Categories

1. [Advanced Modeling](https://datascienceplus.com/category/advanced-modeling/)

Tags

1. [Data Management](https://datascienceplus.com/tag/data-management/)
2. [Linear Regression](https://datascienceplus.com/tag/linear-regression/)
3. [R Programming](https://datascienceplus.com/tag/rstats/)

We have previously [added a set of company identity-agnostic predictors](https://datascienceplus.com/nyc-buses-company-level-predictors-with-r/), such as the number of drivers a company employs, or the number of vehicles in the fleet with a hydraulic lift, and so on. we took this approach, rather than having each company as a unique predictor, so that the addition of a new contractor would not (necessarily) confuse our model.

I understand that this means company history, specific vehicle history, and so on, are not included. Nevertheless, we want to see how much information is in the new predictors.

**Data exploration**

in\_csv <- "../output/intermediate/ii\_times.csv"

ii\_times <- read\_csv(in\_csv) %>%

filter(reported\_before\_resolved == 1) %>% #we kept these flags for exploration, but they are troublesome for our real prediction. If we know a case has already been resolved, our model looks better than it is at predicting delays.

mutate(reported\_before\_resolved = NULL)

#first we look at the correlation of the predictors with the time\_delayed:

corrr\_analysis <- ii\_times %>%

select(-Busbreakdown\_ID) %>%

correlate() %>%

focus(time\_delayed) %>%

rename(feature = rowname) %>%

arrange(desc(abs(time\_delayed))) %>%

mutate(feature = as\_factor(feature))

corrr\_analysis %>% print(n=61)

*## # A tibble: 45 x 2*

*## feature time\_delayed*

*##*

*## 1 Boro\_Bronx -0.166*

*## 2 vehicle\_total\_max\_riders -0.159*

*## 3 vehicle\_total\_with\_attendants 0.159*

*## 4 vehicle\_total\_reg\_seats -0.157*

*## 5 drivers\_total\_attendant 0.147*

*## 6 Has\_Contractor\_Notified\_Parents 0.140*

*## 7 service\_type\_d2d 0.140*

*## 8 Number\_Of\_Students\_On\_The\_Bus -0.139*

*## 9 Boro\_Manhattan 0.125*

*## 10 Boro\_StatenIsland -0.106*

*## 11 Reason\_HeavyTraffic -0.106*

*## 12 vehicle\_total\_with\_lifts 0.106*

*## 13 Reason\_MechanicalProblem 0.100*

*## 14 vehicle\_total\_ambulatory\_seats -0.0960*

*## 15 drivers\_staff\_servSchool 0.0716*

*## 16 rush\_min\_from\_peak 0.0648*

*## 17 Reason\_FlatTire 0.0582*

*## 18 Boro\_Queens 0.0550*

*## 19 Reason\_Accident 0.0503*

*## 20 drivers\_num\_servPreK -0.0478*

*## 21 Reason\_DelayedbySchool -0.0471*

*## 22 drivers\_staff\_servPreK -0.0468*

*## 23 School\_Age 0.0441*

*## 24 drivers\_numServ\_prek -0.0425*

*## 25 Boro\_Westchester 0.0400*

*## 26 drivers\_numServ\_school 0.0390*

*## 27 rush\_within -0.0385*

*## 28 Reason\_WontStart 0.0383*

*## 29 Boro\_Brooklyn 0.0378*

*## 30 Reason\_WeatherConditions 0.0367*

*## 31 Boro\_NassauCounty 0.0345*

*## 32 Has\_Contractor\_Notified\_Schools 0.0334*

*## 33 Boro\_NewJersey 0.0265*

*## 34 time\_am -0.0257*

*## 35 rush\_between 0.0254*

*## 36 Active\_Vehicles -0.0164*

*## 37 rush\_outside 0.0152*

*## 38 Reason\_ProblemRun 0.0115*

*## 39 Have\_You\_Alerted\_OPT -0.0113*

*## 40 Reason\_LatereturnfromFieldTrip 0.0109*

*## 41 drivers\_total\_driver -0.00878*

*## 42 Boro\_RocklandCounty 0.00809*

*## 43 Boro\_Connecticut -0.00787*

*## 44 vehicle\_total\_disabled\_seats 0.00572*

*## 45 drivers\_num\_servSchool -0.00182*

#NR: there are some NAs here. I need to have a look at them. These are the no-information columns.

corrr\_analysis %>%

filter(!is.na(time\_delayed)) %>%

ggplot(aes(x = time\_delayed, y = fct\_reorder(feature, desc(time\_delayed)))) +

geom\_point() +

# Positive Correlations - Contribute to churn

geom\_segment(aes(xend = 0, yend = feature),

color = magma(4)[[3]],

data = corrr\_analysis %>% filter(time\_delayed > 0)) +

geom\_point(color = magma(4)[[3]],

data = corrr\_analysis %>% filter(time\_delayed > 0)) +

# Negative Correlations - Prevent churn

geom\_segment(aes(xend = 0, yend = feature),

color = magma(4)[[1]],

data = corrr\_analysis %>% filter(time\_delayed < 0)) +

geom\_point(color = magma(4)[[1]],

data = corrr\_analysis %>% filter(time\_delayed < 0)) +

# Vertical lines

geom\_vline(xintercept = 0, color = plasma(3)[[2]], size = 1, linetype = 2) +

geom\_vline(xintercept = -0.25, color = plasma(3)[[2]], size = 1, linetype = 2) +

geom\_vline(xintercept = 0.25, color = plasma(3)[[2]], size = 1, linetype = 2) +

# Aesthetics

theme\_bw() +

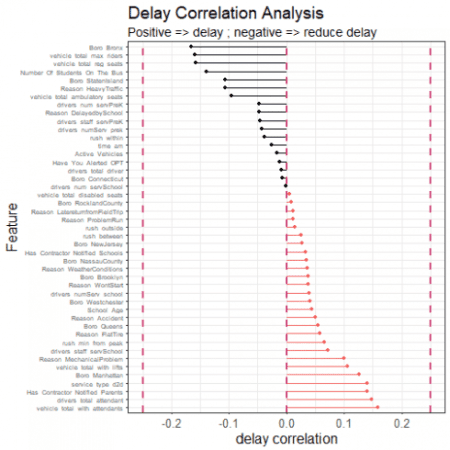
labs(title = "Delay Correlation Analysis",

subtitle = paste("Positive => delay",

"; negative => reduce delay"),

x = "delay correlation", y = "Feature") +

theme(text=element\_text(family="Arial", size=16), axis.text.y = element\_text(size=7))

[](https://i2.wp.com/datascienceplus.com/wp-content/uploads/2018/11/unnamed-chunk-1-1.png?ssl=1)

There are already some interesting insights from these data. Drivers appear more likely to alert OPT for delays which end up shorter. Companies with a lot of ambulatory seats in their fleet suffer shorter delays. Services closer to peak hour suffer *lower delays*, something of a surprise. Companies with a lot of attendants and hydraulic lifts suffer longer delays. Having a high number of pre-K services disposes a company towards less delay; having a high number of school services disposes them towards more delay. And so on. We also see that there is structure from the original data, including different delays for different boroughs, and for different reasons.

One of the first questions we might ask of all these predictors is: are they correlated with *each other*? If they are not, then perhaps we could, at some stage, run a naive Bayesian classification. Let's take a look at the cross-correlations between some of the predictors. We include 'time\_delayed' for reference:

corrr\_high <- corrr\_analysis %>%

mutate(feature = as.character(feature)) %>%

filter(feature != "reported\_before\_resolved") %>%

filter(abs(time\_delayed) >= quantile(abs(corrr\_analysis$time\_delayed), 0.8) ) %>%

dplyr::select(feature) %>%

as.data.frame()

ii\_times %>%

select(time\_delayed, corrr\_high[[1]]) %>%

ggcorr(nbreaks = 8, low = "#924A51", mid = "#70A5D4", high = "#FE9F14",

max\_size = 20,

min\_size = 2,

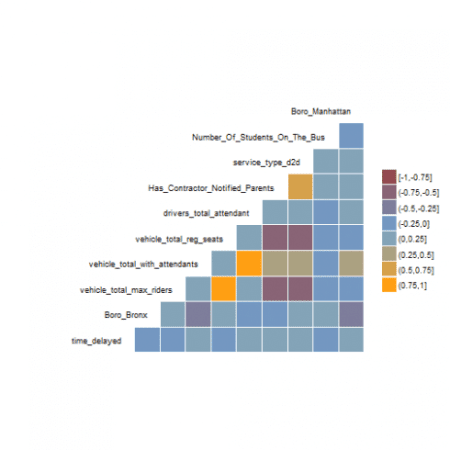
size = 3,

hjust = 1,

vjust = 0.5,

angle = 0,

layout.exp = 4)

[](https://i2.wp.com/datascienceplus.com/wp-content/uploads/2018/11/unnamed-chunk-2-1-1.png?ssl=1)

Even in this reduced set of 'more important' predictors, we can see that many predictors correlate more strongly with each other than with the target variable (time\_delayed). This seems pretty bad news for the independence assumption of naive Bayes. The data set also might be a candidate for dimensionality reduction, but at this stage I wanted to keep things as human-transparent as possible.

**Building a Cubist regression model with the extra predictors**

This next part is straightforward. We go through essentially the same process that we did before, but this time we are armed with the extra predictors. If these predictors are useful, our RMSE should be lower and our \(R^2\) should be higher.

halfFold <- createFolds(ii\_times$time\_delayed, k = 5, list=TRUE)

ii\_small <- ii\_times[halfFold[[2]],]

ii\_test <-ii\_times[halfFold[[3]],]

ii\_train <- ii\_times %>%

anti\_join(ii\_test, by = "Busbreakdown\_ID")

rctrl\_manual <- trainControl(method = "none", returnResamp = "all")

rctrl\_repcv <- trainControl(method = "repeatedcv", number = 2, repeats = 5, returnResamp = "all")

#---There is actually a problem here. If I do not use the reported\_before\_resolved flag, I should remove those points from the data set. Do this and retry.

#---cubist

cubist\_grd <- expand.grid(

committees = 75,

neighbors = 0

)

set.seed(849)

cubist\_cv <- train(time\_delayed ~ . -Busbreakdown\_ID,

data = ii\_small,

method = "cubist",

metric="RMSE",

na.action = na.pass,

trControl = rctrl\_repcv,

tuneGrid = cubist\_grd,

control = Cubist::cubistControl(unbiased = T, rules = 30, sample = 0)

)

cubist\_cv

*## Cubist*

*##*

*## 40665 samples*

*## 46 predictor*

*##*

*## No pre-processing*

*## Resampling: Cross-Validated (2 fold, repeated 5 times)*

*## Summary of sample sizes: 20333, 20332, 20333, 20332, 20331, 20334, ...*

*## Resampling results:*

*##*

*## RMSE Rsquared MAE*

*## 11.74162 0.3502251 8.273603*

*##*

*## Tuning parameter 'committees' was held constant at a value of 75*

*##*

*## Tuning parameter 'neighbors' was held constant at a value of 0*

#comm75, rules30: 0.355, RMSE 11.72

cubist\_cv$finalModel$splits %>% glimpse

*## Observations: 8,341*

*## Variables: 8*

*## $ committee 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...*

*## $ rule 1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, ...*

*## $ variable Number\_Of\_Students\_On\_The\_Bus, vehicle\_total\_with\_l...*

*## $ dir >, <=, <=, <=, >, <=, >, >, >, <=, >, <=, >, <=, >,...*

*## $ value 4, 4, 225, 5449, 0, 389, 601, 7, 131, 5449, 4, 82, ...*

*## $ category , , , , , , , , , , , , , , , , , , , , , , , ,*

*## $ type "type2", "type2", "type2", "type2", "type2", "type2...*

*## $ percentile 0.8056068, 0.3884421, 0.4687077, 0.5004795, 0.94132...*

cubist\_cv$finalModel$coefficients %>% glimpse

*## Observations: 2,234*

*## Variables: 48*

*## $ `(Intercept)` 26.4, 59.3, 11.1, 19.1, 17.7, ...*

*## $ Has\_Contractor\_Notified\_Schools NA, 1.5, NA, 0.7, NA, NA, 3.3,...*

*## $ Has\_Contractor\_Notified\_Parents 0.6, 1.3, 1.5, 0.5, NA, 2.5, N...*

*## $ Have\_You\_Alerted\_OPT 1.4, -0.6, -4.5, -2.4, 0.4, -1...*

*## $ Number\_Of\_Students\_On\_The\_Bus -0.14, -0.07, -0.14, -0.08, -0...*

*## $ School\_Age NA, NA, NA, NA, NA, NA, NA, NA...*

*## $ Reason\_Accident 4, 6, NA, 2, NA, NA, NA, NA, 7...*

*## $ Reason\_DelayedbySchool -2, NA, NA, NA, NA, NA, NA, NA...*

*## $ Reason\_FlatTire 2, 4, NA, NA, NA, NA, NA, NA, ...*

*## $ Reason\_HeavyTraffic -2.4, -1.1, NA, -0.4, -3.7, NA...*

*## $ Reason\_LatereturnfromFieldTrip NA, 1.1, NA, NA, NA, NA, NA, 1...*

*## $ Reason\_MechanicalProblem 2.2, 2.5, NA, 0.8, NA, NA, NA,...*

*## $ Reason\_ProblemRun NA, 2, NA, NA, NA, NA, NA, NA,...*

*## $ Reason\_WeatherConditions NA, 3.9, NA, NA, 5.0, NA, NA, ...*

*## $ Reason\_WontStart 2, 2, NA, NA, NA, NA, NA, NA, ...*

*## $ Boro\_Bronx 0.6, 0.6, -1.1, -1.3, 4.2, -2....*

*## $ Boro\_Brooklyn 2.4, -0.4, NA, 2.5, 0.6, 1.4, ...*

*## $ Boro\_Connecticut NA, NA, NA, NA, NA, NA, NA, NA...*

*## $ Boro\_Manhattan 2.3, 1.5, -1.0, -0.4, 0.6, NA,...*

*## $ Boro\_NassauCounty 4, 2, NA, NA, NA, NA, NA, NA, ...*

*## $ Boro\_NewJersey NA, NA, NA, NA, NA, NA, NA, NA...*

*## $ Boro\_Queens 3.9, 0.7, NA, NA, 0.8, NA, -4....*

*## $ Boro\_RocklandCounty NA, NA, NA, NA, NA, NA, NA, NA...*

*## $ Boro\_StatenIsland NA, 1.9, NA, NA, NA, NA, NA, N...*

*## $ Boro\_Westchester 2.5, NA, NA, NA, NA, NA, NA, N...*

*## $ drivers\_numServ\_school -2.0, NA, NA, NA, -0.3, NA, NA...*

*## $ drivers\_numServ\_prek -2.4, NA, NA, NA, NA, NA, NA, ...*

*## $ drivers\_total\_driver 0.0330, -0.4627, 0.1162, 0.523...*

*## $ drivers\_total\_attendant -0.1228, -0.4552, 0.2071, 0.29...*

*## $ drivers\_num\_servPreK -0.592, NA, -0.463, -0.872, -0...*

*## $ drivers\_num\_servSchool NA, NA, NA, NA, -1.7894, NA, 1...*

*## $ drivers\_staff\_servPreK 0.303, NA, NA, NA, -0.982, 0.0...*

*## $ drivers\_staff\_servSchool 0.0372, 0.3984, -0.2191, -0.42...*

*## $ Active\_Vehicles 0.4074, 0.0722, 0.0733, 0.1984...*

*## $ vehicle\_total\_max\_riders -0.00330, 0.00047, 0.02326, 0....*

*## $ vehicle\_total\_reg\_seats -0.00571, -0.00160, -0.01979, ...*

*## $ vehicle\_total\_disabled\_seats NA, -0.0183, -0.0368, -0.0802,...*

*## $ vehicle\_total\_ambulatory\_seats NA, NA, NA, NA, NA, NA, NA, 0....*

*## $ vehicle\_total\_with\_lifts 14.131, 0.130, 0.120, 0.460, 1...*

*## $ vehicle\_total\_with\_attendants -0.2296, -0.0095, -0.0022, -0....*

*## $ service\_type\_d2d -1.6, 0.5, NA, -1.8, -0.6, -0....*

*## $ time\_am -5.0, -0.5, NA, NA, NA, NA, 4....*

*## $ rush\_within -2.3, 1.3, NA, NA, -1.3, -2.8,...*

*## $ rush\_between -4.8, -1.2, NA, -1.3, -3.2, -5...*

*## $ rush\_outside -1.6, 0.6, NA, -0.4, -3.5, -2....*

*## $ rush\_min\_from\_peak NA, 0.019, NA, 0.006, NA, 0.02...*

*## $ committee "1", "1", "1", "1", "1", "1", ...*

*## $ rule "1", "2", "3", "4", "5", "6", ...*

cubist\_cv$finalModel$usage %>% head

*## Conditions Model Variable*

*## 1 48 86 vehicle\_total\_reg\_seats*

*## 2 41 20 vehicle\_total\_ambulatory\_seats*

*## 3 37 75 vehicle\_total\_with\_lifts*

*## 4 35 49 Reason\_HeavyTraffic*

*## 5 28 90 vehicle\_total\_max\_riders*

*## 6 24 86 vehicle\_total\_with\_attendants*

cubist\_man <- train(time\_delayed ~ . -Busbreakdown\_ID,

data = ii\_train,

method = "cubist",

metric="RMSE",

na.action = na.pass,

trControl = rctrl\_manual,

tuneGrid = cubist\_grd,

control = Cubist::cubistControl(unbiased = T, rules = 30, sample = 0)

)

ii\_withpred <- ii\_test %>%

mutate(

time\_predicted = predict(cubist\_man, newdata = ii\_test)

)

ii\_withpred %>%

summarise(

rms = (mean((time\_predicted-time\_delayed)^2))^0.5,

sum\_sq = sum((time\_predicted-time\_delayed)^2),

tot\_sum\_sq = sum((time\_delayed - mean(time\_delayed))^2),

r\_sq = 1- sum\_sq / tot\_sum\_sq

) #0.371, RMSE 11.6

*## # A tibble: 1 x 4*

*## rms sum\_sq tot\_sum\_sq r\_sq*

*##*

*## 1 11.5 5422077. 8556551. 0.366*

ii\_overfit <- ii\_train %>%

mutate(

time\_predicted = predict(cubist\_man, newdata = ii\_train)

)

ii\_overfit %>%

summarise(

rms = (mean((time\_predicted-time\_delayed)^2))^0.5,

sum\_sq = sum((time\_predicted-time\_delayed)^2),

tot\_sum\_sq = sum((time\_delayed - mean(time\_delayed))^2),

r\_sq = 1- sum\_sq / tot\_sum\_sq

) #0.374, RMSE 11.5: the overfit is not too bad.

*## # A tibble: 1 x 4*

*## rms sum\_sq tot\_sum\_sq r\_sq*

*##*

*## 1 11.5 21412743. 34357934. 0.377*

protec <- ii\_withpred %>%

mutate(Reason\_Other = 1L) %>% #dummy flag to get the name in

select(-contains("Reason")) %>%

colnames()

pal <- brewer.pal(10, "Paired")

ii\_withpred %>%

mutate(Reason\_Other = ifelse(Reason\_Accident + Reason\_DelayedbySchool + Reason\_FlatTire + Reason\_HeavyTraffic + Reason\_LatereturnfromFieldTrip + Reason\_MechanicalProblem + Reason\_ProblemRun + Reason\_WeatherConditions + Reason\_WontStart == 1, 0L, 1L)) %>%

gather(key = Reason, value = dummy, -one\_of(protec)) %>%

filter(dummy == 1) %>%

mutate(dummy = NULL) %>%

filter(row\_number() %% 10 == 0) %>%

ggplot(aes(x = time\_predicted, y = time\_delayed, fill = Reason, color = Reason)) +

geom\_point(alpha = 0.75, shape = 21) +

scale\_fill\_manual(values = pal) +

geom\_abline(intercept = 0, slope = 1, color = "red") +

coord\_cartesian(expand = c(0,0), x = c(0,60)) +

theme\_classic() +

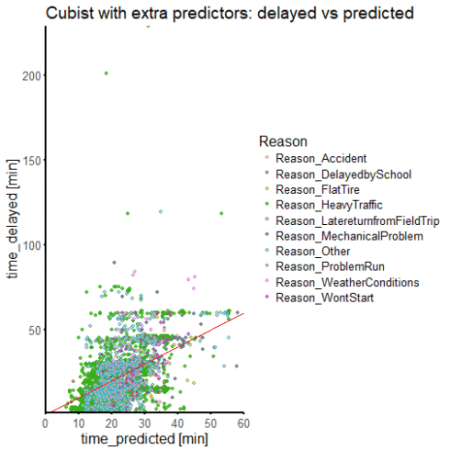
labs(x="time\_predicted [min]",y="time\_delayed [min]") +

theme(text=element\_text(family="Arial", size=16)) +

theme(axis.line = element\_line(colour = 'black', size = 1)) +

theme(axis.ticks = element\_line(colour = "black", size = 1)) +

ggtitle("Cubist with extra predictors: delayed vs predicted")

[](https://i1.wp.com/datascienceplus.com/wp-content/uploads/2018/11/unnamed-chunk-3-1-14.png?ssl=1)

We have more than doubled our \(R^2\) value, although we are still below 0.5. Our time\_delayed vs time\_predicted is starting to cluster more around the line of perfect prediction. We are still mispredicting a lot of delay times, but we are certainly doing better than before.

While there is possibly more to be done to improve our predictors, we can guess that we might not get too much further with this approach. We know that our data have a problem related to how delays have been reported, and it isn't going to go away. So perhaps we ought to try the problem as a classification: how well can we predict if delay time will be above some arbitrary threshold? This is the approach we take next.