BSA Football initial models

Naren Prakash

2024-11-18

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
pbp_2018 <- clean_names(read_csv("pbp_merged_2018.csv"))
```

```
## New names:
## * `` -> `...1`
```

```
## Rows: 17783 Columns: 61
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (20): game_id, home_team, away_team, season_type, posteam, posteam_type...
## dbl  (36): ...1, Unnamed: 0_pbp, play_id, old_game_id, week, yardline_100, q...
## lgl   (3): run_location, run_gap, was_pressure
## date  (1): game_date
## time  (1): time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
pbp_2019 <- clean_names(read_csv("pbp_merged_2019.csv"))
```

```
## New names:
## Rows: 17730 Columns: 61
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (20): game_id, home_team, away_team, season_type, posteam, posteam_type... dbl
## (36): ...1, Unnamed: 0_pbp, play_id, old_game_id, week, yardline_100, q... lgl
## (3): run_location, run_gap, was_pressure date (1): game_date time (1): time
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
pbp_2020 <- clean_names(read_csv("pbp_merged_2020.csv"))
```

```
## New names:
## Rows: 18142 Columns: 61
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (20): game_id, home_team, away_team, season_type, posteam, posteam_type... dbl
```

```
## (36): ...1, Unnamed: 0_pbp, play_id, old_game_id, week, yardline_100, q... lgl
## (3): run_location, run_gap, was_pressure date (1): game_date time (1): time
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
pbp_2021 <- clean_names(read_csv("pbp_merged_2021.csv"))
```

```
## New names:
## * `` -> `...1`
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 18849 Columns: 61
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (20): game_id, home_team, away_team, season_type, posteam, posteam_type...
## dbl  (36): ...1, Unnamed: 0_pbp, play_id, old_game_id, week, yardline_100, q...
## lgl   (3): run_location, run_gap, was_pressure
## date  (1): game_date
## time  (1): time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
pbp_2022 <- clean_names(read_csv("pbp_merged_2022.csv"))
```

```
## New names:
## Rows: 18155 Columns: 61
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (20): game_id, home_team, away_team, season_type, posteam, posteam_type... dbl
## (36): ...1, Unnamed: 0_pbp, play_id, old_game_id, week, yardline_100, q... lgl
## (3): run_location, run_gap, was_pressure date (1): game_date time (1): time
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
pbp_2023 <- clean_names(read_csv("pbp_merged_2023.csv"))
```

```
## New names:
## Rows: 16000 Columns: 61
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (20): game_id, home_team, away_team, season_type, posteam, posteam_type... dbl
## (36): ...1, Unnamed: 0_pbp, play_id, old_game_id, week, yardline_100, q... lgl
## (3): run_location, run_gap, was_pressure date (1): game_date time (1): time
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
full <- pbp_2018 %>% full_join(pbp_2019) %>% full_join(pbp_2020) %>% full_join(pbp_2021) %>% full_join(
```

```
## Joining with `by = join_by(x1, unnamed_0_pbp, play_id, game_id, old_game_id,
## home_team, away_team, season_type, week, posteam, posteam_type, defteam,
```

```
## side_of_field, yardline_100, game_date, quarter_seconds_remaining,
## game_seconds_remaining, qtr, down, time, ydstogo, play_type, yards_gained,
## no_huddle, qb_scramble, pass_length, pass_location, air_yards,
## yards_after_catch, run_location, run_gap, total_home_score, total_away_score,
## posteam_score, defteam_score, score_differential, no_score_prob, ep, epa, wp,
## def_wp, home_wp, away_wp, wpa, season, play_type_nfl, out_of_bounds,
## unnamed_0_pbp_part, nflverse_game_id, possession_team, offense_formation,
## offense_personnel, defenders_in_box, defense_personnel, number_of_pass_rushers,
## ngs_air_yards, time_to_throw, was_pressure, route, defense_man_zone_type,
## defense_coverage_type)`
## Joining with `by = join_by(x1, unnamed_0_pbp, play_id, game_id, old_game_id,
## home_team, away_team, season_type, week, posteam, posteam_type, defteam,
## side_of_field, yardline_100, game_date, quarter_seconds_remaining,
## game_seconds_remaining, qtr, down, time, ydstogo, play_type, yards_gained,
## no_huddle, qb_scramble, pass_length, pass_location, air_yards,
## yards_after_catch, run_location, run_gap, total_home_score, total_away_score,
## posteam_score, defteam_score, score_differential, no_score_prob, ep, epa, wp,
## def_wp, home_wp, away_wp, wpa, season, play_type_nfl, out_of_bounds,
## unnamed_0_pbp_part, nflverse_game_id, possession_team, offense_formation,
## offense_personnel, defenders_in_box, defense_personnel, number_of_pass_rushers,
## ngs_air_yards, time_to_throw, was_pressure, route, defense_man_zone_type,
## defense_coverage_type)`
## Joining with `by = join_by(x1, unnamed_0_pbp, play_id, game_id, old_game_id,
## home_team, away_team, season_type, week, posteam, posteam_type, defteam,
## side_of_field, yardline_100, game_date, quarter_seconds_remaining,
## game_seconds_remaining, qtr, down, time, ydstogo, play_type, yards_gained,
## no_huddle, qb_scramble, pass_length, pass_location, air_yards,
## yards_after_catch, run_location, run_gap, total_home_score, total_away_score,
## posteam_score, defteam_score, score_differential, no_score_prob, ep, epa, wp,
## def_wp, home_wp, away_wp, wpa, season, play_type_nfl, out_of_bounds,
## unnamed_0_pbp_part, nflverse_game_id, possession_team, offense_formation,
## offense_personnel, defenders_in_box, defense_personnel, number_of_pass_rushers,
## ngs_air_yards, time_to_throw, was_pressure, route, defense_man_zone_type,
## defense_coverage_type)`
## Joining with `by = join_by(x1, unnamed_0_pbp, play_id, game_id, old_game_id,
## home_team, away_team, season_type, week, posteam, posteam_type, defteam,
## side_of_field, yardline_100, game_date, quarter_seconds_remaining,
## game_seconds_remaining, qtr, down, time, ydstogo, play_type, yards_gained,
## no_huddle, qb_scramble, pass_length, pass_location, air_yards,
## yards_after_catch, run_location, run_gap, total_home_score, total_away_score,
## posteam_score, defteam_score, score_differential, no_score_prob, ep, epa, wp,
## def_wp, home_wp, away_wp, wpa, season, play_type_nfl, out_of_bounds,
## unnamed_0_pbp_part, nflverse_game_id, possession_team, offense_formation,
## offense_personnel, defenders_in_box, defense_personnel, number_of_pass_rushers,
## ngs_air_yards, time_to_throw, was_pressure, route, defense_man_zone_type,
## defense_coverage_type)`
## Joining with `by = join_by(x1, unnamed_0_pbp, play_id, game_id, old_game_id,
## home_team, away_team, season_type, week, posteam, posteam_type, defteam,
## side_of_field, yardline_100, game_date, quarter_seconds_remaining,
## game_seconds_remaining, qtr, down, time, ydstogo, play_type, yards_gained,
## no_huddle, qb_scramble, pass_length, pass_location, air_yards,
## yards_after_catch, run_location, run_gap, total_home_score, total_away_score,
## posteam_score, defteam_score, score_differential, no_score_prob, ep, epa, wp,
## def_wp, home_wp, away_wp, wpa, season, play_type_nfl, out_of_bounds,
```

```
## unnamed_0_pbp_part, nflverse_game_id, possession_team, offense_formation,
## offense_personnel, defenders_in_box, defense_personnel, number_of_pass_rushers,
## ngs_air_yards, time_to_throw, was_pressure, route, defense_man_zone_type,
## defense_coverage_type)`
```

```
trainIndex <- createDataPartition(full$route, p = 0.7, list = FALSE)

train <- full[trainIndex, ]
test <- full[-trainIndex, ]

train <- subset(train, select = -c(run_gap, run_location, yards_after_catch, x1, unnamed_0_pbp,play_id,
train <- train %>%
  mutate_if(is.character, as.factor) %>%
  dplyr::select(where(~ !any(is.na(.))))
```

```
test <- subset(test, select = -c(run_gap, run_location, yards_after_catch, x1, unnamed_0_pbp,play_id, o
test <- test %>%
  mutate_if(is.character, as.factor) %>%
  dplyr::select(where(~ !any(is.na(.))))
```

```
nb <- naiveBayes(route ~ ., data = train)

y_pred <- predict(nb, newdata = test)

confusionMatrix(y_pred, test$route)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction ANGLE CORNER CROSS FLAT   GO HITCH   IN  OUT POST SCREEN SLANT WHEEL
##     ANGLE    586      0   332  476    2   270   70  371    0     29    44     5
##     CORNER     0     36    37    1   35     2    8   18   27      0     1     1
##     CROSS     60    171   928  384  234   477  304  572  162    107    88    38
##     FLAT     267      0   527 1885    4   169   15  311    0    545    71     4
##     GO         3    384   240   12 1735    74  100  121  373      2     7    36
##     HITCH    133     68   525  249  280  2962  778 1593  356     23   821    29
##     IN        16     64   130   54  118   245  262  195  108      6    11    16
##     OUT       49     28   346   58   79   285  193  422   95      8    89     7
##     POST       0    190   121    3  516    38   84   68  375      0     6    18
##     SCREEN    31      0   321 1000    3    13    0   13    0   2177    24     0
##     SLANT     47     14   300  292  122   625  232  811   38     16   999    20
##     WHEEL      0      2     3    0    5     0    0    1    0      0     0     2
##
## Overall Statistics
##
##                Accuracy : 0.3866
##                  95% CI : (0.3813, 0.392)
##     No Information Rate : 0.1613
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3104
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
```

```
##
##                      Class: ANGLE Class: CORNER Class: CROSS Class: FLAT
## Sensitivity               0.49161      0.037618      0.24357     0.42705
## Specificity               0.94808      0.995811      0.90785     0.93063
## Pos Pred Value            0.26819      0.216867      0.26326     0.49631
## Neg Pred Value            0.97967      0.971061      0.89876     0.91030
## Prevalence                0.03726      0.029914      0.11909     0.13797
## Detection Rate            0.01832      0.001125      0.02901     0.05892
## Detection Prevalence      0.06830      0.005189      0.11018     0.11872
## Balanced Accuracy         0.71985      0.516714      0.57571     0.67884
##                      Class: GO Class: HITCH Class: IN Class: OUT Class: POST
## Sensitivity            0.55378      0.57403   0.12805    0.09386     0.24446
## Specificity            0.95315      0.81906   0.96784    0.95501     0.96572
## Pos Pred Value         0.56203      0.37892   0.21388    0.25437     0.26427
## Neg Pred Value         0.95163      0.90908   0.94202    0.86569     0.96209
## Prevalence             0.09793      0.16129   0.06395    0.14054     0.04795
## Detection Rate         0.05423      0.09259   0.00819    0.01319     0.01172
## Detection Prevalence   0.09649      0.24434   0.03829    0.05186     0.04435
## Balanced Accuracy      0.75347      0.69655   0.54795    0.52444     0.60509
##                      Class: SCREEN Class: SLANT Class: WHEEL
## Sensitivity                0.74734      0.46229    1.136e-02
## Specificity                0.95168      0.91562    9.997e-01
## Pos Pred Value             0.60776      0.28413    1.538e-01
## Neg Pred Value             0.97409      0.95919    9.946e-01
## Prevalence                 0.09105      0.06755    5.501e-03
## Detection Rate             0.06805      0.03123    6.252e-05
## Detection Prevalence       0.11197      0.10990    4.064e-04
## Balanced Accuracy          0.84951      0.68896    5.055e-01
```

```
library(h2o)
```

```
## Warning: package 'h2o' was built under R version 4.4.2
```

```
##
## ----------------------------------------------------------------------
##
## Your next step is to start H2O:
##     > h2o.init()
##
## For H2O package documentation, ask for help:
##     > ??h2o
##
## After starting H2O, you can use the Web UI at http://localhost:54321
## For more information visit https://docs.h2o.ai
##
## ----------------------------------------------------------------------
```

```
##
## Attaching package: 'h2o'
```

```
## The following objects are masked from 'package:data.table':
##
##     hour, month, week, year
```

```
## The following objects are masked from 'package:lubridate':
##
##     day, hour, month, week, year
```

```
## The following objects are masked from 'package:stats':
##
##     cor, sd, var
```

```
## The following objects are masked from 'package:base':
##
##     %*%, %in%, &&, ||, apply, as.factor, as.numeric, colnames,
##     colnames<-, ifelse, is.character, is.factor, is.numeric, log,
##     log10, log1p, log2, round, signif, trunc
```

```
# Initialize the H2O cluster
h2o.init()
```

```
##  Connection successful!
##
## R is connected to the H2O cluster:
##     H2O cluster uptime:         10 minutes 50 seconds
##     H2O cluster timezone:       America/Los_Angeles
##     H2O data parsing timezone:  UTC
##     H2O cluster version:        3.44.0.3
##     H2O cluster version age:    10 months and 28 days
##     H2O cluster name:           H2O_started_from_R_naren_mlb655
##     H2O cluster total nodes:    1
##     H2O cluster total memory:   6.75 GB
##     H2O cluster total cores:    16
##     H2O cluster allowed cores:  16
##     H2O cluster healthy:        TRUE
##     H2O Connection ip:          localhost
##     H2O Connection port:        54321
##     H2O Connection proxy:       NA
##     H2O Internal Security:      FALSE
##     R Version:                  R version 4.4.1 (2024-06-14 ucrt)
```

```
## Warning in h2o.clusterInfo():
## Your H2O cluster version is (10 months and 28 days) old. There may be a newer version available.
## Please download and install the latest version from: https://h2o-release.s3.amazonaws.com/h2o/latest
```

```
# Convert the training dataset to an H2O frame
imputed_h2 <- as.h2o(train)
```

```
##   |                                                                              |
```

```
# Define the AutoML settings, limiting to tree-based models
automl <- h2o.automl(
  y = "route",                     # Response variable
  training_frame = imputed_h2,      # Training dataset
  max_runtime_secs = 60,            # Maximum runtime (in seconds)
  exclude_algos = c("DeepLearning", "GLM", "StackedEnsemble")  # Exclude non-tree-based models
)
```

```
##   |                                                                              |
## 00:46:32.686: AutoML: XGBoost is not available; skipping it.  |
```

```
# Extract the best model (one of the tree-based models)
final_model <- automl@leader

# Convert the test dataset to an H2O frame
testing <- as.h2o(test)
```

```
##   |                                                                      |
```

```
# Make predictions on the test data
predictions <- predict(final_model, testing)
```

```
##   |                                                                      |
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): Test/Validation
## dataset column 'offense_personnel' has levels not trained on: ["0 RB, 4 TE, 1
## WR", "1 RB, 0 TE, 3 WR,1 DB", "1 RB, 0 TE, 3 WR,1 DL", "1 RB, 3 TE, 0 WR,1 LB",
## "2 QB, 2 RB, 2 TE, 0 WR", "2 QB, 6 OL, 1 RB, 2 TE, 0 WR", "4 RB, 1 TE, 0 WR",
## "6 OL, 0 RB, 2 TE, 2 WR"]
```

```
## Warning in doTryCatch(return(expr), name, parentenv, handler): Test/Validation
## dataset column 'defense_personnel' has levels not trained on: ["2 DL, 2 LB, 6
## DB, 1 WR", "3 DL, 2 LB, 5 DB, 1 WR", "3 DL, 5 LB, 2 DB, 1 OL", "6 DL, 5 LB, 0
## DB"]
```

```
# Display predictions
head(predictions)
```

```
##   predict       ANGLE      CORNER       CROSS        FLAT          GO
## 1   HITCH 0.002257819 0.001333023 0.010658690 0.004138312 0.006233827
## 2     OUT 0.060271915 0.003943180 0.041703969 0.131594046 0.006109005
## 3      GO 0.001559473 0.020262228 0.008127529 0.002042808 0.926205195
## 4  CORNER 0.006874723 0.444505678 0.309404234 0.011244697 0.109657234
## 5   HITCH 0.005991998 0.005758866 0.153819179 0.015749394 0.009893108
## 6    POST 0.003736748 0.006241197 0.021848325 0.005808893 0.082809289
##         HITCH          IN         OUT        POST      SCREEN       SLANT
## 1 0.659019970 0.163444720 0.091427255 0.003006899 0.003577947 0.053938749
## 2 0.089692754 0.096556431 0.404995945 0.006285171 0.003276688 0.150604265
## 3 0.003450666 0.004824502 0.008503945 0.012740040 0.001817957 0.002217387
## 4 0.010944960 0.011757950 0.030068656 0.045618328 0.003156106 0.006037435
## 5 0.391989114 0.067650707 0.314920062 0.005248242 0.005118472 0.018230493
## 6 0.053097538 0.102872938 0.261832011 0.433579706 0.006513476 0.017524285
##          WHEEL
## 1 0.0009627888
## 2 0.0049666314
## 3 0.0082482700
## 4 0.0107300000
## 5 0.0056303664
## 6 0.0041355931
```

```
summary(final_model)
```

```
## Model Details:
## ==============
##
## H2OMultinomialModel: gbm
## Model Key:  GBM_1_AutoML_4_20241118_04632
## Model Summary:
##   number_of_trees number_of_internal_trees model_size_in_bytes min_depth
## 1              42                      504             2892587        15
##   max_depth mean_depth min_leaves max_leaves mean_leaves
## 1        15   15.00000        121        449   364.16864
##
## H2OMultinomialMetrics: gbm
```

```
## ** Reported on training data. **
##
## Training Set Metrics:
## =====================
##
## Extract training frame with `h2o.getFrame("AutoML_4_20241118_04632_training_train_sid_a440_1")`
## MSE: (Extract with `h2o.mse`) 0.3421219
## RMSE: (Extract with `h2o.rmse`) 0.5849119
## Logloss: (Extract with `h2o.logloss`) 0.9227366
## Mean Per-Class Error: 0.3655024
## AUC: (Extract with `h2o.auc`) NaN
## AUCPR: (Extract with `h2o.aucpr`) NaN
## R^2: (Extract with `h2o.r2`) 0.9543798
## Confusion Matrix: Extract with `h2o.confusionMatrix(<model>,train = TRUE)`)
## ========================================================================
## Confusion Matrix: Row labels: Actual class; Column labels: Predicted class
##        ANGLE CORNER CROSS  FLAT   GO HITCH   IN  OUT POST SCREEN SLANT WHEEL
## ANGLE   1656      2   200   470    0   245    9  158    0     36     7     0
## CORNER     0   1154   140     1  567    75   73   79  137      0     7     1
## CROSS    190     75  5702   800  306   790  153  465  132    224    53     2
## FLAT     202      4   363  8421   14   317   25  235    9    657    53     0
## GO         1     81   154     4 6292   199  101  156  237      0    86     0
## HITCH    113     14   560   228  106  9230  214  905  105     16   549     2
## IN        41     36   395    44   80  1241 2090  510  153      2   183     1
## OUT      253     55   824   586  133  2381  218 5384  156     23   479     0
## POST       0     67   132     1  674   162  122  181 2219      0    21     2
## SCREEN     8      0    64   824    0    32    2   19    0   5846     5     0
## SLANT     30      3    90   144   28   978   50  408   28     24  3261     0
## WHEEL      0     14    33    34   55    52   11   41   17      1    14   140
## Totals  2494   1505  8657 11557 8255 15702 3068 8541 3193   6829  4718   148
##         Error              Rate
## ANGLE  0.4050 =   1,127 / 2,783
## CORNER 0.4834 =   1,080 / 2,234
## CROSS  0.3587 =   3,190 / 8,892
## FLAT   0.1824 =  1,879 / 10,300
## GO     0.1394 =   1,019 / 7,311
## HITCH  0.2335 =  2,812 / 12,042
## IN     0.5624 =   2,686 / 4,776
## OUT    0.4868 =  5,108 / 10,492
## POST   0.3803 =   1,362 / 3,581
## SCREEN 0.1403 =     954 / 6,800
## SLANT  0.3535 =   1,783 / 5,044
## WHEEL  0.6602 =       272 / 412
## Totals 0.3117 = 23,272 / 74,667
##
## Hit Ratio Table: Extract with `h2o.hit_ratio_table(<model>,train = TRUE)`
## =======================================================================
## Top-10 Hit Ratios:
##     k hit_ratio
## 1   1  0.688323
## 2   2  0.883684
## 3   3  0.956340
## 4   4  0.985643
## 5   5  0.996196
```

```
## 6   6  0.999156
## 7   7  0.999839
## 8   8  0.999973
## 9   9  1.000000
## 10 10  1.000000
##
##
##
##
##
## H2OMultinomialMetrics: gbm
## ** Reported on cross-validation data. **
## ** 5-fold cross-validation on training data (Metrics computed for combined holdout predictions) **
##
## Cross-Validation Set Metrics:
## =====================
##
## Extract cross-validation frame with `h2o.getFrame("AutoML_4_20241118_04632_training_train_sid_a440_1
## MSE: (Extract with `h2o.mse`) 0.472678
## RMSE: (Extract with `h2o.rmse`) 0.6875158
## Logloss: (Extract with `h2o.logloss`) 1.394882
## Mean Per-Class Error: 0.6159273
## AUC: (Extract with `h2o.auc`) NaN
## AUCPR: (Extract with `h2o.aucpr`) NaN
## R^2: (Extract with `h2o.r2`) 0.9369708
## Hit Ratio Table: Extract with `h2o.hit_ratio_table(<model>,xval = TRUE)`
## =======================================================================
## Top-10 Hit Ratios:
##     k hit_ratio
## 1   1  0.467743
## 2   2  0.691323
## 3   3  0.823483
## 4   4  0.901375
## 5   5  0.944326
## 6   6  0.971098
## 7   7  0.987156
## 8   8  0.993839
## 9   9  0.997254
## 10 10  0.998621
##
##
##
##
## Cross-Validation Metrics Summary:
##                                mean        sd  cv_1_valid  cv_2_valid
## accuracy                   0.465734  0.001663    0.466988    0.466318
## auc                             NA  0.000000          NA          NA
## err                        0.534265  0.001663    0.533012    0.533682
## err_count               7978.400000 24.684004 7960.000000 7970.000000
## logloss                    1.402171  0.005930    1.405412    1.395664
## max_per_class_error        0.990018  0.005940    0.989796    0.986667
## mean_per_class_accuracy    0.382614  0.002988    0.386587    0.384509
## mean_per_class_error       0.617386  0.002988    0.613413    0.615491
## mse                        0.470717  0.002075    0.471573    0.469794
```

```
## pr_auc                          NA  0.000000          NA          NA
## r2                        0.937224  0.000707    0.936762    0.937761
## rmse                      0.686087  0.001513    0.686712    0.685415
##                         cv_3_valid  cv_4_valid  cv_5_valid
## accuracy                  0.463336    0.464743    0.467287
## auc                             NA          NA          NA
## err                       0.536664    0.535257    0.532713
## err_count              8014.000000 7993.000000 7955.000000
## logloss                   1.410315    1.401923    1.397542
## max_per_class_error       0.989011    0.984615    1.000000
## mean_per_class_accuracy   0.381188    0.378905    0.381880
## mean_per_class_error      0.618812    0.621095    0.618120
## mse                       0.473011    0.471590    0.467616
## pr_auc                          NA          NA          NA
## r2                        0.936914    0.936514    0.938170
## rmse                      0.687758    0.686724    0.683824
##
## Scoring History:
##              timestamp   duration number_of_trees training_rmse
## 1  2024-11-18 00:47:24 52.197 sec               0       0.91667
## 2  2024-11-18 00:47:25 52.710 sec               5       0.81341
## 3  2024-11-18 00:47:25 53.242 sec              10       0.75006
## 4  2024-11-18 00:47:26 53.768 sec              15       0.69621
## 5  2024-11-18 00:47:26 54.266 sec              20       0.66711
## 6  2024-11-18 00:47:27 54.777 sec              25       0.64051
## 7  2024-11-18 00:47:27 55.276 sec              30       0.62080
## 8  2024-11-18 00:47:28 55.761 sec              35       0.60474
## 9  2024-11-18 00:47:28 56.250 sec              40       0.58918
## 10 2024-11-18 00:47:29 56.444 sec              42       0.58491
##    training_logloss training_classification_error training_auc training_pr_auc
## 1           2.48491                        0.88837           NA              NA
## 2           1.76351                        0.46277           NA              NA
## 3           1.48965                        0.43606           NA              NA
## 4           1.29128                        0.41143           NA              NA
## 5           1.18913                        0.38803           NA              NA
## 6           1.09971                        0.36538           NA              NA
## 7           1.03552                        0.34722           NA              NA
## 8           0.98442                        0.33157           NA              NA
## 9           0.93571                        0.31560           NA              NA
## 10          0.92274                        0.31168           NA              NA
##
## Variable Importances: (Extract with `h2o.varimp`)
## =================================================
##
## Variable Importances:
##                  variable relative_importance scaled_importance percentage
## 1           ngs_air_yards        49820.445312          1.000000   0.467103
## 2           time_to_throw        19372.673828          0.388850   0.181633
## 3         possession_team        15778.593750          0.316709   0.147936
## 4       defense_personnel         7446.482910          0.149466   0.069816
## 5   defense_coverage_type         4339.363770          0.087100   0.040685
## 6       offense_formation         2779.407959          0.055789   0.026059
## 7       offense_personnel         2279.188721          0.045748   0.021369
## 8        defenders_in_box         1580.870361          0.031731   0.014822
```

```
## 9   defense_man_zone_type         1359.008423          0.027278   0.012742
## 10 number_of_pass_rushers         1289.365967          0.025880   0.012089
## 11           was_pressure          612.861511          0.012301   0.005746
```