Naive Wildfires

Bryce Anderson, Chad Baxter, Bradley Dufour

April 12, 2018

1 Initial review

Upon receiving the data set on wildfires an initial analysis showed that it was a geospatial database for use with ArcGIS. This means it included not just data but shapes, geometries and other cartography related data. It would have been interesting to write this data challenge as an ArcGIS plugin. Since geospatial analysis is not a concern any non-essential cartographic data can be ignored.

2 Processing

Immediately the data could be stripped down to a few important columns. This insured the Naive Bayes classifier did not over generalize when the data was one-hot encoded. The most painful part of this process was normalization. Once the data was one-hot encoded the year column greatly overpowered the rest of the data. The simple fix for this was to subtract the oldest year from every other year leaving a year range of about 0–23. This reduced the effect of the year field on the classifier. The best option for the duration of a fire was days. Thankfully the fire discovery date and fire contained date were stored in the database as Julian Days. This made it easy to convert the two dates into a time span of days by subtracting the discovery date from the containment date.

The data set provided had many errors including having fires with a duration longer than three years. The group decided to remove any duration outliers that were longer than one hundred days. There were many null values which had to be handled. Because duration was so important any row that was missing either the discovery date or the date the fire was contained were thrown out.

3 Classifiers

SciKit Learn was used to create the Naive Bayes classifier. The module *GaussianNB* allowed us to model the data with a standard Bayes function. The categorical data had to be one-hot encoded for SKLearn to accept it, this was done using the *get_dummies* function of Pandas. Originally the columns that were kept included the FIPS code for each fire. This allowed for fine grained precinct based identification of location but also expanded too much during one-hot encoding. It was removed in favor of the *state* data.

Unfortunately the predictor was not as flexible as a self implemented one. This created a lot of friction when defining a *predict* function. It was not as simple to provide the output designed in the specification as SKLearn kind of eats the targets / labels / answers and does not give them back. For that reason the predict function does not quite match that of the specification.

4 Conclusions

Based on the results Naive Bayes is a bad predictor of wildfires. At least for this dataset. With the amount of cleaning and mixed data-types as well as the large amount of columns Bayes was too general. The data was under-fit.

5 Reflection

As a reflection, SKLearn was a bad choice. it did what it was supposed to, but not the way the group wanted or needed it. It was unfortunate that time ran out for the implementation of other classifiers.