Homework 3

PSTAT 131/231

Contents

Classification

For this assignment, we will be working with part of a Kaggle data set that was the subject of a machine learning competition and is often used for practicing ML models. The goal is classification; specifically, to predict which passengers would survive the Titanic shipwreck

Load the data from `data/titanic.csv` into *R* and familiarize yourself with the variables it contains using the codebook.

```
library(dplyr)
titanic <- read.csv("titanic.csv", header=TRUE)

titanic$pclass <- as.factor(titanic$pclass)
titanic$survived <- as.factor(titanic$survived)
titanic$sex <- as.factor(titanic$sex)
titanic <- titanic %>%  mutate(survived=relevel(survived,ref="Yes"))
levels(titanic$survived)
```

```
## [1] "Yes" "No"
```

```
#getwd()
head(titanic)
```

```
##   passenger_id survived pclass
## 1            1       No      3
## 2            2      Yes      1
## 3            3      Yes      3
## 4            4      Yes      1
## 5            5       No      3
## 6            6       No      3
##                                                 name    sex age sib_sp parch
## 1                             Braund, Mr. Owen Harris   male  22      1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38      1     0
## 3                              Heikkinen, Miss. Laina female  26      0     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35      1     0
## 5                            Allen, Mr. William Henry   male  35      0     0
## 6                                    Moran, Mr. James   male  NA      0     0
##             ticket    fare cabin embarked
```

```
## 1        A/5 21171  7.2500  <NA>        S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250  <NA>        S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500  <NA>        S
## 6           330877  8.4583  <NA>        Q
```

Notice that `survived` and `pclass` should be changed to factors. When changing `survived` to a factor, you may want to reorder the factor so that *"Yes"* is the first level.

Make sure you load the `tidyverse` and `tidymodels`!

Remember that you'll need to set a seed at the beginning of the document to reproduce your results.

Question 1

Split the data, stratifying on the outcome variable, `survived.` You should choose the proportions to split the data into. Verify that the training and testing data sets have the appropriate number of observations. Take a look at the training data and note any potential issues, such as missing data.

Why is it a good idea to use stratified sampling for this data?

```r
#Analyze the data fetched for datatype and null/missing values
#count of null values in a column
colSums(is.na(titanic))
```

```
## passenger_id     survived       pclass         name          sex          age
##            0            0            0            0            0          177
##       sib_sp        parch       ticket         fare        cabin     embarked
##            0            0            0            0          687            2
```

```r
#position of null values in a column
apply(is.na(titanic), 2, which)
```

```
## $passenger_id
## integer(0)
##
## $survived
## integer(0)
##
## $pclass
## integer(0)
##
## $name
## integer(0)
##
## $sex
## integer(0)
##
## $age
##   [1]   6  18  20  27  29  30  32  33  37  43  46  47  48  49  56  65  66  77
##  [19]  78  83  88  96 102 108 110 122 127 129 141 155 159 160 167 169 177 181
##  [37] 182 186 187 197 199 202 215 224 230 236 241 242 251 257 261 265 271 275
```

```
##  [55] 278 285 296 299 301 302 304 305 307 325 331 335 336 348 352 355 359 360
##  [73] 365 368 369 376 385 389 410 411 412 414 416 421 426 429 432 445 452 455
##  [91] 458 460 465 467 469 471 476 482 486 491 496 498 503 508 512 518 523 525
## [109] 528 532 534 539 548 553 558 561 564 565 569 574 579 585 590 594 597 599
## [127] 602 603 612 613 614 630 634 640 644 649 651 654 657 668 670 675 681 693
## [145] 698 710 712 719 728 733 739 740 741 761 767 769 774 777 779 784 791 793
## [163] 794 816 826 827 829 833 838 840 847 850 860 864 869 879 889
##
## $sib_sp
## integer(0)
##
## $parch
## integer(0)
##
## $ticket
## integer(0)
##
## $fare
## integer(0)
##
## $cabin
##   [1]   1   3   5   6   8   9  10  13  14  15  16  17  18  19  20  21  23  25
##  [19]  26  27  29  30  31  33  34  35  36  37  38  39  40  41  42  43  44  45
##  [37]  46  47  48  49  50  51  52  54  57  58  59  60  61  64  65  66  68  69
##  [55]  70  71  72  73  74  75  77  78  79  80  81  82  83  84  85  86  87  88
##  [73]  90  91  92  94  95  96  99 100 101 102 104 105 106 107 108 109 110 112
##  [91] 113 114 115 116 117 118 120 121 122 123 126 127 128 130 131 132 133 134
## [109] 135 136 139 141 142 143 144 145 146 147 148 150 151 153 154 155 156 157
## [127] 158 159 160 161 162 163 164 165 166 168 169 170 172 173 174 176 177 179
## [145] 180 181 182 183 185 187 188 189 190 191 192 193 197 198 199 200 201 202
## [163] 203 204 205 207 208 209 211 212 213 214 215 217 218 220 221 222 223 224
## [181] 226 227 228 229 230 232 233 234 235 236 237 238 239 240 241 242 243 244
## [199] 245 247 248 250 251 254 255 256 257 259 260 261 262 265 266 267 268 271
## [217] 272 273 275 277 278 279 280 281 282 283 284 286 287 288 289 290 291 294
## [235] 295 296 297 301 302 303 305 307 309 313 314 315 316 317 318 321 322 323
## [253] 324 325 327 329 331 334 335 336 339 343 344 345 347 348 349 350 351 353
## [271] 354 355 356 358 359 360 361 362 363 364 365 366 368 369 372 373 374 375
## [289] 376 377 379 380 381 382 383 384 385 386 387 388 389 390 392 393 396 397
## [307] 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 414 415 416
## [325] 417 418 419 420 421 422 423 424 425 426 427 428 429 432 433 434 437 438
## [343] 440 441 442 443 444 445 447 448 449 451 452 455 456 459 460 462 464 465
## [361] 466 467 468 469 470 471 472 473 475 477 478 479 480 481 482 483 484 486
## [379] 489 490 491 492 494 495 496 498 500 501 502 503 504 507 508 509 510 511
## [397] 512 514 515 518 519 520 522 523 525 526 527 529 530 531 532 533 534 535
## [415] 536 538 539 542 543 544 546 547 548 549 550 552 553 554 555 556 558 560
## [433] 561 562 563 564 565 566 567 568 569 570 571 574 575 576 577 579 580 581
## [451] 583 585 587 589 590 591 593 594 595 596 597 598 599 601 602 603 604 605
## [469] 606 607 608 609 611 612 613 614 615 616 617 618 620 621 623 624 625 627
## [487] 629 630 632 634 635 636 637 638 639 640 641 643 644 645 647 649 650 651
## [505] 652 653 654 655 656 657 658 659 661 662 664 665 666 667 668 669 671 673
## [523] 674 675 676 677 678 679 681 683 684 685 686 687 688 689 692 693 694 695
## [541] 696 697 698 703 704 705 706 707 709 710 714 715 719 720 721 722 723 724
## [559] 726 727 728 729 730 732 733 734 735 736 737 739 740 744 745 747 748 750
## [577] 751 753 754 755 756 757 758 759 761 762 763 765 767 768 769 770 771 772
```

```
## [595] 774 775 776 778 779 781 784 785 786 787 788 789 791 792 793 794 795 796
## [613] 798 799 800 801 802 804 805 806 808 809 811 812 813 814 815 817 818 819
## [631] 820 822 823 825 826 827 828 829 831 832 833 834 835 837 838 839 841 842
## [649] 843 844 845 846 847 848 849 851 852 853 855 856 857 859 860 861 862 864
## [667] 865 866 867 869 870 871 874 875 876 877 878 879 881 882 883 884 885 886
## [685] 887 889 891
##
## $embarked
## [1]  62 830
```

```
set.seed(3435)
library(tidymodels)

titanic_split <- initial_split(titanic, prop = 0.80,
                               strata = survived)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)

summary(titanic)
```

```
##   passenger_id   survived  pclass      name               sex
##  Min.   :  1.0   Yes:342   1:216   Length:891         female:314
##  1st Qu.:223.5   No :549   2:184   Class :character   male  :577
##  Median :446.0             3:491   Mode  :character
##  Mean   :446.0
##  3rd Qu.:668.5
##  Max.   :891.0
##
##       age            sib_sp          parch           ticket
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
##  1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character
##  Median :28.00   Median :0.000   Median :0.0000   Mode  :character
##  Mean   :29.70   Mean   :0.523   Mean   :0.3816
##  3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.000   Max.   :6.0000
##  NA's   :177
##       fare           cabin             embarked
##  Min.   :  0.00   Length:891         Length:891
##  1st Qu.:  7.91   Class :character   Class :character
##  Median : 14.45   Mode  :character   Mode  :character
##  Mean   : 32.20
##  3rd Qu.: 31.00
##  Max.   :512.33
##
```

```
str(titanic)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ passenger_id: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ survived    : Factor w/ 2 levels "Yes","No": 2 1 1 1 2 2 2 2 1 1 ...
##  $ pclass      : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ name        : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)
##  $ sex         : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
```

```
##  $ age         : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ sib_sp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ parch       : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ ticket      : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ fare        : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ cabin       : chr  NA "C85" NA "C123" ...
##  $ embarked    : chr  "S" "C" "S" "S" ...
```

```
str(titanic_train)
```

```
## 'data.frame':    712 obs. of  12 variables:
##  $ passenger_id: int  1 5 7 8 14 15 17 19 21 25 ...
##  $ survived    : Factor w/ 2 levels "Yes","No": 2 2 2 2 2 2 2 2 2 2 ...
##  $ pclass      : Factor w/ 3 levels "1","2","3": 3 3 1 3 3 3 3 3 2 3 ...
##  $ name        : chr  "Braund, Mr. Owen Harris" "Allen, Mr. William Henry" "McCarthy, Mr. Timothy J"
##  $ sex         : Factor w/ 2 levels "female","male": 2 2 2 2 2 1 2 1 2 1 ...
##  $ age         : num  22 35 54 2 39 14 2 31 35 8 ...
##  $ sib_sp      : int  1 0 0 3 1 0 4 1 0 3 ...
##  $ parch       : int  0 0 0 1 5 0 1 0 0 1 ...
##  $ ticket      : chr  "A/5 21171" "373450" "17463" "349909" ...
##  $ fare        : num  7.25 8.05 51.86 21.07 31.27 ...
##  $ cabin       : chr  NA NA "E46" NA ...
##  $ embarked    : chr  "S" "S" "S" "S" ...
```

```
str(titanic_test)
```

```
## 'data.frame':    179 obs. of  12 variables:
##  $ passenger_id: int  6 9 12 13 26 34 38 39 44 45 ...
##  $ survived    : Factor w/ 2 levels "Yes","No": 2 1 1 2 1 2 2 2 1 1 ...
##  $ pclass      : Factor w/ 3 levels "1","2","3": 3 3 1 3 3 2 3 3 2 3 ...
##  $ name        : chr  "Moran, Mr. James" "Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)" "Bonnel
##  $ sex         : Factor w/ 2 levels "female","male": 2 1 1 2 1 2 2 1 1 1 ...
##  $ age         : num  NA 27 58 20 38 66 21 18 3 19 ...
##  $ sib_sp      : int  0 0 0 0 1 0 0 2 1 0 ...
##  $ parch       : int  0 2 0 0 5 0 0 0 2 0 ...
##  $ ticket      : chr  "330877" "347742" "113783" "A/5. 2151" ...
##  $ fare        : num  8.46 11.13 26.55 8.05 31.39 ...
##  $ cabin       : chr  NA NA "C103" NA ...
##  $ embarked    : chr  "Q" "S" "S" "S" ...
```

Stratifying on the outcome variable, `survived.` provides better coverage of the population isrepresented in the sampling.

Question 2

Using the **training** data set, explore/describe the distribution of the outcome variable `survived`.

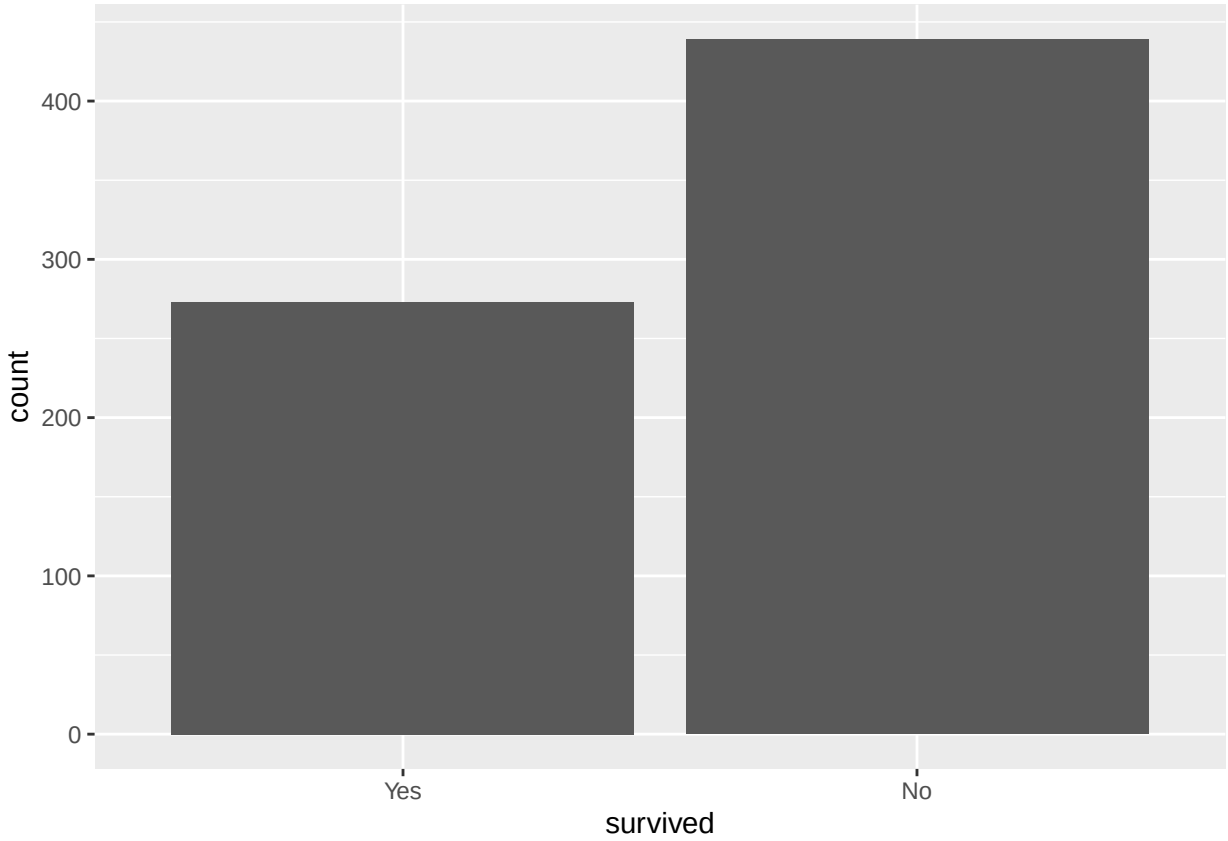
```
summary(titanic_train)
```

```
##   passenger_id   survived  pclass      name              sex
##  Min.   :  1.0   Yes:273   1:176   Length:712         female:247
```

```
##  1st Qu.:214.8   No :439   2:155   Class :character   male  :465
##  Median :435.5             3:381   Mode  :character
##  Mean   :436.9
##  3rd Qu.:656.2
##  Max.   :891.0
##
##       age            sib_sp           parch          ticket
##  Min.   : 0.67   Min.   :0.0000   Min.   :0.000   Length:712
##  1st Qu.:20.00   1st Qu.:0.0000   1st Qu.:0.000   Class :character
##  Median :28.00   Median :0.0000   Median :0.000   Mode  :character
##  Mean   :29.89   Mean   :0.5379   Mean   :0.375
##  3rd Qu.:38.00   3rd Qu.:1.0000   3rd Qu.:0.000
##  Max.   :80.00   Max.   :8.0000   Max.   :6.000
##  NA's   :148
##       fare            cabin             embarked
##  Min.   :  0.000   Length:712         Length:712
##  1st Qu.:  7.918   Class :character   Class :character
##  Median : 14.454   Mode  :character   Mode  :character
##  Mean   : 33.113
##  3rd Qu.: 31.069
##  Max.   :512.329
##
```

Total observations in the Train data set = 712 , where 273 survived and 439 did not. We do have null/missing values in some predictors.not all. below is the visualization of training data for actual outcome.

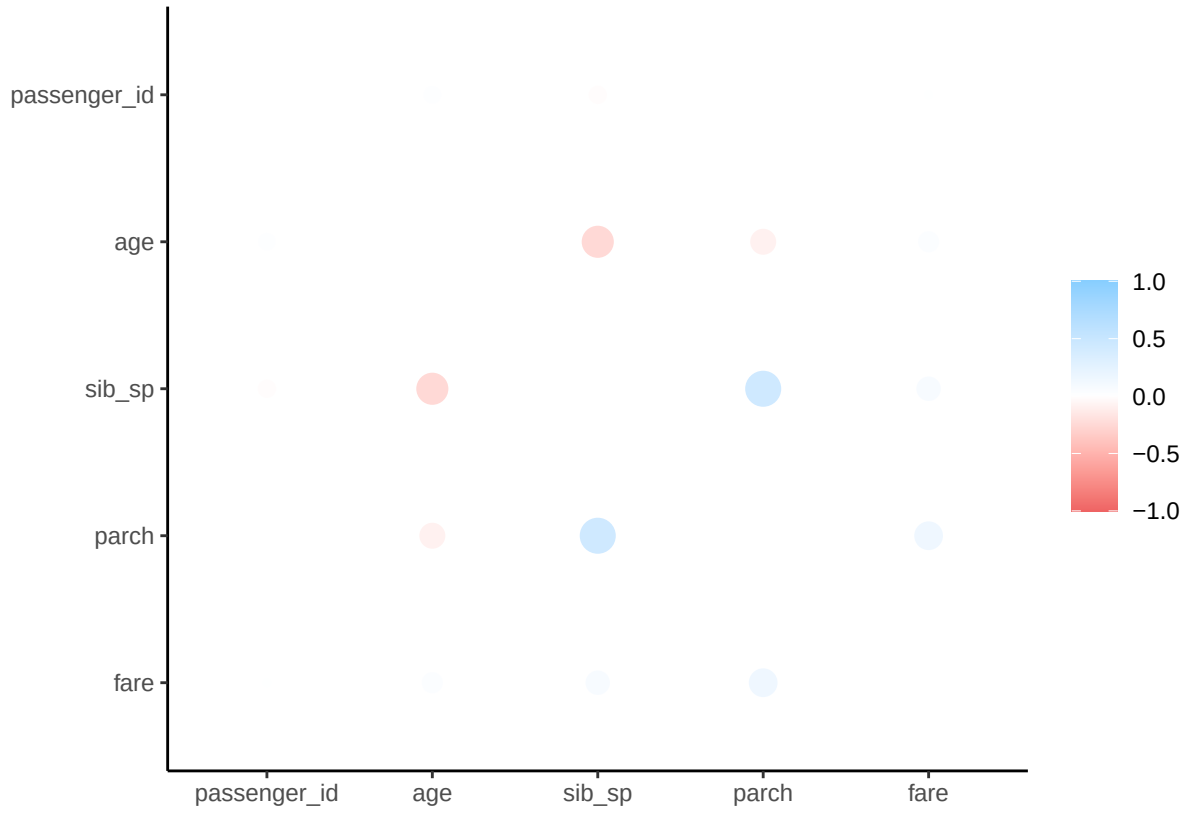
```
titanic_train %>%
  ggplot(aes(x =survived)) +
  geom_bar()
```



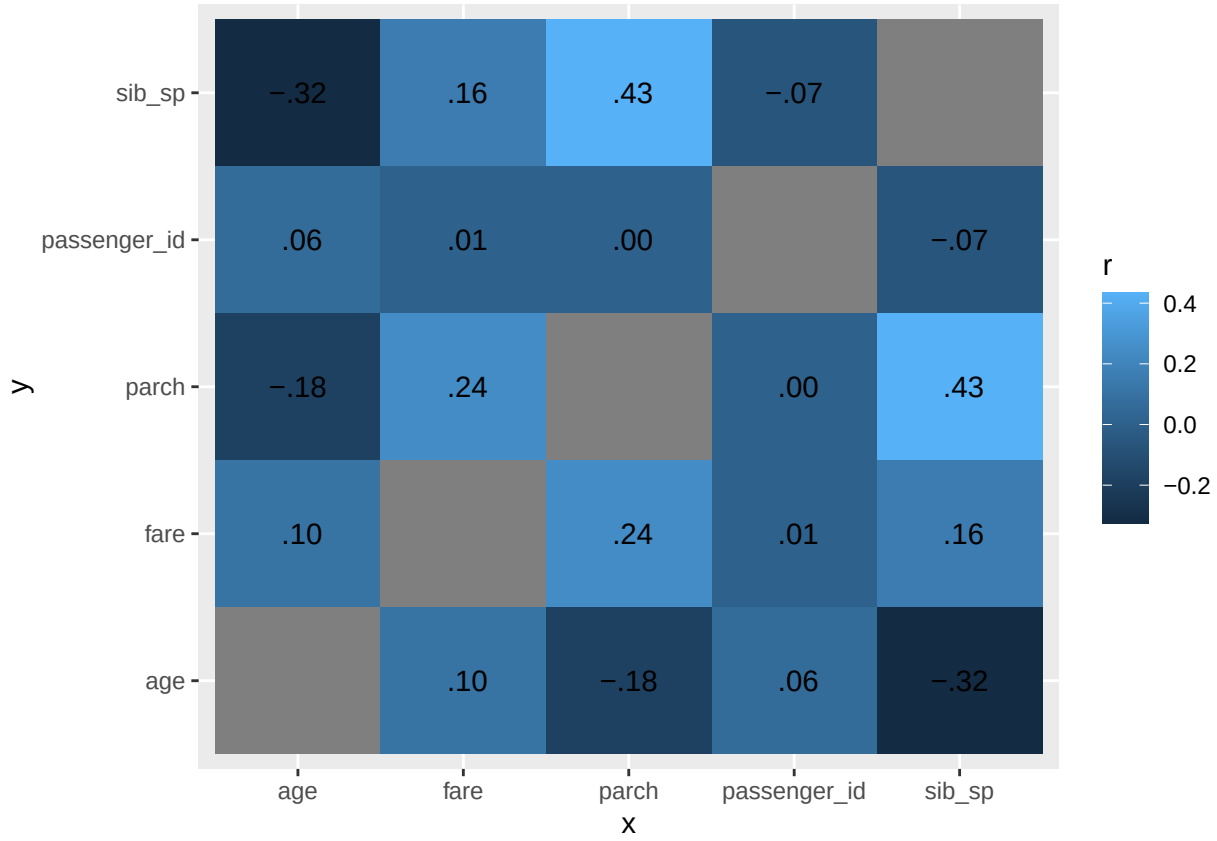
Question 3

Using the **training** data set, create a correlation matrix of all continuous variables. Create a visualization of the matrix, and describe any patterns you see. Are any predictors correlated with each other? Which ones, and in which direction?

```
library(discrim)
library(poissonreg)
library(corrr)
library(klaR) # for naive bayes
tidymodels_prefer()

data<-titanic_train
data <- data %>%
        select(where(is.numeric))
cor_data <- data %>%
  correlate()
rplot(cor_data)
```




```
cor_data %>%
  stretch() %>%
  ggplot(aes(x, y, fill = r)) +
  geom_tile() +
  geom_text(aes(label = as.character(fashion(r))))
```



Notice that the upper and lower triangles of the matrix are identical; that's a common feature of correlation matrices. The grey squares represent the variances of the variables. Again, we see that only sib_sp(# of siblings / spouses aboard the Titanic) and parch(# of parents / children aboard the Titanic) have much of any correlation with each other, and it's only about 0.45

Question 4

Using the **training** data, create a recipe predicting the outcome variable `survived`. Include the following predictors: ticket class, sex, age, number of siblings or spouses aboard, number of parents or children aboard, and passenger fare.

Recall that there were missing values for `age`. To deal with this, add an imputation step using `step_impute_linear()`. Next, use `step_dummy()` to **dummy** encode categorical predictors. Finally, include interactions between:

- Sex and passenger fare, and
- Age and passenger fare.

You'll need to investigate the `tidymodels` documentation to find the appropriate step functions to use.

```
data_recipe <- recipe(survived ~ pclass + sex + age + sib_sp + parch + fare, data = titanic_train)%>%
  step_impute_linear(age)%>%
  step_dummy(all_nominal_predictors())%>%
  step_interact(terms = ~ Sex:fare + age:fare)
data_recipe
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          6
##
## Operations:
##
## Linear regression imputation for age
## Dummy variables from all_nominal_predictors()
## Interactions with Sex:fare + age:fare
```

```
summary(data_recipe)
```

```
## # A tibble: 7 x 4
##   variable type    role      source
##   <chr>    <chr>   <chr>     <chr>
## 1 pclass   nominal predictor original
## 2 sex      nominal predictor original
## 3 age      numeric predictor original
## 4 sib_sp   numeric predictor original
## 5 parch    numeric predictor original
## 6 fare     numeric predictor original
## 7 survived nominal outcome   original
```

Question 5

Specify a **logistic regression** model for classification using the `"glm"` engine. Then create a workflow. Add your model and the appropriate recipe. Finally, use `fit()` to apply your workflow to the **training** data.

Hint: Make sure to store the results of `fit()`. You'll need them later on.

```
log_reg <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")

log_wkflow <- workflow() %>%
  add_model(log_reg) %>%
  add_recipe(data_recipe)

log_fit <- fit(log_wkflow, titanic_train)

log_fit %>%
  tidy()
```

```
## # A tibble: 8 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) -4.26      0.550       -7.75 9.34e-15
## 2 age          0.0476    0.00959      4.96 7.05e- 7
```

```
## 3 sib_sp       0.407     0.123        3.30 9.54e- 4
## 4 parch        0.265     0.151        1.76 7.92e- 2
## 5 fare        -0.00366   0.00277     -1.32 1.86e- 1
## 6 pclass_X2    1.21      0.338        3.59 3.28e- 4
## 7 pclass_X3    2.35      0.348        6.75 1.46e-11
## 8 sex_male     2.74      0.223       12.3  1.40e-34
```

Question 6

Repeat Question 5, but this time specify a linear discriminant analysis model for classification using the "MASS" engine.

```
lda_mod <- discrim_linear() %>%
  set_mode("classification") %>%
  set_engine("MASS")

lda_wkflow <- workflow() %>%
  add_model(lda_mod) %>%
  add_recipe(data_recipe)

lda_fit <- fit(lda_wkflow, titanic_train)

lda_fit
```

```
## == Workflow [trained] ==========================================================
## Preprocessor: Recipe
## Model: discrim_linear()
##
## -- Preprocessor ----------------------------------------------------------------
## 3 Recipe Steps
##
## * step_impute_linear()
## * step_dummy()
## * step_interact()
##
## -- Model -----------------------------------------------------------------------
## Call:
## lda(..y ~ ., data = data)
##
## Prior probabilities of groups:
##      Yes       No
## 0.383427 0.616573
##
## Group means:
##          age    sib_sp     parch     fare pclass_X2 pclass_X3  sex_male
## Yes 28.67760 0.4798535 0.4175824 50.26357 0.2673993 0.3296703 0.3260073
## No  29.97601 0.5740319 0.3485194 22.44772 0.1867882 0.6628702 0.8564920
##
## Coefficients of linear discriminants:
##                     LD1
## age         0.029487276
## sib_sp      0.227095533
## parch       0.171177814
```

```
## fare      -0.002226149
## pclass_X2  0.780454996
## pclass_X3  1.526222537
## sex_male   2.118215683
```

Question 7

Repeat Question 5, but this time specify a quadratic discriminant analysis model for classification using the `"MASS"` engine.

```
qda_mod <- discrim_quad() %>%
  set_mode("classification") %>%
  set_engine("MASS")

qda_wkflow <- workflow() %>%
  add_model(qda_mod) %>%
  add_recipe(data_recipe)

qda_fit <- fit(qda_wkflow, titanic_train)
```

Question 8

Repeat Question 5, but this time specify a naive Bayes model for classification using the `"klaR"` engine. Set the `usekernel` argument to `FALSE`.

```
nb_mod <- naive_Bayes() %>%
  set_mode("classification") %>%
  set_engine("klaR") %>%
  set_args(usekernel = FALSE)

nb_wkflow <- workflow() %>%
  add_model(nb_mod) %>%
  add_recipe(data_recipe)

nb_fit <- fit(nb_wkflow, titanic_train)
```

Question 9

Now you've fit four different models to your training data.

Use `predict()` and `bind_cols()` to generate predictions using each of these 4 models and your **training** data. Then use the *accuracy* metric to assess the performance of each of the four models.

Which model achieved the highest accuracy on the training data?

```
#Method1
log_res<-predict(log_fit, new_data = titanic_train, type = "prob")
log_res <- bind_cols(log_res, titanic_train%>% select(survived))

augment(log_fit, new_data = titanic_train) %>%
  conf_mat(truth = survived, estimate = .pred_class)
```

```
##           Truth
## Prediction Yes  No
##        Yes 194  52
##        No   79 387
```

```
log_reg_acc <- augment(log_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
log_reg_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.816
```

```
#Method2
lda_res<-predict(lda_fit, new_data = titanic_train, type = "prob")
lda_res <- bind_cols(lda_res, titanic_train%>% select(survived))
augment(lda_fit, new_data = titanic_train) %>%
  conf_mat(truth = survived, estimate = .pred_class)
```

```
##           Truth
## Prediction Yes  No
##        Yes 192  56
##        No   81 383
```

```
lda_acc <- augment(lda_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
lda_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.808
```

```
#Method3
qda_res<-predict(qda_fit, new_data = titanic_train, type = "prob")
qda_res <- bind_cols(qda_res, titanic_train%>% select(survived))
augment(qda_fit, new_data = titanic_train) %>%
  conf_mat(truth = survived, estimate = .pred_class)
```

```
##           Truth
## Prediction Yes  No
##        Yes 200  57
##        No   73 382
```

```
qda_acc <- augment(qda_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
qda_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.817
```

```r
#Method4
nb_res<-predict(nb_fit, new_data = titanic_train, type = "prob")
nb_res <- bind_cols(nb_res, titanic_train%>% select(survived))
augment(nb_fit, new_data = titanic_train) %>%
  conf_mat(truth = survived, estimate = .pred_class)
```

```
##           Truth
## Prediction Yes  No
##        Yes 188  61
##        No   85 378
```

```r
nb_acc <- augment(nb_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)
nb_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.795
```

```r
accuracies <- c(log_reg_acc$.estimate, lda_acc$.estimate,
                nb_acc$.estimate, qda_acc$.estimate)
models <- c("Logistic Regression", "LDA", "Naive Bayes", "QDA")
results <- tibble(accuracies = accuracies, models = models)
results %>%
  arrange(-accuracies)
```

```
## # A tibble: 4 x 2
##   accuracies models
##        <dbl> <chr>
## 1      0.817 QDA
## 2      0.816 Logistic Regression
## 3      0.808 LDA
## 4      0.795 Naive Bayes
```

Question 10

Fit the model with the highest training accuracy to the **testing** data. Report the accuracy of the model on the **testing** data.

Again using the **testing** data, create a confusion matrix and visualize it. Plot an ROC curve and calculate the area under it (AUC).

How did the model perform? Compare its training and testing accuracies. If the values differ, why do you think this is so?

```r
predict(qda_fit, new_data = titanic_test, type = "prob")
```

```
## # A tibble: 179 x 2
##        .pred_Yes .pred_No
##            <dbl>    <dbl>
```

```
##  1 0.0536         0.946
##  2 0.0338         0.966
##  3 0.990          0.00985
##  4 0.0822         0.918
##  5 0.00000000149  1.00
##  6 0.0997         0.900
##  7 0.0776         0.922
##  8 0.251          0.749
##  9 0.995          0.00493
## 10 0.758          0.242
## # ... with 169 more rows
```

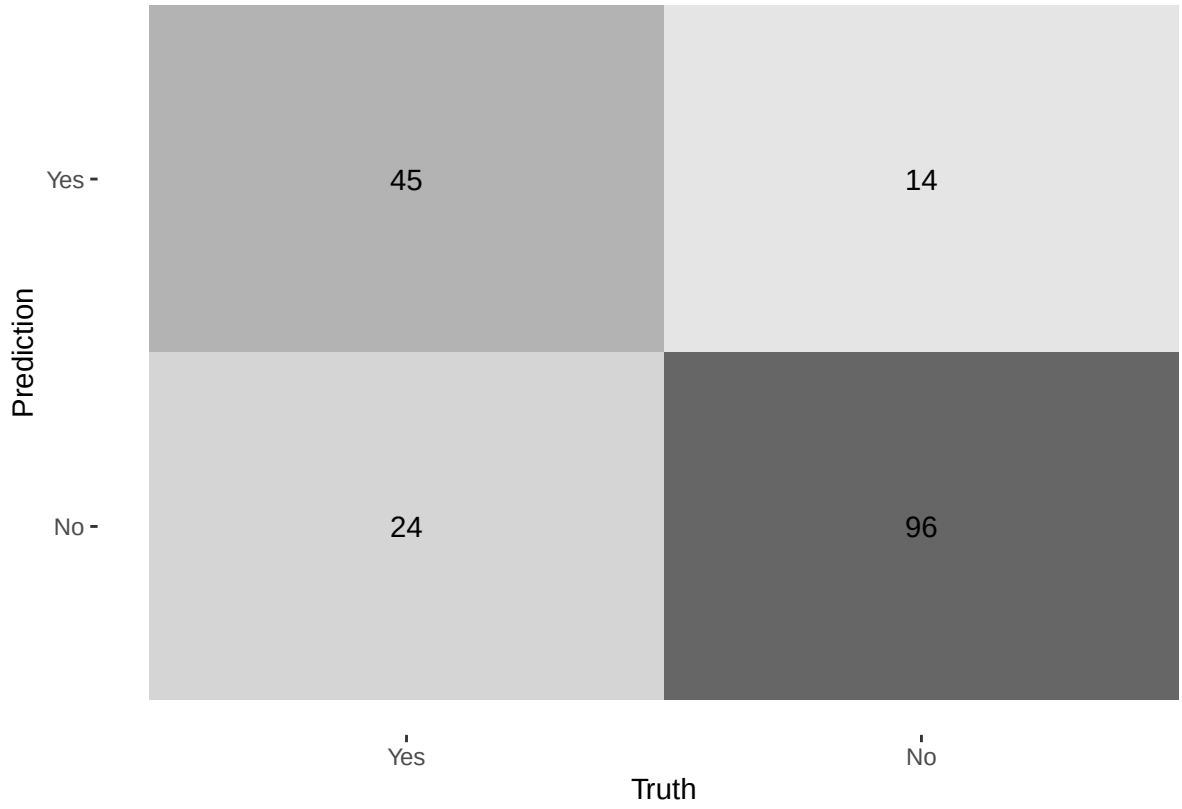
```
augment(qda_fit, new_data = titanic_test) %>%
  conf_mat(truth = survived, estimate = .pred_class)
```

```
##           Truth
## Prediction Yes No
##        Yes  45 14
##        No   24 96
```

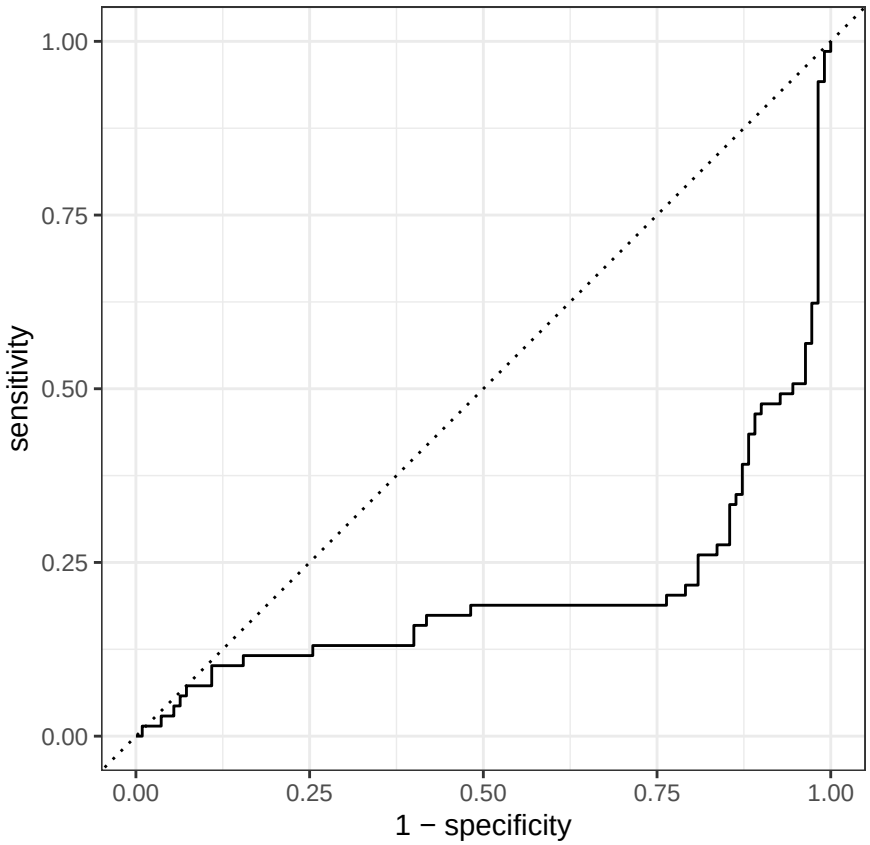
```
multi_metric <- metric_set(accuracy, sensitivity, specificity)

augment(qda_fit, new_data = titanic_test) %>%
  multi_metric(truth = survived, estimate = .pred_class)
```

```
## # A tibble: 3 x 3
##   .metric     .estimator .estimate
##   <chr>       <chr>          <dbl>
## 1 accuracy    binary         0.788
## 2 sensitivity binary         0.652
## 3 specificity binary         0.873
```

```
augment(qda_fit, new_data = titanic_test) %>%
  conf_mat(truth = survived, estimate = .pred_class) %>%
  autoplot(type = "heatmap")
```




```
augment(qda_fit, new_data = titanic_test) %>%
  roc_curve(survived, .pred_No) %>%
  autoplot()
```




```
qda_train_acc <- augment(qda_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate = .pred_class)

qda_test_acc <- augment(qda_fit, new_data = titanic_test) %>%
  accuracy(truth = survived, estimate = .pred_class)

accuracies <- c(qda_train_acc$.estimate, qda_test_acc$.estimate )
models <- c("Train Data Accuracy", "Test Data Accuracy")
results <- tibble(accuracies = accuracies, models = models)
results %>%
  arrange(-accuracies)
```

```
## # A tibble: 2 x 2
##   accuracies models
##        <dbl> <chr>
## 1      0.817 Train Data Accuracy
## 2      0.788 Test Data Accuracy
```

QDA model perfrom really good. Test data accuracy(79%) is very close to the training data accuracy(81%).