**Parsyl Take Home Assessment**

By Joe Hutchings

**Model Process Summary**

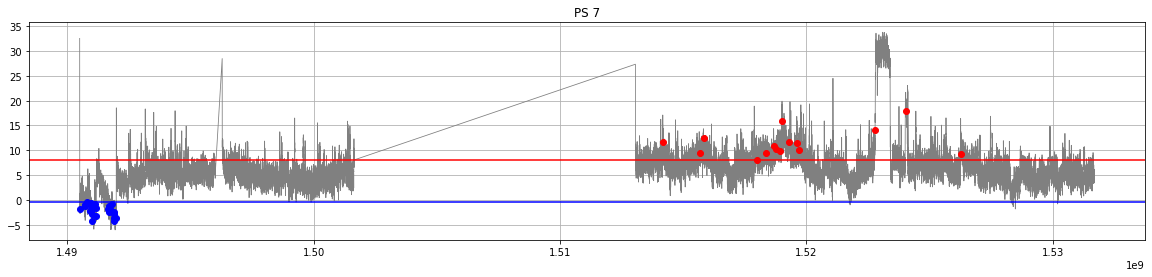
*Data Quality Check*

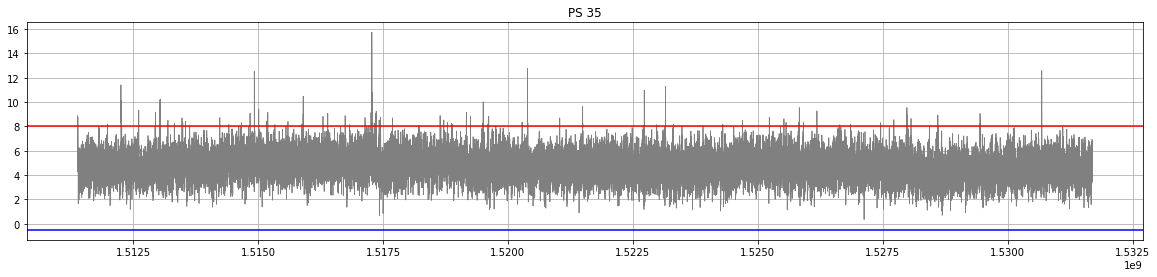
Before creating any models, I checked the data for missing values and anomalies. The time series data had no missing values, but the context data with the fridge makes and models did have missing values. In the end, I did not take into account the fridge data, but given more time I would explore the utility of incorporating the fridge data into the predictive modeling. In the light event data, some of the records had a stop time that was earlier than the start time. I calculated the median start to stop time difference and added that to the start times that had earlier stop times to get estimated stop times. There’s more to that story if anyone is interested. Two of the locations had some temperature readings around 300 degrees at the very end of their time series. I simply removed those records. None of the remaining readings had a temperature reading over 40 degrees.

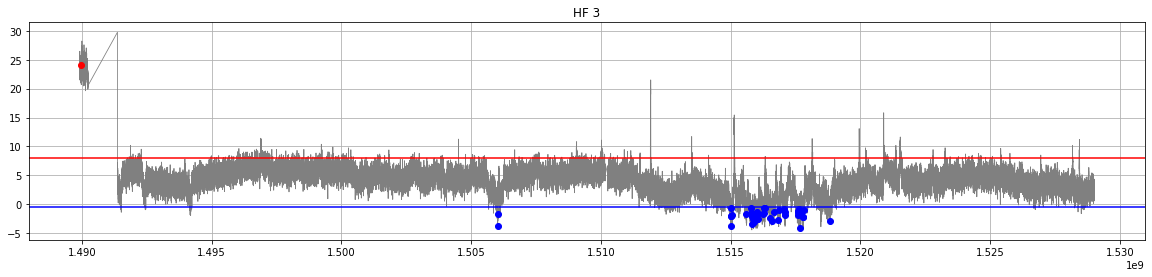
Graphs of the time series showed that temperatures were more likely to go over the heat threshold than go below the cold threshold. Some locations have big gaps of time with no data. Perhaps the fridges had been taken offline during those periods, or the environment-sensing devices were unable to capture data.

*Identifying Cold and Heat Alarms*

I was able to engineer cold and heat alarms based on the criteria given to identify them. I created graphs that showed the time series with the alarms overlaid. The alarms looked correct. Here are some of the graphs of the temperature readings by timestamp per location (red dots indicate a heat alarm, blue dots indicate cold alarms):







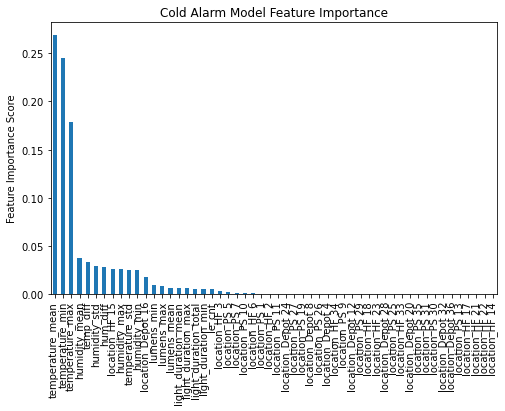
*Getting Forecasts*

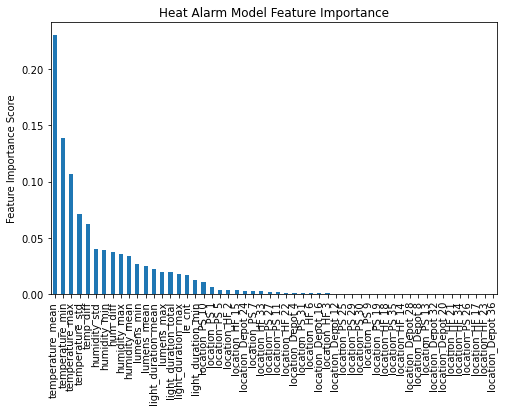
I chose to aggregate the data at the daily level (per location), and then forecast the daily data. I derived metrics per day, such as maximum, mean, and minimum temperatures per day, number of light events per day, and whether or not a cold alarm or heat alarm was triggered during the day. I got forecasts of the daily features using the Neural Prophet forecasting algorithm going out the next 7 days. I created a separate model for each feature at each location, which added up to many models, which were all automated. Why the Neural Prophet algorithm? For the sake of time, I arbitrarily chose a single algorithm and that was it. With more time other algorithms could be tested. In order to move through the project quickly, I purposefully neglected to test the accuracy of the forecasts. That would be something to remedy in order to validate the final results. The forecasts over the next 7 days are to be used in predicting the likelihood of an alarm.

*Predicting Alarms*

I used all of the historical data to train a model to predict which days were likely to have an alarm based on the daily statistics of temperature, humidity, and light event readings. I created one model to predict cold alarms and another one to predict heat alarms. I chose to use a random forest model because it is a well-proven model. Given more time, other models could be tested. I trained the models using all of the historical data, purposefully neglecting to validate the models on held-out test data. Again, given more time, …

Both models essentially memorized the training data, correctly predicting 100% of their target variables. This was actually encouraging because the training data was very imbalanced where days with alarms were fairly rare events. Here are variable importance plots (which also show the features used to predict if there was an alarm during the day):





I proceeded to apply the models to the forecasted data. Rather than getting a single probability of an alarm over an entire week, I got probabilities of alarms for each of the seven days over the forecasted seven days. I then used the highest likelihood of alarm during those days as the likelihood of an alarm during the week. I recognize the weakness in this approach and that it doesn’t capture the true probability of an alarm during the week, which is likely higher than the highest daily probability, but it is a starting point.

In the end, none of the locations had a likelihood of alarm during the forecasted seven days that exceeded the threshold of 0.5. Likelihoods of alarm ranged from 0.0 to 0.48. Here are the predicted likelihoods of alarms during the next 7 days by location:

location cold\_alarm\_likelihood heat\_alarm\_likelihood

0 Depot 12 0.00 0.00

1 Depot 16 0.00 0.00

2 Depot 20 0.00 0.00

3 Depot 24 0.00 0.01

4 Depot 28 0.01 0.00

5 Depot 32 0.00 0.01

6 Depot 36 0.01 0.00

7 Depot 4 0.00 0.00

8 Depot 8 0.00 0.00

9 HF 14 0.00 0.00

10 HF 15 0.27 0.08

11 HF 17 0.01 0.00

12 HF 18 0.01 0.02

13 HF 2 0.00 0.00

14 HF 21 0.00 0.00

15 HF 22 0.01 0.16

16 HF 23 0.00 0.00

17 HF 3 0.01 0.00

18 HF 33 0.00 0.00

19 HF 34 0.01 0.00

20 HF 6 0.01 0.00

21 PS 1 0.01 0.02

22 PS 10 0.00 0.00

23 PS 11 0.01 0.00

24 PS 13 0.01 0.02

25 PS 19 0.00 0.00

26 PS 25 0.00 0.00

27 PS 26 0.00 0.00

28 PS 27 0.01 0.48

29 PS 29 0.01 0.00

30 PS 30 0.01 0.00

31 PS 31 0.00 0.01

32 PS 35 0.01 0.00

33 PS 5 0.00 0.00

34 PS 7 0.03 0.08

35 PS 9 0.01 0.00

Questions and Answers about the Project and the Model

a. How well does your model work? What are its strengths and weaknesses?

The model appears to work fairly well. I would have to go back and create models with train vs. test sets to properly answer that question. One of its strengths is that it performed very well on the training data. Maybe it performed too well, memorizing the training data, which is one of its weaknesses. Another weakness is that I created it to predict the likelihood of alarm at the daily level rather than the weekly level, which was the request.

b. What does your model tell you about risk factors that impact alarms?

According to the variable importance plots, the mean temperature during the day is very predictive of alarms. A tentative takeaway from that is that it is best to keep the temperature of the fridge as stable as possible between the heat and cold thresholds to avoid alarms. Humidity had a smaller impact than temperature, light events had a smaller impact than humidity, and location had a smaller impact than light events.

c. If you had more time, what else would you like to have explored?

I would like to have tested the forecasting models and classification models more thoroughly. I also would have liked to devise other levels of time aggregation for forecasting and classification.

d. Can you briefly describe how your model might be put into production and any

concerns to be aware of?

Putting this model in production would require code that could automate the following:

* Extracting and compiling recent data, essentially real-time data
* Checking the data for quality
* Putting that data through a model to forecast metrics over the next 7 days
* Applying the classification model to the forecasted metrics
* Devising a proper way to get the likelihood of alarm during those 7 days
* Producing output in a location and format that could then be used to display the predictions

The production code would also need error handling functions.