Capstone Blog Post

Predicting 'Down' Turbines With Real Wind Farm Data

Max Davis

10/20/2019

The Problem

Is it possible to predict when a wind turbine will not produce power when it should? That is the problem this project seeks to address. Wind turbines producing less than optimally is a natural problem for a wind farm, and insights about this problem may be gained from a large dataset on many turbines over time.

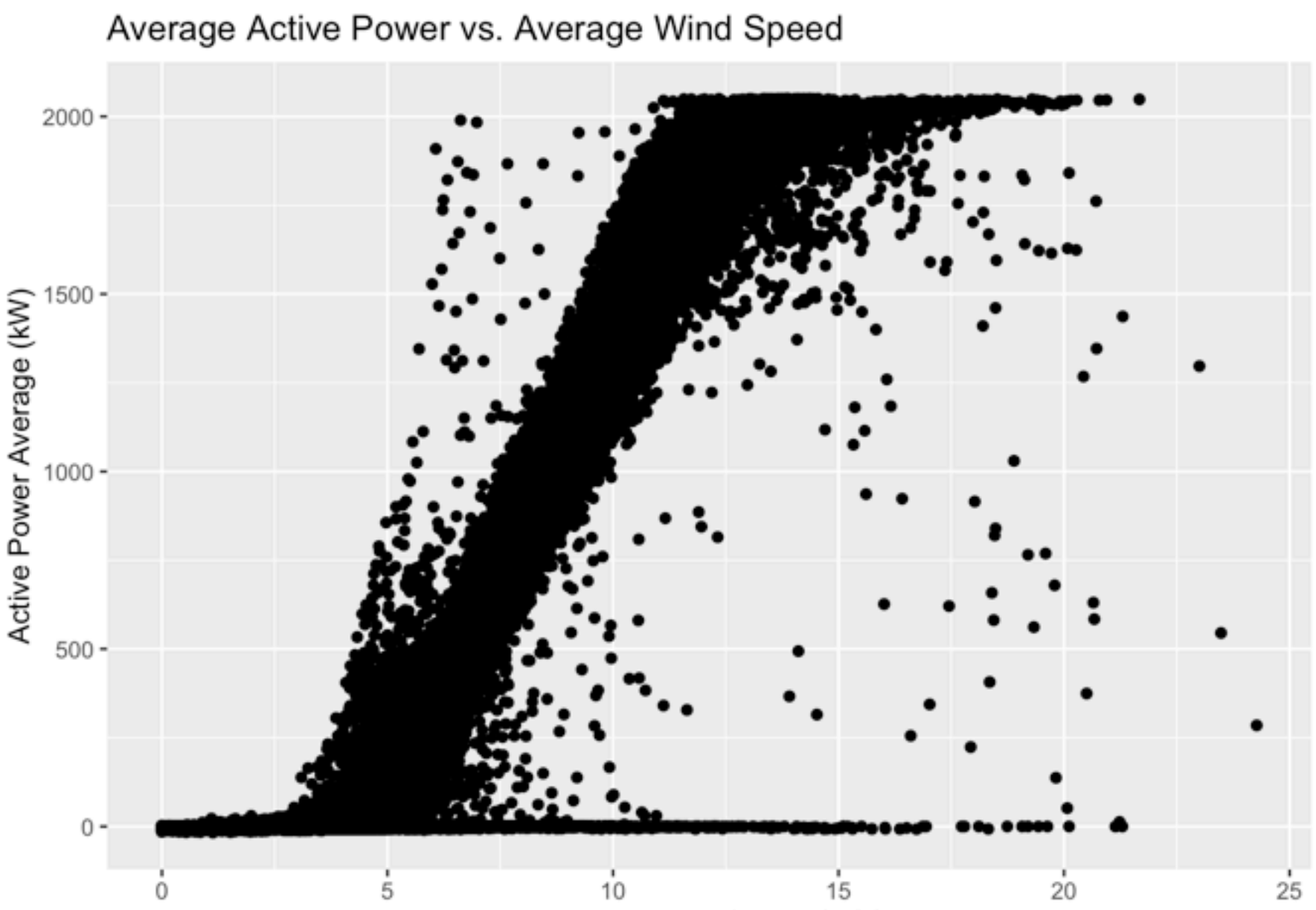
The Data

The Haute Borne Wind Farm in the north of France has released some data from 4 of its 7 turbines. This data covers approximately one year, starting in early 2017 until early 2018. Separate observations are listed for a given sensor's minimum, maximum, and average reading. This capstone project is an exploration of that data, with some machine learning techniques applied to make predictions about it. Included in the observations is "Active Power" in both a minimum, maximum, and average measurement. I decided to look carefully at this measurement of how much power a turbine is producing. Throughout the project Acive Power is the variable of interest, and expecially looking at Active Power in terms of how much wind there is, and thereby, whether or not a turbine is producing as designed. As I will demonstrate, the overwhelming trend of this data is that Wind Speed and Active Power are highly positively correlated, but there are interesting exceptions.

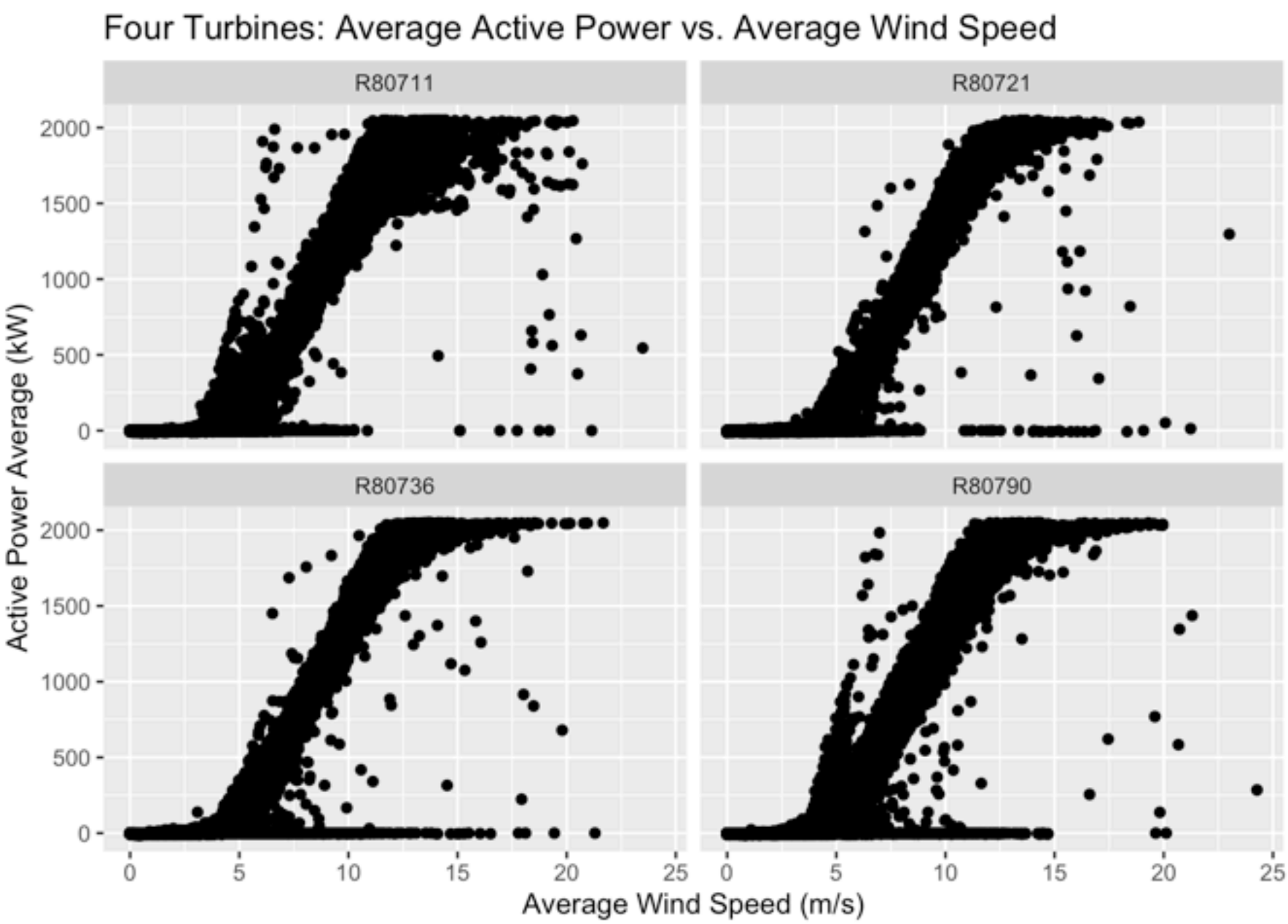
A wind farm is not a closed system but rather connected to a grid with fluctuations in Active Power that may have nothing to do with either wind or variables measured. The technicalities of this relationship are beyond the scope of this project and my own knowledge of the field. So I will focus on the dataset at hand as a starting point, knowing the results will not be definitive, but may be illuminating.

The first step was to get to know the data, clean it up, and rename the columns for easier readability. A more descriptive name for each variable and its units were provided in a key along with the dataset.

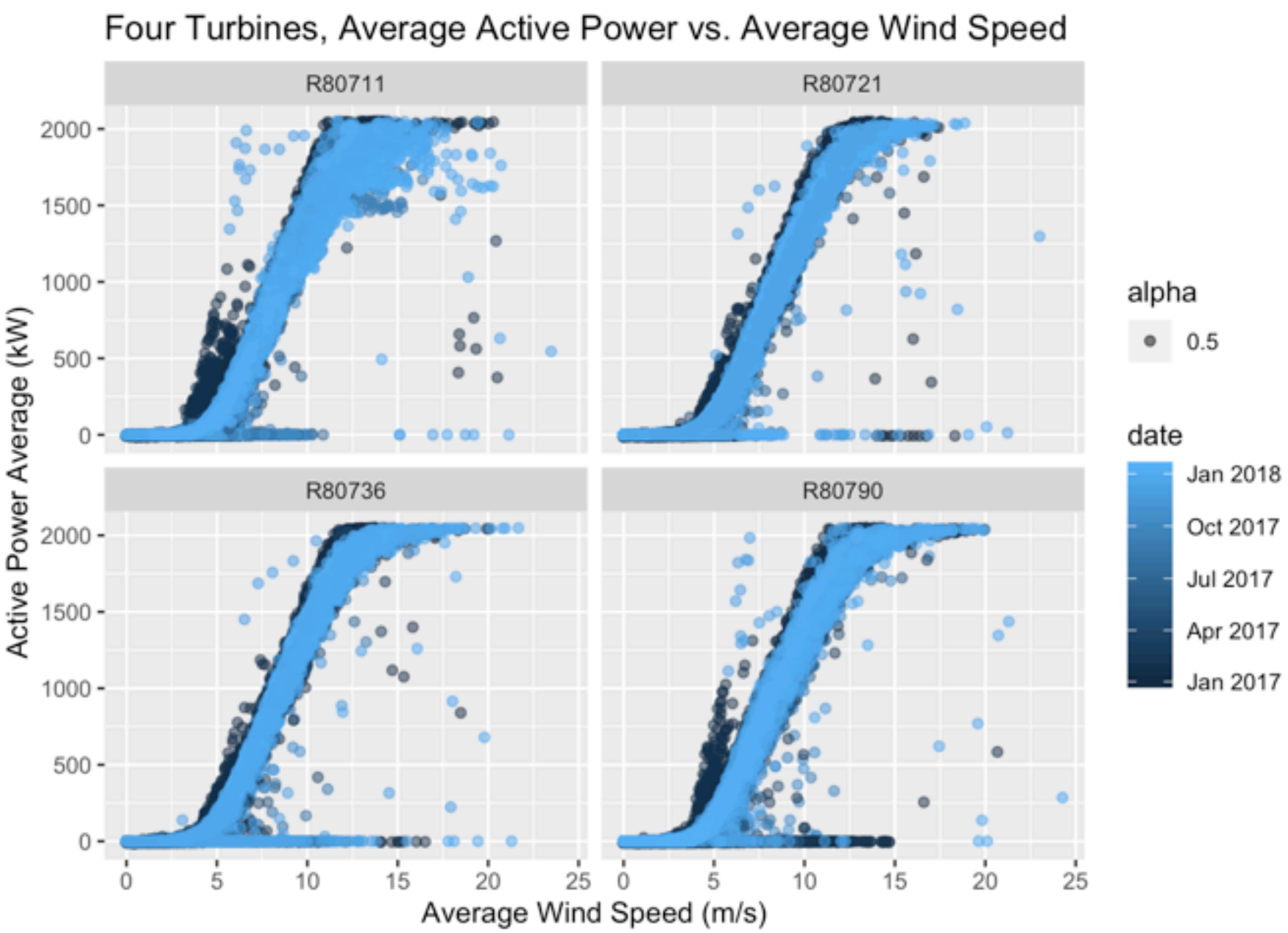
To begin exploring the relationship between Wind Speed and Active Power, I made some plots. These serve as a picture of the data. I used visualizion as a strategy for getting some insights into how to proceed with a model that might be able to predict some particular state of a turbine. The first is a simple display of all the data, showing Active Power Average against Wind Speed Average. The positive correlation is immediately visible.



Despite the clear upward trend, the data is not completely uniform. There are cases where the sensors were showing low Wind Speed, and unusually high Active Power, appearing as clusters to the left of the main trend line. On the other hand, there is a good amount of data displaying little Active Power, when Wind Speed is relatively high, the pooling at the bottom, and to the right. To better understand the anomalies, I seperated this aggregate data into facets. This clearly sets the individual wind turbines side by side. Two of the turbines had a more pronounced deviation from the main trend of the data. This was in the form of a steeply positive sloping branch to the left of the main branch.

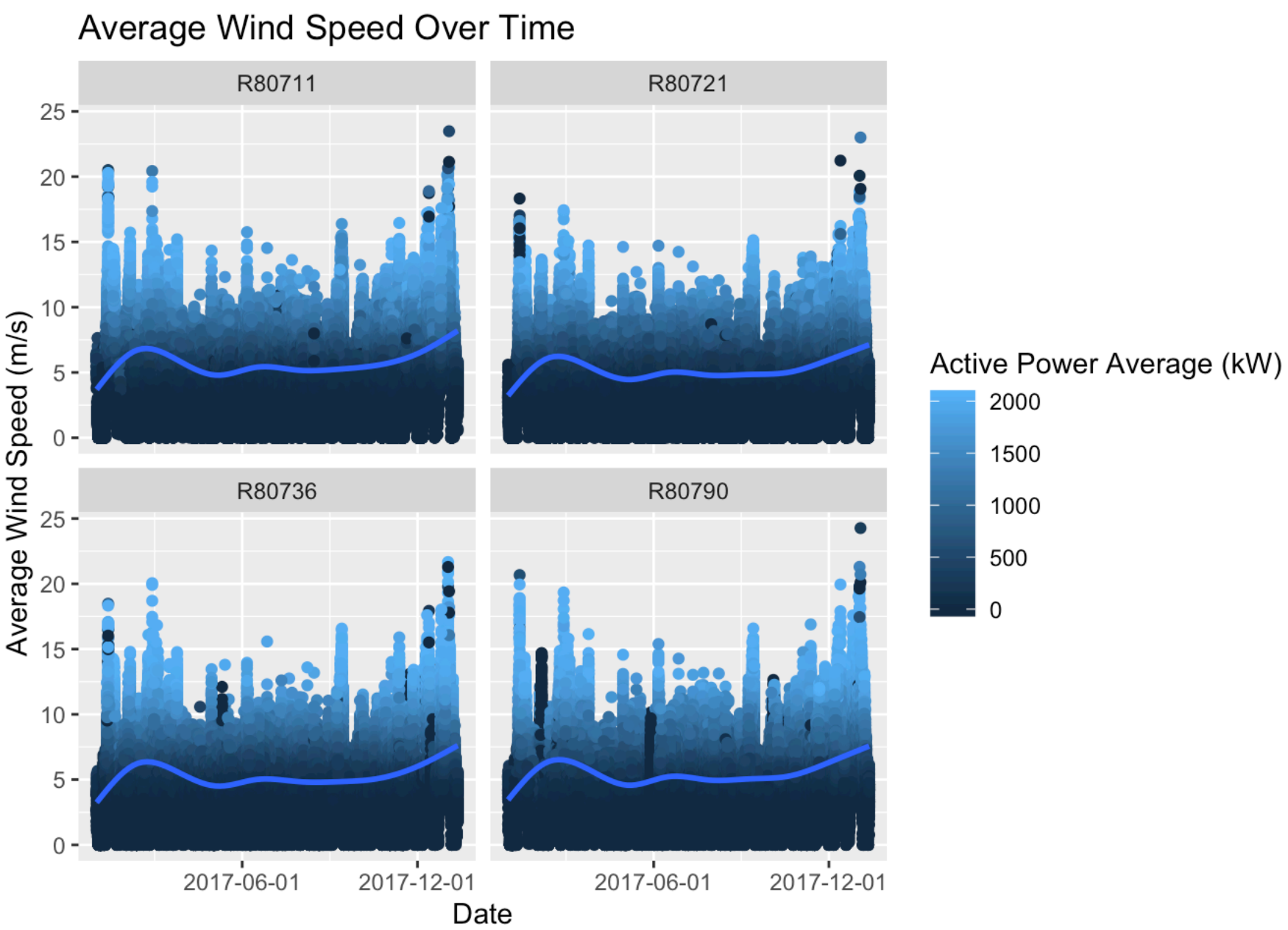


To keep the same basic visualization and try to learn what made those areas different, I started mapping different variables to color, and found that by mapping Date to color, it was apparent that the breakaway groupings in each of the turbines mostly seem to be in the chronologically earlier part of the sample, appearing below as dark blue spikes to the left of the main grouping (dark blue corresponds to earlier dates).



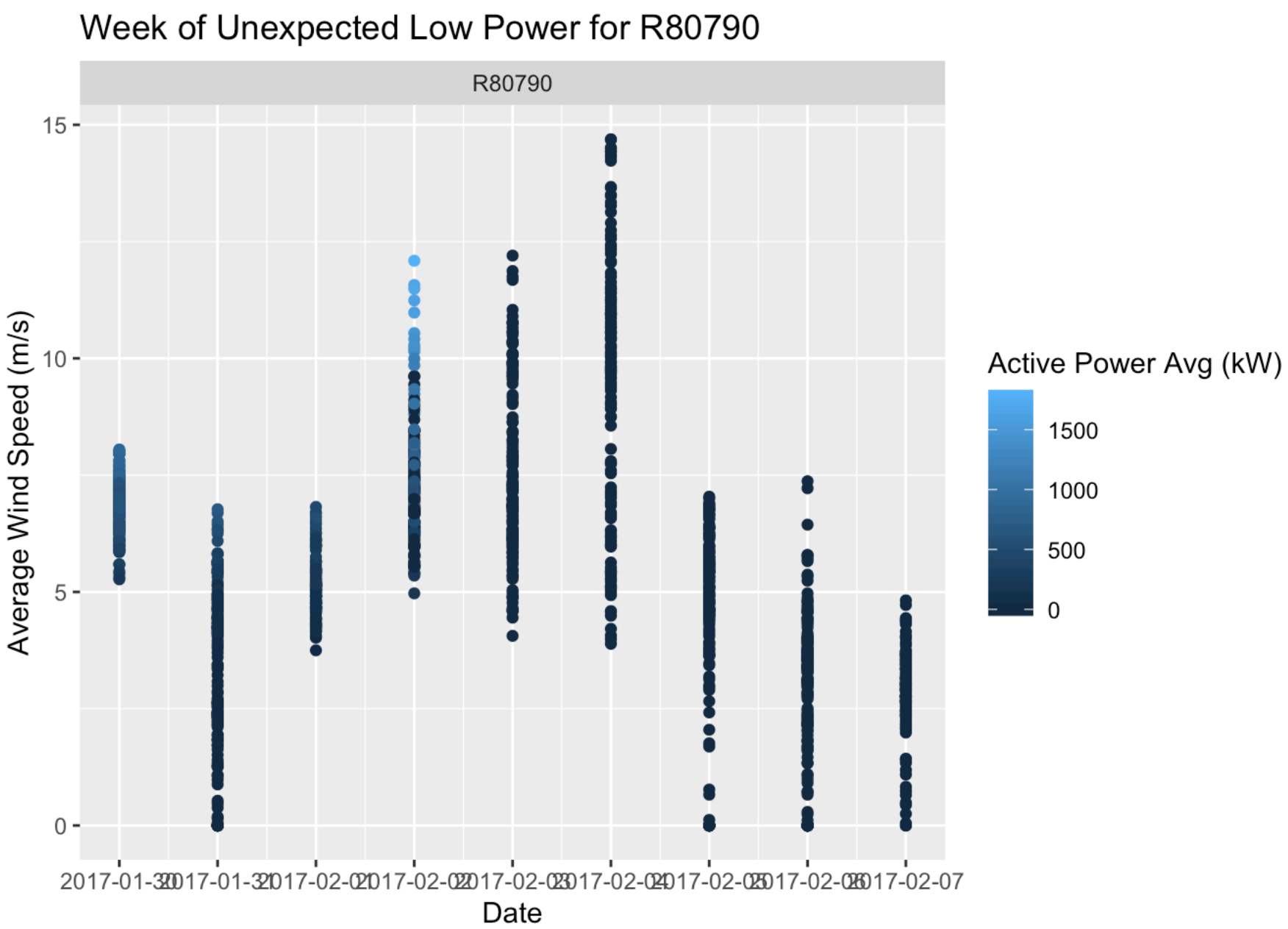
Could it be the turbines ran more efficiently sooner after installation? Its tempting to think so, but the very highest points of efficiency (where there is wind speed around 5 m/s, and high active power) are later in the time series, or lighter in color. This plot also raised another question: why might turbine R80711 have a significantly larger range of active power readings at the higher wind speeds? Above ~12.5 m/s, the correlation between wind speed and active power becomes much weaker for that turbine.

These are interesting problems, but to get a different perspective, I tried introducing time into the exploration, in particular comparing Wind Speeds over time. For a new set of plots, I mapped Date on the X-axis, and Wind Speed on the Y-axis, and this time mapped Active Power to color. Over a full year period, the resulting plot showed that wind rises and falls over time with a generally corresponding increase and decrease of Active Power as observed in the previous plots. But in this plot, some dark spikes indicate low to zero Active Power at high Wind Speeds:



As an experiment, I isolated the darkest spike in Turbine R80790's time series. Since this dark spike represents an unexpected lack of Active Power at high Wind Speed for a given amount of time, I filtered out a smaller date range on either side of the dark spike (9 days) to see exactly which days are in question.

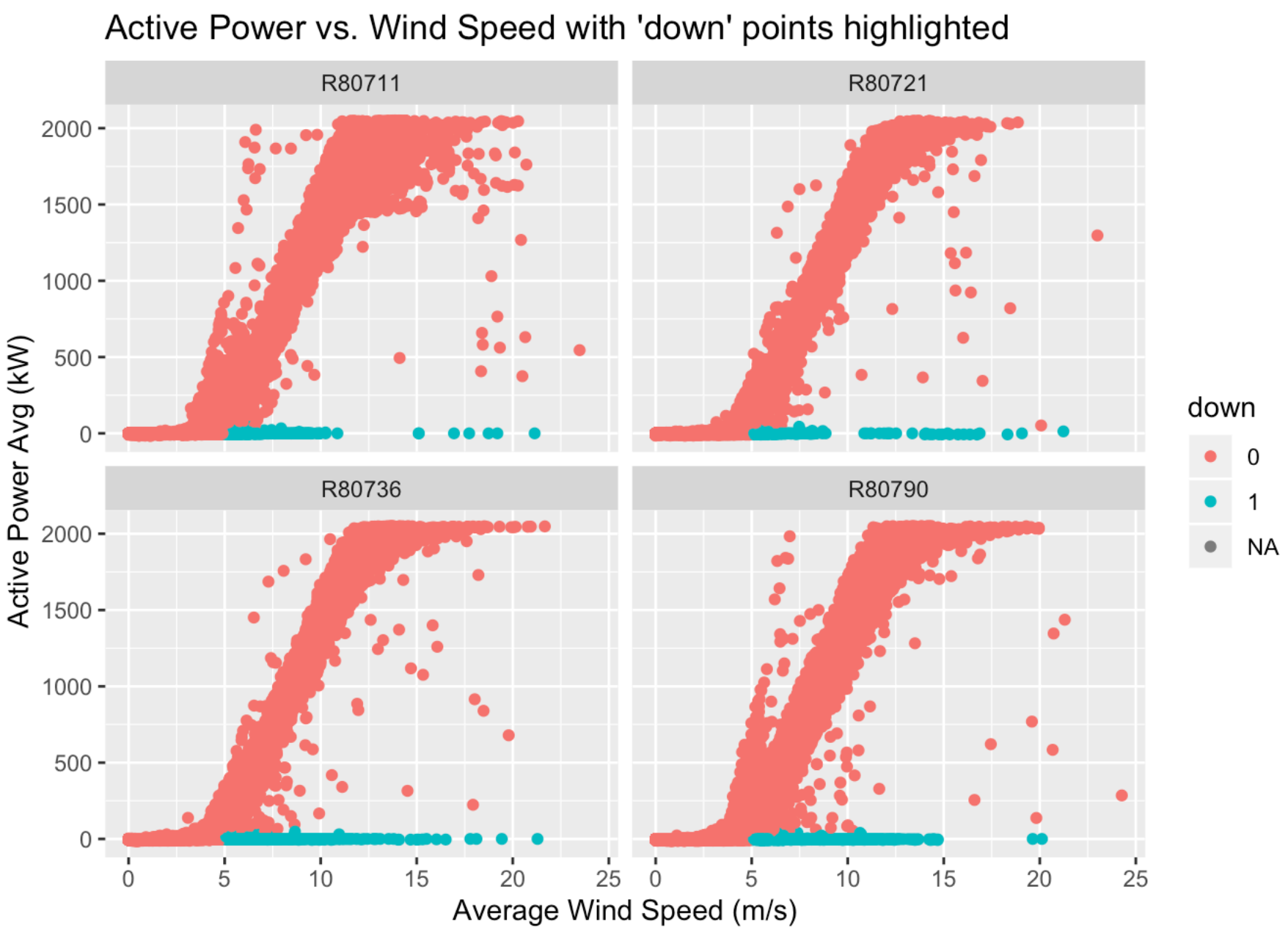
The resulting plot reveals that this condition of low power and high wind occured over just a two day period, February 3rd and 4th of 2017, when despite the windy conditions, and the three other turbines producing power, Turbine R80790 was not producing. Further exploration might determine whether it was down for service (which would require access to service records), or perhaps we can find out if there might have been a predictor for that condition.



Feature Engineering for Machine Learning

This visualization and exploration has suggested to me some parameters for a machine learning approach. I have already begun to think of this Active Power versus Wind Speed as a measure of efficiency. Since the dataset does not have a sensor reading for “efficiency” as I understand it based on this exploration, I will create a new variable. This variable will tell us for each of the data points, whether a turbine is below an Average Active Power level of 50kW while Average Wind Speed is over the supposedly functional level of 5 m/s. I will call this binary variable “down”, and a “1” in this column indicates that the conditions are met, and a “0” that they are not. So “1” in this column describes an undesirable state for the turbine. These parameters involve assumptions about what a Turbine “should” be doing, but as we will see they do seem to entirely cover the cluster of data that caught our attention in the exploration, Machine learning may be useful for predicting cases of “down”, and with some further work, of preventing them.

After creating this “down” variable, we can be visualize it. Here is the same plot of Wind Speed and Active Power, now with instances of “down” highlighted in light green. These green points are the data we are interested in predicting. When the wind is blowing, and there is no power being produced. Green points are a snapshot of a turbine not doing what it is supposed to.



Machine Learning

The positive cases of “down” is only about 1.2% of the dataset. By using an automated resampling techniques, we can get a version of this dataset that is more balanced so that we can start creating models to predict our “down” variable. We are going to look at all of the variables and see how statistically significant they are in relationship to “down”. We use a model called logistic regression, that can be trained by measuring the relationship between all of the variables in the dataset and the new “down” variable we have created, and based on what it learns from this measuring process, it will tell us whether each instance of the “down” variable will be “1” or “0”. This result can be compared to the actual value in the “down” column of the test set to let us know how accurate our model is. Or how good it is at predicting “down” turbines.

After further cleaning up the balanced data, I have split the data into two sets, one to train the model, and one to test it.

The logistic regression model is fed the training set and determines which variables are significant. The stars next to a variable shows how significant that variable is in predicting whether “down” is 0 or 1:

```
##
## Call:
## glm(formula = down ~ ., family = binomial, data = train)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -4.1401  -0.0001   0.0000   0.1781   4.4611
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -1.400e+01  2.011e+00  -6.961 3.37e-12
## Pitch_angle_avg                 1.019e-01  2.964e-02   3.438 0.000585
## Pitch_angle_min                -4.348e-02  2.831e-02  -1.536 0.124564
## Pitch_angle_max                 9.837e-02  3.429e-02   2.868 0.004126
## Pitch_angle_std                -1.498e-01  7.686e-02  -1.949 0.051321
## Hub_temperature_avg            -6.434e-01  3.435e-01  -1.873 0.061082
## Hub_temperature_min             1.460e-01  3.127e-01   0.467 0.640706
## Hub_temperature_max             2.856e-01  3.301e-01   0.865 0.386794
## Hub_temperature_std            -2.900e+00  8.092e-01  -3.584 0.000338
## Generator_converter_speed_avg  -1.160e-01  3.649e-02  -3.178 0.001483
## Generator_converter_speed_min   1.037e-01  4.332e-02   2.394 0.016662
## Generator_converter_speed_max  -4.099e-02  2.681e-02  -1.529 0.126318
## Generator_converter_speed_std  -7.327e-03  7.933e-02  -0.092 0.926406
## Converter_torque_avg           -1.197e-02  3.709e-03  -3.226 0.001255
## Converter_torque_min           -5.006e-03  3.499e-03  -1.431 0.152453
## Converter_torque_max           -1.265e-02  3.210e-03  -3.941 8.13e-05
## Converter_torque_std            1.589e-02  4.787e-03   3.319 0.000903
## Generator_speed_avg             1.614e-01  5.092e-02   3.169 0.001529
## Generator_speed_min            -3.523e-02  4.748e-02  -0.742 0.458100
## Generator_speed_max             5.189e-02  3.054e-02   1.699 0.089310
## Generator_speed_std             5.114e-02  1.059e-01   0.483 0.629196
## Generator_bearing_1_temperature_avg  1.033e-01  9.628e-01   0.107 0.914549
## Generator_bearing_1_temperature_min  5.458e-01  5.411e-01   1.009 0.313188
## Generator_bearing_1_temperature_max -4.687e-01  1.182e+00  -0.397 0.691613
## Generator_bearing_1_temperature_std  6.417e-01  2.653e+00   0.242 0.808876
## Generator_bearing_2_temperature_avg -8.108e-01  1.270e+00  -0.638 0.523219
## Generator_bearing_2_temperature_min -2.351e-01  1.228e+00  -0.191 0.848196
## Generator_bearing_2_temperature_max  7.200e-01  9.941e-01   0.724 0.468929
## Generator_bearing_2_temperature_std -1.051e+00  2.837e+00  -0.370 0.711035
## Generator_stator_temperature_avg     1.844e+00  7.521e-01   2.453 0.014186
## Generator_stator_temperature_min    -7.001e-01  4.554e-01  -1.537 0.124230
## Generator_stator_temperature_max    -1.190e+00  5.578e-01  -2.134 0.032876
## Generator_stator_temperature_std     1.371e+00  1.361e+00   1.007 0.313947
## Gearbox_bearing_1_temperature_avg    4.306e-01  7.661e-01   0.562 0.574027
## Gearbox_bearing_1_temperature_min   -1.045e+00  7.659e-01  -1.364 0.172590
## Gearbox_bearing_1_temperature_max    7.620e-01  6.475e-01   1.177 0.239269
## Gearbox_bearing_1_temperature_std   -3.977e+00  1.784e+00  -2.229 0.025794
## Gearbox_bearing_2_temperature_avg    1.188e+00  9.630e-01   1.234 0.217191
## Gearbox_bearing_2_temperature_min   -1.200e+00  1.004e+00  -1.195 0.232239
## Gearbox_bearing_2_temperature_max   -1.902e-01  8.157e-01  -0.233 0.815669
## Gearbox_bearing_2_temperature_std    4.377e-01  2.397e+00   0.183 0.855105
## Gearbox_inlet_temperature_avg       -9.998e-02  1.098e-01  -0.911 0.362378
## Gearbox_inlet_temperature_min        3.254e-02  1.112e-01   0.293 0.769874
## Gearbox_inlet_temperature_max       -4.180e-02  1.182e-01  -0.354 0.723574
## Gearbox_inlet_temperature_std       -6.745e-02  2.715e-01  -0.248 0.803768
```

```
## Gearbox_oil_sump_temperature_avg    -1.703e-01  2.298e-01  -0.741 0.458528
## Gearbox_oil_sump_temperature_min     3.511e-01  2.449e-01   1.433 0.151753
## Gearbox_oil_sump_temperature_max     2.363e-02  2.215e-01   0.107 0.915028
## Gearbox_oil_sump_temperature_std    -3.264e-03  6.009e-01  -0.005 0.995666
## Nacelle_angle_avg                    1.496e-02  4.061e-03   3.684 0.000229
## Nacelle_angle_min                   -1.219e-02  2.690e-03  -4.533 5.81e-06
## Nacelle_angle_max                   -3.828e-03  3.531e-03  -1.084 0.278321
## Nacelle_angle_std                    2.607e-02  1.126e-02   2.317 0.020530
## Nacelle_temperature_avg             -1.854e-01  5.958e-01  -0.311 0.755614
## Nacelle_temperature_min             -1.090e+00  6.521e-01  -1.671 0.094751
## Nacelle_temperature_max              1.041e+00  6.371e-01   1.635 0.102118
## Nacelle_temperature_std             -2.963e+00  1.964e+00  -1.508 0.131465
## Outdoor_temperature_avg             -2.608e+00  8.462e-01  -3.083 0.002052
## Outdoor_temperature_min              3.183e+00  8.033e-01   3.963 7.41e-05
## Outdoor_temperature_max             -3.234e-01  7.546e-01  -0.429 0.668241
## Outdoor_temperature_std              1.738e+00  2.135e+00   0.814 0.415565
## Grid_frequency_avg                  -1.226e+00  2.061e+00  -0.595 0.551864
## Grid_frequency_min                  -1.514e+00  2.098e+00  -0.721 0.470655
## Grid_frequency_max                   1.238e+00  1.036e+00   1.194 0.232482
## Grid_frequency_std                  -4.041e+00  4.091e+00  -0.988 0.323248
## Grid_voltage_avg                     1.016e-01  1.419e-01   0.716 0.474008
## Grid_voltage_min                     9.572e-02  1.216e-01   0.787 0.431226
## Grid_voltage_max                    -8.591e-02  3.820e-02  -2.249 0.024524
## Grid_voltage_std                     2.792e-01  2.542e-01   1.098 0.272133
## Rotor_speed_avg                     -3.925e+00  5.550e+00  -0.707 0.479461
## Rotor_speed_min                     -6.735e+00  3.955e+00  -1.703 0.088558
## Rotor_speed_max                     -8.005e-01  1.673e+00  -0.479 0.632256
## Rotor_speed_std                     -4.259e+00  7.496e+00  -0.568 0.569886
## Rotor_bearing_temperature_avg       -2.323e+00  1.741e+00  -1.334 0.182229
## Rotor_bearing_temperature_min       -4.153e+00  1.593e+00  -2.607 0.009135
## Rotor_bearing_temperature_max        6.820e+00  1.677e+00   4.067 4.77e-05
## Rotor_bearing_temperature_std        2.001e+00  4.312e+00   0.464 0.642623
## Torque_avg                           6.646e-04  3.118e-03   0.213 0.831186
## Torque_min                          -2.061e-03  6.836e-04  -3.015 0.002573
## Torque_max                           1.246e-02  3.340e-03   3.730 0.000191
## Torque_std                          -1.143e-02  4.125e-03  -2.770 0.005605
##
## (Intercept)                         ***
## Pitch_angle_avg                     ***
## Pitch_angle_min
## Pitch_angle_max                     **
## Pitch_angle_std                     .
## Hub_temperature_avg                 .
## Hub_temperature_min
## Hub_temperature_max
## Hub_temperature_std                 ***
## Generator_converter_speed_avg       **
## Generator_converter_speed_min       *
## Generator_converter_speed_max
## Generator_converter_speed_std
```

```
## Converter_torque_avg                  **
## Converter_torque_min
## Converter_torque_max                  ***
## Converter_torque_std                  ***
## Generator_speed_avg                   **
## Generator_speed_min
## Generator_speed_max                   .
## Generator_speed_std
## Generator_bearing_1_temperature_avg
## Generator_bearing_1_temperature_min
## Generator_bearing_1_temperature_max
## Generator_bearing_1_temperature_std
## Generator_bearing_2_temperature_avg
## Generator_bearing_2_temperature_min
## Generator_bearing_2_temperature_max
## Generator_bearing_2_temperature_std
## Generator_stator_temperature_avg    *
## Generator_stator_temperature_min
## Generator_stator_temperature_max    *
## Generator_stator_temperature_std
## Gearbox_bearing_1_temperature_avg
## Gearbox_bearing_1_temperature_min
## Gearbox_bearing_1_temperature_max
## Gearbox_bearing_1_temperature_std   *
## Gearbox_bearing_2_temperature_avg
## Gearbox_bearing_2_temperature_min
## Gearbox_bearing_2_temperature_max
## Gearbox_bearing_2_temperature_std
## Gearbox_inlet_temperature_avg
## Gearbox_inlet_temperature_min
## Gearbox_inlet_temperature_max
## Gearbox_inlet_temperature_std
## Gearbox_oil_sump_temperature_avg
## Gearbox_oil_sump_temperature_min
## Gearbox_oil_sump_temperature_max
## Gearbox_oil_sump_temperature_std
## Nacelle_angle_avg                     ***
## Nacelle_angle_min                     ***
## Nacelle_angle_max
## Nacelle_angle_std                     *
## Nacelle_temperature_avg
## Nacelle_temperature_min               .
## Nacelle_temperature_max
## Nacelle_temperature_std
## Outdoor_temperature_avg               **
## Outdoor_temperature_min               ***
## Outdoor_temperature_max
## Outdoor_temperature_std
## Grid_frequency_avg
## Grid_frequency_min
```

```
## Grid_frequency_max
## Grid_frequency_std
## Grid_voltage_avg
## Grid_voltage_min
## Grid_voltage_max                    *
## Grid_voltage_std
## Rotor_speed_avg
## Rotor_speed_min                     .
## Rotor_speed_max
## Rotor_speed_std
## Rotor_bearing_temperature_avg
## Rotor_bearing_temperature_min       **
## Rotor_bearing_temperature_max       ***
## Rotor_bearing_temperature_std
## Torque_avg
## Torque_min                          **
## Torque_max                          ***
## Torque_std                          **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13601  on 10217  degrees of freedom
## Residual deviance:  1628  on 10137  degrees of freedom
## AIC: 1790
##
## Number of Fisher Scoring iterations: 15
```

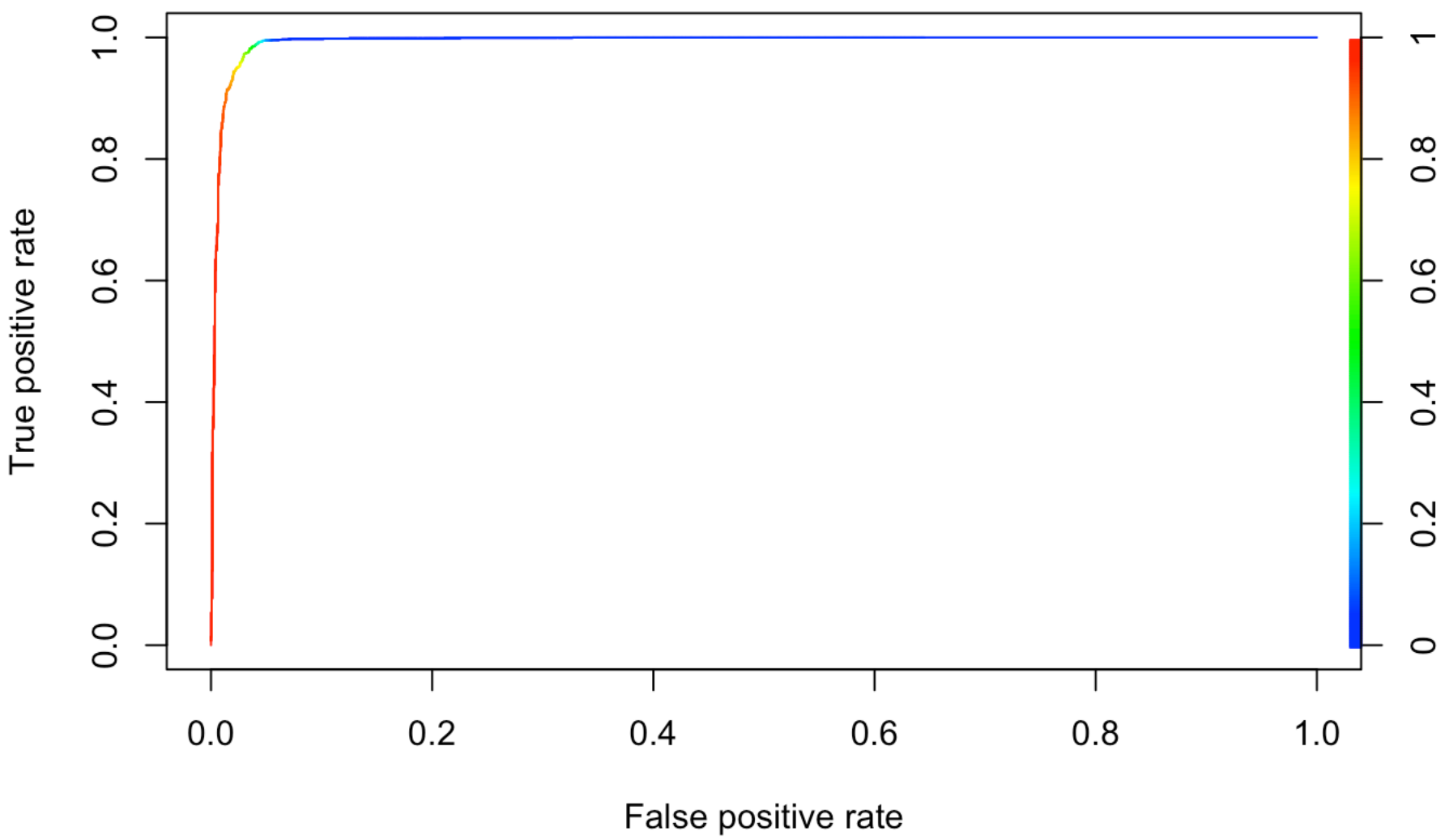
Predictions

So the model has learned how to predict “down” by using the data in the training set. To test it, we will feed it the test portion that we set aside earlier. It will look at what it already knows about those significant or insignificant variables in the test set, and without looking at whether “down” is actually 1 or 0, it will make a prediction. To see how accurate it is, we compare its performance in terms of actual versus predicted values, specifically, true positives against false positives. Which can be visualized in a Reciever Operating Characteristic (ROC) curve. The area under the curve represents the model’s accuracy:

```
##
##    0    1
## 3394 2107
```

```
##
##      FALSE TRUE
##   0  3270  124
##   1    34 2073
```

```
## [1] 0.993741
```



A function can quickly calculate the area under this ROC curve as follows:

```
## [1] 0.993741
```

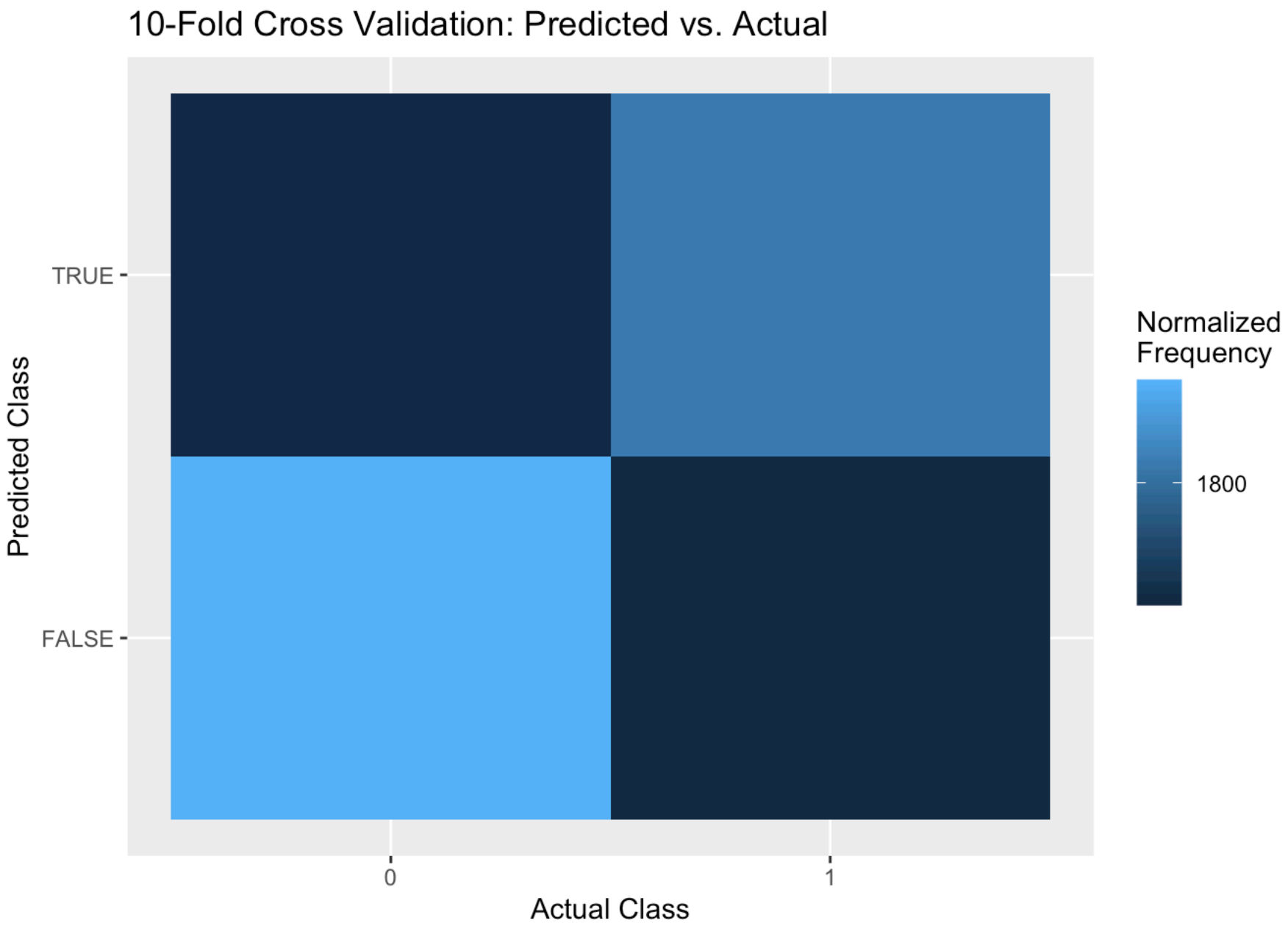
```
##
##       FALSE TRUE
##    0   3270  124
##    1     34 2073
```

The area under the curve is just over 99%, in other words, more accurate than our baseline of 98.85%

So clearly logistic regression produces a result, though it may be problematic and I will discuss this further in the conclusion. To try a different approach I will use a type of machine learning called cross-validation. Specifically, this will be a 10-fold cross validation, so the training data will be split into 10 groups, and a model will be created ten times, with each of the ten groups acting as a test set once, and each of them being a part of the 90% of the data considered a training set.

And visualize the confusion matrix for this as well.

```
##
## predCV    0    1
##      0 3270   34
##      1  124 2073
```



Cross validation gives us a model that is about 97% accurate. This may seem accurate, but it does not beat our logistic regression model which was above 99% accurate, and is not even as accurate as our baseline of 98.85%.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3270   34
##          1  124 2073
##
##                Accuracy : 0.9713
##                  95% CI : (0.9665, 0.9755)
##     No Information Rate : 0.617
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9397
##
##  Mcnemar's Test P-Value : 1.437e-12
##
##             Sensitivity : 0.9635
##             Specificity : 0.9839
##          Pos Pred Value : 0.9897
##          Neg Pred Value : 0.9436
##              Prevalence : 0.6170
##          Detection Rate : 0.5944
##    Detection Prevalence : 0.6006
##       Balanced Accuracy : 0.9737
##
##        'Positive' Class : 0
##
```

Conclusion

The Results

If we predicted a “0” for all rows on our dataset, our prediction would be 98.85% accurate. That is the baseline prediction accuracy. The Logistic regression method proved a more accurate way to predict our classifier at over 99% accuracy. The 10-fold Cross Validation was highly accurate at predicting cases of “down” at about 97%. Considering the baseline, it cannot be considered an improvement. But as a more thorough prediction model, it still gives us a good picture of how predictable this “down” feature is in this dataset.

Implementation

The “down” feature, which is a simplification, nevertheless labels a circumstance that a client should be interested in learning more about. Applied to a dataset with the same variables, and excluding wind and power readings, the logistic regression model presented here could tell us whether a turbine is functioning above a reasonable performance threshold or whether it is not. This model or similar could be used in a set of incomplete data, such as data on these turbines or turbines with the same sensor reading in which wind and power columns contained a high number of missing values. Or as a first step toward further exploration of more complicated problems in this data.

Improvements

While the data can be fitted to a machine learning model, and a result produced, the high accuracy of the predicitions indicates that there may be unexplored problems with modeling the data in this way. One possibility may be that a wind turbine's system of sensors is designed to predict errors in a turbine. In other words, every system monitored by the sensors has a close relationship to either wind speed or power. The temperatures of various components, the vane positions, the pitch angles, all of these componenets are designed in relationship to the wind and the power distribution.

A more nuanced approach to prediction with this data could include some kind of time series analysis, in which certain features in the moments or days before a condition of "down" or similar is registered may be used to predict the failing wind turbine. Machine learning in this kind of technical field would also benefit from collaboration with someone with more specific domain knowledge than I have. A model designed with these things in mind, and perhaps combined with a time series analysis to create a model that could serve as more of an "early warning" predictor might be of more use. Moreover, the labels that I made, while doing a good job of isolating a specific cluster of data, might be improved in many ways. For example, by taking into account records of decisions to power down a turbine for other reasons like maintenance, or the overall wind farm production capacity. Assigning a more informed set of parameters to our "down" feature would also make this model more useful to a client.