Charger Study: A Data Science Approach to Predict the Outcome of an EV Charger Getting Certified

Ayushi Nayak[1], Kritee Saxena[2]

Abstract: ***Charger study aims to predict results of certification of Electric Vehicle (EV) DC fast chargers accurately by applying machine learning techniques to historical data. The historical data consists of rows where each row consists of several statistics for both the EV-maker charger and a 3rd party charger. The historical data is generated using web scraping libraries such as Selenium and Beautiful Soup. Based on the scraped data, data cleaning and feature engineering is done to generate several features of a charger like connectors used, power rating, voltage rating, current rating etc. Finally, the features are represented in a vector format and fed as inputs to different Machine Learning classifier algorithms like Multinomial Logistic Regression, SVM, Gradient Boosting Classifier and DecisionTreeClassifier. After the classification, accuracy is measured by calculating percentage of correct predictions and percentage of correct no-certification for mistake correction predictions. Error analysis is performed using techniques like Region under Curve to tune hyper-parameters and to identify the features which are more prominent/useful in accurately predicting the results.***

1. Introduction

EV Supply Equipment industry panders to a fast growing market. While the production of EVs is a tedious and time taking endeavor with space for improvement, once an EV is launched in the market there is a deficiency of viable charges. This results third party companies like PlugIn India and Charge Point, in producing chargers to suit the EV type not unlike a smart phone USB Type Micro and C charger manufactured by companies other than the phone manufacturer. Instead of producing their own product, EV companies certify them in compliance to their vehicles. In this project, we try to predict the results of certification of different chargers of an EV by refining historical data, performing data analysis, feature engineering and finally evaluating the performance of different machine learning models.

2. Related Work

Not many previous works have attempted to predict the results of certification of EV chargers. So, we decided to take reference from predicting results from a similar field: the outcome of a football match. Ben Ulmer and Matthew Fernandez of Stanford University[1] used game day data and current team performance achieving error rates of linear classifier (:48), Random Forest (:50), and SVM (:50). Another work that we went through was by Timmaraju et al[2]. They were able to incorporate features such as corner kicks and shots attempted which is why they were able to obtain an accuracy of 60% using a RBF-SVM. Joseph et al used another approach to the problem. They used Bayesian Nets to predict the results of matches played by Tottenham Hotspur during 1995-1997. Their results show huge variations in accuracy (38%-59%). However, their approach provides a different insight into feature selection. Finally, we took further inspiration from the famous Kaggle[3] competition called March Madness[4] where different approached like converting the dataset into feature vectors and then evaluation of different machine learning models are performed.

3. The Data Set

Data forms the most integral part for any Machine Learning Algorithm. It is the data that trains the Algorithm to make predictions. The more accurate data fed, the more accurate are the results. Hence, collecting and preparing a data set forms a crucial part.

For our project, we decided to build the dataset over the period of 2013 to 2017. We wanted to incorporate features like connectors used, power rating, voltage rating, current rating etc. We also wanted to take into account the ratings of the charger makers as usually better rated makers have better chances of certification. The reason for choosing 2013 wasdue to the availability of all the statistics like connectors used, power rating, voltage rating, current rating etc. for EV manufacturers like Nissan and Mitsubishi on Amazon[5] for every certified charger from 2013 onwards. Thus, having a dataset which consisted of features that could describe a charger was the primary motivation of including the features that have been incorporated. We also introduced a feature of EV-maker charger as it is usually seen that the home EV-maker charger has an advantage over a 3rd party Original Equipment Manufacturer (OEM).

4. Methodology

Here are the steps that we followed while collecting and preparing the data.

4.1 Feature Selection

In order to generate the data set, the first thing we had to decide upon is what features we need for our problem i.e. which all features are relevant to the problem of accurately predicting the result of a certification. Statistics such as connectors used, power rating, voltage rating, current rating etc.come to the mind as these are important aspects of an EV charger. For obtaining all such statistics relevant to a certification, we decided to scrape the data from the websiteof Renault Group [6] as it contained an authentic and detailed summary for every charger certified. Also, the ability of the charger is another thing which has a huge influence on the outcome of the certification. For this, we decided to build a feature called charger rating for each

OEM. We decided to use the ratings given by Amazon. In order to calculate the charger rating for an OEM, we fetched the ratings of each charger manufactured by the OEM and then calculated the effective rating of the charger from that OEM using the durability of strokes by an individual charger socket (M_i). Total durability of an individual charger (Tt) and the Charger Rating (Pr). We calculated the Effective Rating (ERi) of a charger as follows:

$$Eri = \frac{Mi}{Tt} * Pr \qquad - (1)$$

Using the individual ERi obtained from (1), the total Charger Rating (Tr) was
calculated as:

$$Tr = \sum_{1}^{n} Eri$$

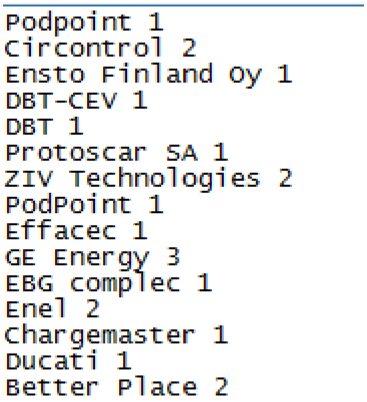
where n is the number of chargers of an OEM.

For fetching the Renault certification of each charger, we used the Renault certified charger website. This way we could calculate the effective charger ratings for all the chargers for a given OEM.

4.2 Data Collection

This step involves in collecting data from various sources that is relevant to the problem statement. We adopted a method known as Scraping. Data scrapping or web scrapping is a method that is used to collect data from web pages into a readable form for e.g. in the form of a spread sheet or notepad etc., for further data analysis.

To collect data, we started off with BeautifulSoup4, a python frame-work which is used for pulling data out of HTML or XML files. This gave us incomplete data as BeautifulSoup4 is a library that extracts data from static web pages (i.e., pages that are purely designed from HTML and XML), and the website we were surfing was a dynamic one. Hence, the data that we collected was incomplete. We then used another framework Selenium to extract information from the dynamic pages. Selenium is a framework that creates a version of the web browser which is controlled by python. Thus, allowing us to extract information from dynamic pages.



```
Podpoint 1
Circontrol 2
Ensto Finland Oy 1
DBT-CEV 1
DBT 1
Protoscar SA 1
ZIV Technologies 2
PodPoint 1
Effacec 1
GE Energy 3
EBG complec 1
Enel 2
Chargemaster 1
Ducati 1
Better Place 2
```

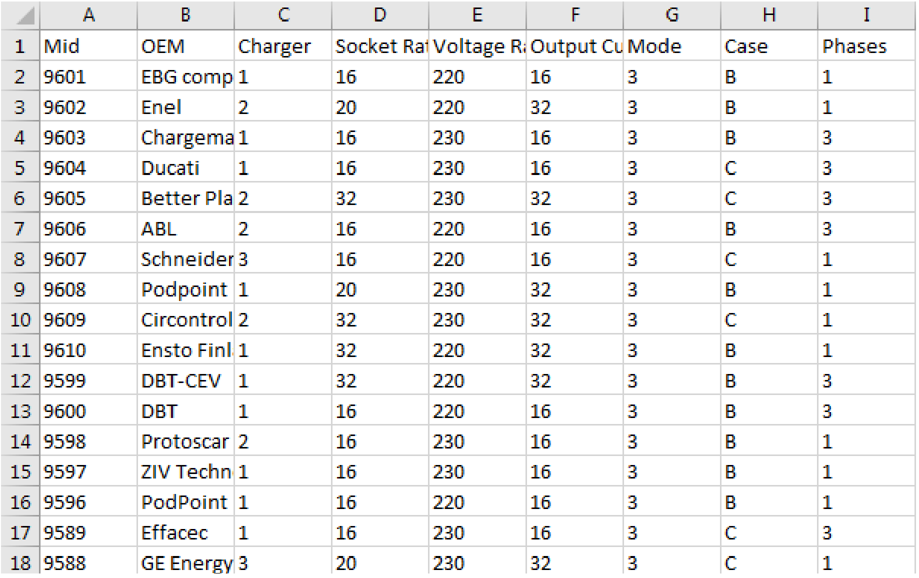
Figure 1: Data obtained after Scraping

4.3 Data Preprocessing

Preprocessing is basically arranging the data in a format in which the data can be visualized. As you can see in the figure 1, the data that we had obtained after scraping, may not be in a format that could be fed to the Machine Learning Algorithms. Hence, the data has to be organized to make it easier to work with. We proceeded with data processing in the following order.

1) Formatting
After scrapping, the data present with us was haphazard. It was not in a format that could be accepted by a Machine Learning Algorithm (as, all machine learning Algorithms accept data in a vector format). We chose to arrange the data, obtained, in a .csv file to visualize the data in a better way. Organizing that data in a .csv file helped us to arrange and segregate the data into rows and columns based on the features selected by us.

2) Data Cleaning
This is the step in which we add or remove data based on the problem requirement. After the data was arranged in a .csv file we came across several records that were redundant and some important records that were



	A	B	C	D	E	F	G	H	I
1	Mid	OEM	Charger	Socket Rat	Voltage R	Output Cu	Mode	Case	Phases
2	9601	EBG comp	1	16	220	16	3	B	1
3	9602	Enel	2	20	220	32	3	B	1
4	9603	Chargema	1	16	230	16	3	B	3
5	9604	Ducati	1	16	230	16	3	C	3
6	9605	Better Pla	2	32	230	32	3	C	3
7	9606	ABL	2	16	220	16	3	B	3
8	9607	Schneider	3	16	220	16	3	C	1
9	9608	Podpoint	1	20	230	32	3	B	1
10	9609	Circontrol	2	32	230	32	3	C	1
11	9610	Ensto Finl	1	32	220	32	3	B	1
12	9599	DBT-CEV	1	32	220	32	3	B	3
13	9600	DBT	1	16	220	16	3	B	3
14	9598	Protoscar	2	16	230	16	3	B	1
15	9597	ZIV Techn	1	16	230	16	3	B	1
16	9596	PodPoint	1	16	220	16	3	B	1
17	9589	Effacec	1	16	230	16	3	C	3
18	9588	GE Energy	3	20	230	32	3	C	1

Figure 2: Data arranged in the .csv format

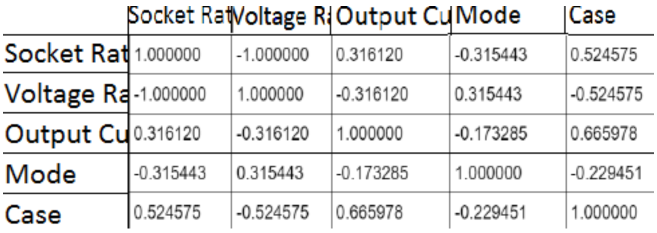
incomplete. Such situations had to be specially handled. In our case, as we had extracted data from one web page, we found that the certification of a few chargers was missing. To handle this situation, we had to plug out data from other web pages to fit into the incomplete records.

3) Sampling

Sometimes, there might be situations where we have a data set with large number of features. Large data sets might sometimes slow down the performance of the algorithms to which the data is fed. There might be cases where the features selected by us are related to each other. In this case, it is sensible to drop some as it can improve the performance significantly.

On analyzing the data obtained, we noticed that there were several features that were similar. To confirm our intuition, we analyzed features by comparing two at a time and visualizing it using plots. Comparing two features at a time got tedious as we had 33 features. This left us with 1122 possibilities to be compared and comparing them manually was next to impossible. Moreover, we ran into few errors while making our analysis. We had to search for other methods to help us getting sent back conclusions.

On further research, we found a technique to analyze features automatically. This could be done by the Correlation Matrix. This matrix finds the correlation between every feature pairwise. In other words, it compares every feature with every other feature present in a data set. There are many Correlation Matrices out of which we chose the Pearson's Correlation Matrix. The Pearson's Correlation quantifies the degree to which a relation can be established between two variables. The correlation compares all the features and gives an output from a scale of -1 to 1. When the output is 1, it implies that the increase in one variable leads to the increase in the other. An output of -1 indicates that, the increase in one variable leads to the decrease in the other. An output of 0 indicates that the variables are independent.



	Socket Rat	Voltage R	Output Cu	Mode	Case
Socket Rat	1.000000	-1.000000	0.316120	-0.315443	0.524575
Voltage Ra	-1.000000	1.000000	-0.316120	0.315443	-0.524575
Output Cu	0.316120	-0.316120	1.000000	-0.173285	0.665978
Mode	-0.315443	0.315443	-0.173285	1.000000	-0.229451
Case	0.524575	-0.524575	0.665978	-0.229451	1.000000

Figure 3: Pearson correlation for the feature set

At first, we got several numbers in a matrix format ranging from 1 to 1, as shown in figure 3. It was difficult for us to decipher the correlation. Using Pandas, the matrix could be highlighted with colors. This helped us to visualize the high and low correlations in a better way. As one can see in figure 4, as the shade for a cell becomes darker it indicates an inverse correlation and as the cells become lighter, it indicates that the two variables are strongly correlated.

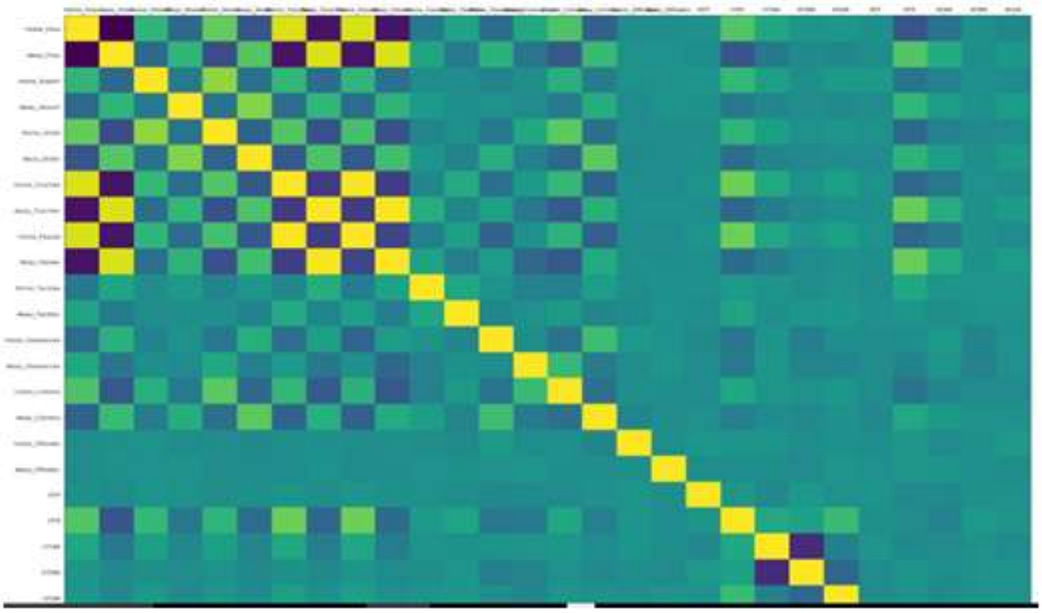


Figure 4: Color Visualization of the matrix shown in figure 3

4.4 Feature Engineering

This is the step in which we modify features set, of the raw data, to represent the data, in a better format, to the predicting models. At first, we chose to decompose the feature named as Total Rating into Standard Charge rating (typically at home), Fast Charge rating (public charge station) and Rapid Charge Rating (high power public charge station) as in an EV charger, the Total Rating may not always reflect the true nature of how it gets certified. There have been instances where an OEM with very high Standard Charge Rating was certified against OEM which had Total Rating more than them. Thus, breaking the total charger rating into 3 sub categories helped us gain more insight into a particular OEM.

4.5 Data Transformation

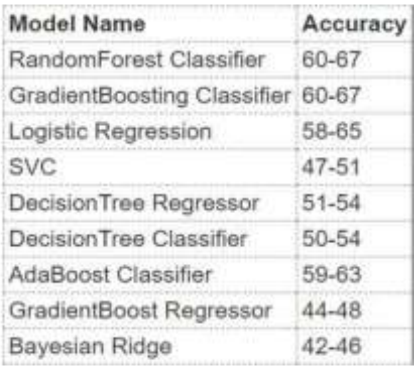
The next step is important as it involves in transforming the dataset into feature vectors, where each charger of an OEM is represented using a feature vector. The feature vector consists of all 33 features, having a dimension of 1x33 suitable to be fed to the Machine Learning Algorithms. We also had two fields named EV Maker charger and 3rd party OEM charger, to distinguish between the two, we assigned 1 to the EV Maker charger and -1 to the 3rd party OEM charger.

5. Results and Evaluation

We implemented several machine learning algorithms with our data set and we got the best accuracy with the Random Forest Classifier and the Gradient Boosting Classifier. The following are the results obtained:

5.1 Results with the Random Forest Classifier

We predicted the certification of the year 2017 using 11 different machine learning models. After evaluating and tuning these 11 models, we found that Gradient Boosting Classifier and Random Forest Classifier gave us the best results, with accuracy ranging from 60% to 67% for both of the models. The models along with their accuracies can be found in the figure 5.

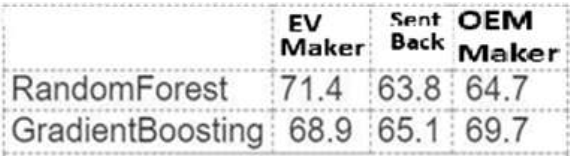


Model Name	Accuracy
RandomForest Classifier	60-67
GradientBoosting Classifier	60-67
Logistic Regression	58-65
SVC	47-51
DecisionTree Regressor	51-54
DecisionTree Classifier	50-54
AdaBoost Classifier	59-63
GradientBoost Regressor	44-48
Bayesian Ridge	42-46

Figure 5: Results of the output for various Machine Learning Algorithms

Due to higher accuracy of Gradient Boosting Classifier and Random Forest Classifier, we focus on these two models for further evaluation. One common problem faced in other works was poor accuracy in predicting charger not certified.

Since our dataset had features which were specific to a particular charger, we found that our accuracy in correctly predicting certification, getting sent back or not compatible was roughly equal. The accuracies for EV Maker Charger, getting sent back and 3rd party OEM charger can be found in the tables below for both the models.



	EV Maker	Sent Back	OEM Maker
RandomForest	71.4	63.8	64.7
GradientBoosting	68.9	65.1	69.7

Figure 6: Results for the chargers given by the Gradient Boosting and Random Forest Classifier

Further, we also were interested to know which features had the highest contribution or weightage while predicting the results for RandomForest Classifier, we found out that Voltage Rating was the most important feature followed by type of connector, while for GradientBoosting Classifier, Charger Rating was the most important feature followed by type of connector.

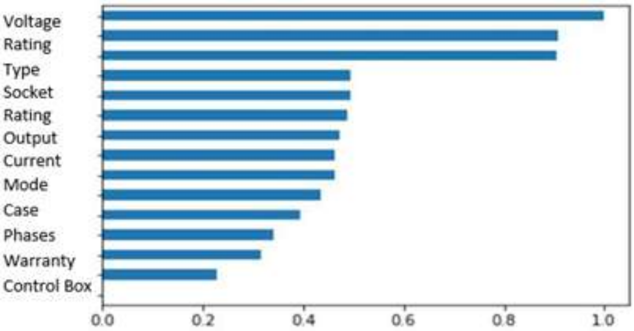


Figure 7: Feature importance for the Random Forest Classifier

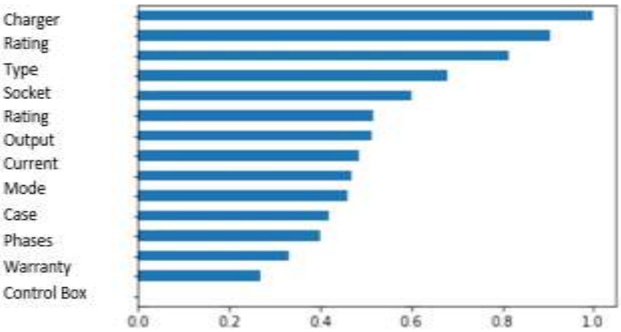


Figure 8: Feature importance for the Gradient Boosting Classifier

6. Future Work

Our model performs comparatively well in predicting charger certification for the time period chosen. However, it remains to be seen how it would performs over a longer time period. The accuracy obtained was for a relatively short period of time (4 years). Also, the accuracy of the models can be further improved by having more accurate data in terms of past statistics of AC slow chargers. Other aspects which can be taken into account are home installation and number of public chargers in an area, which could have a significant impact on the accuracy of the models.

References

[1] Ben Ulmer and Matthew Fernandez, Predicting Soccer Match Results in the English Premier League.
[2] S. Timmaraju, A. Palnitkar, & V. Khanna, Game ON! Predicting English Premier League Match Outcomes, 2013.
[3] Kaggle March Machine Learning Mania - https://www.kaggle.com/c/march-machine-learning-mania-2017
[4] Adit Deshpande, Applying Machine Learning to March Madness - Applying Machine Learning To March Madness
[5] Amazon site- amazon.com
[6] Renault Group, Renault Motors, Renault Zoe-Renault.com