Project 2: Change in Weather

Austin Derbique A01967241

Goal: The goal of this project is to determine how accurate the forecast models are of the National Weather Service. For this project, I chose San Diego, California as my location. This is what I chose for assignment 4 which is why I am using it in project 2. In addition to accuracy, the secondary goal is to monitor temperatures over time to see if in fact it is getting colder outside.

Methodology:

Data Collection: For this assignment I collected data for roughly two weeks from the period of November 15th to November 30th. To pull the weather information, I used an Amazon Web Server (AWS) ec2 which is a virtual private server running linux. Then, I created a cron job to execute my python script every 15 minutes. Temperature information is stored in the directory called temperature and weather data stored in the directory called weather. The script used for this is called collectdata.py

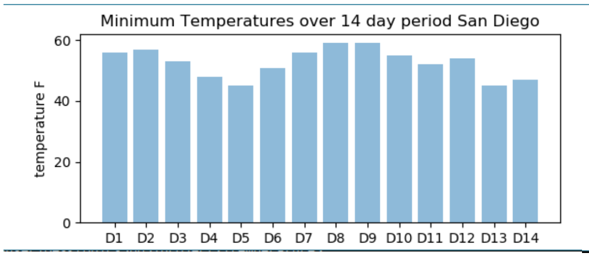
Data Parsing: For the parsing of data, I broke down the task into two parts. One for the daily min-max information and one for the three-hour predictions. Both are similar scripts but are modified to fit each other's needs. Essentially I create a dictionary with the key being the timestamp and the value being a list of predicted temperatures for that time stamp beginning with the actual measurement followed by the hourly or daily predictions. For instance: (pulled form daily_results.csv)

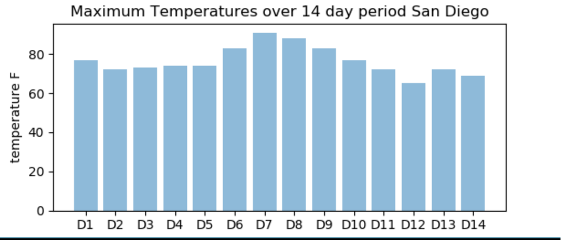
Date	d0	d1	d2	d3	d4	d5	d6
20171122	56	56	56	56	56	56	57

In the above table, d0 is the current measurement, d1 is the prediction for the day 1 day out,…,d6 is the prediction 6 days prior. This is the same format used for the 3 hourly data used in the other script. These scripts are called daily_min_max.py and three_hour_predictoins.py.

Data Processing: In order to calculate errors and other useful characteristics of this data, we must first process it. In the two scripts previously mentioned, I calculate the averages of the predictions and determine the error percentage based on that prediction and the actual recorded value for that time.

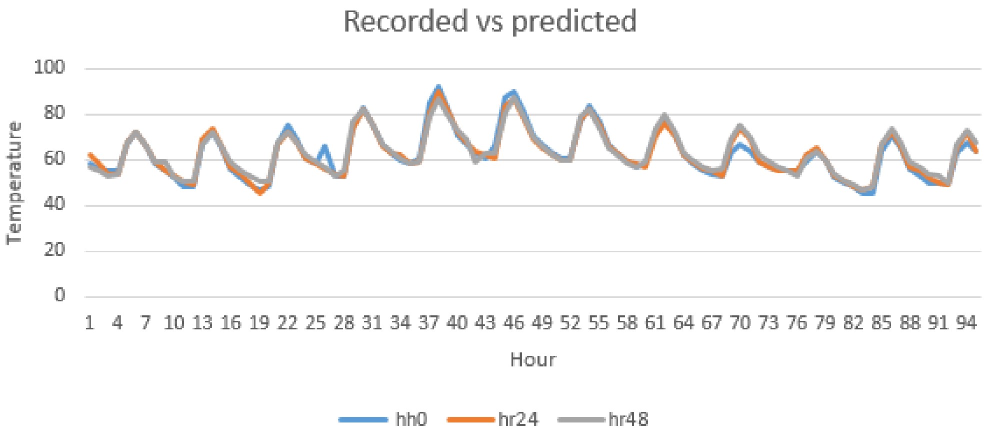
Results: There are several graphs displayed below. These are some of the results graphed in matplotlib and excel based on the data processed in the python scripts I wrote. First, we will see the daily min and max temperatures for one week:



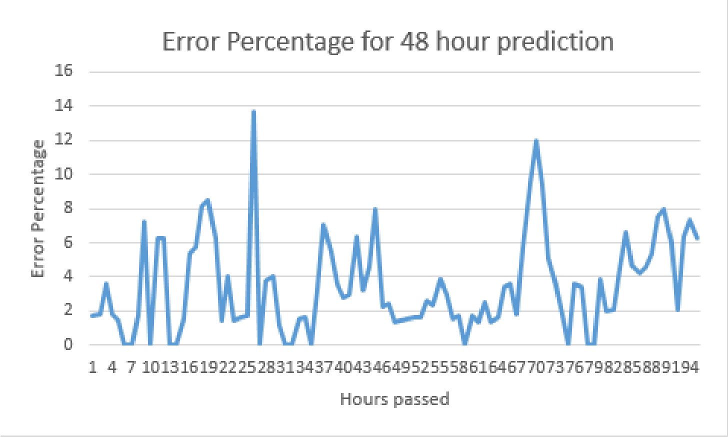


D1 is the starting date of collecting data and D14 is after the end of two weeks of collecting data. There appears to be a correlation between warmer days having a higher max and min and colder days having a cooler max and min for that given day. It is expected that there is not a great 'cooling' in the temperatures as this is San Diego where it is warm year round.

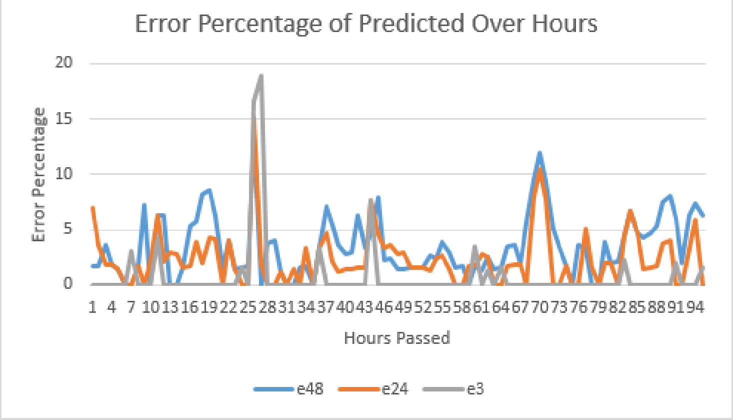
Next is a graph of the recorded air temperature for that day along with a 24 hour prediction of that temperature and a 48 hour prediction of that temperature.



They appear to be quite accurate. In order to get a better idea of the accuracy, error results are shown below:



This graph tells a different story. It can be seen that the error percentage is as bad as 14% on some days. This is for the 48 hour prediction. To see how predictions improve the closer to the actual time, one final graph is displayed.



The blue line is the error at 2 days out, orange is 1 day, and gray is 3 hours. It is evident that the forecast models become more accurate the closer to the time of measurement.

More results are available in daily_results.csv, hour_predictions.csv, and 48hour_predictions.csv.

Evaluation:

What can be seen from this data is that the accuracy of the predictions further our (48 hours, or 6 days) are less accurate than predictions closer to the recorded temperature at the target time (1 hour). This would make sense as models are not perfect and fine tune their predictions the closer to the given time.

It also makes sense that the weather did not change much being that the location recorded is Southern California. The results would likely be different if recorded in Montana or Utah.

What's Next?

Something that would be interesting is to take these modeling techniques and record data for an entire year. You might be able to see if predictions are more accurate in certain types of the year. You could also use these techniques to find stations with a high error percentage rate. This might be due to sensor issues or something related.