Finetune Package for xgboost predict home runs

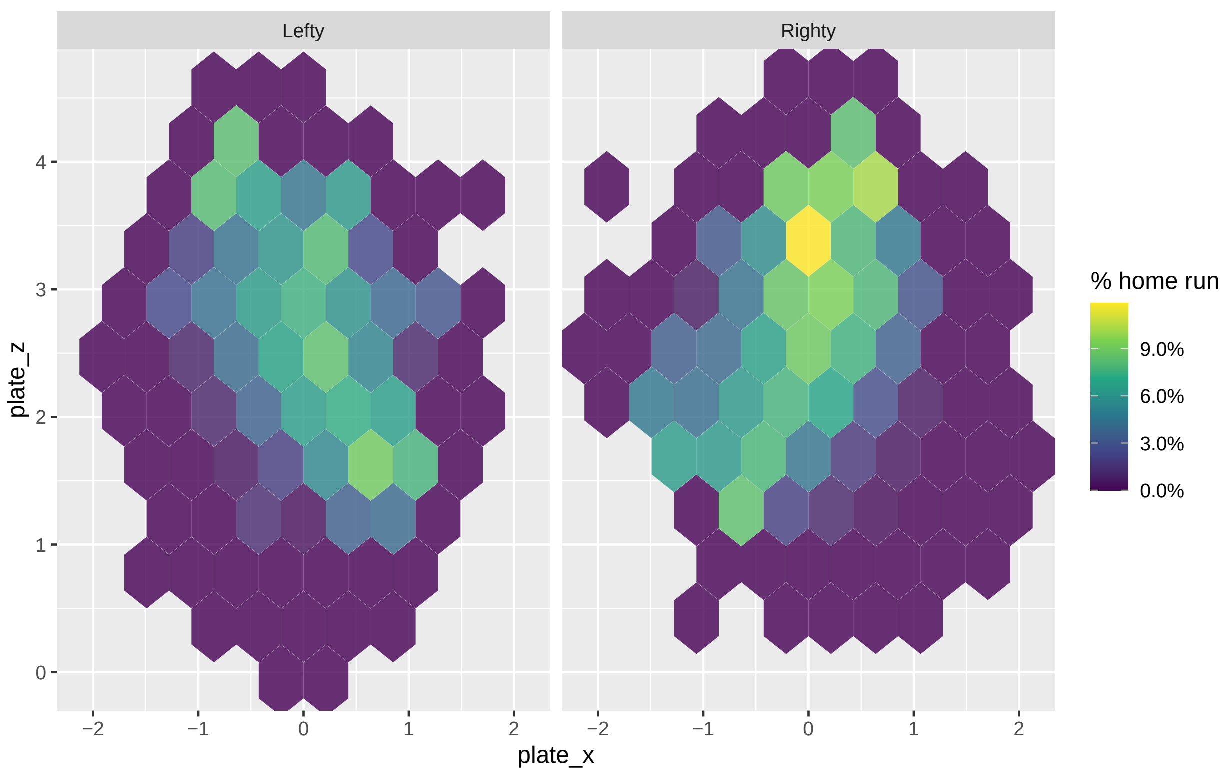
Andrew vanderWilden

8/12/2021

```
df <- read_csv('train_home_run.csv')
```

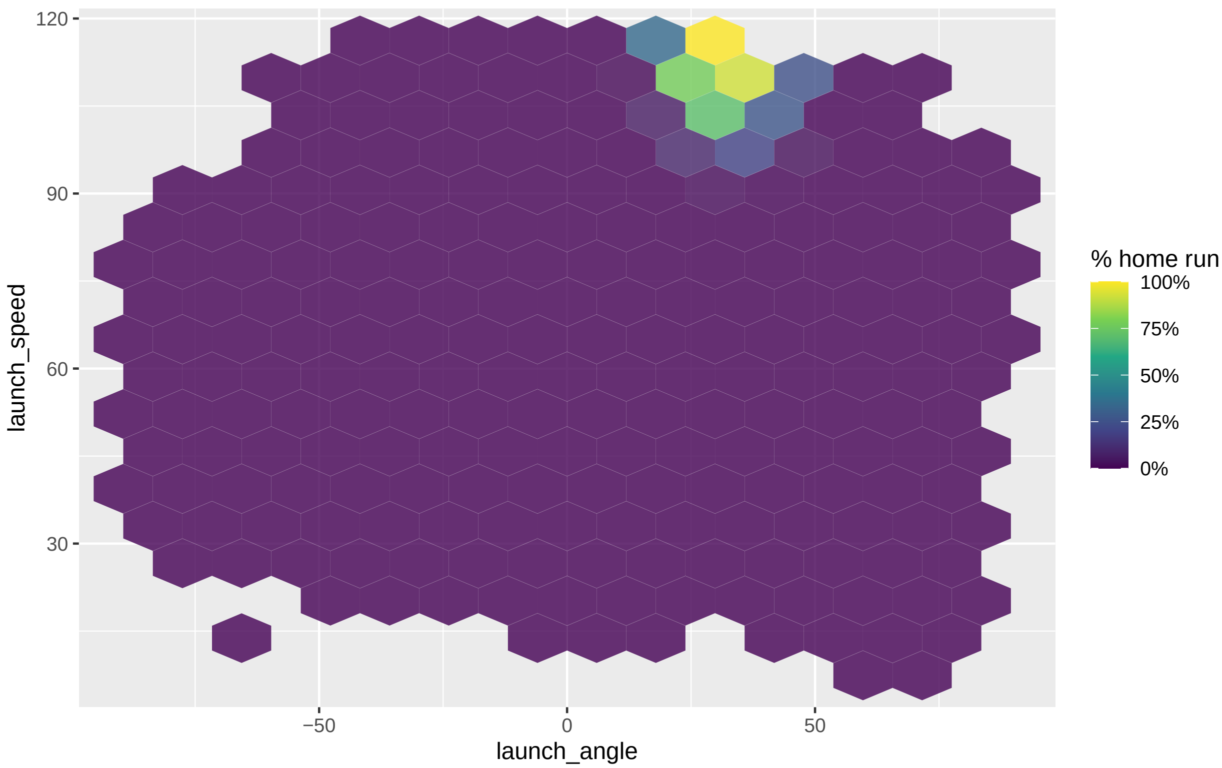
How are home runs distributed in the physical space around home plate? Is it different for righties and lefties?

```
df %>%
  mutate(is_batter_lefty = if_else(is_batter_lefty == 1, 'Lefty', 'Righty')) %>%
  ggplot(aes(plate_x, plate_z, z = is_home_run)) +
  facet_wrap(~is_batter_lefty) +
  stat_summary_hex(alpha = .8, bins = 10) +
  scale_fill_viridis_c(labels = scales::percent_format()) +
  labs(fill = '% home run')
```



What about launch angle and velocity?

```r
df %>%
  ggplot(aes(launch_angle, launch_speed, z = is_home_run)) +
  stat_summary_hex(alpha = 0.8, bins = 15) +
  scale_fill_viridis_c(labels = scales::percent_format()) +
  labs(fill = '% home run')
```



Build a model

```r
set.seed(123)
bb_split <- df %>%
  mutate(is_home_run = if_else(as.logical(is_home_run), 'HR', 'no'),
         is_home_run = factor(is_home_run)) %>%
  initial_split(strata = is_home_run)

bb_train <- training(bb_split)
bb_test <- testing(bb_split)

set.seed(234)
bb_folds <- vfold_cv(bb_train, strata = is_home_run)
bb_folds
```

```
## #  10-fold cross-validation using stratification
## # A tibble: 10 x 2
##    splits               id
##    <list>               <chr>
```

```
##  1 <split [31214/3469]> Fold01
##  2 <split [31214/3469]> Fold02
##  3 <split [31214/3469]> Fold03
##  4 <split [31215/3468]> Fold04
##  5 <split [31215/3468]> Fold05
##  6 <split [31215/3468]> Fold06
##  7 <split [31215/3468]> Fold07
##  8 <split [31215/3468]> Fold08
##  9 <split [31215/3468]> Fold09
## 10 <split [31215/3468]> Fold10
```

```r
bb_rec <-
  recipe(is_home_run ~ launch_angle + launch_speed + plate_x + plate_z +
           bb_type + bearing + pitch_mph +
           is_pitcher_lefty + is_batter_lefty +
           inning + balls + strikes + game_date,
         data = bb_train) %>%
  step_date(game_date, features = c('week'), keep_original_cols = FALSE) %>% # week of the year
  step_unknown(all_nominal_predictors()) %>%
  step_dummy(all_nominal_predictors(), one_hot = TRUE) %>% # one hot for xgb
  step_impute_median(all_numeric_predictors(), -launch_angle, -launch_speed) %>%
  step_impute_linear(launch_angle, launch_speed,
                     impute_with = imp_vars(plate_x, plate_z, pitch_mph)) %>%  # use linear regression
  step_nzv(all_predictors())

# prep just to see that it works
prep(bb_rec)
```

```
## Data Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor         13
##
## Training data contained 34683 data points and 15255 incomplete rows.
##
## Operations:
##
## Date features from game_date [trained]
## Unknown factor level assignment for bb_type, bearing [trained]
## Dummy variables from bb_type, bearing [trained]
## Median Imputation for plate_x, plate_z, pitch_mph, ... [trained]
## Linear regression imputation for launch_angle, launch_speed [trained]
## Sparse, unbalanced variable filter removed bb_type_unknown, bearing_unknown [trained]
```

```r
xgb_spec <-
  boost_tree(
    trees = tune(),
    min_n = tune(),
    mtry = tune(),
    learn_rate = 0.01
```

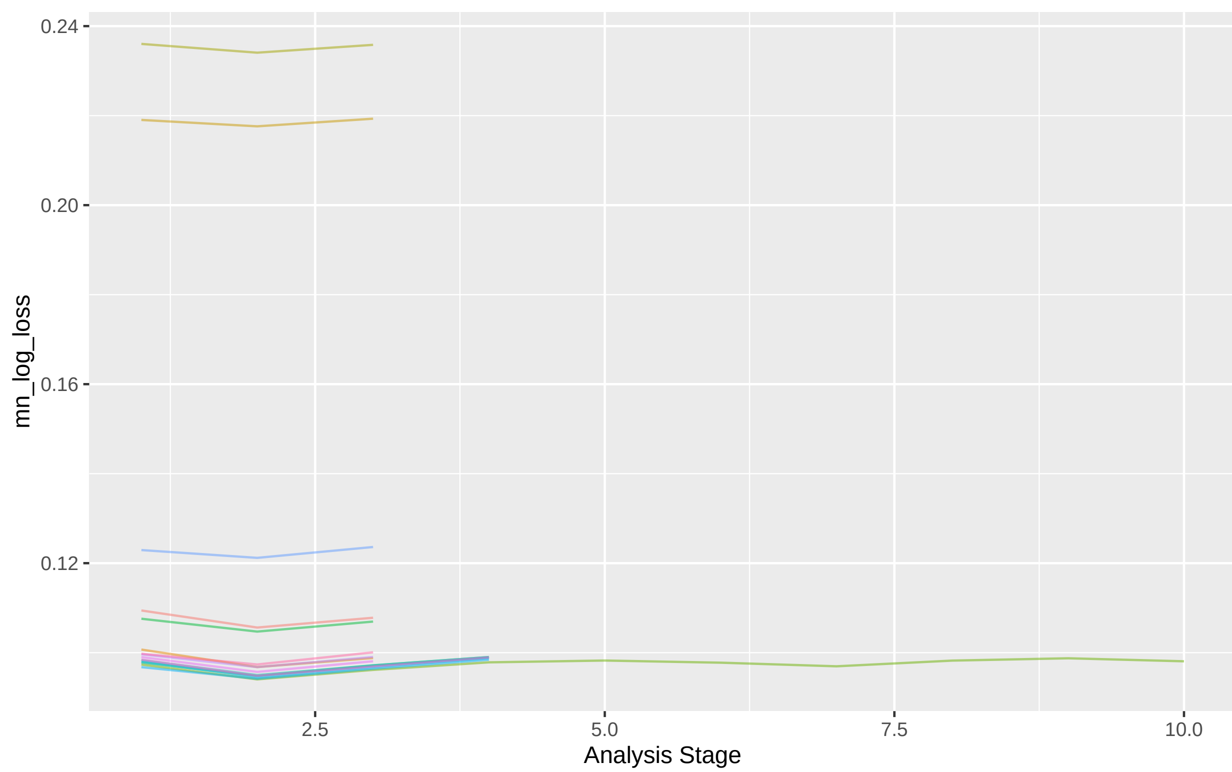
```
  ) %>%
  set_engine('xgboost') %>%
  set_mode('classification')

xgb_wf <- workflow(bb_rec, xgb_spec)
```

Use racing to tune XGB

```
library(finetune)

# for parallel processing
cl <- parallel::makePSOCKcluster(4)
doParallel::registerDoParallel(cl)

set.seed(345)
# anova tuning eliminates obviously bad sets of params after a few resamples saving time and computing p
xgb_res <- tune_race_anova(
  xgb_wf,
  resamples = bb_folds,
  grid = 15,
  metrics = metric_set(mn_log_loss),
  control = control_race(verbose_elim = TRUE)
)

xgb_res
```

```
## # Tuning results
## # 10-fold cross-validation using stratification
## # A tibble: 10 x 5
##    splits               id     .order .metrics          .notes
##    <list>               <chr>   <int> <list>            <list>
##  1 <split [31214/3469]> Fold01      2 <tibble [15 x 7]> <tibble [0 x 1]>
##  2 <split [31214/3469]> Fold02      3 <tibble [15 x 7]> <tibble [0 x 1]>
##  3 <split [31215/3468]> Fold10      1 <tibble [15 x 7]> <tibble [0 x 1]>
##  4 <split [31215/3468]> Fold07      4 <tibble [6 x 7]>  <tibble [0 x 1]>
##  5 <split [31214/3469]> Fold03      5 <tibble [1 x 7]>  <tibble [0 x 1]>
##  6 <split [31215/3468]> Fold04      8 <tibble [1 x 7]>  <tibble [0 x 1]>
##  7 <split [31215/3468]> Fold05      6 <tibble [1 x 7]>  <tibble [0 x 1]>
##  8 <split [31215/3468]> Fold06      9 <tibble [1 x 7]>  <tibble [0 x 1]>
##  9 <split [31215/3468]> Fold08     10 <tibble [1 x 7]>  <tibble [0 x 1]>
## 10 <split [31215/3468]> Fold09      7 <tibble [1 x 7]>  <tibble [0 x 1]>
```

```r
finetune::plot_race(xgb_res)
```




```r
show_best(xgb_res)
```

```
## # A tibble: 1 x 9
##    mtry trees min_n .metric     .estimator   mean     n std_err .config
##   <int> <int> <int> <chr>       <chr>       <dbl> <int>   <dbl> <chr>
## 1     6  1536    11 mn_log_loss binary     0.0981    10 0.00174 Preprocessor1_M~
```

Final Fit

```r
xgb_last <- xgb_wf %>%
  finalize_workflow(select_best(xgb_res, 'mn_log_loss')) %>%
  last_fit(bb_split)

xgb_last
```

```
## # Resampling results
## # Manual resampling
## # A tibble: 1 x 6
##   splits                 id               .metrics  .notes .predictions .workflow
##   <list>                 <chr>            <list>    <list> <list>       <list>
## 1 <split [34683/11561]> train/test split <tibble ~ <tibb~ <tibble [11~ <workflo~
```

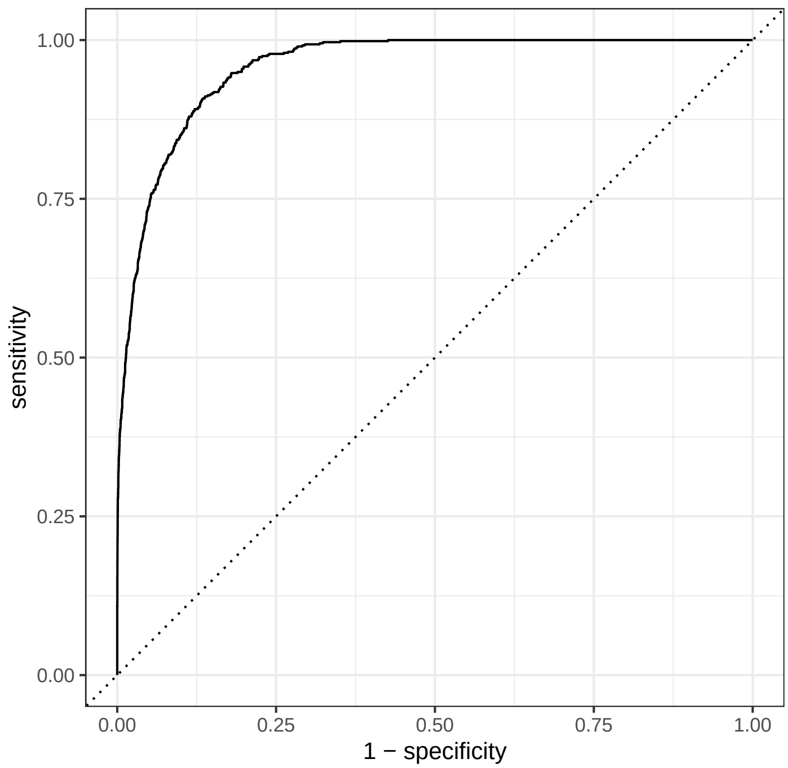
```r
collect_metrics(xgb_last)
```

```
## # A tibble: 2 x 4
##   .metric  .estimator .estimate .config
##   <chr>    <chr>          <dbl> <chr>
## 1 accuracy binary         0.964 Preprocessor1_Model1
## 2 roc_auc  binary         0.957 Preprocessor1_Model1
```

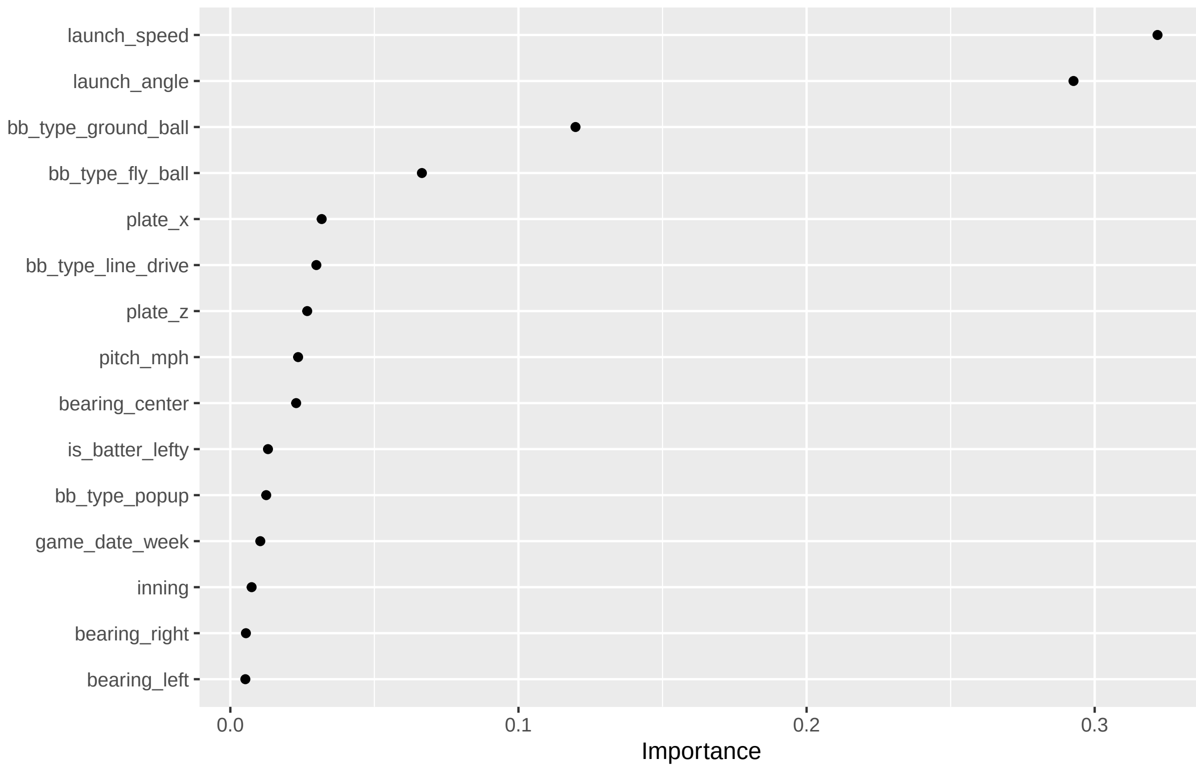
```r
mean(df$is_home_run)
```

```
## [1] 0.05291497
```

```r
roc_res <- roc_curve(xgb_last %>% collect_predictions(), truth = is_home_run,.pred_HR)

autoplot(roc_res)
```




```r
# variable importance plot
library(vip)
extract_workflow(xgb_last) %>%
  extract_fit_parsnip() %>%
  vip(geom = 'point', num_features = 15)
```


launch_speed
launch_angle
bb_type_ground_ball
bb_type_fly_ball
plate_x
bb_type_line_drive
plate_z
pitch_mph
bearing_center
is_batter_lefty
bb_type_popup
game_date_week
inning
bearing_right
bearing_left
0.0
0.1
0.2
0.3
Importance