# Machine Learning Engineer Nanodegree

# Capstone Project Report

## Salvador Núñez

## February 24, 2018

# Project Overview

Advancements in the electric automobile industry are contributing to the disruption of the energy industry. Electric utilities are increasingly interested in identifying the load from electric vehicles (EVs) in order to develop a better informed investment strategy and create incentives for residential customers to charge their EVs during certain times in the day. This project analyzes the electricity consumption of houses to identify a) if there is an electric vehicle (EV) charging at each house and b) during which time intervals the EV is charging at the house. A [dataset](https://www.gridcure.com/wp2/wp-content/uploads/2016/05/EV_files-2.zip) consisting of 1,590 houses has been downloaded from GridCure’s [website](https://www.gridcure.com/)[[1]](#footnote-1), containing labels to denote which houses are charging their EVs and when. The website also contains a separate unlabeled test dataset with an additional 699 houses. The goal of this project is to create a series of data transformations and machine learning models that correctly predict which houses have EVs and when the EVs are charging. The performance of the models are benchmarked against a validation data set separated from the original labeled data set. Finally, these transformations and model are used to submit predictions for the 699 test houses.

# Problem Statement

This project will be approached as a supervised learning classification problem. The first strategy used to solve this project is to find ways to decrease the imbalance of the classes. While 31% of the houses in the training dataset are labeled at charging during at least 1 time interval, that corresponds to only 2% of the intervals or data points yielding a positive label. Training a model to accurately predict a positive label when it only occurs 2% of the time in the training data set will be very difficult. To help with this, first houses that have EVs are predicted. Then, using only that subset, time intervals when EVs are charging are predicted. By doing so, the prevalence of a positive label is increased from 2% to 8%. This four-fold increase in a positive label will make the training much easier.

The second strategy used is to augment the provided training data with implicit temporal information. The data contains one categorical variable for the 1590 “House IDs” and 2880 continuous variables consisting of two months of energy (kWh) data taken at 30 minute intervals (i.e. 2x24x60 + 1 =2,881). However, we know that means there are 60 24 hour cycles in the data, but those cycles are not represented in the dataset. Therefore, the day and the time of day corresponding to each interval are explicitly added to the data.

Using the daily cycles in energy consumption, consumption archetypes can be defined. Companies like Opower Inc, have developed [load curve archetypes](https://blogs.oracle.com/utilities/load-curve-archetypes)[[2]](#footnote-2) derived from unsupervised learning techniques like k-means clustering. Load curve archetypes are also applied here with the objective of creating clusters that effectively discriminate between EV houses and non-EV houses.

Finally, Gradient Boosting Decision Trees (GBDT) to classify the houses and the time intervals. A GBDT model was applied since they [often perform well](https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/)[[3]](#footnote-3) on imbalanced datasets.

# Metrics

The performance of the model(s) developed for this project will be evaluated based on the F1 score of the trained model on the validation dataset. The F1 score is a better evaluation metric than accuracy when considering a model trained on uneven class distribution. The F1 score is calculated as the harmonic mean of precision and recall.



{\displaystyle F\_{1}=2\cdot {\frac {1}{{\tfrac {1}{\mathrm {recall} }}+{\tfrac {1}{\mathrm {precision} }}}}=2\cdot {\frac {\mathrm {precision} \cdot \mathrm {recall} }{\mathrm {precision} +\mathrm {recall} }}}



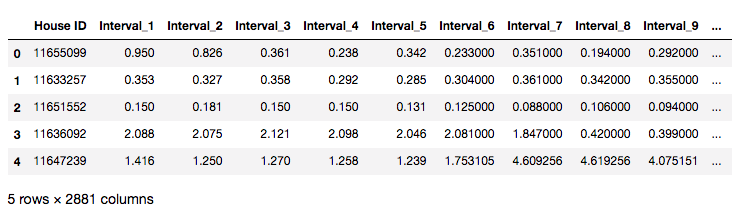


High recall attempts to minimize false negatives, whereas precision tries to minimize false positives. Accuracy works best if false positives and false negatives have similar cost. In this case, a false negative (incorrectly predicting a house or time interval is not charging an EV) is much more expensive than a false positive (incorrectly predicting that a house or time interval is charging an EV, when it isn’t).

# Analysis and Methodology

## Data Exploration

The “wide” data format for the data set does not conform to the traditional format for scikit learn with a matrix **X** and a target vector **y** corresponding to exactly 1 prediction per row. It also contains 4 houses missing between 48 and 144 data points.



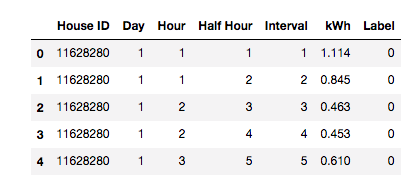
## Exploratory Visualizations

Preliminary exploration of the raw dataset shows a long tail distribution of the energy consumption of the 30-minute interval, as well as the distribution of energy consumption for houses over the entire 2 months. This indicates that there may be some outliers in the data, but the data doesn’t seem to be multimodal.

## Data Preprocessing

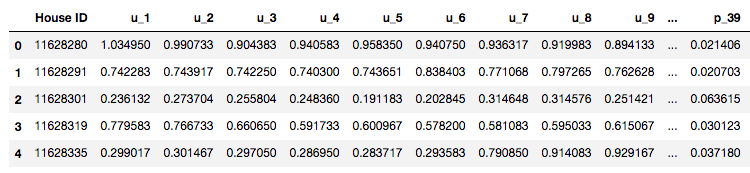
All training data set and the training labels are combined and pivoted to “long” format. Then, the implicit temporal dimensions are explicitly added with the assumption that sequence in the interval column names can be translated to 60 cycles of 48 30-minute intervals. With this temporal information, the missing values are filled for the 4 houses missing data points. These are filled with the average energy value for that house during that time of day. This results in the following data set without any null values:



## House Classification

The data is later aggregated at the house level and new labels are created at the house level. If a house has **any** time interval positively labeled for charging EVs, it is a positively labeled “EV house”. If not, it is a “non-EV house”. These house aggregations are further described with the mean and standard deviation of the kWh the house consumes every half hour. The means are also divided by the average total daily consumption to produce a metric for the average percent of daily energy spent in that half hour. This last summary statistic, the percent of daily energy spent, is particularly helpful in isolating the load shape of the house, irrespective of the total amount of energy the house may consume. These columns are named with the following convention, where “i” is an interval between 1 and 48, corresponding to all the half hour intervals in a day:

* u\_i – mean during the ith interval for that house
* s\_i – standard deviation value during the ith interval for that house
* p\_i – average percent of the daily consumption during the ith interval for that house

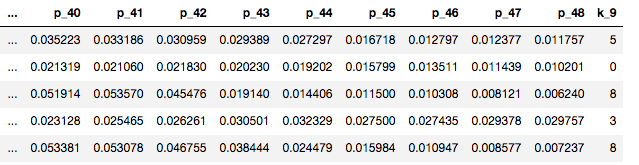


Before any additional processing, this data set aggregated at the house level is split between training and validation groups, using a test size of 0.25.

Then, an additional column is added to the separated training data set: a cluster assignment. The data set is clustered through k-means using only the columns describing the percent of daily consumption during the ith interval. The “optimal” k was selected based on the k that would maximize the ratio comparing the cluster with the highest percent of positive labels with the cluster with the lowest percent of negative labels, for each set of clusters produced by k ranging from 2 to 10.



In this case, the optimal “k” was 9 and the column “k\_9” was added to the aforementioned dataset, assigning a cluster from 0 to 8 to each house.

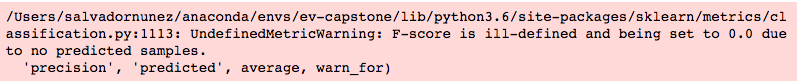


This modeling exercise also yield the centers for each of the 9 clusters in 48 dimensions, corresponding to each of the 48 intervals. These centers will be used later for time or interval classification.

Next, a Gradient Boosted Decistion Tree model (GBDT) is trained on this data set, resulting in a training F1 score of 0.981. However, this F1 score is on the training data set and has a fair degree of overfitting. Instead, an F1 score must be obtained from the validation data set and compared against the benchmark.

To do so, a k-nearest neighbors (KNN) algorithm is trained on the training data set and used to predict a cluster for the validation data set. This is because the validation data set was excluded from the k-means exercise in order keep the validation data set unexposed to any supervised or unsupervised training technique. A KNN model with n = 5 is fitted to the training data set on the mean (u), standard deviation (s), and percentage (p) columns to predict one of the 9 clusters. The fitted model is then used to predict a cluster for the validation data set. Once column “k\_9” is appended to the validation dataset using KNN, the trained GBDT model is used to predict EV houses in the validation data set. This yields a validation F1 score of 0.753.

Benchmarking this score against a naïve predictor that assumes that there are no EV owners in the dataset did not make too much sense because this benchmark yields 0.00.



Therefore, two more naïve predictors were used: (a) a naïve predictor that sequentially alternates between predicting 0 and 1, and; (b) naïve predictor that predicts that assumes that all houses are EV houses. These scores were 0.367 and 0.487, respectively. The model does a much better job at predicting EV houses than these benchmarks.

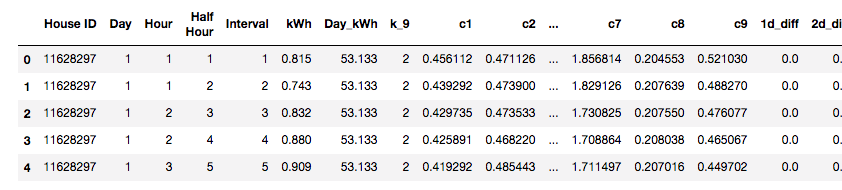
## Time Classification

In order to classify the time intervals when the EVs are charging, the data set at the “interval” level is also augmented with additional features while keeping the same training and validation split used for house classification.

Additionally, only the houses which are positively classified as EV houses are subsequently used to train the classifier at the interval level. A negatively labeled house implies that all the time intervals for that house are not EV-charging intervals. Furthermore, as discussed earlier, this increases the prevalence of positively labeled charging intervals in the data set, making it easier to train the classification model.

The features added to the interval-level data set are meant to capture information on each house, as well as information that describes differences in energy consumption at that time for that house compared to other days. Existing models to detect EV ownership, like the one describes by Fischer et al[[4]](#footnote-4)., have used features in the data such as increases/decreases in power load by certain pre-determined amounts (e.g. 1-2 kWh), a particular frequency of such increases/decreases, and temporal spacing between such events.

Therefore, the following features were added to the interval dataset: (a) the distance (or difference) between the energy consumption with respect to each of the 9 clusters defined by the house classification, and; (b) the difference in energy consumption compared to the same house at the same time in the past 7 days. In comparing the energy value with the cluster center, the unit for the former is kWh whereas the latter is % daily energy. Thus, to compare them, the percent daily energy use for each cluster in that half hour interval is scaled to the total daily energy consumption for that day. Then, the difference is taken and the columns are labeled c1 through c9. Comparing the energy values for the same house in the past 7 days is more straight forward. This comparison is done for 7 days to capture any weekly cycles hidden within the data. These columns are labeled 1d\_diff to 7d\_diff. However, the first 7 days for each house will be missing some values. Since many classification models do not accept missing or NA values in the training data, these missing values are filled with zero. Overall, this results in the following data set:



Compared to the raw data set that essentially would have only contained **2** columns, a House ID and a kWh value, this data set contains **24** columns, resulting in significantly more information that can be used to train a classification model. A separate GBDT model is trained on this dataset at the interval level. Once the time intervals are classified for the subset of data classified as EV-houses, the remaining records in the training data (all of which implicitly are predicted as zero) are combined into the data set. Overall, this yields a F1 score on the training data of 0.724.

The same data augmentation, classification, and recombination that was performed on the training data set is then performed on the validation data set. Overall, this yields a F1 score on the validation data of 0.568. The corresponding F1 benchmark scores are 0.000, 0.048, 0.050 for all 0, alternating 01, and all 1. The F1 score performs significantly better than the benchmarks, but it’s still not a great score.

# Refinement and Final Results

To refine and improve the results, GridSearch is used to optimize parameters for the GradientBoostingClassifier. For the house classification, shuffle split cross validation was used with an f1\_scorer, iterating over the 2\*3^4 (162) parameter combination. This resulted in an increase of the F1 score for the house data from 0.753 to 0.770.

The time classification is a much larger dataset that takes much longer to train. While the house training set consisted of 1192 rows and 146 columns, the time training data set consisted of 1,028,160 rows and 24 columns. Since the time training data set is several orders of magnitude greater than the house training data set, it will take much longer to train. The complexity of solving this grid search would quickly increase as the complexity and the number of operations would increase by O(n^2). Therefore, fewer parameter combinations were attempted in this grid search: only 3^2=9 combinations. This also resulted in a small increase of the F1 score for the house data from 0.565 to 0.586.

These optimized classifiers were then applied to the test data and transformed into the desired format for submission.

# Conclusion

By separating and sequencing the classification of EV houses and then EV time intervals, revealing implicit temporal data, using k-means clustering, K-Nearest Neighbors, and Gradient Boosting Decision Trees, time intervals where an EV is charging in a home were successfully classified. This approach was developed from intuition as an alternative to traditional nonintrusive appliance load monitoring (NIALM) to disaggregate EV loads from other appliances and therefore detect EV owners. The house classification F1 score was 0.753, 54% higher than 0.487, the highest house-level benchmark. The time classification F1 score was 0.586, 18x higher than 0.050, the highest time-level benchmark.

## 

According to the ranked feature importance for the trained house classifier, understanding the variance or standard deviation of the % of daily energy consumed per half hour (interval) were among the most important features. The House ID number itself was not excluded from the model and ranked as being the 6th most important feature.

## 

According to the ranked feature importance for the trained time interval classifier, the absolute value of the energy consumption was the most important feature. The House ID number was the 2nd most predictive feature, followed by daily difference in energy consumption compared to 3 days ago, 2 days ago, 4 days ago, and finally, 1 day ago. The interval number and the cluster assignment were the 7th and 8th most important features, respectively.

Originally, a few other models were tested with the data set including Support Vector classifiers (SVC), Random Forrest classifiers, and multilayer perceptron (MLP). Gradient Boosted Decision Trees (GBDT) outperformed all of these models with the default parameters. Therefore, refining the model focused on optimizing the parameters for the classifier. However, more can be done to improve the results. Further parameter tuning could be done by training the classifier with a more powerful machine (more cores, GPU, etc.) in AWS. Doing so on a laptop (3.1 GHz Intel Core i7, 16 GB 2133 MHz) is feasible but impractical due to time constraints. Finally, a completely different modelling approach can be pursued which applies more traditional ways of disaggregating time series data such as [harmonic feature analysis and multiple-class support vector machines](http://ieeexplore.ieee.org/abstract/document/6150170/)[[5]](#footnote-5) or [using hidden Markov modeling (HMM) and residual analysis](http://ieeexplore.ieee.org/abstract/document/6406788/).[[6]](#footnote-6) These alternative modeling approaches are likely to obtain better results but are outside of the scope of Udacity’s nanodegree program.

1. Optional Predictive Modeling Challenge. GridCure, <https://www.gridcure.com/contact/> [↑](#footnote-ref-1)
2. Fischer, Barry. “We plotted 812,000 energy usage curves on top of each other. This is the powerful insight we discovered.” *Opower, Inc.* October 13, 2014. <https://blogs.oracle.com/utilities/load-curve-archetypes> [↑](#footnote-ref-2)
3. 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset. <https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/> [↑](#footnote-ref-3)
4. Fischer, et al. Identifying Electric Vehicle Owners. United States Patent US 9.576,245 B2. United States Patent and Trademark Office. Feb. 21, 2017. [↑](#footnote-ref-4)
5. Jiang et al. “An Approach of Household Power Appliance Monitoring Based on Machine Learning”. Intelligent Computation Technology and Automation (ICICTA) – IEEE. Jan 14,2012. [↑](#footnote-ref-5)
6. Pattem, S. “Unsupervised Disaggregation for Non-intrusive Load Monitoring ”. Machine Learning and Applications (ICMLA) –IEEE. Dec 12,2012. [↑](#footnote-ref-6)