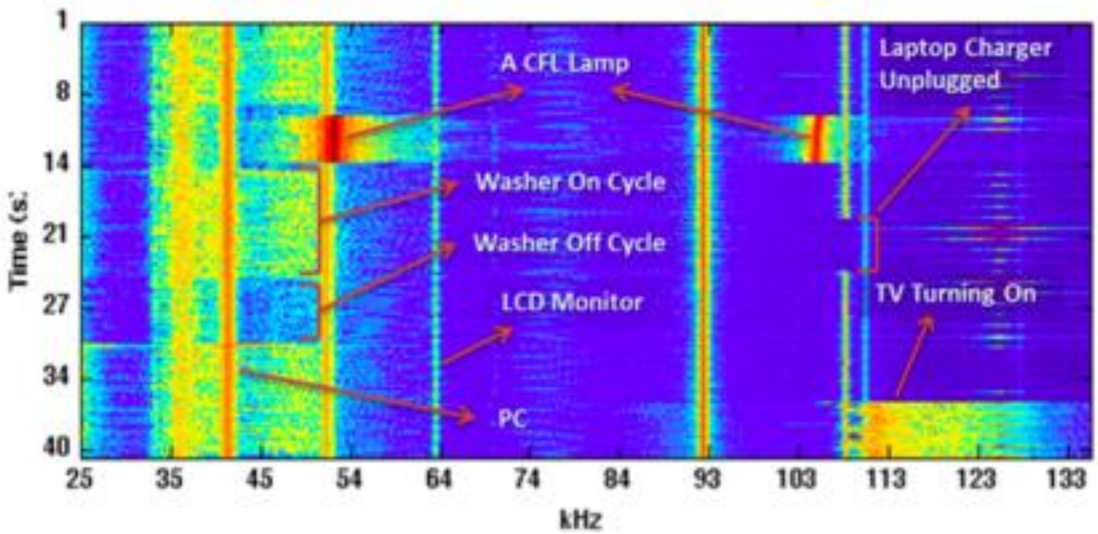


Energy Disaggregation Product Proof-of-Concept

Using Data Science to Develop a New Product

December 29, 2015

Prepared by:
Group 5
Jason Dilmanian
Adam Gonthier
Joe Morreale
Pavan Sudhir Nallam

Data source

All data from this project is from the kaggle.com competition, sponsored by Belkin. We also use several images and concepts from the competition in our write-up and project descriptions.
https://www.kaggle.com/c/belkin-energy-disaggregation-competition

Table of Contents

Appendices

Business Understanding

Product Introduction

As energy prices increase, and people become more concerned about global warming, consumers are more interested in understanding their own energy consumption, and where their energy costs are going. We seek to provide consumers with a tool that they can use to monitor their energy consumption in real time, and provide a breakdown of their electrical usage. This tool will provide consumers with a dashboard that they can view which will show them how long their appliances have been running, the total power consumption associated with that appliance, and the energy used by that appliance. The ultimate goal is to disaggregate household energy consumption into individual appliances, to give home-owners more insight into their electricity usage.

Using this dashboard, consumers will be better able to monitor their energy uses and make intelligent decisions about what appliances and tools they run at home. This will cover major appliances in the home, such as computers, air conditioning, televisions, radios, etc. By receiving this information, consumers will be able to take actions to reduce their energy bills, since they will have a customized breakdown for their own home, and will know when things are running. For example, a consumer could learn that their AC units are running longer every day than they expect, and that it is costing them extra money every month. They could then take steps to reprogram their thermostats and shut the AC off while they are at work, for example.

Data Usage Definition

Belkin has developed hardware to read certain power characteristics for electricity coming into a home. This reads electromagnetic interference (EMI) patterns through frequency and time, as well as other power characteristics, such as current and voltage. From the data files, which include the electrical features and target variable labels, we have built a prediction model to learn and estimate which appliances are on in each home in real time. The model, in turn, is used to inform the current usage profile of the appliances in a home, which are then either displayed on the dashboard or used to build the historical usage and energy breakdown in the home.

The data for this product comes from Belkin, via kaggle.com. The data comes in several files, including EMI patterns, real/complex power, and currents/voltages. The labels indicate at which time each appliance turns on and at which time each appliance is off. Further details are provided in the Data Understanding section.

Data Mining Approach

As a data mining problem, this becomes a supervised classification problem. The target variable at any given time is the appliance which is on or off. Or, to be more precise, there are a series of target variables that represent each appliance, and we have essentially built a model for each appliance. We have developed software in Python that reads in the current electricity profile, and uses the model to predict if that particular appliance is on or not at any given time. For example, the targets can represent “toaster”, “lamp”, “Air Conditioning” or “washing machines”, and there is a separate prediction model built for each of these devices. The product would then evaluate the incoming power, and predict whether each of these devices are on or off at any given time.

Project Scope

The project presented here should be considered a proof of concept for the ultimate product that would be created, given further time and resources. We limit ourselves to predicting the on/off state of each of the appliances in the sample homes, rather than conducting the full energy disaggregation analysis that would be present in the final product, which would include power usage prediction by appliance, as well as monthly forecasts. Our data, modeling, and evaluation descriptions all consider the on/off prediction only, though we discuss the full final product in the Deployment section; this would be what we consider to ultimately be the first generation marketable product.

We also do not consider the product in terms of return on investment and product profitability. Though critical to the decision whether to launch the product or note, our approach does not consider these aspects as it is primarily product design, rather than using data science to guide a marketing strategy, or customer ad targeting (for instance). For purposes of this project, we assume a priori that there is a sufficient market and that a successfully implemented product would generate sufficient revenues or return on investment to justify the project from a business perspective.

Business Recommendation

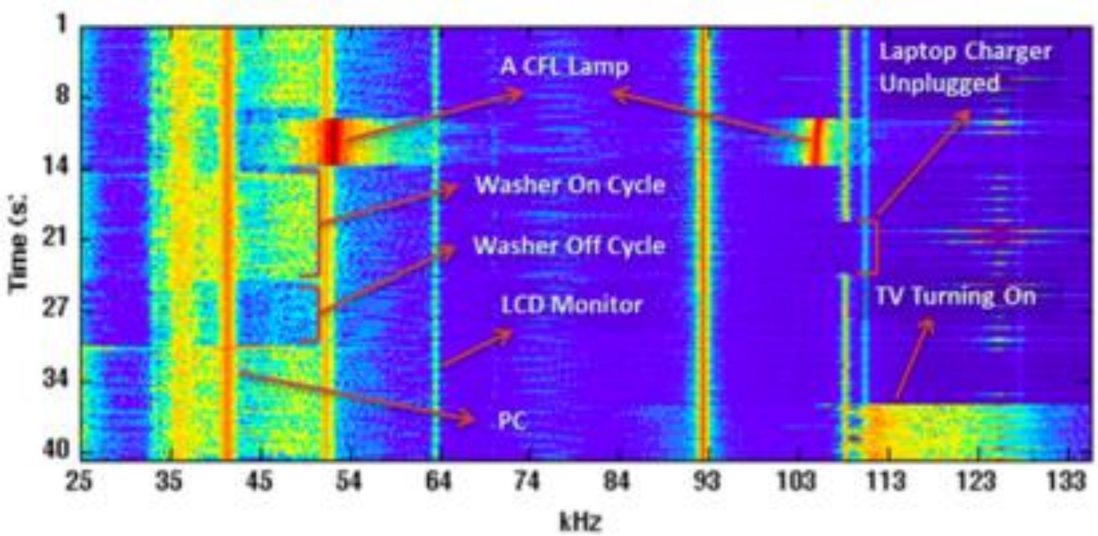
Though the initial modeling and prediction results (highlighted in the Evaluation section) are highly promising, given other challenges associated with this product, we ultimately do not believe it to be a viable consumer product for two main reasons. First, the training method we have outlined in the Deployment section would be highly burdensome to a home-owner (requiring an intensive setup period, as well as ongoing monitoring). Secondly, though our modeling results accurately predict the state of the appliances, our model takes a very long time to run. Since this is anticipated to be a "real-time" product, we find that our model "as-is" would prevent it from being a viable real-time product.

This sort of product may be more viable for institutional uses, such as large offices or campuses, rather than single family residential uses. These types of facilities may have an energy manager who can use, train and accurately interpret the model outputs; this product could be used in lieu of electricity sub-meters for equipment such as pumps, chillers or major fans.

In spite of this, for informational purposes, rest of this report considers the initial intended use, that of a residential home-owner.

Data Understanding

The data source is entirely supplied by Belkin via the kaggle.com data science competition website. Data is provided as a means of producing prediction models for identifying which appliances are on in a home at any given time based on electromagnetic interference (EMI) and electrical power. The EMI is determined by custom hardware built by the Ubicomp Lab at the University of Washington[1]; an example of which is shown below. This plot shows only the EMI pattern, plotting the interference in the corresponding frequency range against time:



This is the basis for the data science approach outlined for this project, and the approach taken to develop our energy disaggregation product. In an ideal world, the presence or absence of particular EMI signatures in a particular frequency would be able to tell us exactly what appliances are running at a given time. However, in practice, this approach is not possible. There are several confounding factors, such as a large number of appliances in the home, each generating their own (and potentially overlapping) EMI patterns, and the fact that a single appliance might have different EMI patterns under different operating modes (e.g. the different cycles of a washing machine). As such, a data science approach is the most

1 www.kaggle.com/c/belkin-energy-disaggregation-competition

appropriate approach to solving this problem, as a probabilistic prediction algorithm count account for these nuances.

Data is provided for four different sample homes, taken over the course of several days at each home. The competition provides both training and testing data sets; however, only the training sets are labeled. The testing data sets are intended to be used to provide results to the competition runners for evaluation; however, as we do not have the test set labels, we disregard the test data for purposes of our project.

Measured data is available in five files for each home:

HF gives frequency interference information with measurements approximately every second.

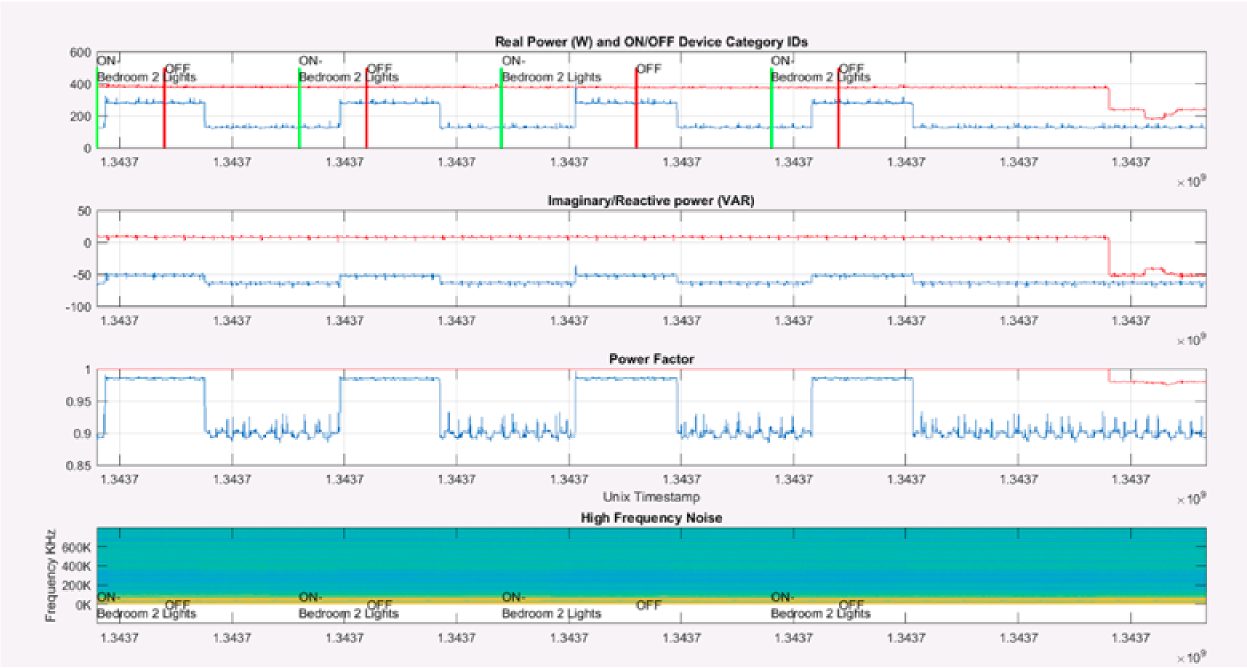
LF1V & LF2V give voltage information (as complex numbers), with measurements approximately every second

LF1I & LF2I give current information (as complex numbers), with measurements approximately every second

The approach taken to convert these files into usable data structures is outlined in the Data Preparation section. Once the data are compiled into the appropriate forms, as described below, they are in the correct format to conduct the data mining analysis and prediction modeling.

Data Preparation

In order to prepare the take into the necessary format for data mining, we break these data files down into electrical information such as Power Factor, Imaginary/Reactive Power and Real Power as shown below. These graphs are generated from MATLAB code provided by the competition organizers, but we have done a similar analysis on the raw data to generate these parameters in Python.



The labeling is provided by default in the training data set, but also included is the TaggingInfo file, which contains a list of devices and at which time they were turned on and turned off (sample below). As the labeling for each appliance is already present in the data, we do not need to do any additional manipulation to get the labels into the correct format, but this file is presented as an additional straightforward way of viewing the appliance statuses. Time is shown below as a Unix timestamp.

Device ID	Device Name	Time On	Time Off
32	Stove	1343331240	1343331270
32	Stove	1343333640	1343333700
32	Stove	1343335230	1343335290
25	Microwave	1343331360	1343331420
25	Microwave	1343333760	1343333790

Zeros in our labeling data indicate times that the device is off and ones indicate times that the device is on.

Practically, this meant importing the tagged training data into a python environment and formatting the data for use by the computer. The training files consisted of a series of lists stored as dictionary key-value pairs. In order to properly format the data, we extracted the list out from the dictionary and paired it with timestamp information and then saved all known features (frequency data, line voltage, line current, and line power factor) to a variable representing a given house. In order to reduce processing time, these operations are performed in parallel and only brought together into the feature tuple right before the model needs them. Mapping the device state to the given timestamps was trickier, as the data only provided the on and off times of each device. In order to perform any modeling, we would need to have an understanding of the device state at all times, so part of the preparation of the target variable involved creating a space of all possible timestamps and individually mapping Boolean on/off state indicators to each time-step. Both the features we tested and the target variables we were seeking insight on were arrayed using Pandas dataframes. This made manipulation easier and helped us focus on the intricate aspects of modeling the data.

Modeling

Owing to the sheer volume of data (81,000 rows, 10 columns for one day's worth of target data alone), model selection was of the utmost importance. For the House 3 7/30 datafile, for example, there were ten target variables. This meant that a simple boolean predictor was out of the question. Since we had multiple target variables with their own unique characteristics, a classification model lent itself naturally to our plans. Specifically, a k-nearest-neighbor (kNN) model predicated on the defining features: 8196 frequency ranges, power, voltage, current, and power factor for each timestep. Training this model took time due to the large dataspace of prediction variables, but eventually, we were able to create a model using an 80/20 cross-validation split for training and test data. Testing the model took even longer (51 seconds to fit vs. 18 minutes to test), but we were eventually able to draw modest conclusions from the input data. k-Nearest-neighbors was selected as a model due to its speed in training the model- key to our completing this project on time. Furthermore, in contrast with a classification tree, we were not trying to separate the target variables based on proximity to shared features. This left the kNN model as the most viable solution.

While we considered turning this project into essentially a large boolean voting algorithm, this was discarded due to implementation considerations. This would have called for creating unique boolean predictor models for each appliance, and given a feature input vector, had each model "vote" on how confident they were that this represented their appliance. These votes would have been tallied, and perhaps fed into a deciding model. Based on the previous success of the voters in the past, the decider would choose a winner from among the voters, the predicted target appliance, and its current state. Unfortunately, time considerations prevented us from fully realizing this implementation.

The kNN model helps the business problem by giving new devices a starting point for analysis when they are first added to a user's profile.

Evaluation

Modeling Results

Due to large volumes of data and due to skewness in the data has made evaluating the model very tricky job. As explained in the modeling section of the report, we are using the supervised KNN classification model, with classification threshold of 0.5. We have ignored the cost function, but we have assumed the cost for error to be same for false positives to true positives (the actual costs of inaccurate prediction are currently not known, other than just affecting the overall reliability/trustworthiness of the product). We have used 3-fold cross-validation for getting overall performance of the data. Since there are no historical models, we can use as baseline model, we have selected baseline models based on the evaluation option.

We have considered 3 options for evaluating the model. Accuracy, ROC curves and Hamming Loss. Each of its results are explained as follows:

Accuracy: On an average we got 99.65% accuracy, which looks like very good, but is not necessarily a complete assessment of the model. We do not have a clear idea of what the impact of false positives and false negatives, so we consider it valuable to examine this as well. For accuracy, see the table below for accuracy ratings for house 3 with 80,000 readings for various k-values.

k(values)	Accuracy
2	99.68
5	99.73
10	99.69
15	99.48

ROC curves: The results of ROC curves with area under curves for different k values are shown below. For k>=5 the model can be considered as conservative, they are making more true positive rate(tpr) is high and false positive rate(fpr) is less. We have used randomly assigned model as a baseline to compare.

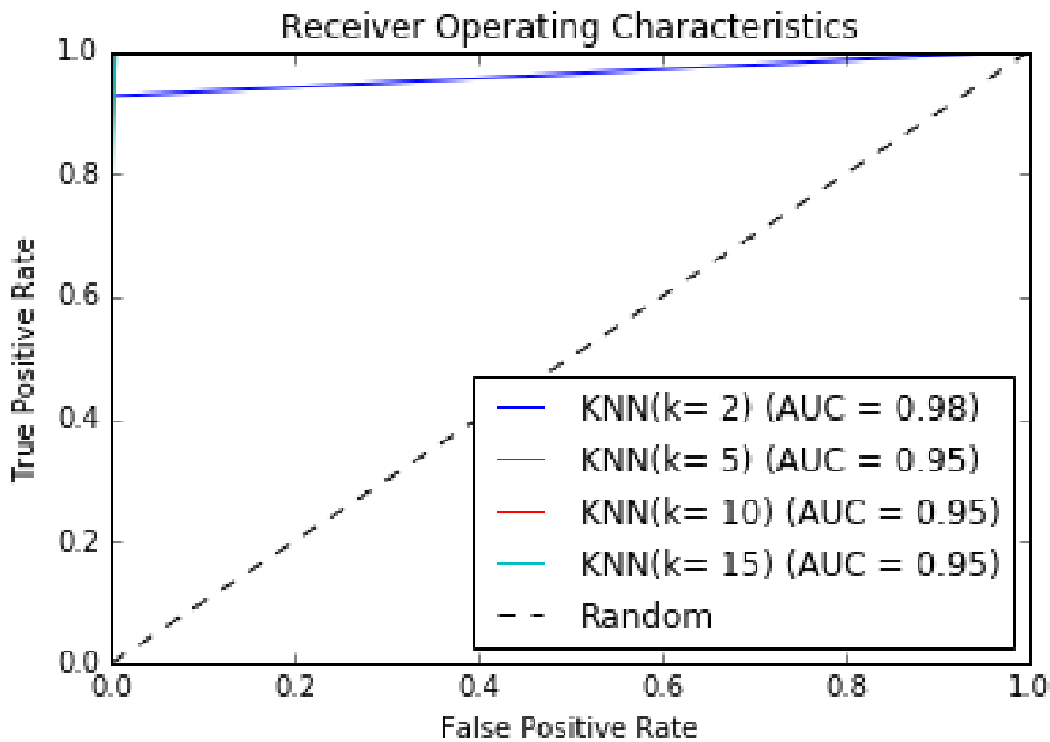
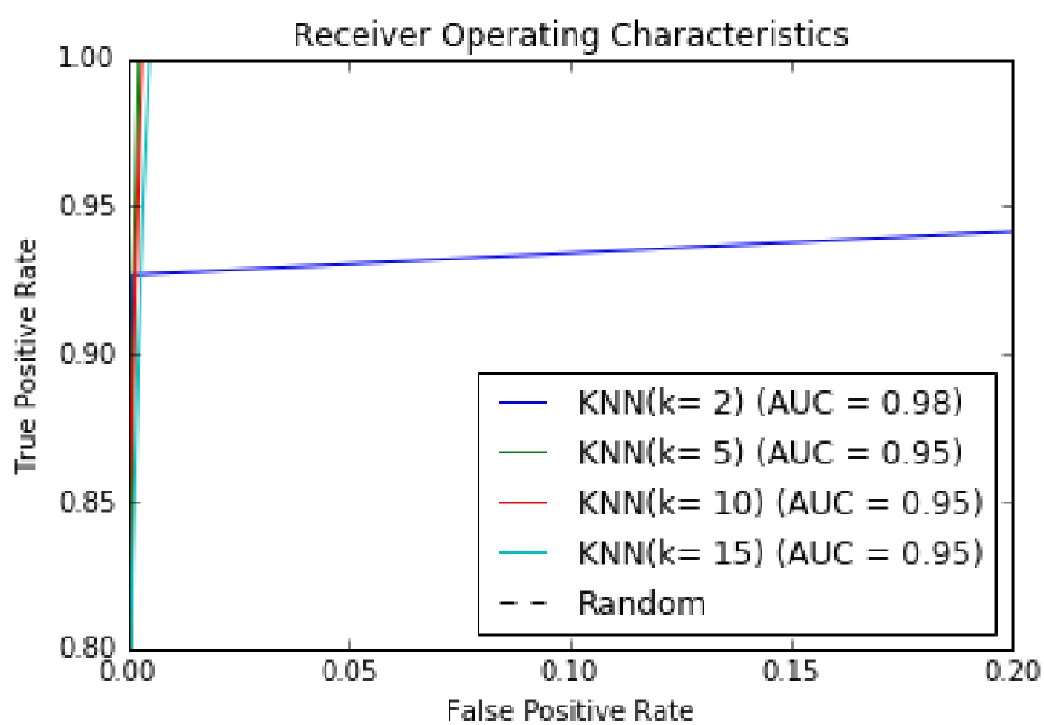


Figure zoomed to top left corner of the ROC curve.



Hamming Loss: The Hamming Loss measures accuracy in a multi-label classification task with the formula:

$$\text{HammingLoss}(x_i, y_i) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{xor(x_i, y_i)}{|L|},$$

where:
|D| is the number of samples (timepoints)
|L| is the number of labels (appliances)
y_i is the ground truth for i
x_i is the prediction for i

Exclusive operator (XOR) increments the hamming loss if xi is not equal to y_i. In order to select the best model we need to look for minimum hamming loss. Hamming loss penalizes the false positive and the false negative equally. If we would have calculated the cost function of our model predictions, cost of false positive would be equal to false negative. So we think hamming loss would be seamless evaluator for our model. See the table below for the Hamming Loss rating for house 3 for various k-values.

k(values)	Hamming Loss
2	0.0003148
5	0.0002654
10	0.0003086
15	0.0005185

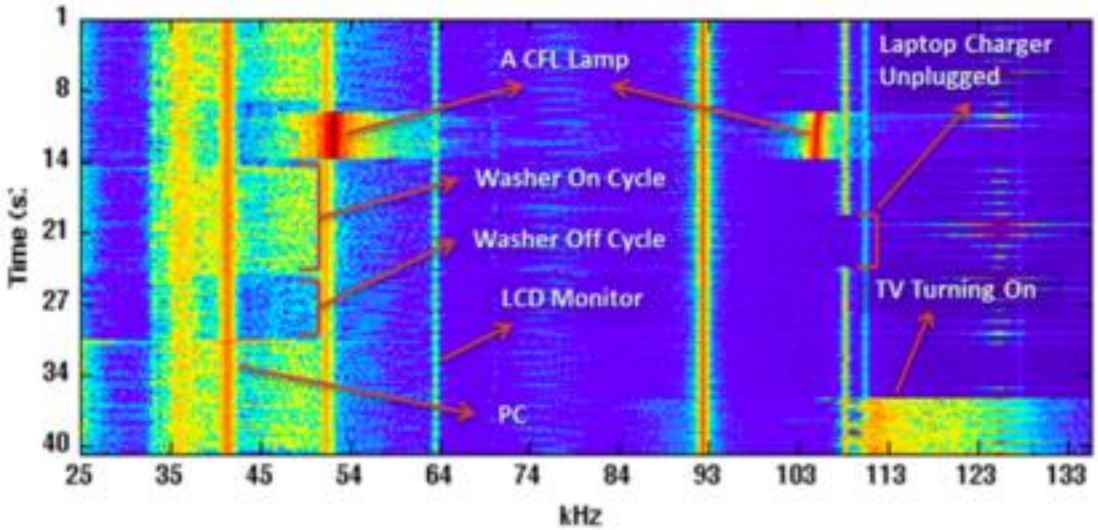
Analysis of our High Prediction Accuracy

An unexpected result of our modeling is the relatively high prediction accuracy. Part of this (upon further reflection on and investigation of the data) likely results from the limited set of appliances in the sample house. From this limited set, each appliance has a very unique EMI pattern, resulting in the high modeling accuracy. For example, refer to the image below[2]:

[2] screenshot of https://www.youtube.com/watch?v=o-SqO8y8XUA&feature=youtu.be, from the Ubicomp lab at UW

This video shows the researcher going around and turning appliances on and off, and the resulting EMI pattern in the upper right. From the EMI pattern shown, it is very clear that each of the devices has a unique pattern in a specific frequency band. As can be seen, its very noticeable when the CFL is turned off and when the TV is turned off.

Another contributing factor to our high prediction accuracy is the “homogeneity” of the data set. Each house has a very specific set of appliances, as discussed above, as well as only running each appliance for a limited time. So, when we split the data set, our resulting “training” and “test” datasets are very similar to each other, and there is not a lot of variation between what a “TV on” instance looks versus the other “TV on” instances. The model, essentially, looks for that specific frequency band to determine whether the device is on or off, and ignores the rest of the data. The image below, reproduced from the Data Understanding section shows this very clearly with the different appliance statuses highlighted.



Generalizability of the Model

Ultimately, because we are using such a small set of appliances, its not clear if the results would be generalizable to a greater scope, but does help in getting very accurate prediction for this proof-of-concept. One way to test whether the prediction models for each appliance are broadly applicable would be to test it against another house, but unfortunately, the sets of appliances available for each house are different. Even more surprisingly, the same house does not run the same appliances on consecutive days, meaning we cannot use one day's data as our training set and test it against the other day's data (refer to chart below that shows sample house 3 and which devices are running each day).

30-Jul	31-Jul	1-Aug
Back Porch Lights	Foyer Lights	Dryer
Bedroom 1 LCD TV	Garage Door Opener	Washer
Bedroom 1 Lights	Garage Lights	Dishwasher
Bedroom 2 Lights	Garbage Disposal	Front Porch Lights
Bonus Room Blueray/DVD	Guest Bath Fan	Master Bath Fan
Bonus Room LED TV	Guest Bath Lights	Master Bath Lights
Bonus Room Lights	Hair Dryer	Master LCD TV/DVR
Bonus Room Wii	Kitchen Lights	Master Closet Lights
Computer	Laundry Room Lights	Master Blueray/DVD
Dining Room Lights	Living Room Audio-DVR-TV	Master Lights
	Living Room Lights	Office Lights
		Microwave
		Oven
		Toaster
		Upstairs Hallway Lights
		Powder Room Lights

In terms of how this would affect the product design, we assume that the models are not likely to be generalizable across multiple houses without further evidence. This would affect how the training would be implemented (this is detailed further in the Product Usage section below).

Deployment

Product Introduction

Ultimately, the intent of this data mining project is as a proof of concept for the energy disaggregation product. Though our analysis for this product indicates it would not be a viable marketable product (refer to our Business Recommendations section), we present here our vision for what the product would look like were it to be fully developed.

The energy disaggregator product would come with a user interface for tablets and/or desktop computers, that provides real-time and historical readouts of the current electricity usage. A prototype user interface is shown below, showing what kind of data would be presented (refer to Appendix 1 for a full size version of the dashboard).

The user interface is broken down into four main parts, each of which makes use of the prediction algorithm. The top left of the user interface shows the total energy usage of each appliance, both as a percentage of total annual energy use and total kilowatt-hours. It bases the calculation on both historical usage (determined by the prediction model) and predicted usage (determined by estimating off of previous months). The bottom left of the interface shows the historical usage and cost by month. This is a handy summary for the homeowner to review their overall patterns. The historical usage would be compared to the utility statement to get costs and accurate monthly usage (since the days between meter readings might not be consistent). Ideally, it would tie into the utility company's database automatically to make it easier on the user.

The right side of the interface shows the current summary of usage. The top is the total power usage at any given moment, essentially the current consumption by all devices in the home. Below that, it shows the current month month-to-date and projected total costs and usage, as well as the same information for the current year (year-to-date and total projected costs and usage). Finally, the bottom right of the chart shows which devices are currently running, and which are currently off, and the power consumption for each device. This allows the homeowner an "at-a-glance" summary of what is going on in their home, and to make real-time decisions to shut off appliances that aren't needed.

The final launch product would include several features described above that aren't discussed as part of our data mining process. As stated above, we limit our study to a proof-of-concept on the ability to identify which devices are active at any given time, as this underpins the rest of the product features.

Product Usage

One of the key points about this product is that any given home has nearly limitless potential devices that could be used or installed. For instance, just think of the range of TVs and lighting that exist, and that a homeowner could buy, each with their own EMI patterns and electrical characteristics. Clearly, generating a database of EMI patterns and electrical characteristics in advance for each product is an intractable problem. As such, when a homeowner purchases this energy disaggregation device, there would an initial training period that they must go through.

Upon initial startup and installation of the product, the software will ask the homeowner to set up the initial set of products based on what is currently in the user's home. The homeowner will be able to set up the appliances in their home by going through and turning each device on for a period of time, and telling the software when that device was first turned on and when it was turned off. Though a potentially time-consuming process, depending on the size of the home and the number of different appliances, this will serve as the initial training set for that particular home. Without this initial set of data, there would be nothing for the model to make its estimates on, as there would be no existing labeled data, and the software would not be able to identify the unique EMI patterns for each device; it would be indistinguishable from random noise without the initial labels.

Once this initial training period has been completed, the product is ready for use and can start monitoring the energy usage of the home. This initial period is likely to be relatively inaccurate, though. For a house with a small data set, and especially an owner who may not be interested in putting all his devices into the database (e.g. does he really care about monitoring when his electric shaver is on?), we would expect a fair amount of noise and difficulty fitting the model to the EMI patterns. One way we would address this, though, is that over time, the device would refine its model as devices are used more and as it collects more data. This would also allow it to monitor for unlabeled patterns (such as that electric shaver) and actively ignore those EMI patterns.

Part of dealing with the issue of unidentified appliances would be the "other" catch-all category. This category would be necessary to square the individual breakdown with the overall energy usage. This category would include usage by any uninstalled devices, as well as usage that was "missed" by the prediction model. Theoretically, this other category could also show negative numbers, if the model is predicting too much usage by installed devices. A growing other category would be a useful flag for the owner, and the software could notify him if the other category reaches a certain threshold to check if more devices have been installed or changes (e.g. if they get a new TV with a new EMI pattern).

Ethical Considerations

One key concern that may be prominent in potential buyers' minds is how their data will be used and or collected. They may not want the company to collect information on their habits, and there is of course a potential security threat. If a hacker is able to access the data, they might (somehow) be able to determine when people are home or not based on what devices are currently running – a potential burglary risk. Though out of the scope of the proof-of-concept project, security would be a priority item to consider in the product launch.

One analogue for the data and privacy concerns are the debate with smart meters[3]. We would expect very similar concerns to be held against this sort of energy disaggregator product. These concerns could be medical (i.e. any specialized medical equipment appearing in the data set), or indicate the presence of expensive gadgets (such as numerous laptops or smartphones) that may appeal to thieves. Ultimately, the kind of energy usage patterns could reveal a lot of information about a person's (or household's) life. Though these concerns could be considered paranoid, it's important to address them proactively in order to prevent the concerns from impacting demand for the product.

One way to head off these concerns would be to have a product that is completely isolated from the web; an entirely local product. This might be viable for a first generation product, but would miss out on some of the potential benefits that this kind of product could have with widespread adoption (large scale estimation and feedback, "community" based learning, or suggestions for the end user from the utility on what sort of actions to take).

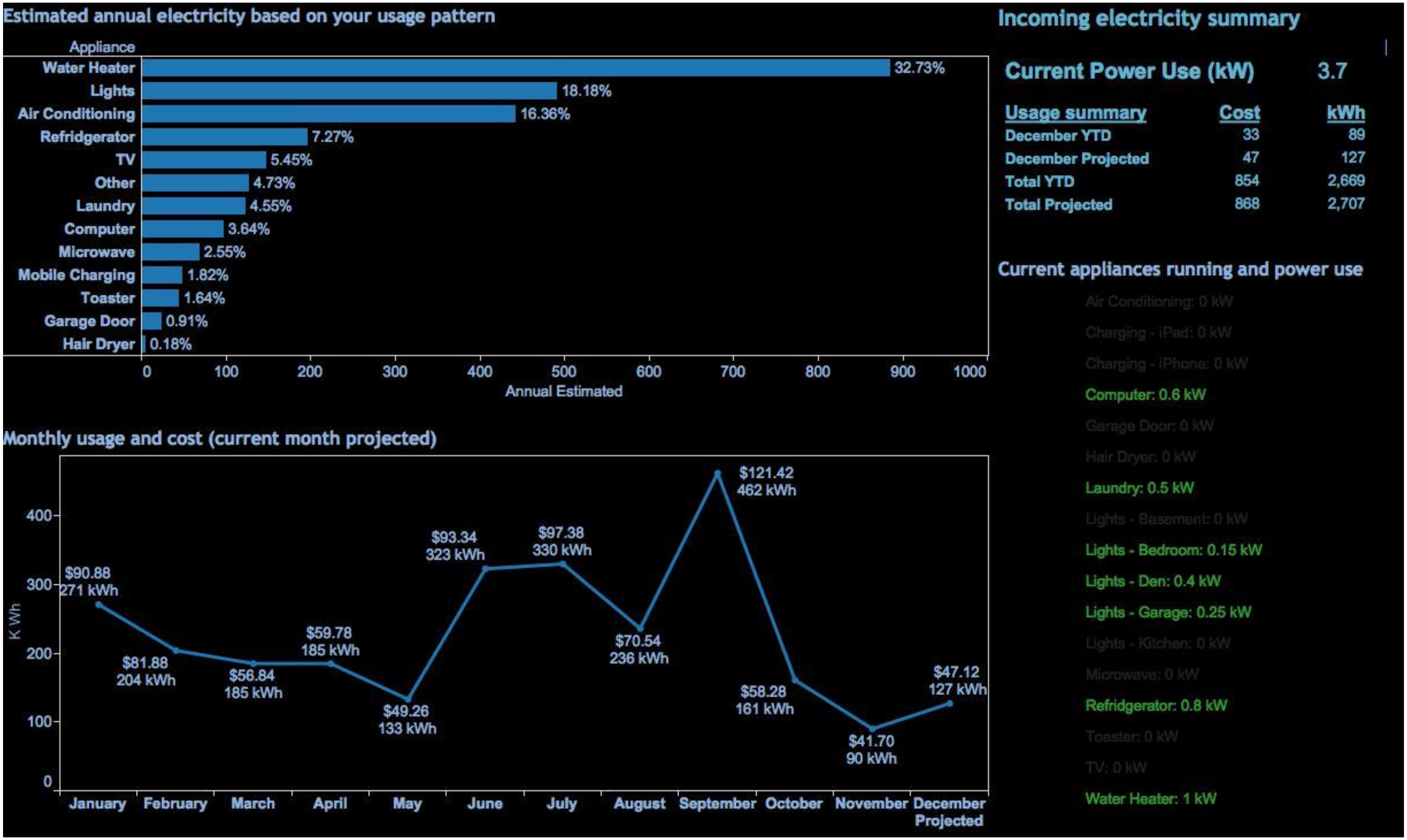
[3] http://www.forbes.com/sites/federicoguerrini/2014/06/01/smart-meters-friends-or-foes-between-economic-benefits-and-privacy-concerns/

Project Risks

This product would have the usual risks associated with product development and launch, but from a data science perspective, there are a couple unique risks that should be considered. First, though our results indicate that the modeling and prediction is fairly accurate (see evaluation section above), the prediction is nonetheless not 100% accurate. For an individual appliance, this may not make a big impact on the overall accuracy of the product, but as the number of appliances gets larger, these inaccuracies could build on each other and cause significant errors in the end-use breakdown. This might lead users to make incorrect assessments about how to optimize their energy performance.

Potential complications from an increasing number of appliances form the basis for another risk with this product. Individual appliances in the data set analyzed tend to have EMI patterns consistently in the same, unique frequency range. With a larger set of appliances, there is more possibility that there is overlapping (or even changing) EMI patterns between appliances. Overlapping patterns may result in inaccurate predictions (e.g. estimating that the TV is on when its actually the dishwasher) and patterns that change when different devices on may result in missed predictions. One potential way to address these issues is to increase the scope of the prediction model to include time of day, season, etc. that is not available in the training datasets. Longer data sets (and the expectation of ongoing refinement of the models) could address the issue with overlapping / changing EMI patterns, but this risk would be present nonetheless. Related, if a device’s EMI pattern changes over time (as it wears down for instance), it might get lost in the model unless the model is renewed regularly.

Appendix 1: Prototype Graphical User Interface



Appendix 2: Individual Project Contribution Summary

Jason:
Project coordination and scheduling
Preliminary data preparation and manipulation methods

Adam:
Development of modeling results into “real-world” interpretation
Discussion on product implementation
Project reporting, tying together and editing

Joe:
Development of the overall data reading / manipulation program
Preliminary model development and evaluation methods
Data Preparation and Modeling sections

Sudhir:
Converted the matlab scripts(from kaggle project) to python to load data.
Tried different models and did a grid search to get the best model configurations.
Worked on different evaluation techniques and generated graphs for the report.