Program Sales Forecast

Eric Chang

2/23/2017

```
require(ggplot2)
source("featurize_data.R")
```

Building initial regression model

Initial

```
dat_lm <- filter(dat, !(programs_ordered == programs_sold)) %>%
          select(-programs_ordered, -time_kickoff, -percent_male) %>%
          filter(coach_pokey != 1) %>%  # take out old coach game
          select(-coach_pokey) %>%
          filter(!(month == "August" & day_of_month == 30 & opponent == "Nebraska")) %>%
          select(-opponent) %>%
          mutate(year = as.integer(as.character(year)))

dat_lm$month <- droplevels(dat_lm$month)

dat_lm_fit <- lm(programs_sold ~ ., dat_lm)
message("R^2: ", summary(dat_lm_fit)$r.squared)
```

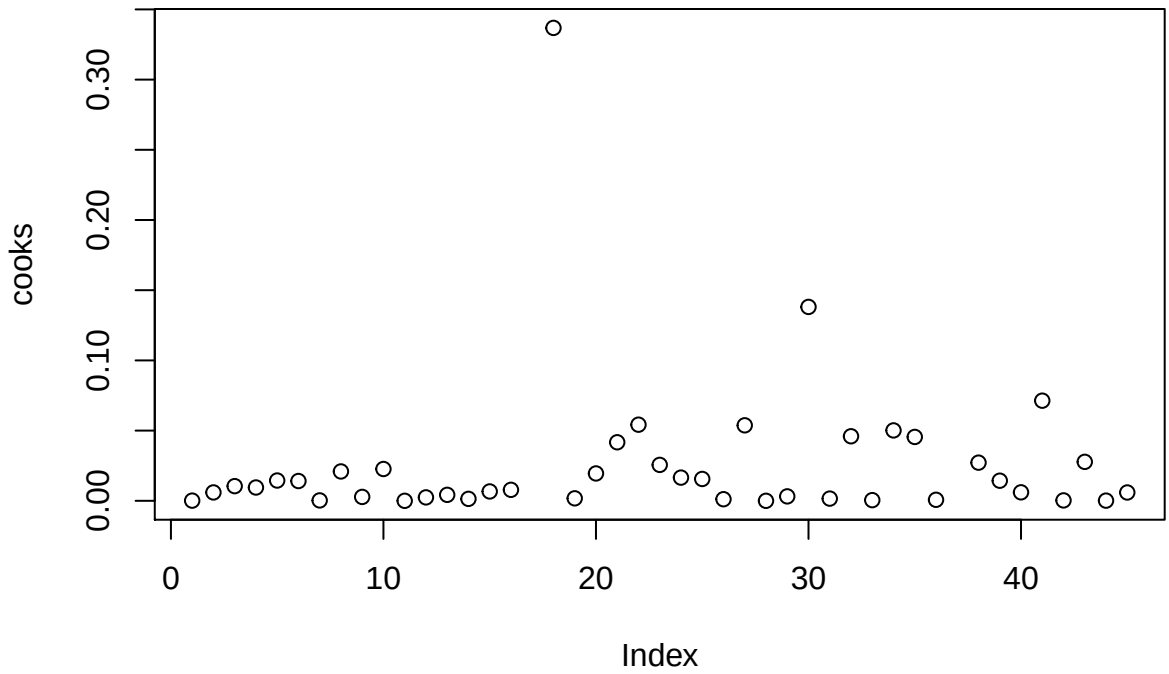
```
## R^2: 0.715193219095598
```

```
message("RMSE: ", sqrt(sum(dat_lm_fit$residuals ^ 2)/nrow(dat_lm)))
```

```
## RMSE: 392.532825306792
```

Outlier Detection

```
cooks <- cooks.distance(dat_lm_fit)
plot(cooks)
```




```
dat_lm[cooks > 4 / nrow(dat_lm), ] %>% na.omit()  # extract outliers using Fox's threshold
```

```
##        month day_of_month year programs_sold homecoming morning_kickoff
## 18  November            7 2015          3525          0               1
## 30 September           13 2008          3128          0               0
##    hour big_ten
## 18   11       0
## 30   17       0
```

Removing August/December

```
dat_lm <- dat_lm %>% filter(!month %in% c("August", "December"))

message("Not cross validated, removed August and December points:")
```

```
## Not cross validated, removed August and December points:
```

```
dat_lm_fit <- lm(programs_sold ~ ., dat_lm)
message("R^2: ", summary(dat_lm_fit)$r.squared)
```

```
## R^2: 0.652413672943868
```

```
message("RMSE: ", sqrt(sum(dat_lm_fit$residuals ^ 2 / nrow(dat_lm) )))
```

```
## RMSE: 401.557747373375
```

Cross validating

```
lm_cross_validation <- function(formula, dat_lm, folds){
    # generate model matrix
    set.seed(123)
    dat_lm$fold <- ceiling(runif(nrow(dat_lm)) * folds)
    dat_lm <- arrange(dat_lm, fold)
    dat_lm$oof_predictions <- NA

    summaries <- list()

    for (i in seq(1, folds)) {
        in_fold <- dat_lm[dat_lm$fold != i,]
        out_of_fold <- dat_lm[dat_lm$fold == i,]

        fit <- lm(formula, data=select(in_fold, -oof_predictions, -fold))
        summaries[[i]] <- summary(fit)

        dat_lm$oof_predictions[dat_lm$fold == i] <-
          predict(fit, select(out_of_fold, -oof_predictions, -fold))
    }

    # calculate r-squared with out of fold predictions
    dat_lm$tss <- (dat_lm$programs_sold - mean(dat_lm$programs_sold))^2
    dat_lm$rss <- (dat_lm$programs_sold - dat_lm$oof_predictions)^2
    rsquared <- 1 - sum(dat_lm$rss)/sum(dat_lm$tss)

    message("R^2 CV: ", rsquared)
    message("RMSE CV: ", sqrt(sum(dat_lm$rss/nrow(dat_lm))))
    return(dat_lm)
}

message("Cross validated:")
```

```
## Cross validated:
```

```
lm_cross_validation("programs_sold ~ .", dat_lm, folds=nrow(dat_lm))
```

```
## R^2 CV: 0.426468378252633
```

```
## RMSE CV: 515.816643390712
```

```
##        month day_of_month year programs_sold homecoming morning_kickoff
## 1  September            8 2012          2500          0               1
## 2  September           11 2010          2390          0               1
## 3    October           27 2012          2440          0               1
## 4  September            7 2013          4037          0               1
## 5   November           12 2011          1969          0               1
## 6    October           24 2009          1536          0               1
## 7    October            8 2011          2404          1               1
## 8  September            5 2009          2691          0               0
## 9   November            7 2015          3525          0               1
## 10 September            3 2011          2885          0               1
## 11 September           13 2008          3128          0               0
## 12 September           24 2011          1911          0               0
```

```
## 13 September            1 2012          3634          0               1
## 14 September           22 2012          2810          0               0
## 15  November            5 2011          2345          0               1
## 16  November           16 2013          2445          0               0
## 17   October            4 2014          2910          1               1
## 18   October           18 2008          2158          1               0
## 19   October            2 2010          3056          1               1
## 20  November            1 2008          2120          0               0
## 21   October           24 2015          3291          0               1
## 22  November           17 2012          2173          0               1
## 23 September           17 2011          3677          0               0
## 24   October            3 2009          1669          1               1
## 25   October           15 2011          2113          0               1
## 26  November           15 2008          1434          0               1
## 27 September           12 2009          3202          0               1
## 28 September            8 2012          3961          1               0
## 29   October           12 2013          2757          0               0
## 30  November           20 2010          2324          0               0
## 31  November            2 2013          2678          0               0
## 32   October           31 2015          2680          0               0
## 33   October            4 2008          2972          0               1
## 34   October           10 2015          2940          0               0
## 35  November            3 2012          2017          0               1
## 36 September            4 2010          3059          0               1
## 37   October           25 2014          3721          0               1
## 38   October           17 2009          2194          0               1
## 39 September            6 2014          2985          0               1
## 40   October           23 2010          2298          0               1
## 41   October           30 2010          2956          0               0
## 42  November            7 2009           763          0               1
## 43 September           28 2013          2930          1               1
##    hour big_ten fold oof_predictions         tss          rss
## 1    11       0    2        3060.891   20709.218  314598.3384
## 2    11       0    2        2638.513   64468.753   61758.9170
## 3    11       1    2        2563.272   41578.055   15195.9452
## 4    11       1    5        3771.568 1940708.171   70454.1963
## 5    11       1    7        2164.361  455499.427   38165.8176
## 6    11       1    7        2197.386 1227457.869  437430.7945
## 7    11       1   10        2685.166   57555.357   79054.0579
## 8    13       0   10        2722.121    2217.753     968.4870
## 9    11       0   11        1837.405  776324.916 2847975.6062
## 10   11       0   13        2903.378   58125.846     337.7418
## 11   17       0   13        2207.106  234346.055  848046.1208
## 12   14       0   14        2813.150  537152.637  813874.1824
## 13   11       0   15        3072.573  980284.195  315200.7911
## 14   17       0   18        3136.223   27586.892  106421.3785
## 15   11       1   18        2317.776   89345.381     741.1694
## 16   15       1   18        2751.567   39563.985   93983.1110
## 17   11       1   20        3362.773   70805.497  205003.3999
## 18   13       1   20        2061.146  236105.590    9380.6415
## 19   11       1   21        2508.213  169820.660  300070.7191
## 20   13       1   23        2071.023  274478.520    2398.7061
## 21   11       1   24        3099.931  418729.381   36507.4214
## 22   11       1   24        2157.294  221753.381     246.6879
```

```
## 23   17       1   25        3583.605 1067281.195    8722.6421
## 24   11       0   26        1812.566  950443.613   20611.0638
## 25   11       1   28        2731.745  281862.218  382845.3878
## 26   11       1   29        1549.939 1463874.892   13441.8186
## 27   11       1   30        2936.278  311467.823   70608.1575
## 28   14       1   30        3562.376 1734734.032  158901.3193
## 29   14       1   30        3468.657   12790.032  506455.6018
## 30   12       1   31        1701.992  102340.474  386894.0151
## 31   14       1   33        3055.713    1162.334  142666.9637
## 32   14       0   34        2476.974    1302.706   41219.6973
## 33   11       1   35        2285.501  107645.032  471280.7064
## 34   15       0   38        3206.985   87671.078   71280.9015
## 35   11       1   39        2587.716  393012.357  325716.7056
## 36   11       0   39        2704.509  172302.218  125663.9978
## 37   11       1   39        2821.137 1160129.381  809753.3454
## 38   11       1   39        2217.890  202416.288     570.7220
## 39   11       0   41        3439.762  116344.451  206808.1609
## 40   11       1   42        2226.271  119651.637    5145.0301
## 41   14       1   42        2202.061   97402.055  568423.8248
## 42   11       0   42        1433.707 3537811.055  449847.3829
## 43   11       1   43        3206.046   81849.218   76201.1374
```

```
message("\n")
```

```
##
```

```
message("Cross validated with polynomial terms, removed outlier")
```

```
## Cross validated with polynomial terms, removed outlier
```

```
formula = "programs_sold ~ . + I(day_of_month^2) + month*day_of_month + I(year^2)"
cv_lm <- lm_cross_validation(formula, dat_lm, folds=nrow(dat_lm)-1)
```

```
## R^2 CV: 0.474666180086055
```

```
## RMSE CV: 493.667284659631
```

```
dat_fit <- lm(programs_sold ~ .+ I(day_of_month^2) + I(year^2) + day_of_month*month, dat_lm)

dat_fit$fitted.values
```

```
##        1        2        3        4        5        6        7        8
## 3056.489 3019.617 2729.014 3051.113 3531.373 2756.368 2261.076 2333.998
##        9       10       11       12       13       14       15       16
## 3438.737 3390.423 2131.644 2123.861 3069.997 3592.679 3976.933 2864.202
##       17       18       19       20       21       22       23       24
## 2355.789 2264.553 3450.200 2558.617 3037.103 3752.666 2221.585 3046.014
##       25       26       27       28       29       30       31       32
## 1985.098 1885.780 2206.742 1717.367 2910.345 1978.232 1064.403 2101.359
##       33       34       35       36       37       38       39       40
## 2919.135 2679.031 2500.386 2673.331 2826.753 2384.339 2015.257 2783.909
##       41       42       43
## 2147.351 2088.951 2806.179
```

Model Evaluation

```
t <- lm(formula, dat_lm[c(-17, -21),])
summary(t)
```

```
##
## Call:
## lm(formula = formula, data = dat_lm[c(-17, -21), ])
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -607.4 -124.8   10.2  140.4  406.6
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   9.262e+07  4.082e+07   2.269 0.031182 *
## monthOctober                  1.148e+03  2.428e+02   4.728 5.83e-05 ***
## monthSeptember                2.361e+03  2.132e+02  11.074 9.66e-12 ***
## day_of_month                 -6.769e+01  1.891e+01  -3.579 0.001282 **
## year                         -9.221e+04  4.059e+04  -2.271 0.030999 *
## homecoming                   -1.649e+02  1.444e+02  -1.142 0.263059
## morning_kickoff               2.115e+02  1.932e+02   1.095 0.282999
## hour                          1.861e+02  5.326e+01   3.494 0.001600 **
## big_ten                       1.027e+03  1.100e+02   9.337 4.28e-10 ***
## I(day_of_month^2)             3.190e+00  7.350e-01   4.340 0.000168 ***
## I(year^2)                     2.295e+01  1.009e+01   2.274 0.030819 *
## monthOctober:day_of_month    -5.479e+01  1.716e+01  -3.194 0.003459 **
## monthSeptember:day_of_month -8.429e+01  1.691e+01  -4.986 2.88e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 248.4 on 28 degrees of freedom
## Multiple R-squared:  0.9035, Adjusted R-squared:  0.8622
## F-statistic: 21.86 on 12 and 28 DF,  p-value: 4.377e-11
```

```
require(xgboost)
dat_lm <- dat_lm[-17,]

# create model matricies taking out 17th observation
train_matrix <- Matrix::sparse.model.matrix(programs_sold ~ . + I(day_of_month^2) + month*day_of_month -
response <- dat_lm$programs_sold

eval_cv <- function(evaluation_log){
    # input: xgb_cross_validation$evaluation_log
    # output: min test rmse, number of iterations, and dataset size
    min_index <- which(evaluation_log$test_rmse_mean == min(evaluation_log$test_rmse_mean))
    message("Minimum test rmse: ", evaluation_log$test_rmse_mean[min_index],
            "\nIterations: ", min_index,
            "\nDataset size: ", length(cv$folds[[1]]) * length(cv$folds))

    out = list("min_rmse" = evaluation_log$test_rmse_mean[min_index],
               "min_index" = min_index,
               "n" = length(cv$folds[[1]]) * length(cv$folds))
    return(out)
```

```
}

## TRAIN
cv <-
    xgb.cv(data=train_matrix,
           label=response,
           objective = "reg:linear",
           eta = .01,
           max_depth = 6,
           subsample = .8,
           colsample_bytree = .6,
           nfold = nrow(dat_lm),
           nrounds = 1500,
           early_stopping_rounds = 1500,
           print_every_n = 5,
           metrics="rmse")
eval_cv(cv$evaluation_log)

eval <- cv$evaluation_log %>% data.frame()
ggplot(eval) +
  geom_line(aes(x=iter, y=train_rmse_mean), size=1, color='darkgray', alpha=1) +
  geom_line(aes(x=iter, y=test_rmse_mean), size=1, color='blue', alpha=.6) +
  geom_point(data=eval[which(eval$test_rmse_mean == min(eval$test_rmse_mean)),],
             aes(x=iter, y=test_rmse_mean), size=3, color='red', alpha=.6) +
  ggtitle("XGBOOST cross validation") +
  ylab("RMSE") + xlab("Iteration #")

#
# xgb_fit <-
#     xgboost(data=train_matrix,
#            label=response,
#            objective = "reg:linear",
#            eta = .05,
#            max_depth = 8,
#            subsample = .8,
#            nfold = 20,
#            nrounds = 125,
#            metrics="rmse")
#
# xgb.importance(model = xgb_fit) %>% xgb.plot.importance()
```