

09 Elastic Net GLM

Fabian Blasch

06/01/2022

1 Load Data

```
# source AUX
source("../Misc/Auxilliary.R")
source("../Misc/model_eval.R")

# packages
get.package(c("lubridate", "glmnet", "glmnetUtils", "tidyverse"))

# load data
dat_bids <- readRDS("../Data/Bid Tab RDS/Bids_df_split.RDS")
```

2 No Feature Engineering

2.1 Variable Removal

```
# exclude variables that are not supposed to be in the model
lapply(dat_bids, \(x){

  # remove
  x$Vendor_Name <- NULL
  x$Contract_ID <- NULL

  # for now remove vendor ID to reduce training time
  # x$Vendor_ID <- NULL

  # return
  return(x)

}) -> dat_bids_mod

# assign training set
dat_train <- dat_bids_mod[["Train"]]
```

2.2 Cross Validation

```
# cross validation

## RAN ONCE ##
# cvfit <- glmnetUtils::cva.glmnet(Win ~., data = dat_train,
#                               family = binomial(link = "logit"),
#                               type.measure = "deviance", nfolds = 10,
```

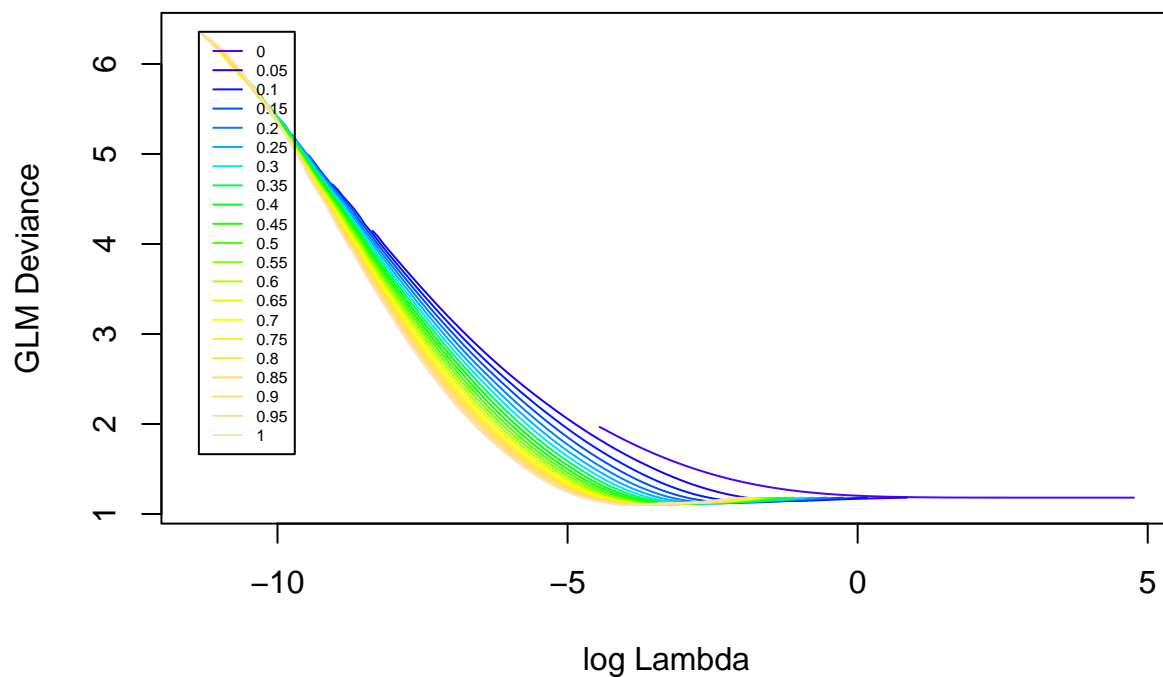
```
#                                     alpha = seq(0, 1, 0.05), nlambda = 100)

# save
# saveRDS(cvfit, "../Data/Models/Glmnets/Raw/cvfit.RDS")

# read file
cvfit <- readRDS("../Data/Models/Glmnets/Raw/cvfit.RDS")

# plot fit
plot(cvfit, c.legend = 0.5, main = "Cross Validation Results")
```

Cross Validation Results



2.3 Best Model

```
# obtain best model
per_matrix <- do.call(rbind, Map(function(x, y){

  cbind("Per" = x$cvm,
        "Lambda" = x$lambda,
        "Alpha" = rep(y, length(x$lambda)))

}, cvfit$modlist, cvfit$alpha))

# best performing paremeters
best_para <- per_matrix[which.min(per_matrix[, "Per"]), ]
```

```

# fit model using best parameters
fit_fin <- glmnetUtils::glmnet(Win ~., data = dat_train,
                             family = binomial(link = "logit"),
                             alpha = best_para["Alpha"],
                             lambda = best_para["Lambda"])

# predict
pred_vals <- predict(fit_fin, dat_train, type = "response")

# pred
Eval_Curve_prel(list(pred_vals),
                dat_train[["Win"]]) -> prelim

# curve
par(mfrow = c(1, 2))
Eval_Curve(prelim, col = "forestgreen", leg_text = "Glmnet")
Eval_Curve(prelim, col = 4, leg_text = "Glmnet", RoC = FALSE,
           act_label = dat_train[["Win"]])

```

