear and SVR models, a new feature was implemented called Charging Time (hours). This new variable was the difference between the Charging End Time and the Charging Start Time passed to hour format:

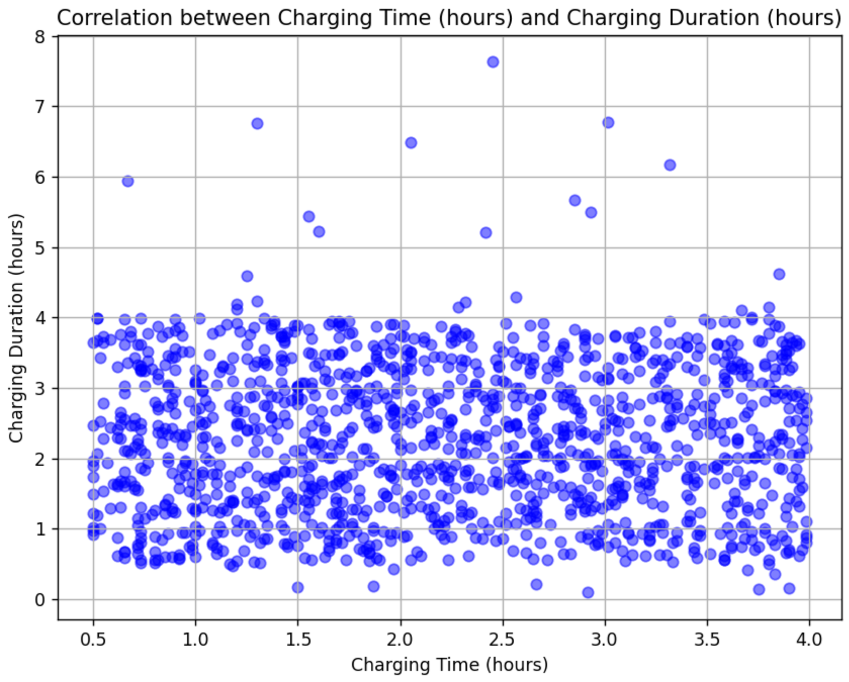
```
df = pd.read_csv(file_path)

df['Charging Start Time'] = pd.to_datetime(df['Charging Start Time'])
df['Charging End Time'] = pd.to_datetime(df['Charging End Time'])

df['Charging Time (hours)'] = (df['Charging End Time'] - df['Charging Start Time']).dt.total_seconds() / 3600
```

pd: our target variable is Charging Duration (hours), which is not the duration of the charge itself, but the "driving time duration". The new feature is the duration of the charging itself.

This new feature was added with the intention and idea of improving the model's performance (as this two variables could be potentially correlated). But the results showed us that there was not any linear correlation between both variables, as we can notice in this scatter plot:



The plot shows us a random point positioning, with no correlation at all.

2.6 & 2.7 Dataset Issues and Model Performance

Despite applying various model improvement techniques, including hyperparameter tuning, feature engineering, and scaling, the models failed to make accurate predictions. The primary issue is that there is no meaningful correlation between the features and the target variable, **Charging Duration (hours)**. Without this correlation, the models couldn't identify patterns or make useful predictions.

Attempts to improve the dataset through new features like Charging Time (hours), imputation strategies, and scaling did not help. Hyperparameter tuning also had no effect, as the models couldn't learn anything meaningful from the data.

In conclusion, the dataset lacks the necessary relationships for effective machine learning, and without addressing these fundamental issues, model performance cannot be improved

Despite this problem, the model that performed the best was the SVR model when the dataset was scaled, and when hyperparameters were tuned using grid search.