

# Setup

```
library(tidyverse)
library(tidymodels)
```

Orig data from Antonio, Almeida, and Nunes (2019) - <https://doi.org/10.1016/j.dib.2018.11.126>  
(<https://doi.org/10.1016/j.dib.2018.11.126>) - Data dictionary -  
<https://github.com/rfordatascience/tidytuesday/tree/master/data/2020/2020-02-11#data-dictionary>  
(<https://github.com/rfordatascience/tidytuesday/tree/master/data/2020/2020-02-11#data-dictionary>)

## Data basics

```
hotels = read_csv(  
  'https://tidymodels.org/start/case-study/hotels.csv'  
) %>%  
  mutate(  
    across(where(is.character), as.factor)  
  )
```

```
## Rows: 50000 Columns: 23
```

```
## — Column specification —————  
## Delimiter: ","  
## chr  (11): hotel, children, meal, country, market_segment, distribution_chan...  
## dbl  (11): lead_time, stays_in_weekend_nights, stays_in_week_nights, adults,...  
## date  (1): arrival_date
```

```
##
```

```
## 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.
```

```
glimpse(hotels)
```

```
## Rows: 50,000
## Columns: 23
## $ hotel              <fct> City_Hotel, City_Hotel, Resort_Hotel, R...
## $ lead_time          <dbl> 217, 2, 95, 143, 136, 67, 47, 56, 80, 6...
## $ stays_in_weekend_nights <dbl> 1, 0, 2, 2, 1, 2, 0, 0, 0, 2, 1, 0, 1, ...
## $ stays_in_week_nights <dbl> 3, 1, 5, 6, 4, 2, 2, 3, 4, 2, 2, 1, 2, ...
## $ adults             <dbl> 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 1, 2, ...
## $ children           <fct> none, none, none, none, none, none, none, chi...
## $ meal               <fct> BB, BB, BB, HB, HB, SC, BB, BB, BB, BB,...
## $ country            <fct> DEU, PRT, GBR, ROU, PRT, GBR, ESP, ESP,...
## $ market_segment    <fct> Offline_TA/T0, Direct, Online_TA, Onlin...
## $ distribution_channel <fct> TA/T0, Direct, TA/T0, TA/T0, Direct, TA...
## $ is_repeated_guest  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ previous_cancellations <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ previous_bookings_not_canceled <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ reserved_room_type <fct> A, D, A, A, F, A, C, B, D, A, A, D, A, ...
## $ assigned_room_type <fct> A, K, A, A, F, A, C, A, D, A, D, D, A, ...
## $ booking_changes    <dbl> 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ deposit_type       <fct> No_Deposit, No_Deposit, No_Deposit, No_...
## $ days_in_waiting_list <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ customer_type      <fct> Transient-Party, Transient, Transient, ...
## $ average_daily_rate <dbl> 80.75, 170.00, 8.00, 81.00, 157.60, 49...
## $ required_car_parking_spaces <fct> none, none, none, none, none, none, non...
## $ total_of_special_requests <dbl> 1, 3, 2, 1, 4, 1, 1, 1, 1, 1, 0, 1, 0, ...
## $ arrival_date       <date> 2016-09-01, 2017-08-25, 2016-11-19, 20...
```

```
hotels %>%
  count(children) %>%
  mutate(prop = n/sum(n))
```

```
## # A tibble: 2 × 3
##   children      n  prop
##   <fct>    <int> <dbl>
## 1 children  4038 0.0808
## 2 none     45962 0.919
```

## Test train split

```
set.seed(123)

splits = initial_split(hotels, strata = children)

hotel_train = training(splits)
hotel_test = testing(splits)
```

## Checking on strata split

```
hotel_train %>%
  count(children) %>%
  mutate(prop = n/sum(n))
```

```
## # A tibble: 2 × 3
##   children      n  prop
##   <fct>    <int> <dbl>
## 1 children  3027 0.0807
## 2 none     34473 0.919
```

```
hotel_test %>%  
  count(children) %>%  
  mutate(prop = n/sum(n))
```

```
## # A tibble: 2 × 3  
##   children      n  prop  
##   <fct>    <int> <dbl>  
## 1 children  1011 0.0809  
## 2 none     11489 0.919
```

## Creating the validation set

```
set.seed(1234)  
val_set = validation_split(hotel_train, strata = children, prop = 0.8)  
val_set
```

```
## # Validation Set Split (0.8/0.2) using stratification  
## # A tibble: 1 × 2  
##   splits      id  
##   <list>    <chr>  
## 1 <split [30000/7500]> validation
```

## Logistic Regression model

```
show_engines("logistic_reg")
```

```
## # A tibble: 6 × 2
##   engine    mode
##   <chr>     <chr>
## 1 glm      classification
## 2 glmnet   classification
## 3 LiblineaR classification
## 4 spark    classification
## 5 keras    classification
## 6 stan     classification
```

```
lr_model = logistic_reg() %>%
  set_engine("glm")

lr_model %>%
  translate()
```

```
## Logistic Regression Model Specification (classification)
##
## Computational engine: glm
##
## Model fit template:
## stats::glm(formula = missing_arg(), data = missing_arg(), weights = missing_arg(),
##             family = stats::binomial)
```

## Recipe

```
holidays = c("AllSouls", "AshWednesday", "ChristmasEve", "Easter",  
             "ChristmasDay", "GoodFriday", "NewYearsDay", "PalmSunday")
```

```
lr_recipe = recipe(children ~ ., data = hotel_train) %>%  
  step_date(arrival_date) %>%  
  step_holiday(arrival_date, holidays = holidays) %>%  
  step_rm(arrival_date) %>%  
  step_rm(country) %>%  
  step_dummy(all_nominal_predictors()) %>%  
  step_zv(all_predictors())
```

```
lr_recipe
```

```
## Recipe  
##  
## Inputs:  
##  
##      role #variables  
## outcome      1  
## predictor     22  
##  
## Operations:  
##  
## Date features from arrival_date  
## Holiday features from arrival_date  
## Delete terms arrival_date  
## Delete terms country  
## Dummy variables from all_nominal_predictors()  
## Zero variance filter on all_predictors()
```

```
lr_recipe %>%
  prep() %>%
  bake(new_data = hotel_train)
```

```
## # A tibble: 37,500 × 76
##   lead_time stays_in_weekend_nights stays_in_week_nigh... adults is_repeated_gue...
##   <dbl>          <dbl>          <dbl> <dbl>          <dbl>
## 1         2             0             1     2             0
## 2        95             2             5     2             0
## 3        67             2             2     2             0
## 4        47             0             2     2             0
## 5        56             0             3     0             0
## 6         6             2             2     2             0
## 7       130             1             2     2             0
## 8        27             0             1     1             0
## 9        46             0             2     2             0
## 10       423             1             1     2             0
## # ... with 37,490 more rows, and 71 more variables: previous_cancellations <dbl>,
## #   previous_bookings_not_canceled <dbl>, booking_changes <dbl>,
## #   days_in_waiting_list <dbl>, average_daily_rate <dbl>,
## #   total_of_special_requests <dbl>, children <fct>, arrival_date_year <dbl>,
## #   arrival_date_AllSouls <dbl>, arrival_date_AshWednesday <dbl>,
## #   arrival_date_ChristmasEve <dbl>, arrival_date_Easter <dbl>,
## #   arrival_date_ChristmasDay <dbl>, arrival_date_GoodFriday <dbl>, ...
```

## Workflow

```
lr_workflow = workflow() %>%
  add_model(lr_model) %>%
  add_recipe(lr_recipe)
```



## Fit

```
lr_fit = lr_workflow %>%  
  fit(data = hotel_train)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
lr_fit
```

```
## == Workflow [trained] =====  
## Preprocessor: Recipe  
## Model: logistic_reg()  
##  
## — Preprocessor —————  
## 6 Recipe Steps  
##  
## • step_date()  
## • step_holiday()  
## • step_rm()  
## • step_rm()  
## • step_dummy()  
## • step_zv()  
##  
## — Model —————  
##  
## Call:  stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)  
##  
## Coefficients:  
##                (Intercept)                lead_time  
##                -2.543e+02                -1.287e+03
```

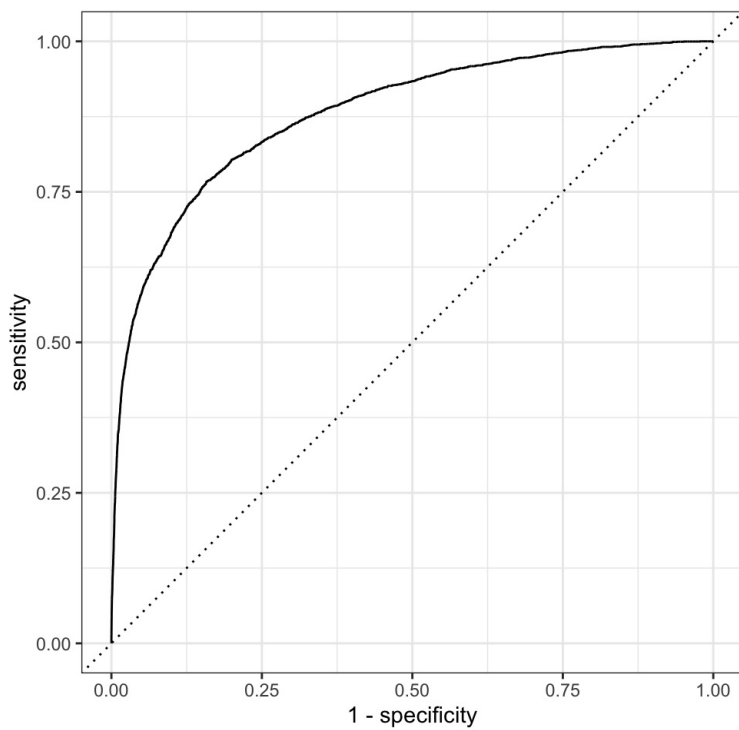
##	stays_in_weekend_nights	stays_in_week_nights
##	5.231e-02	-3.433e-02
##	adults	is_repeated_guest
##	7.328e-01	3.962e-01
##	previous_cancellations	previous_bookings_not_canceled
##	2.147e-01	3.728e-01
##	booking_changes	days_in_waiting_list
##	-2.396e-01	6.415e-03
##	average_daily_rate	total_of_special_requests
##	-1.049e-02	-4.936e-01
##	arrival_date_year	arrival_date_AllSouls
##	1.344e-01	1.006e+00
##	arrival_date_AshWednesday	arrival_date_ChristmasEve
##	2.019e-01	5.328e-01
##	arrival_date_Easter	arrival_date_ChristmasDay
##	-9.749e-01	-6.875e-01
##	arrival_date_GoodFriday	arrival_date_NewYearsDay
##	-1.593e-01	-1.185e+00
##	arrival_date_PalmSunday	hotel_Resort_Hotel
##	-6.243e-01	9.581e-01
##	meal_FB	meal_HB
##	-6.348e-01	-3.799e-02
##	meal_SC	meal_Undefined
##	1.285e+00	-1.938e-01
##	market_segment_Complementary	market_segment_Corporate
##	-1.345e+01	-1.213e+01
##	market_segment_Direct	market_segment_Groups
##	-1.314e+01	-1.217e+01
##	market_segment_Offline_TA.TO	market_segment_Online_TA
##	-1.353e+01	-1.362e+01
##	distribution_channel_Direct	distribution_channel_GDS
##	-4.351e-01	1.384e+01

```
##      distribution_channel_TA.T0      distribution_channel_Undefined
##      3.958e-02      -1.933e+01
##      reserved_room_type_B      reserved_room_type_C
##      -1.247e+00      -2.512e+00
##      reserved_room_type_D      reserved_room_type_E
##      1.170e+00      4.205e-01
##      reserved_room_type_F      reserved_room_type_G
##      -1.483e+00      -2.270e+00
##      reserved_room_type_H      reserved_room_type_L
##      -3.492e+00      1.327e+01
##      assigned_room_type_B      assigned_room_type_C
##      -5.158e-01      -1.922e+00
##
## ...
## and 34 more lines.
```

```
# collect_metrics(lr_fit)

lr_perf = lr_fit %>%
  augment(new_data = hotel_train) %>%
  select(children, starts_with(".pred"))

lr_perf %>%
  yardstick::roc_curve(
    children,
    .pred_children
  ) %>%
  autoplot()
```



```
lr_perf %>%  
  roc_auc(children, .pred_children)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>       <dbl>
## 1 roc_auc binary      0.881
```

```
lr_perf %>%
  precision(children, .pred_class)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>       <dbl>
## 1 precision binary      0.719
```

```
lr_perf %>%
  conf_mat(children, .pred_class)
```

```
##           Truth
## Prediction children none
##   children    1075   420
##   none        1952 34053
```

## Using validation split

```
lr_val_fit = lr_workflow %>%
  fit_resamples(val_set)

lr_val_fit
```

```
## # Resampling results
## # Validation Set Split (0.8/0.2) using stratification
## # A tibble: 1 × 4
##   splits          id      .metrics      .notes
##   <list>         <chr>    <list>      <list>
## 1 <split [30000/7500]> validation <tibble [2 × 4]> <tibble [0 × 1]>
```

```
collect_metrics(lr_val_fit)
```

```
## # A tibble: 2 × 6
##   .metric .estimator mean    n std_err .config
##   <chr>   <chr>    <dbl> <int>   <dbl> <chr>
## 1 accuracy binary    0.936     1      NA Preprocessor1_Model1
## 2 roc_auc  binary    0.865     1      NA Preprocessor1_Model1
```

## Fitting a lasso model

For the mixture argument 1 -> Lasso, 0 -> Ridge, other -> elastic net.

```
lasso_model = logistic_reg(penalty = tune(), mixture = 1) %>%
  set_engine("glmnet")

lasso_model %>%
  translate()
```

```
## Logistic Regression Model Specification (classification)
##
## Main Arguments:
##   penalty = tune()
##   mixture = 1
##
## Computational engine: glmnet
##
## Model fit template:
## glmnet::glmnet(x = missing_arg(), y = missing_arg(), weights = missing_arg(),
##   alpha = 1, family = "binomial")
```

```
lasso_model %>%
  parameters()
```

```
## Collection of 1 parameters for tuning
##
## identifier   type    object
##   penalty penalty nparam[+]
```

```
lasso_recipe = recipe(children ~ ., data = hotel_train) %>%  
  step_date(arrival_date) %>%  
  step_holiday(arrival_date, holidays = holidays) %>%  
  step_rm(arrival_date) %>%  
  step_rm(country) %>%  
  step_dummy(all_nominal_predictors()) %>%  
  step_zv(all_predictors()) %>%  
  step_normalize(all_predictors())  
  
lasso_recipe %>%  
  prep() %>%  
  bake(new_data = hotel_train)
```



```
## # A tibble: 37,500 × 76
##   lead_time stays_in_weekend_nights stays_in_week_nigh... adults is_repeated_gue...
##   <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 -0.858 -0.938 -0.767 0.337 -0.213
## 2 0.160 1.09 1.32 0.337 -0.213
## 3 -0.146 1.09 -0.245 0.337 -0.213
## 4 -0.365 -0.938 -0.245 0.337 -0.213
## 5 -0.267 -0.938 0.278 -3.59 -0.213
## 6 -0.814 1.09 -0.245 0.337 -0.213
## 7 0.544 0.0735 -0.245 0.337 -0.213
## 8 -0.584 -0.938 -0.767 -1.63 -0.213
## 9 -0.376 -0.938 -0.245 0.337 -0.213
## 10 3.75 0.0735 -0.767 0.337 -0.213
## # ... with 37,490 more rows, and 71 more variables: previous_cancellations <dbl>,
## # previous_bookings_not_canceled <dbl>, booking_changes <dbl>,
## # days_in_waiting_list <dbl>, average_daily_rate <dbl>,
## # total_of_special_requests <dbl>, children <fct>, arrival_date_year <dbl>,
## # arrival_date_AllSouls <dbl>, arrival_date_AshWednesday <dbl>,
## # arrival_date_ChristmasEve <dbl>, arrival_date_Easter <dbl>,
## # arrival_date_ChristmasDay <dbl>, arrival_date_GoodFriday <dbl>, ...
```

```
lasso_workflow = workflow() %>%
  add_model(lasso_model) %>%
  add_recipe(lasso_recipe)
```

## Tuning the model

```
lasso_res = lasso_workflow %>%
  tune_grid(
    val_set,
    grid = tibble(
      penalty = 10^seq(-4, -1, length.out = 30)
    ),
    control = control_grid(save_pred = TRUE),
    metrics = metric_set(roc_auc)
  )
```

```
lasso_res
```

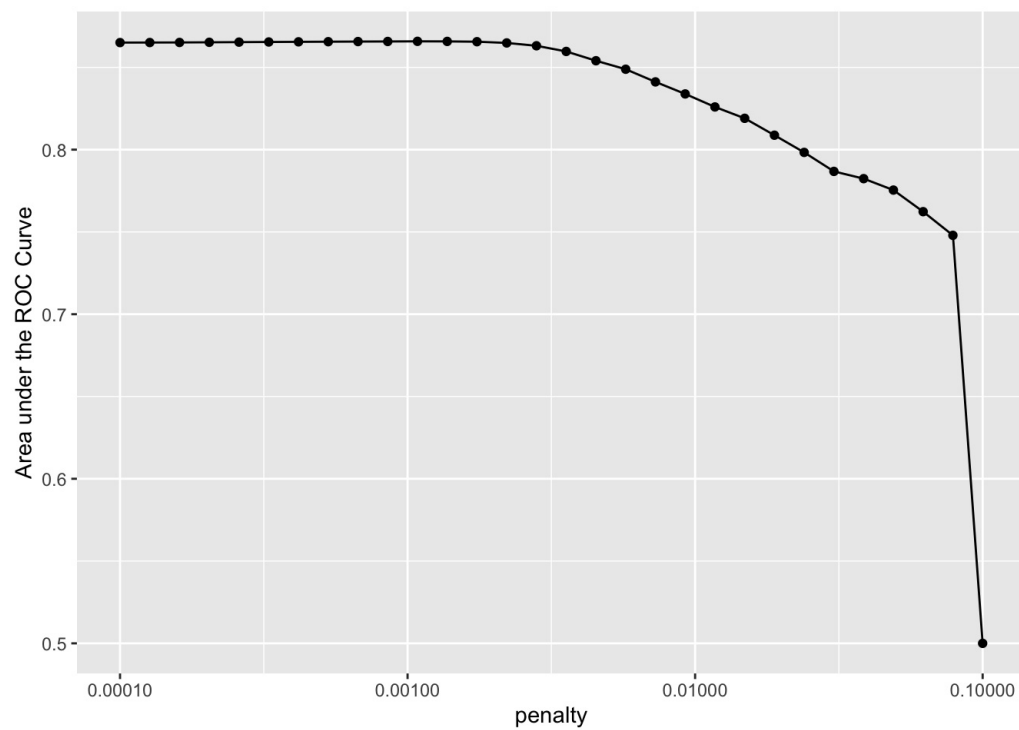
```
## # Tuning results
## # Validation Set Split (0.8/0.2) using stratification
## # A tibble: 1 × 5
```

##	splits	id	.metrics	.notes	.predictions
##	<list>	<chr>	<list>	<list>	<list>
##	1 <split [30000/7500]>	validation	<tibble [30 × 5]>	<tibble [0...	<tibble [225,00...

```
lasso_res %>%
  collect_metrics()
```

```
## # A tibble: 30 × 7
##   penalty .metric .estimator mean      n std_err .config
##   <dbl> <chr>   <chr>    <dbl> <int>   <dbl> <chr>
## 1 0.0001  roc_auc  binary    0.865     1      NA Preprocessor1_Model01
## 2 0.000127 roc_auc  binary    0.865     1      NA Preprocessor1_Model02
## 3 0.000161 roc_auc  binary    0.865     1      NA Preprocessor1_Model03
## 4 0.000204 roc_auc  binary    0.865     1      NA Preprocessor1_Model04
## 5 0.000259 roc_auc  binary    0.865     1      NA Preprocessor1_Model05
## 6 0.000329 roc_auc  binary    0.865     1      NA Preprocessor1_Model06
## 7 0.000418 roc_auc  binary    0.866     1      NA Preprocessor1_Model07
## 8 0.000530 roc_auc  binary    0.866     1      NA Preprocessor1_Model08
## 9 0.000672 roc_auc  binary    0.866     1      NA Preprocessor1_Model09
## 10 0.000853 roc_auc  binary    0.866     1      NA Preprocessor1_Model10
## # ... with 20 more rows
```

```
lasso_res %>%
  collect_metrics() %>%
  ggplot(aes(x = penalty, y = mean)) +
    geom_point() +
    geom_line() +
    ylab("Area under the ROC Curve") +
    scale_x_log10(labels = scales::label_number())
```



```
lasso_res %>%  
  show_best("roc_auc", n=10)
```

```
## # A tibble: 10 × 7
##   penalty .metric .estimator mean      n std_err .config
##   <dbl> <chr>   <chr>   <dbl> <int>   <dbl> <chr>
## 1 0.00108 roc_auc binary    0.866     1      NA Preprocessor1_Model11
## 2 0.00137 roc_auc binary    0.866     1      NA Preprocessor1_Model12
## 3 0.000853 roc_auc binary    0.866     1      NA Preprocessor1_Model10
## 4 0.000672 roc_auc binary    0.866     1      NA Preprocessor1_Model09
## 5 0.000530 roc_auc binary    0.866     1      NA Preprocessor1_Model08
## 6 0.00174 roc_auc binary    0.866     1      NA Preprocessor1_Model13
## 7 0.000418 roc_auc binary    0.866     1      NA Preprocessor1_Model07
## 8 0.000329 roc_auc binary    0.865     1      NA Preprocessor1_Model06
## 9 0.000259 roc_auc binary    0.865     1      NA Preprocessor1_Model05
## 10 0.000204 roc_auc binary    0.865     1      NA Preprocessor1_Model04
```

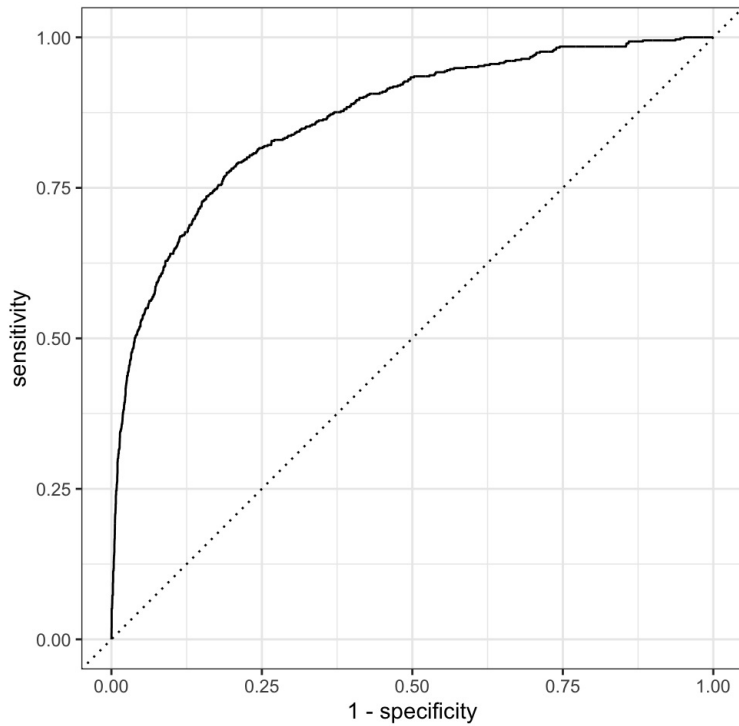
```
lasso_best = lasso_res %>%
  collect_metrics() %>%
  mutate(mean = round(mean, 2)) %>%
  arrange(desc(mean), desc(penalty)) %>%
  slice(1)
```

```
lasso_best
```

```
## # A tibble: 1 × 7
##   penalty .metric .estimator mean      n std_err .config
##   <dbl> <chr>   <chr>   <dbl> <int>   <dbl> <chr>
## 1 0.00174 roc_auc binary    0.87     1      NA Preprocessor1_Model13
```

```
lasso_best_pred = lasso_res %>%  
  collect_predictions(parameters = lasso_best)
```

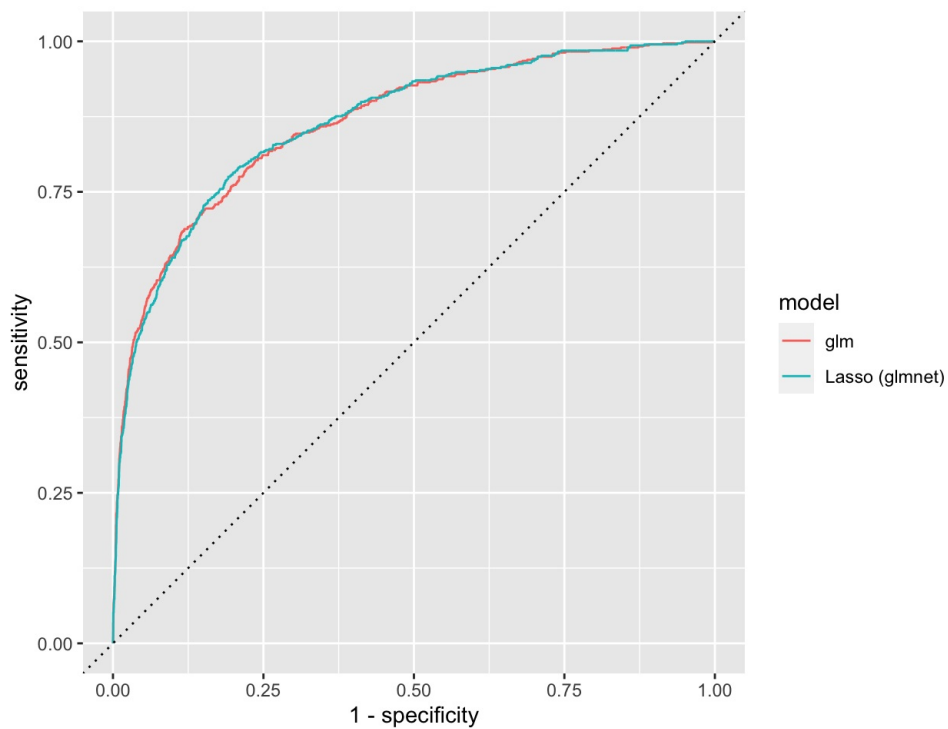
```
lasso_best_pred %>%  
  roc_curve(children, .pred_children) %>%  
  autoplot()
```



Comparing models

```
lr_val_fit = lr_workflow %>%  
  fit_resamples(  
    val_set,  
    control = control_resamples(save_pred=TRUE)  
  )  
  
bind_rows(  
  lr_val_fit %>%  
    collect_predictions() %>%  
    roc_curve(children, .pred_children) %>%  
    mutate(model = "glm"),  
  lasso_best_pred %>%  
    roc_curve(children, .pred_children) %>%  
    mutate(model = "Lasso (glmnet)")  
) %>%  
  ggplot(aes(x=1-specificity, y=sensitivity, col=model)) +  
    geom_path() +  
    geom_abline(lty = 3) +  
    coord_equal()
```





Random Forest

```
rf_model = rand_forest(mtry = tune(), min_n = tune(), trees = 100) %>%  
  set_engine("ranger", num.threads = 4) %>%  
  set_mode("classification")  
  
rf_recipe = recipe(children ~ ., data = hotel_train) %>%  
  step_date(arrival_date) %>%  
  step_holiday(arrival_date) %>%  
  step_rm(arrival_date)  
  
rf_workflow = workflow() %>%  
  add_model(rf_model) %>%  
  add_recipe(rf_recipe)  
  
rf_model %>%  
  parameters()
```

```
## Collection of 2 parameters for tuning  
##  
## identifier type object  
## mtry mtry nparam[?]  
## min_n min_n nparam[+]  
##  
## Model parameters needing finalization:  
## # Randomly Selected Predictors ('mtry')  
##  
## See `?dials::finalize` or `?dials::update.parameters` for more information.
```

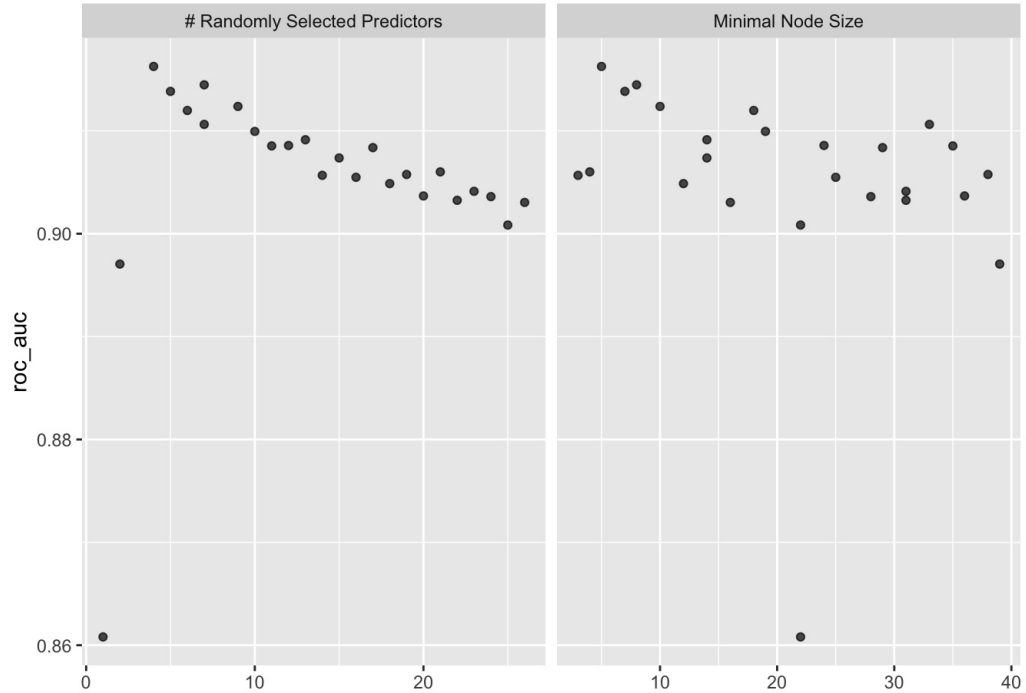
```
rf_res = rf_workflow %>%  
  tune_grid(  
    val_set,  
    grid = 25,  
    control = control_grid(save_pred = TRUE),  
    metrics = metric_set(roc_auc)  
  )
```

```
## i Creating pre-processing data to finalize unknown parameter: mtry
```

```
rf_res %>%  
  show_best(metric = "roc_auc")
```

```
## # A tibble: 5 × 8  
##   mtry min_n .metric .estimator mean      n std_err .config  
##   <int> <int> <chr>    <chr>    <dbl> <int>   <dbl> <chr>  
## 1     4     5 roc_auc  binary    0.916     1     NA Preprocessor1_Model08  
## 2     7     8 roc_auc  binary    0.914     1     NA Preprocessor1_Model23  
## 3     5     7 roc_auc  binary    0.914     1     NA Preprocessor1_Model13  
## 4     9    10 roc_auc  binary    0.912     1     NA Preprocessor1_Model11  
## 5     6    18 roc_auc  binary    0.912     1     NA Preprocessor1_Model20
```

```
autoplot(rf_res)
```



```
rf_best = rf_res %>%  
  select_best(metric = "roc_auc")  
  
rf_best_pred = rf_res %>%  
  collect_predictions(parameters = rf_best)
```

## Compare

```
bind_rows(  
  lr_val_fit %>%  
    collect_predictions() %>%  
    roc_curve(children, .pred_children) %>%  
    mutate(model = "glm"),  
  lasso_best_pred %>%  
    roc_curve(children, .pred_children) %>%  
    mutate(model = "Lasso (glmnet)"),  
  rf_best_pred %>%  
    roc_curve(children, .pred_children) %>%  
    mutate(model = "Random Forest (ranger)")  
) %>%  
ggplot(aes(x=1-specificity, y=sensitivity, col=model)) +  
  geom_path() +  
  geom_abline(lty = 3) +  
  coord_equal()
```

