Titanic EDA and Modeling Survival

Matthew Houser

7/26/2020

Setup: Convert character features to factors, add column that says "Survived" or "Died" rather than 1 and 0 for graphical reasons. Rename columns for simplicity.

```
library(tidyverse)
library(discrim)
library(readr)
library(tidymodels)
titanic <- read.csv("https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv") %>%
  mutate_if(is.character, factor) %>%
  mutate(Survived_text = case_when(Survived == 1 ~ "Survived", Survived == 0 ~ "Died"))

names(titanic) <- c("Survived", "Pclass", "Name", "Sex", "Age", "SibSp", "ParCh", "Fare", "Survived_tex
```

Inspect the dataset. We have 8 unique features: Survived, and integer indicating whether they survived or died, what passenger class their ticket was, their name, sex, and age, how many siblings or spouses were on board with them, how many parents or children were on board with them, and their ticket fare.

```
str(titanic)
```

```
## 'data.frame':    887 obs. of  9 variables:
##  $ Survived     : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass       : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name         : Factor w/ 887 levels "Capt. Edward Gifford Crosby",..: 602 823 172 814 733 464 700
##  $ Sex          : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age          : num  22 38 26 35 35 27 54 2 27 14 ...
##  $ SibSp        : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ ParCh        : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare         : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Survived_text: chr  "Died" "Survived" "Survived" "Survived" ...
```

Let's find how many people in the dataset Survived

```
titanic %>%
  count(Survived_text)
```

```
##   Survived_text   n
## 1          Died 545
## 2      Survived 342
```

Let's facet survival by sex

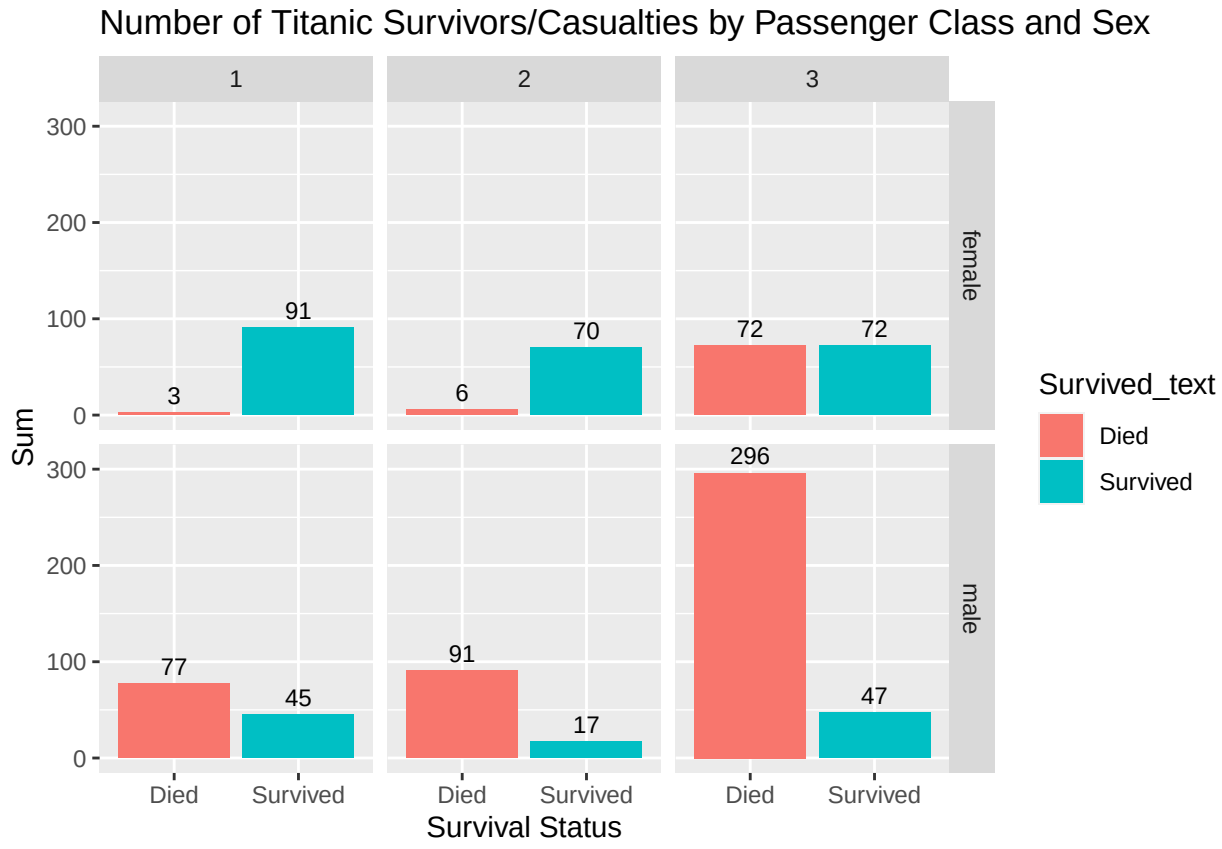
```
titanic %>%
  group_by(Sex) %>%
  count(Survived_text)
```

```
## # A tibble: 4 x 3
```

```
## # Groups:   Sex [2]
##   Sex    Survived_text     n
##   <fct>  <chr>         <int>
## 1 female Died             81
## 2 female Survived        233
## 3 male   Died            464
## 4 male   Survived        109
```

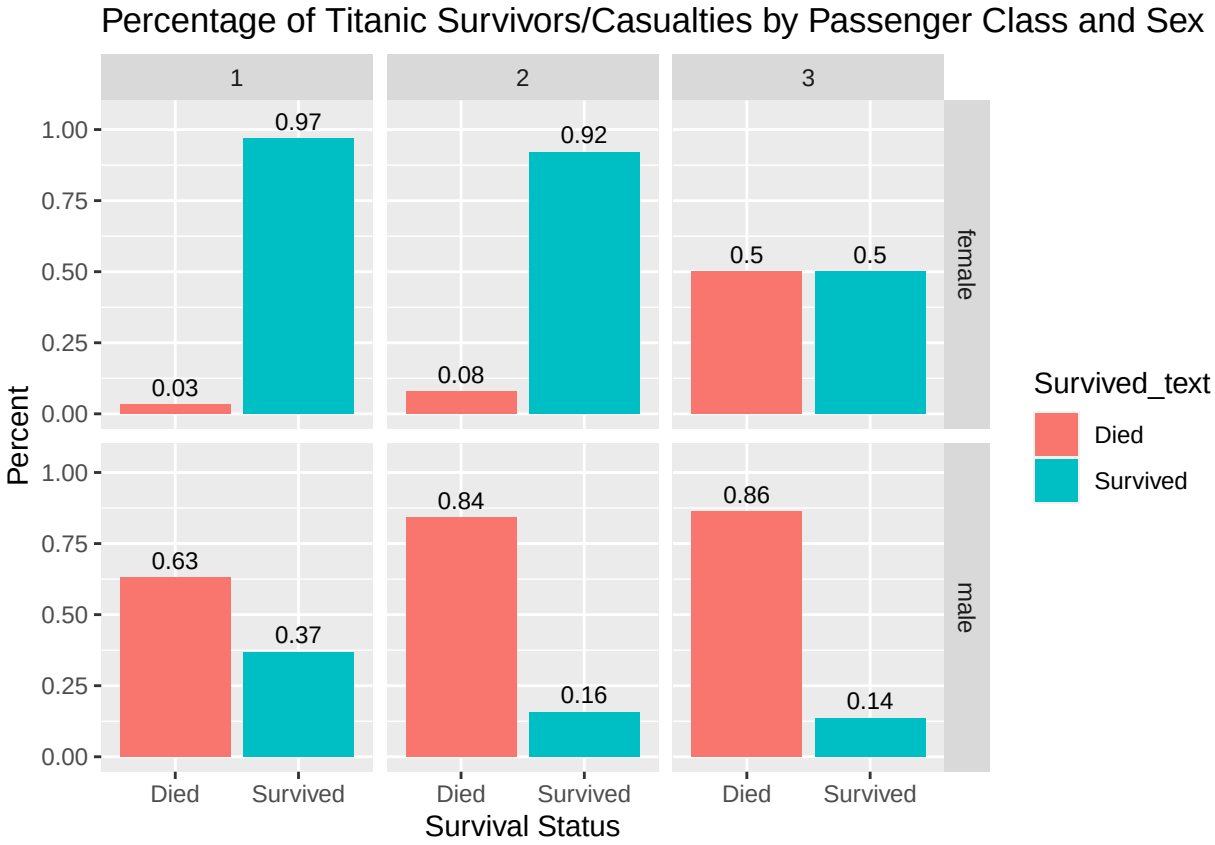
Exploratory Data Analysis:

```
titanic %>%
  group_by(Sex, Pclass, Survived_text) %>%
  tally() %>%
  ggplot(aes(Survived_text, n, fill = Survived_text)) +
  geom_col() +
  facet_grid(Sex ~ Pclass) +
  geom_text(aes(label = n, vjust = -0.5, hjust = .5), size = 3) +
  scale_y_continuous(limits = c(0, 310)) +
  xlab("Survival Status") +
  ggtitle("Number of Titanic Survivors/Casualties by Passenger Class and Sex")
```

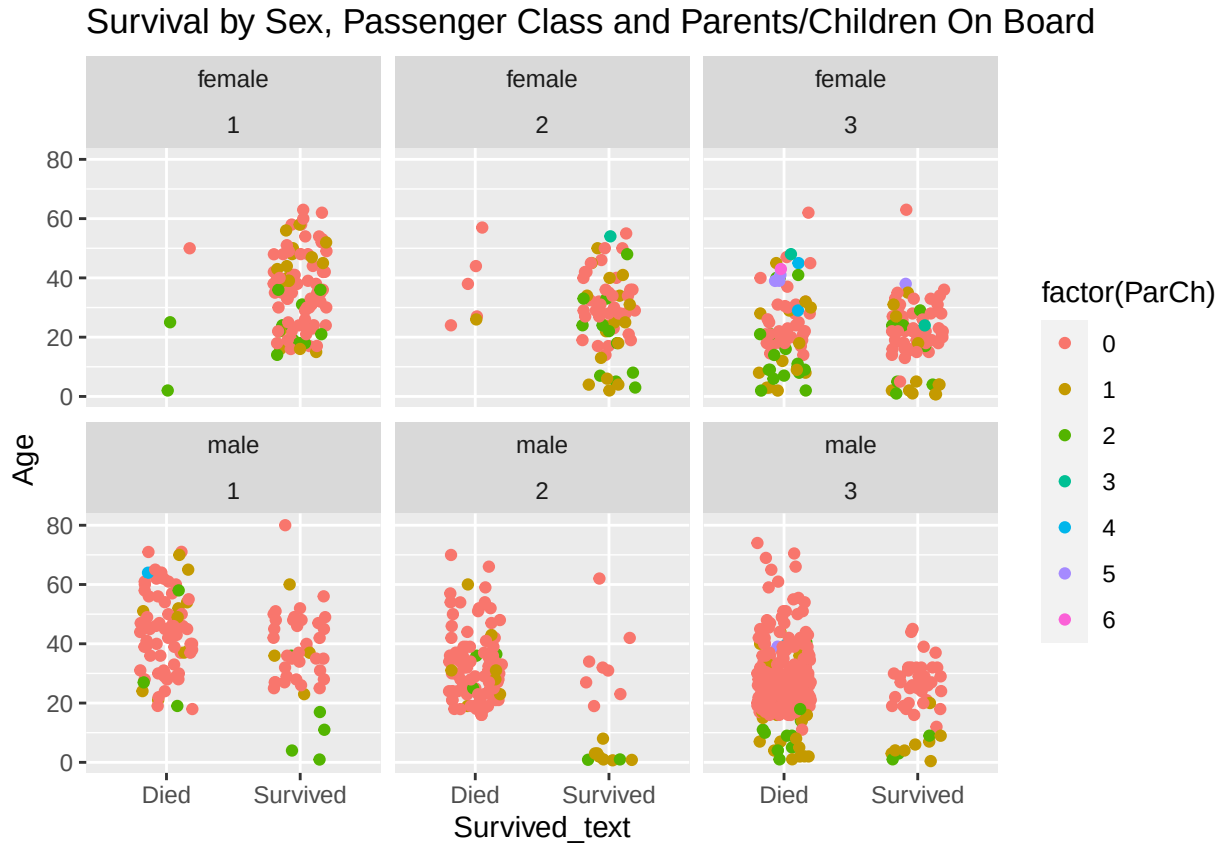



```
titanic %>%
  group_by(Pclass, Sex, Survived_text) %>%
  tally() %>%
  mutate(percent = n/sum(n)) %>%
  ggplot(aes(Survived_text, percent, fill = Survived_text)) +
```

```
  geom_col() +
  facet_grid(Sex ~ Pclass) +
  geom_text(aes(label = round(percent, 2), vjust = -0.5, hjust = .5), size = 3) +
  scale_y_continuous(limits = c(0, 1.05)) +
  xlab("Survival Status") +
  ylab("Percent")+
  ggtitle("Percentage of Titanic Survivors/Casualties by Passenger Class and Sex")
```




```
titanic %>%
  ggplot(aes(Survived_text, Age)) +
  geom_jitter(aes(color = factor(ParCh)), width = .2) +
  facet_wrap(Sex ~ Pclass) +
  ggtitle("Survival by Sex, Passenger Class and Parents/Children On Board")
```




```
titanic %>%
  ggplot(aes(Survived_text, Age)) +
  geom_jitter(aes(color = factor(SibSp)), width = .25) +
  facet_wrap(Sex ~ Pclass) +
  ggtitle("Survival by Sex, Passenger Class and Siblings/Spouses On Board")
```


```
#Convert Survived column to a factor in order for classification to work
titanic$Survived <- as.factor(titanic$Survived)

#Set seed for reproducibility, split data into training and testing sets
set.seed(1)
titanic_split <- titanic %>%
  initial_split(prop = .8, strata = Survived)

train <- training(titanic_split)
test <- testing(titanic_split)
```

```
#Create recipe, model specifications and workflow:
titanic_recipe <- recipe(Survived ~ Sex + Age + SibSp + Pclass + ParCh + Fare, data = train) %>%
  step_downsample(Survived)

glm_spec <- logistic_reg() %>%
  set_engine("glm")

rf_spec <- rand_forest() %>%
  set_engine("ranger") %>%
  set_mode("classification")

lda_spec <- discrim_linear() %>%
  set_engine("MASS") %>%
  translate()
```

```r
qda_spec <- discrim_regularized(frac_common_cov = 0, frac_identity = 0) %>%
  set_engine("klaR") %>%
  translate()

titanic_wf <- workflow() %>%
  add_recipe(titanic_recipe)
```

```r
#Fit the logistic model
titanic_glm <- titanic_wf %>%
  add_model(glm_spec) %>%
  fit(data = train)

titanic_glm
```

```
## == Workflow [trained] ==========================================================================
## Preprocessor: Recipe
## Model: logistic_reg()
##
## -- Preprocessor ----------------------------------------------------------------------------------
## 1 Recipe Step
##
## * step_downsample()
##
## -- Model -----------------------------------------------------------------------------------------
##
## Call:  stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)
##
## Coefficients:
## (Intercept)      Sexmale          Age        SibSp       Pclass        ParCh
##    5.711010    -2.778730    -0.036951    -0.520437    -1.216848    -0.232116
##        Fare
##    0.004249
##
## Degrees of Freedom: 547 Total (i.e. Null);  541 Residual
## Null Deviance:       759.7
## Residual Deviance: 499.4     AIC: 513.4
```

```r
#Fit the LDA model
titanic_lda <- titanic_wf %>%
  add_model(lda_spec) %>%
  fit(data= train)

titanic_lda
```

```
## == Workflow [trained] ==========================================================================
## Preprocessor: Recipe
## Model: discrim_linear()
##
## -- Preprocessor ----------------------------------------------------------------------------------
## 1 Recipe Step
##
## * step_downsample()
##
## -- Model -----------------------------------------------------------------------------------------
## Call:
```

```
## lda(..y ~ ., data = data)
##
## Prior probabilities of groups:
##   0   1
## 0.5 0.5
##
## Group means:
##     Sexmale      Age     SibSp   Pclass     ParCh     Fare
## 0 0.8430657 29.46533 0.5510949 2.558394 0.3102190 21.09373
## 1 0.3211679 28.82573 0.4416058 1.937956 0.4379562 47.89468
##
## Coefficients of linear discriminants:
##                   LD1
## Sexmale -2.0279621175
## Age     -0.0220301547
## SibSp   -0.2633583132
## Pclass  -0.8457512755
## ParCh   -0.1086210387
## Fare     0.0004654823
```

```
#Fit the QDA
titanic_qda <- titanic_wf %>%
  add_model(qda_spec) %>%
  fit(data = train)

titanic_qda
```

```
## == Workflow [trained] ==========================================================================
## Preprocessor: Recipe
## Model: discrim_regularized()
##
## -- Preprocessor ----------------------------------------------------------------------------
## 1 Recipe Step
##
## * step_downsample()
##
## -- Model ----------------------------------------------------------------------------
## Call:
## rda(formula = ..y ~ ., data = data, lambda = ~0, gamma = ~0)
##
## Regularization parameters:
##  gamma lambda
##      0      0
##
## Prior probabilities of groups:
##   0   1
## 0.5 0.5
##
## Misclassification rate:
##        apparent: 19.343 %
```

```
#Fit the random forest
titanic_rf <- titanic_wf %>%
  add_model(rf_spec) %>%
  fit(data = train)
```

```
titanic_rf
```

```
## == Workflow [trained] ==========================================================================
## Preprocessor: Recipe
## Model: rand_forest()
##
## -- Preprocessor ----------------------------------------------------------------------------
## 1 Recipe Step
##
## * step_downsample()
##
## -- Model -----------------------------------------------------------------------------------
## Ranger result
##
## Call:
##  ranger::ranger(formula = ..y ~ ., data = data, num.threads = 1,      verbose = FALSE, seed = sample
##
## Type:                             Probability estimation
## Number of trees:                  500
## Sample size:                      548
## Number of independent variables:  6
## Mtry:                             2
## Target node size:                 10
## Variable importance mode:         none
## Splitrule:                        gini
## OOB prediction error (Brier s.):  0.1314029
```

```
#Use the training model on the test set and then show confusion matrix
results <- test %>%
  bind_cols(predict(titanic_glm, test) %>%
              rename(.pred_glm = .pred_class)) %>%
  bind_cols(predict(titanic_lda, test) %>%
              rename(.pred_lda = .pred_class)) %>%
  bind_cols(predict(titanic_qda, test) %>%
              rename(.pred_qda = .pred_class)) %>%
  bind_cols(predict(titanic_rf, test) %>%
              rename(.pred_rf = .pred_class))

#Confusion matrix for logistic regression
results %>%
  conf_mat(truth = Survived, estimate = .pred_glm)
```

```
##           Truth
## Prediction  0  1
##          0 91 18
##          1 17 50
```

```
#Confusion matrix for LDA
results %>%
  conf_mat(truth = Survived, estimate = .pred_lda)
```

```
##           Truth
## Prediction  0  1
##          0 90 18
##          1 18 50
```

```r
#Confusion matrix for QDA
results %>%
  conf_mat(truth = Survived, estimate = .pred_qda)
```

```
##           Truth
## Prediction  0  1
##          0 92 18
##          1 16 50
```

```r
#Confusion matrix for Random Forest
results %>%
  conf_mat(truth = Survived, estimate = .pred_rf)
```

```
##           Truth
## Prediction  0  1
##          0 94 17
##          1 14 51
```

```r
#Find sensitivity for each model
sens_glm <- sensitivity(results, truth = Survived, estimate = .pred_glm)
sens_lda <- sensitivity(results, truth = Survived, estimate = .pred_lda)
sens_qda <- sensitivity(results, truth = Survived, estimate = .pred_qda)
sens_rf <- sensitivity(results, truth = Survived, estimate = .pred_rf)

c(sens_glm$.estimate, sens_lda$.estimate, sens_qda$.estimate, sens_rf$.estimate)
```

```
## [1] 0.8425926 0.8333333 0.8518519 0.8703704
```

```r
#Find specificity for each model
spec_glm <- specificity(results, truth = Survived, estimate = .pred_glm)
spec_lda <- specificity(results, truth = Survived, estimate = .pred_lda)
spec_qda <- specificity(results, truth = Survived, estimate = .pred_qda)
spec_rf <- specificity(results, truth = Survived, estimate = .pred_rf)

c(spec_glm$.estimate, spec_lda$.estimate, spec_qda$.estimate, spec_rf$.estimate)
```

```
## [1] 0.7352941 0.7352941 0.7352941 0.7500000
```

```r
#Find accuracy for each model
acc_glm <- accuracy(results, truth = Survived, estimate = .pred_glm)
acc_lda <- accuracy(results, truth = Survived, estimate = .pred_lda)
acc_qda <- accuracy(results, truth = Survived, estimate = .pred_qda)
acc_rf <- accuracy(results, truth = Survived, estimate = .pred_rf)

c(acc_glm$.estimate, acc_lda$.estimate, acc_qda$.estimate, acc_rf$.estimate)
```

```
## [1] 0.8011364 0.7954545 0.8068182 0.8238636
```