Predicting 2019's Inclement Weather Using Machine Learning Methods

May 17th, 2020

Andrew Thompson

Introduction

What was the prediction task and why is it important?

While the science of meteorology has existed since the mid-1800s[1], the act of trying to predict the weather has been around at least since Egyptian rain-ceremonies in 3,500 B.C.[2], if not since the dawn of humanity[3]. Today, various machine learning methods are used in varying degrees to predict the weather, often using the temperature and pressure on a given day in a given place.

The reason predicting *inclement* weather events in 2019 was chosen was for multiple reasons. One, being able to use machine learning methods to predict weather is a very useful task that even ancient peoples could appreciate; that is to say, it is a classic use of machine learning techniques to showcase what was learned in class. Two, specifically inclement weather in 2019 was chosen, because the data is readily available online, and a 'will this date have bad weather, yes or no' classifier would be easier than trying to do numerical weather prediction accurately with limited data (less data means less time to train models too). Three, the weather already happened. This makes 'predicting' and evaluating the effectiveness easier. Four, it is open-ended. In the future, the model(s) could be given multiple years of climate data and hopefully made more accurate. The task is worthwhile, while not overly ambitious.

This report contains the methods used, the results and discussion, a conclusion, acknowledgements, with references used at the end.

Methods

Describe the data: From where you got it, what all it included, what were the features and target, what pre-processing you needed to do.

From where:

The data was freely acquired using Google Cloud Platform's BigQuery[4] with guidance from one of the system's programmers Dylan Walseth[5] (Andrew's former roommate).

What it all included:

The raw data (about 5 gigabytes) included a variety of would-be features that had to be cut, which will be described in the pre-processing bolded section. The raw data, acquired through SQL queries limiting the datasets to data acquired daily in 2019, is hosted online at Reference 10.

The features and target:

The features included in the data set were bother numeric and nominal. The features were the numeric daily temperature (in Fahrenheit), numeric daily pressure (in millibars), the nominal date, and the nominal state (in the United States of America) where the measurement occurred.

The target was a "Yes" or "No" nominal value if inclement weather occurred that day. The definition for inclement weather (event_type) is in the next bolded section.

What pre-processing was needed:

A lot. The manual and automated pre-processing of data took at least 8 hours. The raw datasets for the National Oceanic and Atmospheric Administration's (NOAA) severe storms and the Environmental Protection Agency's (EPA) daily temperature measurements and daily pressure measurements were a couple million rows each. The unedited dataset for severe storms, included 33 columns including:

- episode_id: a unique integer for the weather event
- state: a string for the state's name. Includes non-states like, the District of Columbia, American Samoa, the North Atlantic, Lake Huron, and Lake Michigan.
- state_fips_code: Federal Information Processing Standard state code[6]
- event_type: the Meteorlogical term for the weather event. There were at least 55, including wildfires, droughts, debris flow, many kinds of floods, thunderstorms, many kinds of wind, tornados, water spouts, hurricanes, tropical storms, hail, heavy snow, ice storms, and many more
- wfo: Weather Forecasting Office[7]
- Multiple columns for the damages, both financial, number of people wounded, and number of people dead
- Multiple columns for the times of the event and the location components
- And more...

The pre-processing started on BigQuery by using an SQL query to limit each of the datasets to 2019. For example, the query done for the EPA's daily temperature measurements was,

SELECT * FROM `bigquery-public-data.epa_historical_air_quality.temperature_daily_summary` WHERE date_local > '2018-12-31' LIMIT 7000000

For each dataset, the remaining ~200,000 rows of data were then downloaded as a .csv. To process the .csv's into one final .csv with the 4 features and 1 target, a series of steps were taken.

Example pre-processing for the target data:

First, the ~25 least-useful columns were removed. The next step was to remove the duplicate episode_ids. This was done to remove multiple bad events in one day, like multiple lightning strikes within a tropical storm that later became a hurricane were many different rows. I only wanted to know a binary yes or no if that day had bad weather. To get the date, the start times were trimmed using Excel's =DATEVALUE(timestamp) command to turn 05/18/2019 3:43pm into 05/18/2019. Then the time was copy and pasted as a value to remove the equation and then duplicate dates removed to remove multiple unique events and leave only dates when at least 1 severe storm occurred. The data was then sorted alphabetically by state, then numerically by date. Thankfully, Excel had commands for most of this.

For the others, a similar process was followed. Leaving 3 separate .csv files that had to be manually merged by copy and pasting so that the states and dates lined up.

The resulting file had the temperature feature missing 1% of the days listed as severe storms, and the pressure feature missing 29% of the days listed as severe storms. While the severe storm list spanned almost all 365 days in a year, no state provided more than 10 months of temperature and pressure data,

and many states provided no data for pressure. To make matters worse, there were at least 4 states that had gaps in the data reported, one as large as two separate months missing and as small as 1 day missing from the chunk of data reported if it was data report. Once properly merged and sorted, the file of 12,585 rows was ready to be loaded into Weka[8].

In Weka, replaceMissingValues was used to fill the remaining gaps with the average value. The data was ready to be classified.

Classification Methods

J48, Naïve Bayes, Bayes net, random forest, k-nearest neighbors, and a multilayer perceptron were used on the 4 features, (nominal) state & date, (numeric) temperature & pressure, to target if inclement weather occurred given the features. Each classifier used the default values with 10-fold cross-validation. The following section shows the results and discusses them.

Results

Include results of data exploration.

The first results were the trends Weka showed in the data (without classifying yet).

50 states and Washington D.C. were included in the states, and 359 days were covered.

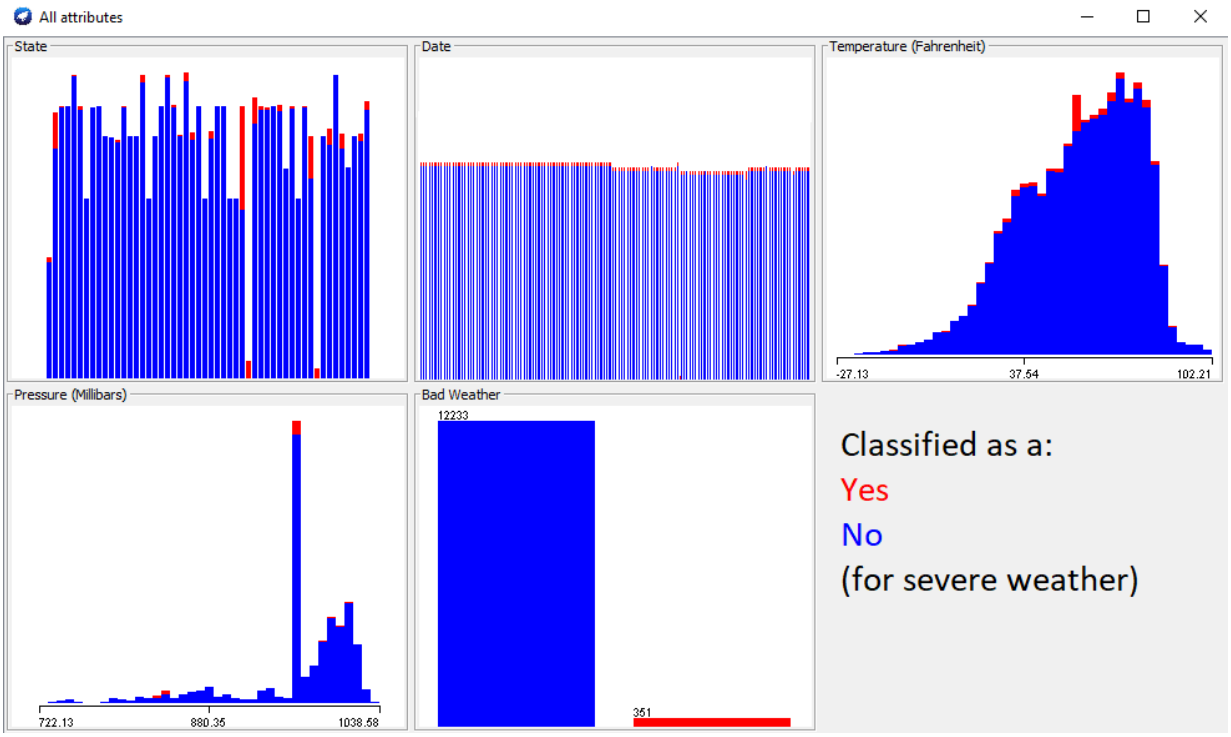

Figure 1. Weka's graphs for the data inputted. Upper left is State, top middle is Date, upper right is Temperature in F., bottom left is Pressure in millibars, bottom middle is Bad Weather classification.

From the graphical representation, some basic data exploration can be done. The state chart shows how irregular the distribution of yeses is. This implies that either some states are magnets for severe weather as defined by the EPA, or perhaps more likely, that reporting standards vary from state to state. In the date chart, you can see how the reporting of weather drops off after about 6 months. One red blip near the bottom middle shows a mistake in my data, 'New mexico' and 'New Mexico' are used, as is 'South dakota' and 'South Dakota'. The temperature chart shows how many of the severe weather events also occur during days with common temperatures. The pressure chart is the easiest to see an issue in. Because there were many missing values, and those values were replaced with the average value, it is easy to see the average being the plurality on the graph. The last chart shows the classification. About 97% of the days are not classified as severe weather days.

What machine learning techniques did you apply and using what tool/software/program? Why did you choose those techniques and tool?

In Weka, J48, Naïve Bayes, Bayes net, random forest, k-nearest neighbors, and a multilayer perceptron were used. I chose a variety of techniques so that I may compare which would do the best job. The multilayer perceptron was added in for fun- it took 5065.99 seconds to build the model, and another 14 hours to cross-validate the folds on my top-of-the-line consumer CPU from 3 years ago (Intel Core i7 7700K). Weka was chosen because I was most familiar with it.

Include evaluation results in the form of tables or graphs.

Selected resulting statistics from each classifier are shown in Table 1 and were used to determine the preferred machine learning method for the task.

Classifier	Weighted Avg. ROC	Correctly Classified Instances	Incorrectly Classified Instances	Relative absolute error	Time (s) to Model:
J48	0.499	97.21%	2.79%	99.86%	0.01
Naïve Bayes	0.874	96.07%	3.93%	93.90%	0
Random Forest	0.893	97.43%	2.57%	81.36%	0.73
Bayes Net	0.867	97.54%	2.46%	66.17%	0.06
kNN	0.699	96.75%	3.25%	60.34%	0
MultiPerceptron	0.663	97.19%	2.81%	52.37%	5065.99

Table 1. Some results from the classifiers. Best result for each column is highlighted.

The Naïve Bayes, Bayes Net, and Random Forest have the best ROCs. Most seem competitive for correctly classifying instances. The multilayer perceptron has the least relative absolute error, however, in terms of time-efficiency, any of the methods besides the multilayer perceptron are preferred. The confusion matrices tell us more, as shown in Figure 2.

J48:

a	b	<-- classified as
12233	0	a = No
351	0	b = Yes

Naïve Bayes:

a	b	<-- classified as
11998	235	a = No
260	91	b = Yes

Random Forest:

a	b	<-- classified as
12233	0	a = No
323	28	b = Yes

Bayes Net:

a	b	<-- classified as
12113	120	a = No
189	162	b = Yes

kNN:

a	b	<-- classified as
12025	208	a = No
201	150	b = Yes

Multilayer Perceptron:

a	b	<-- classified as
12196	37	a = No
317	34	b = Yes

Figure 2. The confusion matrices for each classifier.

With these results, there is enough information to draw some useful conclusions.

Discussion

Discussion about the results (do some error analysis if possible) and conclusions.

From the results, insights can be drawn. For example, J48 did very well just by classifying each day not having bad weather, perhaps because the chance of weather being severe is so low. A primitive, yet valid strategy for predicting inclement weather. The Naïve Bayes, Random Forest, and Bayes Net each did well, and the ROCs for each reflect this. Which did the best of those three? Well, firstly, we can kick out the Naïve Bayes because the Bayes Net and Random Forest had a lower relative absolute error. Naïve doing worse than the Net is due to Naïve Bayes requiring that the features are independent, but air temperature affects air pressure! Guillaume Amontons showed this near the end of the 1600s, and his equation Amonton's Law, with three others, was incorporated into the famous ideal gas law[9]:

$$PV = nRT \qquad \text{(equ. 1)}$$

This means that Naïve Bayes will always under-perform the Bayes Net with these features! This leaves using either the Random Forest or Bayes Net to predict severe weather. Either could work well. While the Random Forest has a higher relative absolute error, whenever it says there will be severe weather, there is a 100% chance (1.00 precision) that there will be severe weather. With the Bayes Net, it certainly does the best for predicting when severe weather occurs, however, it can only find about half of the cases when it happens which could lead to distrust in weather predictions from the uninformed masses. Bayes Net is probably more preferable. More testing could be done over multiple years of data to show this in the future.

Issues with the data:

Firstly, the inclement weather definition seems to vary a lot state to state. This is especially a problem if this data is reported and defined manually. Likely much worse, for whatever reason the temperature and pressure datasets were not complete. This caused the use of an average value to replace missing values. If dates without temperature and pressure data (but classified as 'yes') were removed, perhaps this could have been remedied. The dates without temperature and pressure classified as 'no' were not included in the initial measurement. This likely skewed the data on top of already missing some data in two key features.

Conclusion

I was amazed that even with so many gaps in the data, that the methods classify so well. This project took some time to do, though, I really enjoyed it. It seems that Random Forest and Bayes Networks work well to classify what days severe weather may occur given the date, state, temperature, and pressure.

Acknowledgements

Thank you to Dylan Walseth for helping me with Dylan's work project at Google. Finding good datasets that are relevant is always difficult and BigQuery made my life easier. Thank you, Dr. Kate, for teaching the machine learning course!! I had a lot of fun this semester, and I appreciate everything you have done. I would have failed the class had you not given me my first ever (in my school 'career') extension (of 14 hours) on assignment 3. I pass this class and graduate this semester because of this. I cannot thank you enough.

References

1. Czerski, H. BBC - Orbit: Earth's Extraordinary Journey: 150 years since the first UK weather 'forecast'. https://www.bbc.co.uk/blogs/23degrees/2011/08/150_years_since_the_first_uk_w.html.
2. Wainwright, G. A. *The Sky-Religion in Egypt: Its Antiquity & Effects*. (Cambridge University Press, 1938).

Frisinger, Howard. "Meteorology Before Aristotle," n.d., American Meteorlogical Society.

4. BigQuery: Cloud Data Warehouse. *Google Cloud* https://cloud.google.com/bigquery.
5. Dylan Walseth | LinkedIn. https://www.linkedin.com/in/dylan-walseth-a0a8ab7b/.
6. FIPS 5-2 - State Codes of the US and Outlying Areas. https://web.archive.org/web/20090705054444/http:/www.itl.nist.gov/fipspubs/fip5-2.htm (2009).

7. US Department of Commerce, N. NWS Weather Forecast Offices.

https://www.weather.gov/srh/nwsoffices?site=tae.

8. WEKA. https://www.cs.waikato.ac.nz/ml/weka/

9. Gas Laws. http://chemed.chem.purdue.edu/genchem/topicreview/bp/ch4/gaslaws3.html.

10. Hosted online here: https://drive.google.com/open?id=15eGD9ETuNCstGm0Bm-_7iHLSYD7QQffA (includes the "WeatherPredictionWeka.csv" which was the dataset used for the classifier).