User’s Guide

Project Name: Indy
Program Name: Energy Data Management

Table of Contents

What is This?

I have deep passion in climate change, environment, clean and renewable energy. I study computer science so that I can one day invent technologies that help climate change and foster renewable energy use.

I naturally got interested in smart home and smart grid products, where our built environments get digitized, which means our daily energy usage becomes useful data to make useful inferences.

As such, for Indy, I built a program that analyzes power consumption data of a house. The data is actually a real data from a house in France, spanning from Dec 2006 to Nov 2010. Let’s take a look at a sample data.

Date	Time	Active_power	Reactive_power	Voltage	Intensity	Meter 1	Meter 2	Meter 3
16/12/2006	17:24:00	4.216	0.418	234.84	18.4	0	1	17

Date: the date when this measurement was made
Time: the minute when this measurement was made
Active_power: real power consumed during this minute (unit: kW)
Reactive_power: unused power in the lines during this minute (unit: kW)
Voltage: average voltage during the minute (unit: V)
Intensity: average intensity during the minute (unit: A)
Meter 1: Energy used in the Kitchen area of this house (unit: kWh)
Meter 2: Energy used in the Laundry area of this house (unit: kWh)
Meter 3: Energy used for Climate Control (HVAC, Water Heater) (unit: kWh)

With this data, the program can do various jobs. The UI is divided into four different sections: Data, Graph, Machine Learning, and House. Data and Graph is under one subtitle “What is This?” Here is a brief explanation of each of these sections:

Data

The program can parse this data into different categories, summed up by hours, days, weeks, months, and seasons. The user can then save each data into tables for a holistic view, as a csv file, which can be opened in text editors, Excel, or Numbers.

Graph

The user can also choose a specific date and take a look at various graphs of these parsed data, which can provide useful insights about usage patterns. The user chooses a date, time frame (day ~ season), meter. Then a graph that corresponds to the chosen parameters are shown.

You can also choose graph type, which will show a year-long data of the chosen time frame (i.e. power consumption of the year of the date divided by day / week / month / season) or a graph that corresponds to the selected time frame.

For example, if week is chosen, the week’s power consumption from Monday to Sunday is shown. If day is chosen, usage for each hour is shown. If month, each day of the month, if season, each month of the season.

Machine Learning

The user can choose any future date from Dec 2010 to Nov 2011, and get a prediction value gained from machine learning process. A time frame is chosen from day to season, and the graph of average value for each day / week / month / season for the four years of data is shown. The predicted value is attached to the graph as a point. This visualization helps to evaluate how far the predicted energy usage is from the average value.

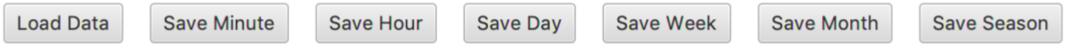
House

When the future date is chosen, the prediction for different meters in the house are also made. The GUI of an house is made, and when the user clicks a certain area, a popup window is made at the location of the click that shows the predicted usage of that area of the house.

Below are detailed guides of each of these sections.

Data

In the data section, we first see an explanation similar to the “What is This?” section above. After that, we see several buttons to click.

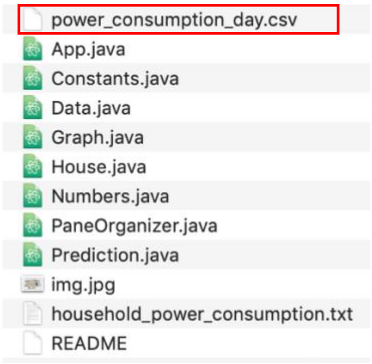


Before doing any other works in other sections, click Load Data button. This button does all the preliminary works of loading, parsing, and categorizing data, as well as training the machine learning model for prediction. This takes some time, around 2 minutes, so on the top left corner of the screen, you will see a status update to see where the process is. Some of the status updates include:

Uploading initial text file…
Parsing daily data…
Training machine learning model…
Data upload complete!

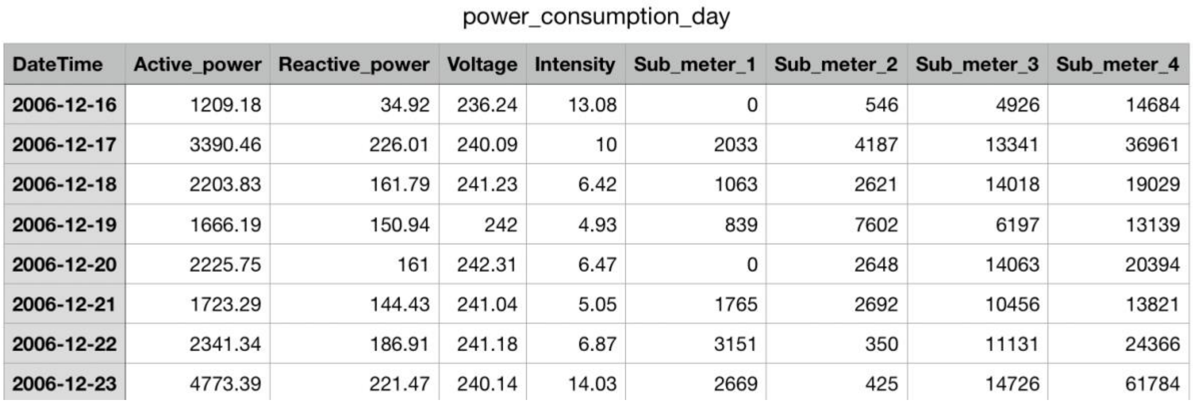
After this step, you can click any of the “Save” buttons. In the project folder, you will see a new csv file created with the name of “power_consumption_’selected time frame’.csv”.

For example, if you clicked “Save Day” button,



power_consumption_day.csv
App.java
Constants.java
Data.java
Graph.java
House.java
Numbers.java
PaneOrganizer.java
Prediction.java
img.jpg
household_power_consumption.txt
README

A new file is made. The data in this file would look like:

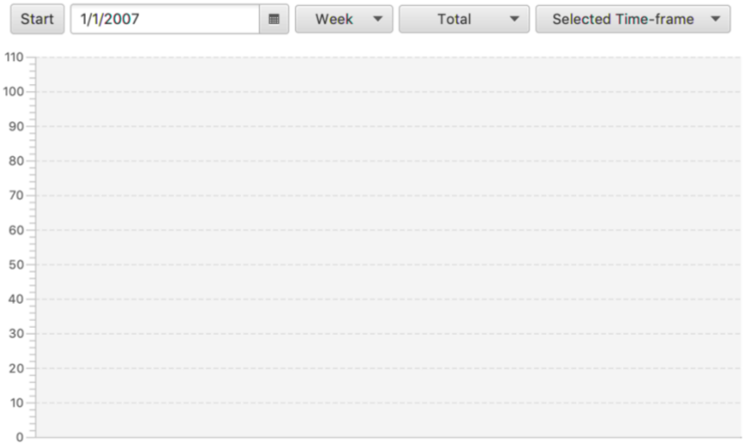


power_consumption_day

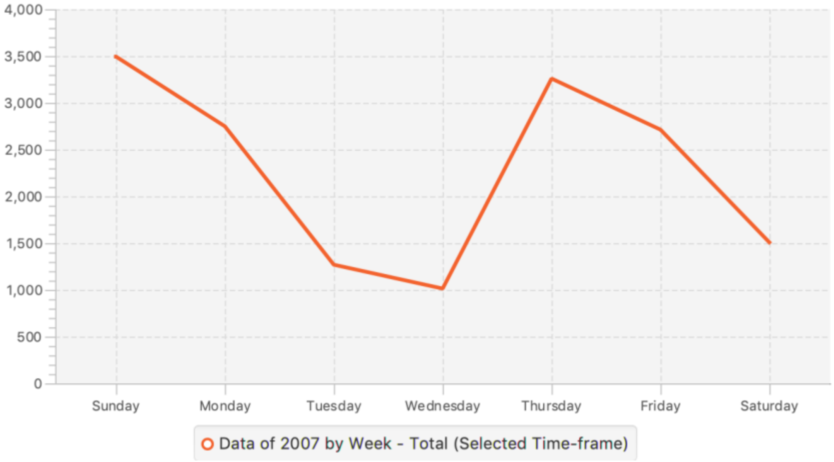
DateTime	Active_power	Reactive_power	Voltage	Intensity	Sub_meter_1	Sub_meter_2	Sub_meter_3	Sub_meter_4
2006-12-16	1209.18	34.92	236.24	13.08	0	546	4926	14684
2006-12-17	3390.46	226.01	240.09	10	2033	4187	13341	36961
2006-12-18	2203.83	161.79	241.23	6.42	1063	2621	14018	19029
2006-12-19	1666.19	150.94	242	4.93	839	7602	6197	13139
2006-12-20	2225.75	161	242.31	6.47	0	2648	14063	20394
2006-12-21	1723.29	144.43	241.04	5.05	1765	2692	10456	13821
2006-12-22	2341.34	186.91	241.18	6.87	3151	350	11131	24366
2006-12-23	4773.39	221.47	240.14	14.03	2669	425	14726	61784

Graph

The Graph section looks like this:

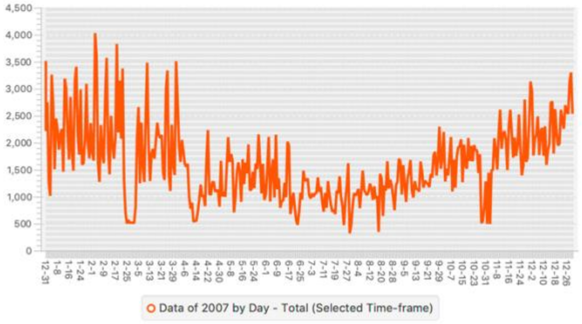


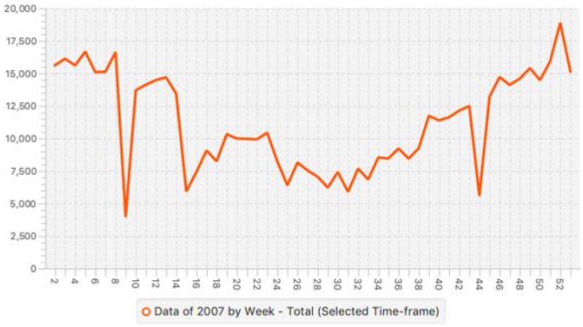
After pressing Load Data from the section above, you can click Start. Then the chart will have a new graph that looks like:

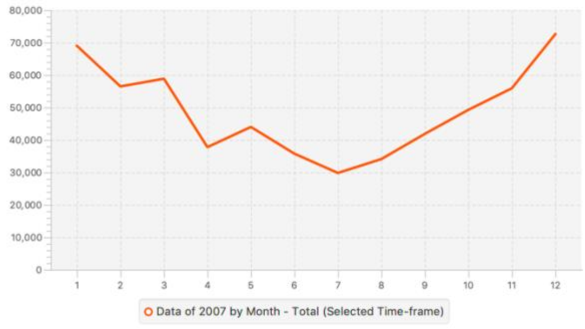


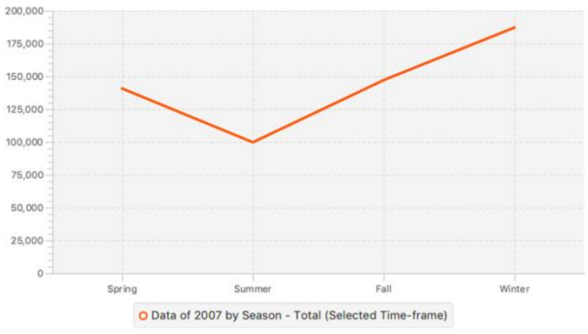
You can now choose any different parameters from the calendar and the menus. Below are examples of each of the different parameters.

Graphs for Year-long (from day to season)

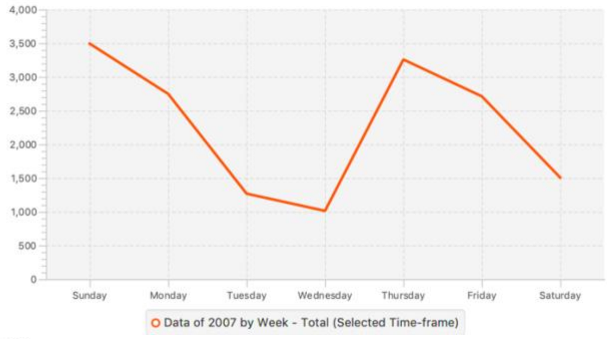


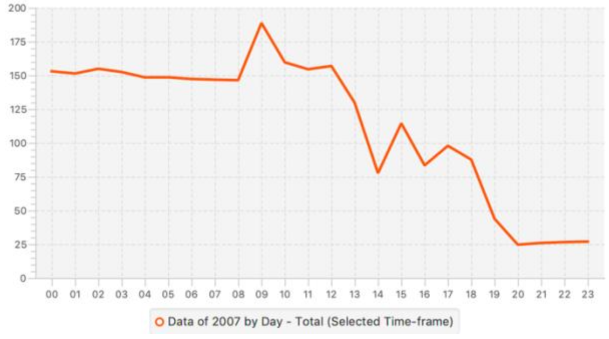


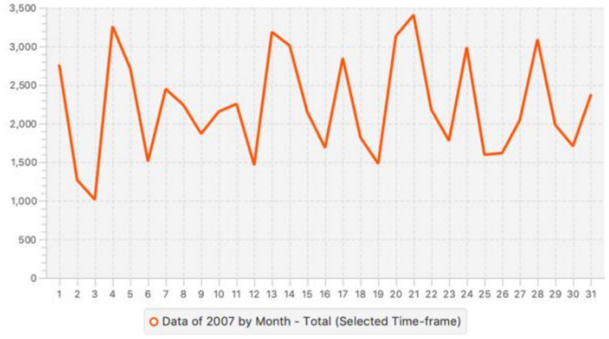


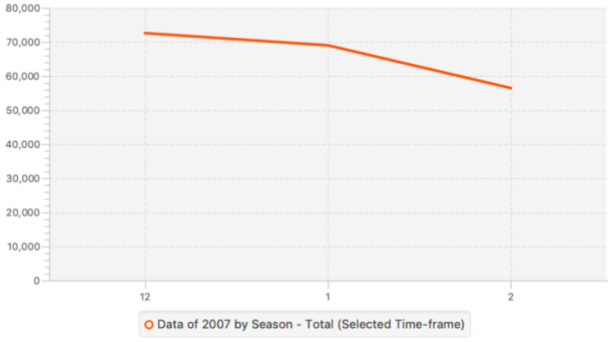


Graphs for Selected time-frame (from day to season)







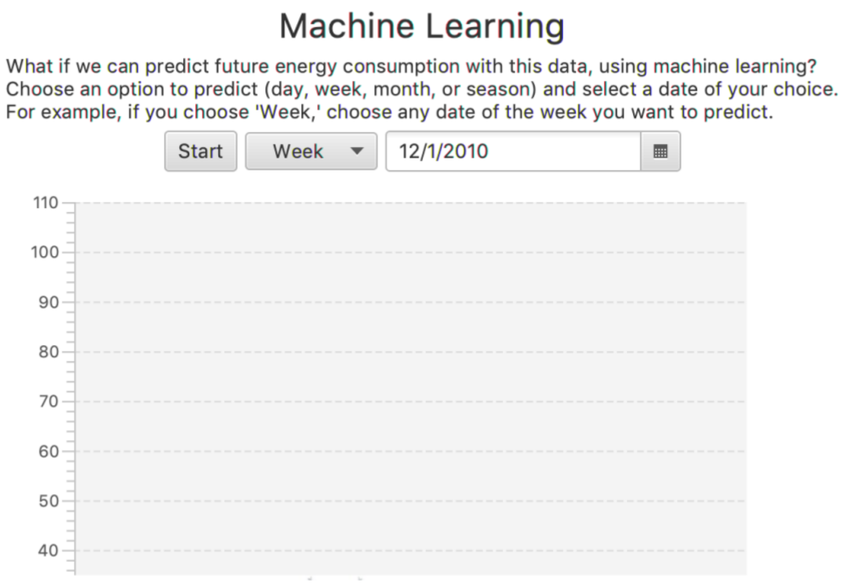


Choosing different meters will simply show different y-values.

Choosing a different date from different year will also change year-long graphs as well. If a date is chosen from the same year as before, the year-long graphs do not change.

Machine Learning

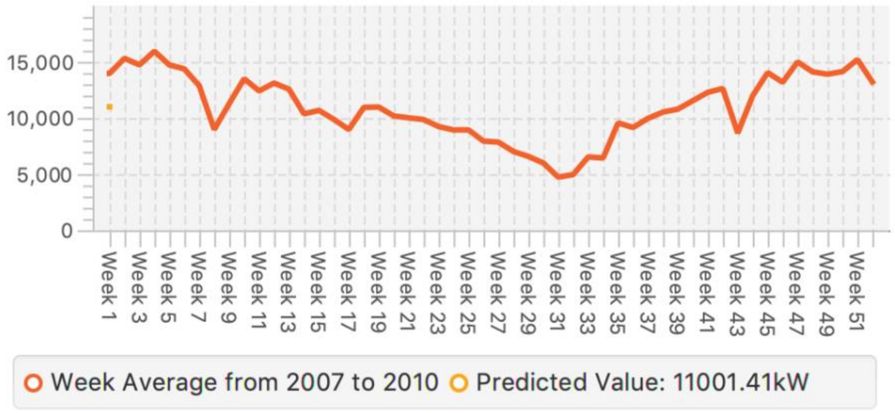
Machine Learning section looks like this.



Machine Learning

What if we can predict future energy consumption with this data, using machine learning?
Choose an option to predict (day, week, month, or season) and select a date of your choice.
For example, if you choose 'Week,' choose any date of the week you want to predict.

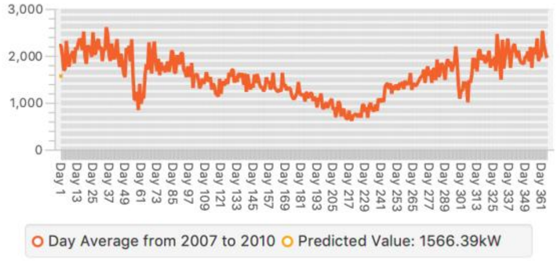
Just as the Graph section, press Start to get a graph and a prediction. It looks like this.

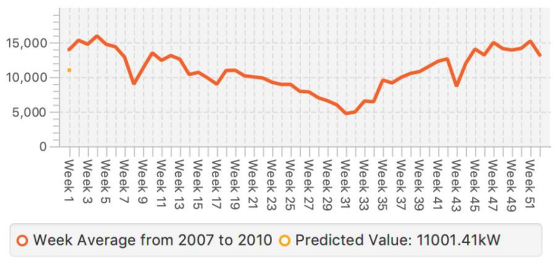


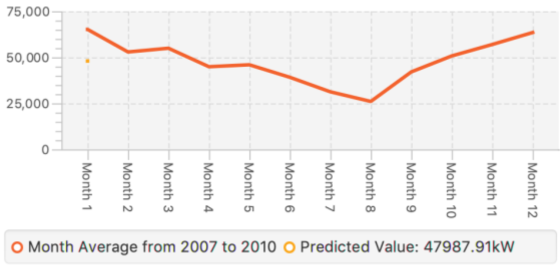
The orange plot is the average of all Week 1 to 52 for the four years of data (2007-2010), and the yellow point is the prediction of the date you picked.

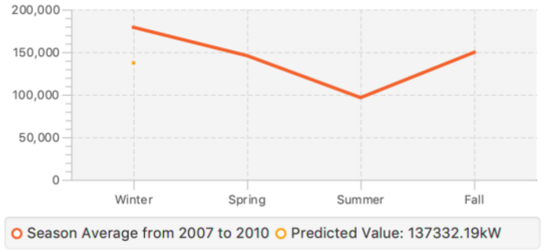
If you change the time frame from week to another, like day, month, or season, they look like this.

In the order of day, week, month, and season:



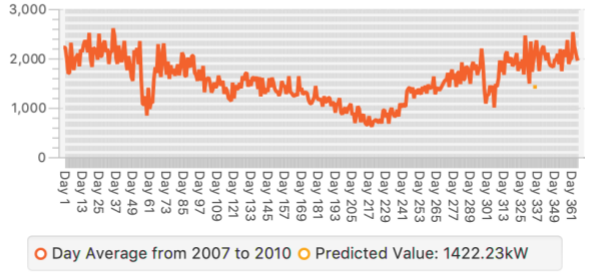


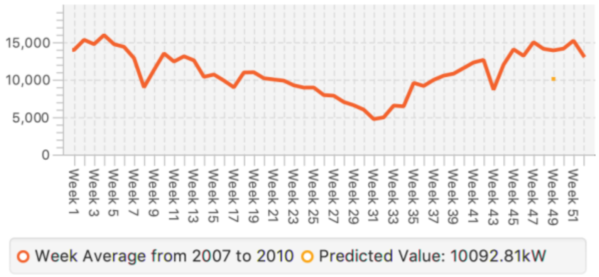


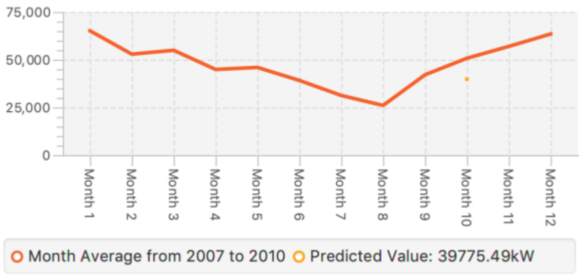


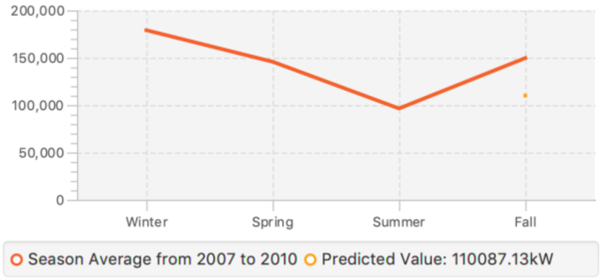
Since the graphs are averages, they do not change even if you change the date. However, the predicted yellow point will move its position, and also have a different value.

For example, if you choose a different date, i.e. May 31st, the graphs look like this.



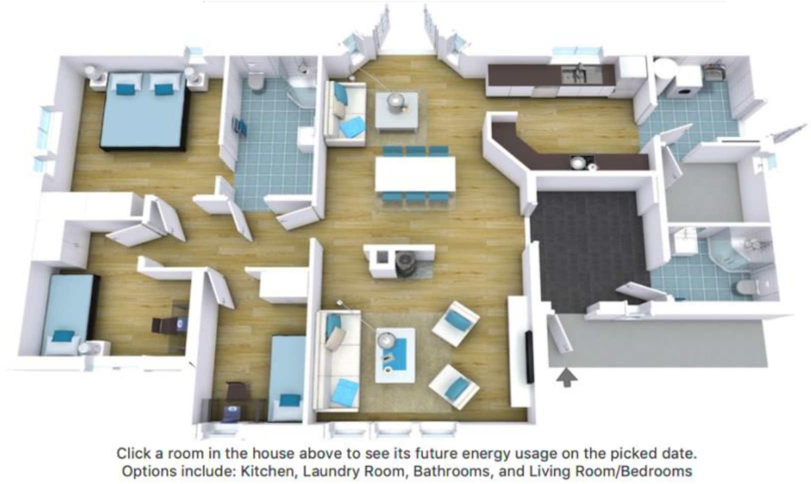






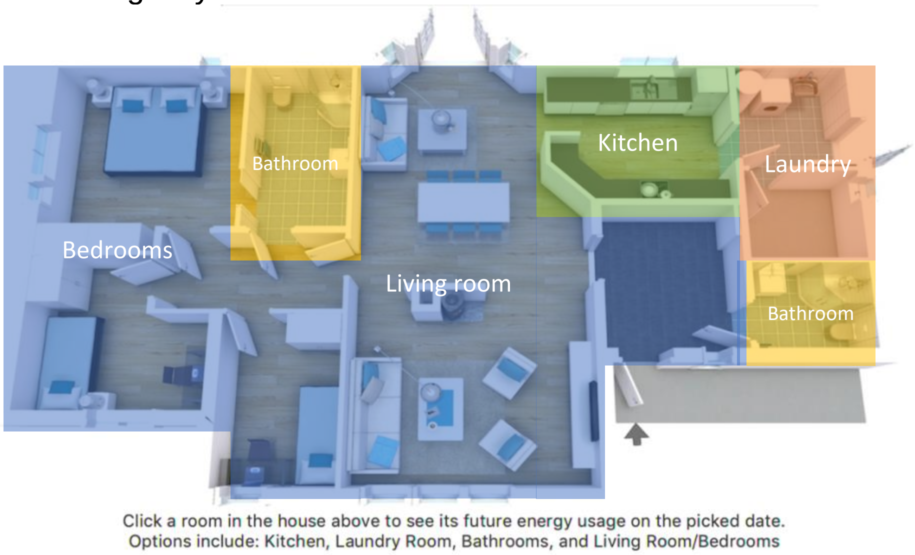
House

The House GUI looks like this.



Click a room in the house above to see its future energy usage on the picked date.
Options include: Kitchen, Laundry Room, Bathrooms, and Living Room/Bedrooms

You can click certain areas in the house and see their energy usage. The areas are divided in the following way.



Click a room in the house above to see its future energy usage on the picked date.
Options include: Kitchen, Laundry Room, Bathrooms, and Living Room/Bedrooms

If you click these certain areas, you will see a popup window that is in the format of:

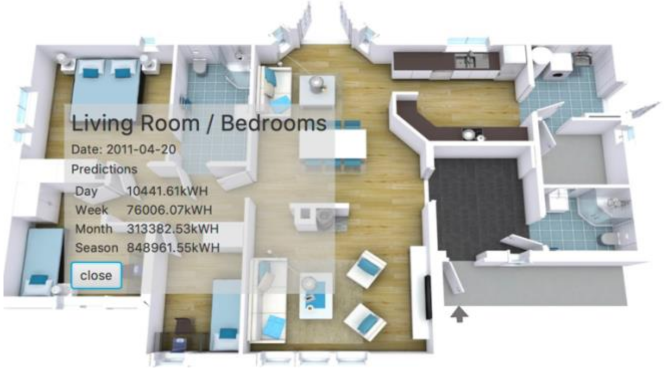
Location

Date: yyyy-mm-dd
Predictions

Day	kWh
Week	kWh
Month	kWh
Season	kWh

Below are the demonstrations of all the area in the house.

Living Room / Bedrooms



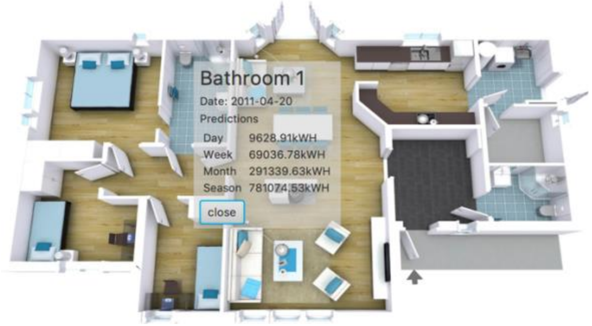
Living Room / Bedrooms

Date: 2011-04-20

Predictions

Day	10441.61kWH
Week	76006.07kWH
Month	313382.53kWH
Season	848961.55kWH

Bathrooms

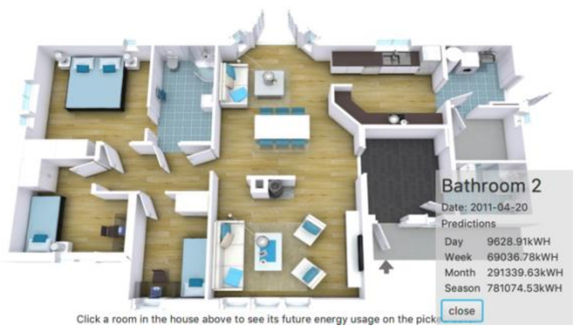


Bathroom 1

Date: 2011-04-20

Predictions

Day	9628.91kWH
Week	69036.78kWH
Month	291339.63kWH
Season	781074.53kWH



Bathroom 2

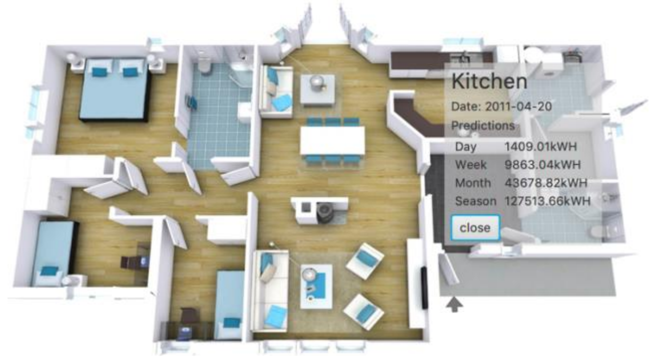
Date: 2011-04-20

Predictions

Day	9628.91kWH
Week	69036.78kWH
Month	291339.63kWH
Season	781074.53kWH

Click a room in the house above to see its future energy usage on the pick

Kitchen



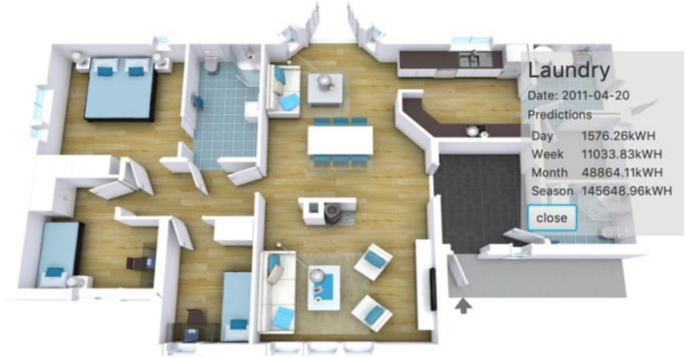
Kitchen

Date: 2011-04-20

Predictions

Day	1409.01kWH
Week	9863.04kWH
Month	43678.82kWH
Season	127513.66kWH

Laundry



Laundry

Date: 2011-04-20

Predictions

Day	1576.26kWH
Week	11033.83kWH
Month	48864.11kWH
Season	145648.96kWH

Limitations

The purpose of this program was to provide an intuitive user interface where the user can analyze his/her energy usage. Through the graphs and data tables, one can find out usage patterns. Through the house GUI one can easily understand which area of the house uses how much energy, and what to do with these findings. Machine Learning itself is a very useful technique, especially in this field, where both the users who consume power can understand their future energy demand and the utility company who has to manage the energy supply. As energy industry is becoming less centralized and distributed through technologies like smart homes, smart grids, and blockchain technologies, understanding individual energy usage and providing useful tools to do so is becoming more important than ever.

However, in terms of performance of predictions, the outcome is not as perfect as more mature technologies out in the world are. I used a technique called time-series autoregression, which builds a multi-layer linear regression for time-series based data like this. It performs error analysis to pick how many past times to use as parameters and build a multi-variable equation.

I used an error analysis tool called Bayesian Information Criterion, which returns a value based on the number of used parameters and the standard deviation of each parameter. The lowest turned out to be around 11, and this value cannot really be said as the best outcome to expect. However, considering the fact that the machine learning algorithms were hard-coded with the current level of programming experience, it may not be said as the worst outcome either.