DS Final Poject

JD

January 31, 2019

Introduction

The goal of this exercise is to build a classification model to predict whether the relative humidity will be high or low at 3PM based on several predictors captured from weather readings at 9AM. For this exercise high humidly is defined as relative humidity greater than or equal to 24.99%

The dataset was downloaded from Kaggle.
https://www.kaggle.com/ktochylin/san-diego-every-minute-weather-indicators-201114/downloads/san-diego-every-minute-weather-indicators-201114.zip/1
The dataset contains weather readings from 2011-09-10 to 2014-09-10 for San Diego, CA. Readings were taken at 1-minute intervals. There are 1.59M rows and 13 features.

The R Package Ada for Stochastic Boosting was used. The package was authored Mark Culp, Kjell Johnson, and George Michailidis and is maintained by Mark Culp
mvculp@mail.wvu.edu

Package description: Performs discrete, real, and gentle boost under both exponential and logistic loss on a given data set. The package ada provides a straightforward, well-documented, and broad boosting routine for classification, ideally suited for small to moderate-sized data sets.

Analysis

Rows with missing features have been dropped.

Since we only need data for the 9AM observations and the relative humidity at 3PM, we take a subset of the data.

The 9AM data is extracted with all features and combined with the relative_humidity3pm which was also extracted.

The feature HighHumidity3pm is derived from relative_humidity3pm and added to the new dataset.

There are 1080 records in the cleaned dataset.

Train and Test data sets were created. The test set is 10% of the data.

There are 972 records in the training dataset.

There are 108 records in the test dataset.

Train 3 models using ADA package. Each using a diffent algorithm to determin which will work best with our data.

Results using the algorithm “discrete”.

```
## Call:
## ada(HighHumidity3pm ~ air_pressure9am + air_temp9am +
avg_wind_direction9am +
##     avg_wind_speed9am + max_wind_direction9am + max_wind_speed9am +
##     min_wind_direction9am + min_wind_speed9am + rain_accumulation9am +
##     +rain_duration9am + relative_humidity9am, data = train_set,
##     iter = 50, type = "discrete")
##
## Loss: exponential Method: discrete   Iteration: 50
##
## Training Results
##
## Accuracy: 0.961 Kappa: 0.862
```

Results using the algorithm “real”.

```
## Call:
## ada(HighHumidity3pm ~ air_pressure9am + air_temp9am +
avg_wind_direction9am +
##     avg_wind_speed9am + max_wind_direction9am + max_wind_speed9am +
##     min_wind_direction9am + min_wind_speed9am + rain_accumulation9am +
##     +rain_duration9am + relative_humidity9am, data = train_set,
##     iter = 50, type = "real")
##
## Loss: exponential Method: real   Iteration: 50
##
## Training Results
##
## Accuracy: 0.94 Kappa: 0.777
```

Results using the algorithm “gentle”.

```
## Call:
## ada(HighHumidity3pm ~ air_pressure9am + air_temp9am +
avg_wind_direction9am +
##     avg_wind_speed9am + max_wind_direction9am + max_wind_speed9am +
##     min_wind_direction9am + min_wind_speed9am + rain_accumulation9am +
##     +rain_duration9am + relative_humidity9am, data = train_set,
##     iter = 50, type = "gentle")
##
## Loss: exponential Method: gentle   Iteration: 50
##
## Training Results
##
## Accuracy: 0.969 Kappa: 0.89
```

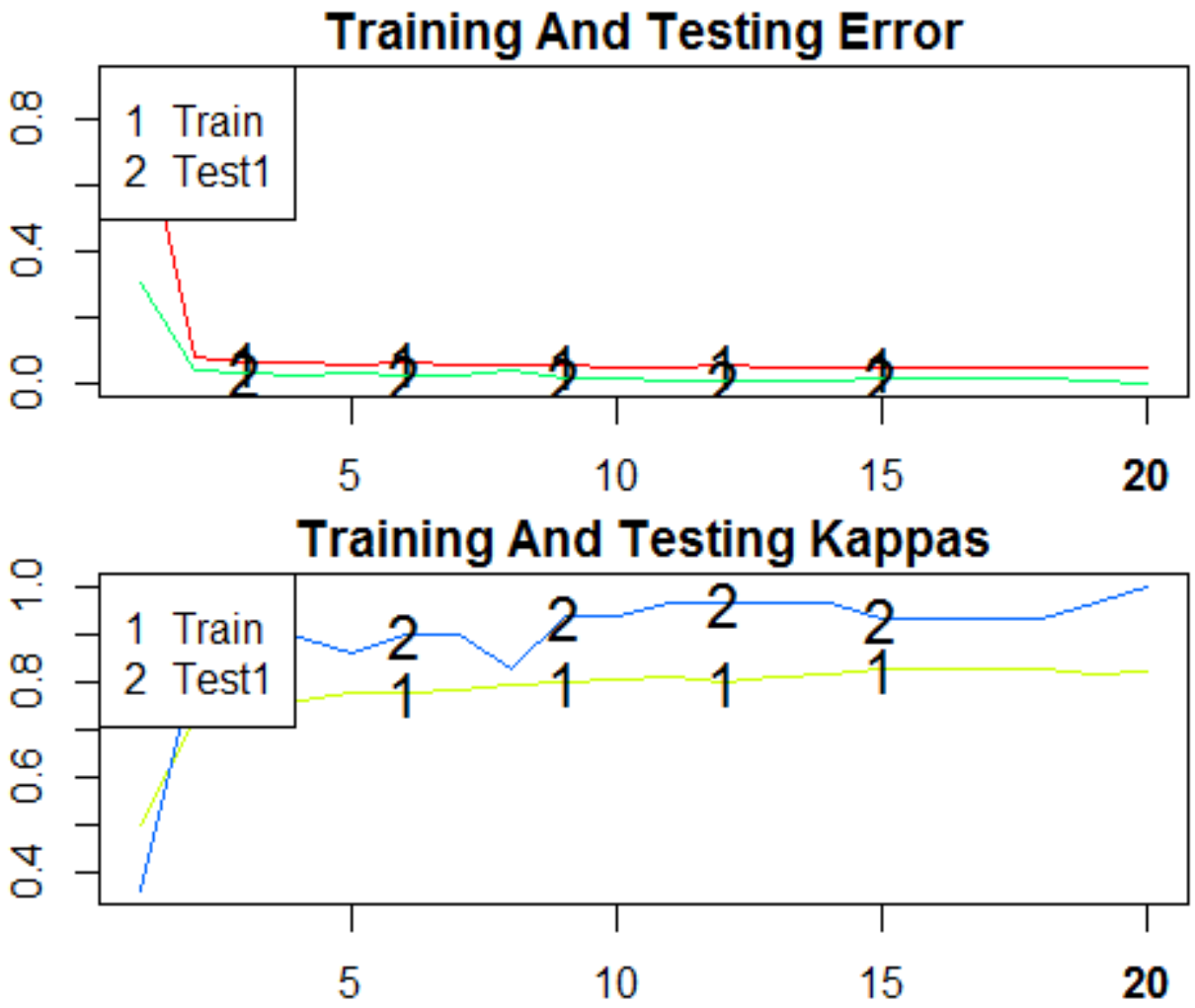
The “gentle” algorithm appears to be the most accurate for this data.

Let’s build 3 additional models with the “gentle” algorithm using different iteration values. This time we will also add the test data to the model so we can see our accuracy and kappa (*the measure of agreement between the predicted classification and actual classification*) for our test set when we use the trained models.

We will show the results and plot the Error and Kappa against the iterations.

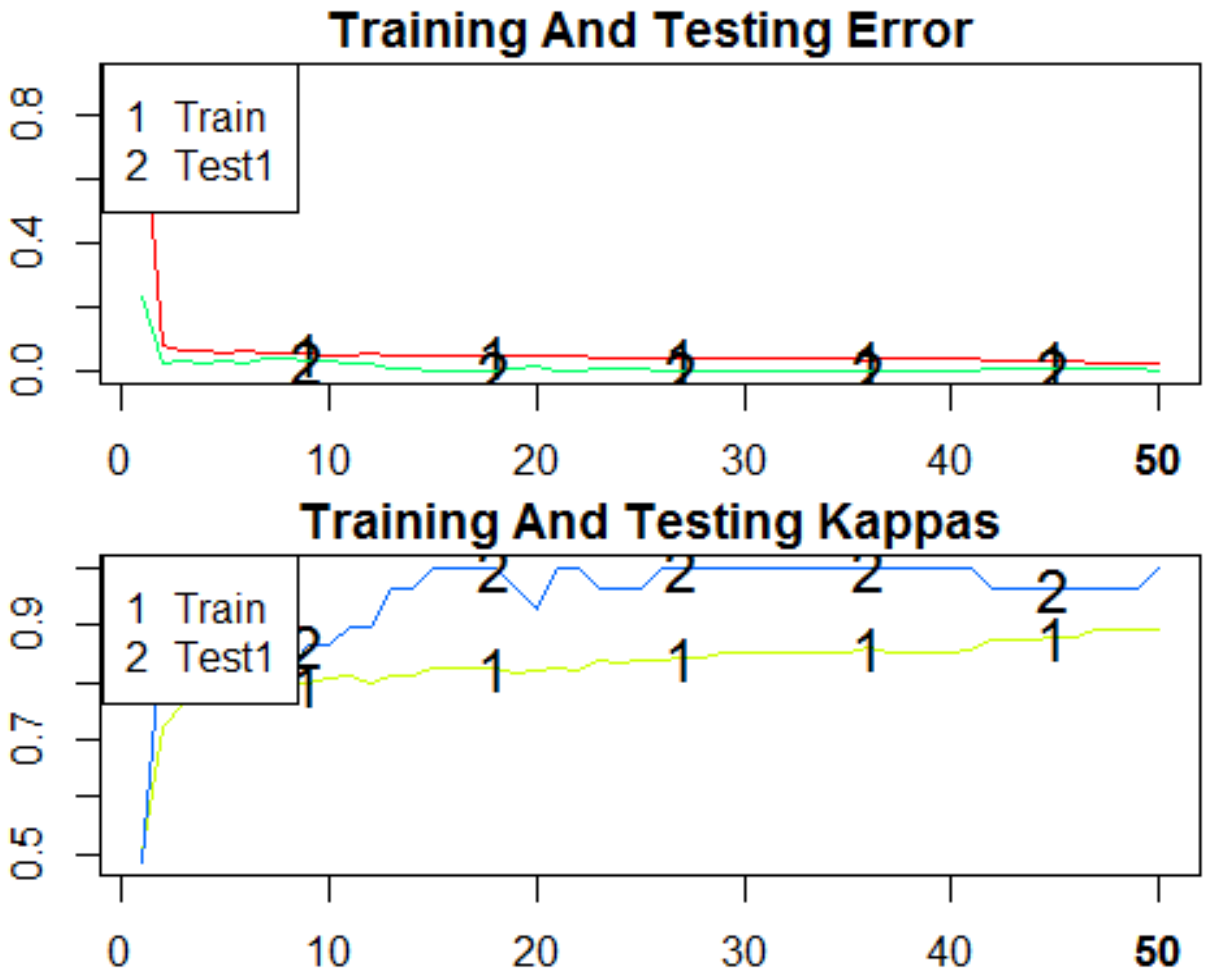
Results using 20 iterations.

```
## Call:
## ada(HighHumidity3pm ~ air_pressure9am + air_temp9am +
avg_wind_direction9am +
##     avg_wind_speed9am + max_wind_direction9am + max_wind_speed9am +
##     min_wind_direction9am + min_wind_speed9am + rain_accumulation9am +
##     +rain_duration9am + relative_humidity9am, data = train_set,
##     iter = 20, type = "gentle")
##
## Loss: exponential Method: gentle   Iteration: 20
##
## Training Results
##
## Accuracy: 0.95 Kappa: 0.821
##
## Testing Results
##
## Accuracy: 1 Kappa: 1
```



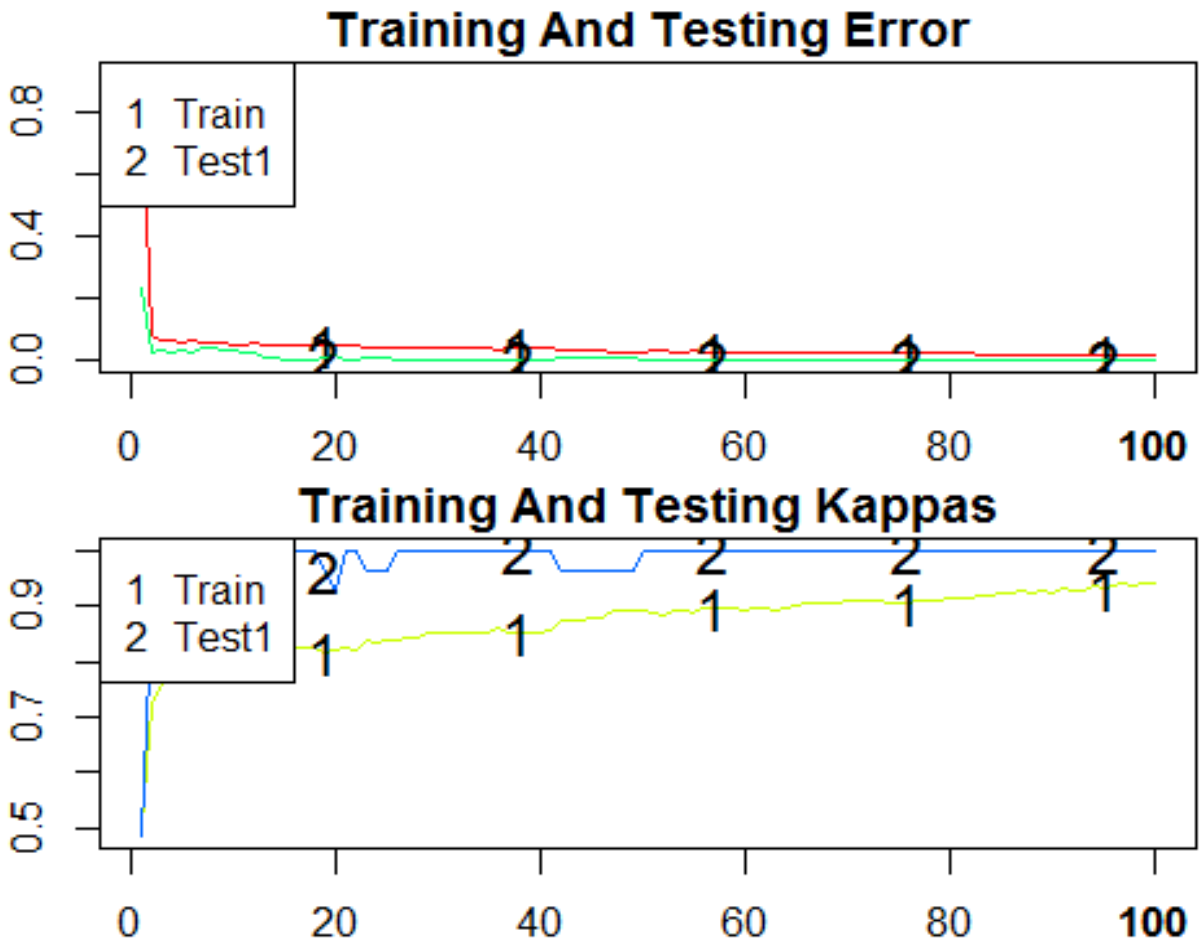
Results using 50 iterations.

```
## Call:
## ada(HighHumidity3pm ~ air_pressure9am + air_temp9am +
avg_wind_direction9am +
##     avg_wind_speed9am + max_wind_direction9am + max_wind_speed9am +
##     min_wind_direction9am + min_wind_speed9am + rain_accumulation9am +
##     +rain_duration9am + relative_humidity9am, data = train_set,
##     iter = 50, type = "gentle")
##
## Loss: exponential Method: gentle   Iteration: 50
##
## Training Results
##
## Accuracy: 0.969 Kappa: 0.89
##
## Testing Results
##
## Accuracy: 1 Kappa: 1
```



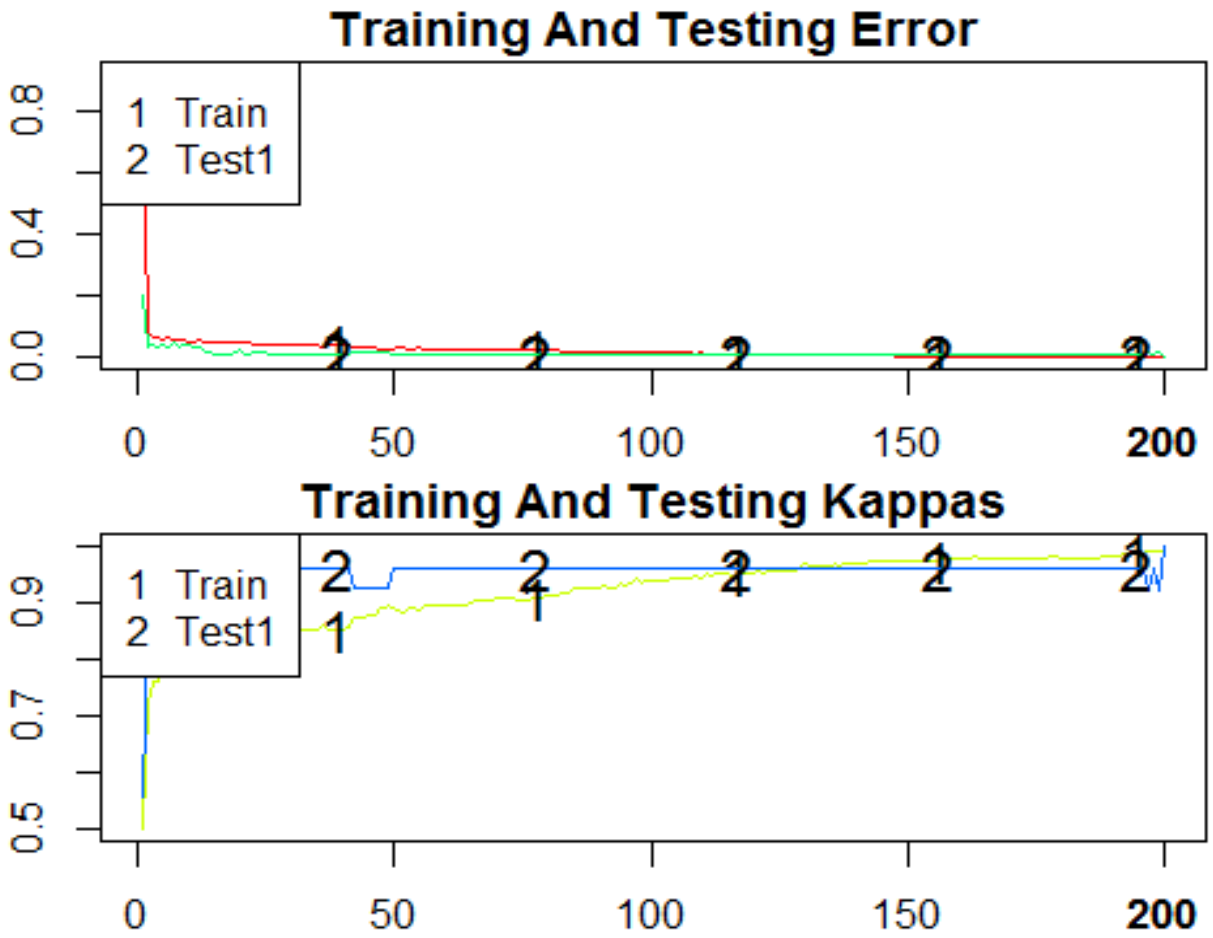
Results using 100 iterations.

```
## Call:
## ada(HighHumidity3pm ~ air_pressure9am + air_temp9am +
avg_wind_direction9am +
##     avg_wind_speed9am + max_wind_direction9am + max_wind_speed9am +
##     min_wind_direction9am + min_wind_speed9am + rain_accumulation9am +
##     +rain_duration9am + relative_humidity9am, data = train_set,
##     iter = 100, type = "gentle")
##
## Loss: exponential Method: gentle   Iteration: 100
##
## Training Results
##
## Accuracy: 0.983 Kappa: 0.94
##
## Testing Results
##
## Accuracy: 1 Kappa: 1
```



Results using 200 iterations.

```
## Call:
## ada(HighHumidity3pm ~ air_pressure9am + air_temp9am +
avg_wind_direction9am +
##     avg_wind_speed9am + max_wind_direction9am + max_wind_speed9am +
##     min_wind_direction9am + min_wind_speed9am + rain_accumulation9am +
##     +rain_duration9am + relative_humidity9am, data = train_set,
##     iter = 200, type = "gentle")
##
## Loss: exponential Method: gentle   Iteration: 200
##
## Training Results
##
## Accuracy: 0.998 Kappa: 0.993
##
## Testing Results
##
## Accuracy: 1 Kappa: 1
```

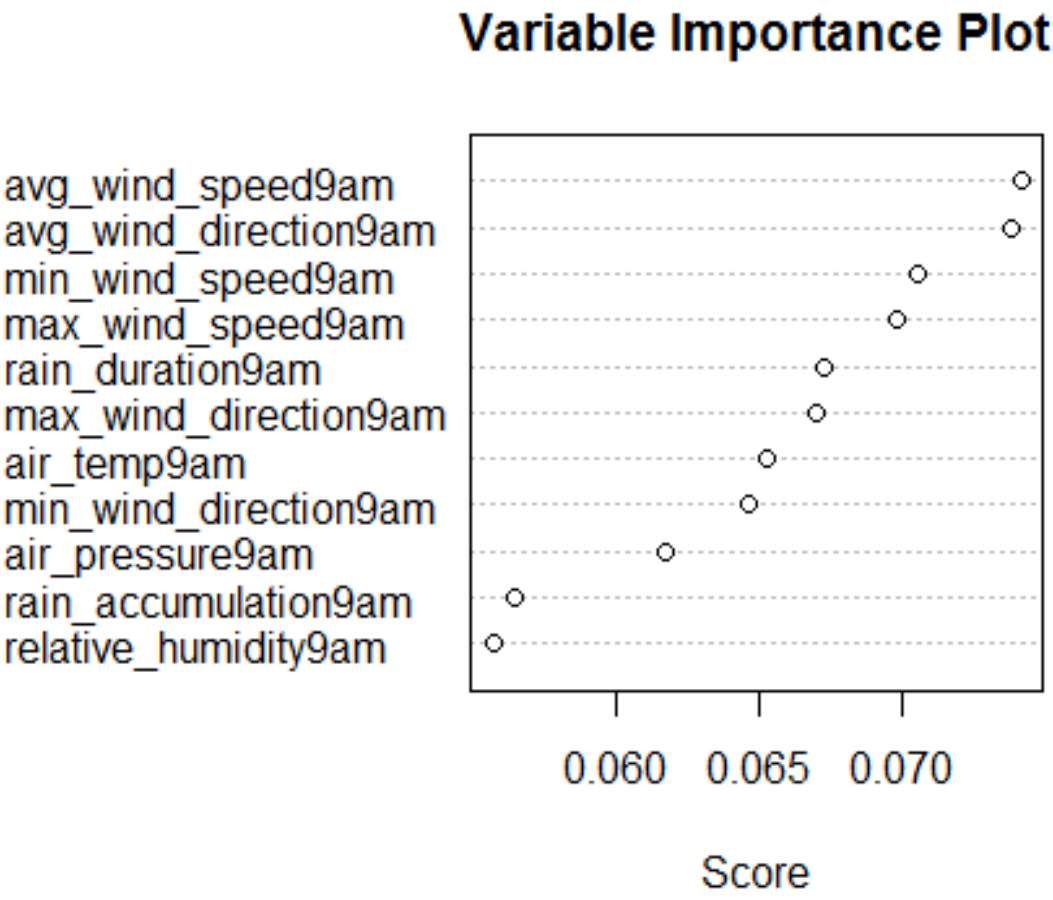


Conclusion

We created models using 3 different algorithms from just one of many available packages. Each algorithm yielded different results thus showing that not all algorithms will perform equally well on a given dataset. Experimentation with different algorithms is required to obtain the best results.

Most models can be tuned. We only experimented with one tuning parameter here. The results show how changing tuning parameters can affect the outcome. In this case the more iterations we go though in training the model the higher the accuracy. However, as we set iterations higher the improvement in accuracy grows smaller. As we go beyond 100 iterations the incremental improvement greatly diminishes. The rate of improvement in the error drops of even earlier in less than 10 iterations.

One final observation we made is that not all features have the same predictive power. The graph below show which features has more and less influence on the predictions.



References:

Introduction to Data Science. Author Rafael A. Irizarry
https://rafalab.github.io/dsbook/