Eli Bolotin
DS 740: Final Project
August 5, 2019

Executive Summary

1. Introduction

This summary describes a machine learning project that is based on the *Rain in Australia* [7] dataset from Kaggle. The goal of this project is to maximize rain prediction accuracy and analyze the importance of data features. The purpose of this document is two-fold: (1) summarize project results and (2) relate these results to the work being done in Atmospheric Sciences division at NASA.

2. About the data

In its original preprocessed form, the aforementioned dataset contains 142k observations, each of which represents a day of weather in various regions of Australia. The observations are ordered chronologically and span 10 years from late 2007 to 2017. There are 23 features in this dataset, each representing a traditional weather statistic. The target variable, 'RainTomorrow', is a binary indicator of whether there will be rain on a subsequent day. The goal of this project is to predict this target variable in order to discover which features are most meaningful in predicting rain.

3. Discussion of features

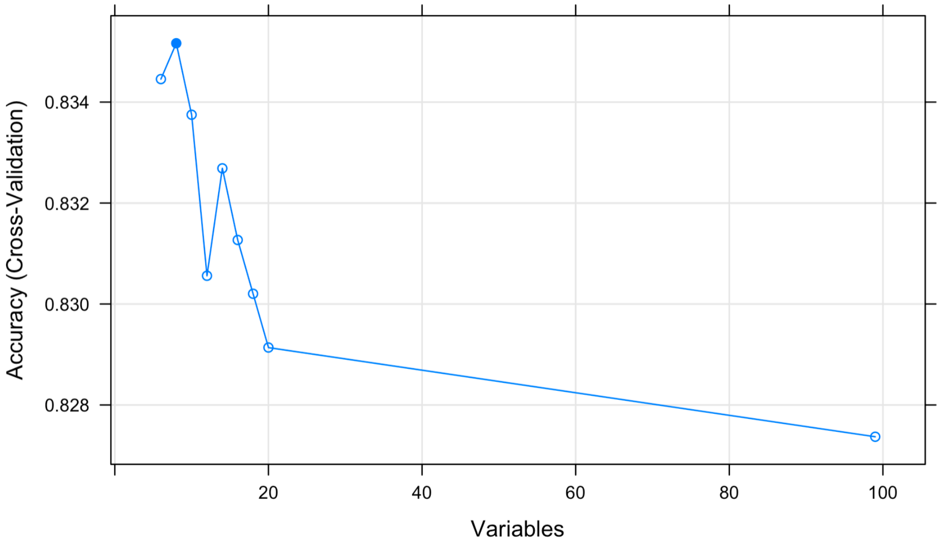
Out of the 23 variables in the original dataset, 6 are categorical variables (aka factors) and the rest are numeric. The factors include *date*, *location*, *wind gust direction* ("the direction of the strongest wind gust in the 24 hours to midnight" [1]), *wind direction* at 9am and 3pm, and *RainToday* (whether it rained today). The 17 numeric variables include daily weather statistics such as: *min/max temperatures, rain fall, evaporation, sunshine, wind gust speeds, wind speeds, humidity, atmospheric pressure, clouds* ("fraction of sky obscured by clouds"), and *air temperature*. The latter five features are recorded twice daily at 9am and 3pm.

As is the case with most datasets, not all predictor variables are useful in predicting the target variable. During the data preprocessing and transformation stage of this project, 2 predictors were dropped, and 4 new predictors created. *Risk_MM* ("the amount of next day rain in MM") was dropped to avoid introducing bias; and *date* was dropped because it is useless in predicting whether it will rain specifically tomorrow. If, however, we were predicting whether there will be rain in a given season or month, date may have been a useful predictor (because dates are partially associated with season intervals). Nevertheless, to test for the possibility that previous weather can "incite" future weather, 4 new features were created. 3 of these features are different lags of *RainToday*, and the fourth feature is *percent of prior days rain*. These columns were intended to test for the (highly unlikely) possibility that merely because it rained yesterday, it might also rain today.

The cleaned and processed training set contains 100 distinct features, the majority of which are encoded versions of categorical columns. After data processing/transformation, feature selection was performed using backwards selection with random forest modeling.

⇒ **Results of feature selection**:

- The optimal number of predictors is 8. These predictors are: *Humidity at 3pm, Sunshine, Wind Gust Speed, Pressure at 3pm, Cloud at 3pm, Pressure at 9am, Rainfall, and Humidity at 9am*
- Prediction accuracy does not substantially increase with the number of predictors used and may even decrease depending on the training sample used.



- The most parsimonious model of 6 predictors (acc = 0.8345) is actually slightly better than the most complex model of 99 predictors (acc = 0.8274).

4. Selection of modeling techniques

Time series autoregressive modeling?

As discussed in the previous section: although the *Rain in Australia* dataset has a date field, using time-series autoregressive forecasting to predict rain is unlikely to yield accurate results. This is because weather and climate patterns are governed by purely physical processes. These events do not occur according to a predetermined cycle that can be predicted accurately with temporal autoregression.

For instance, a region of Europe that typically experiences a continental climate (hot summers, cold winters) may also experience the near-opposite seasonal climates (cool summers, warm winters) for extended periods, and may even have blended seasons. This is why predicting weather events such as rain or snow on the basis that they merely happen in the same region at different points in time will produce inaccurate predictions. I tested this notion by creating the “new predictors” described in section 3. As evident from the results of recursive feature elimination, these predictors were not useful.

Note: An autoregressive technique that might be useful in predicting weather is spatial autoregression (based on spatial autocorrelation), which “measures and analyzes the degree of dependency among observations in a geographic space” [8].

Chosen methods

The choice of training methods for this project is based on 3 primary reasons:

- The decision boundary between classes (*rain tomorrow* = yes/no) is linear.
- There are many features (100) and we only want to select the important ones, automatically. That is, we want to compare an embedded method.
- Data is relatively wide (100 predictors) and deep (~53.5k training observations), so we want use parallel processing.

Among the many possible choices, the primary methods compared in this analysis are bagging and support vector machines (SVM). Bagging (bootstrap aggregation) is an ensemble method based on decision trees. SVM is a generalization of the maximal margin classifier [5]. The advantages in using each method are the following:

Advantages of Bagging [4]

- Decreases variance inherent to decision trees
- Uses bootstrapping to generate many training sets and averages results
- Can handle qualitative variables
- Can be trained in parallel

Advantages of SVM

- Linear kernel
- Tunable with cost parameter to avoid overfitting
- Can be trained in parallel
- Efficient algorithm

In addition to bagging and SVM, this project also compared 2 secondary methods: neural networks and LDA (linear discrimination analysis).

⇒ Results of modeling:

- The best model is a linear SVM with cost = 0.81, having a prediction accuracy on the test data of 85.37%.
- The 2nd best model was a neural network of 3 layers, followed by LDA, and then bagging.

5. Conclusion

Results

The methodology and process used in this analysis results in:

- A prediction accuracy of ~85.4% on the test data of ~10.7k observations (~25% of the training data size of ~43k).
- A prediction accuracy of ~85.3% on the full data of ~53.6k observations.

The best model was an SVM with a cost of 0.81, and was trained on the 8 predictors mentioned in section 3.

Thoughts & Takeaways

As of 2018-19, "a three-day forecast today is as good as a one-day forecast 10 years ago, thanks to the massive computing power of supercomputers that can consolidate trillions of data points on atmospheric conditions into simple simulations" [6]. So, even supercomputer generated models, fit on enormous datasets - cannot predict weather reliably more than 3 days in advance. The challenge is that “many crucial weather phenomena, such as precipitation, are largely determined by cloud processes, which occur on much smaller [and more dynamic] scales” [6] than are currently being observed and modeled.

Therefore, to predict weather with greater than 85% accuracy requires more detailed data. Important features include [2][3]:

1) Measurements of traditional weather variables at different altitudes
2) Chemical composition of air
3) Oceanic readings such as fluid levels
4) Soil and surface level moisture readings

All of these variables need to be captured frequently – ideally constantly (i.e. live) – to maximize the reliability of a model. Furthermore, in addition to having the right predictors, for forecasting to be effective, the model’s parameters/weights need to be continuously updated as weather conditions change. So, the model should be designed with continuous/live backward propagation in mind.

NASA’s Mesoscale Modeling and Dynamics Group is currently developing such models, based on the investigation of “cloud and precipitation systems from the scale of individual clouds and thunderstorms through mesoscale convective systems and cyclonic storms, and up to the scale of the impact of these systems on regional and global climate”[9].

Notes regarding Kaggle:

- I noticed that on Kaggle, some people posted kernels with ~99% accuracy, but had incorrectly pre-processed their train/test data by encoding certain categorical features as ordinal variables. The results of such kernels are obviously invalid.
- Other kernels had not dropped the column *Risk_MM* (which was meant to be excluded).

6. References

- [1] Rain in Australia dataset, https://www.kaggle.com/jsphyg/weather-dataset-rattle-package
- [2] Weather variables, https://content.meteoblue.com/nl/specifications/weather-variables
- [3] Nat. Geo Weather https://www.nationalgeographic.org/encyclopedia/weather/
- [4] p. 316-317, An Introduction to Statistical Learning: With Applications in R
- [5] p. 337, An Introduction to Statistical Learning: With Applications in R
- [6] Why Predicting Weather is Still So Difficult, https://observer.com/2018/03/weather-forecast-predictions-still-difficult/
- [7] Rain in Australia dataset, https://www.kaggle.com/jsphyg/weather-dataset-rattle-package
- [8] Spatial Autocorrelation, https://en.wikipedia.org/wiki/Spatial_analysis#Spatial_autocorrelation
- [9] NASA Mesoscale Modeling and Dynamics Group, https://cloud.gsfc.nasa.gov