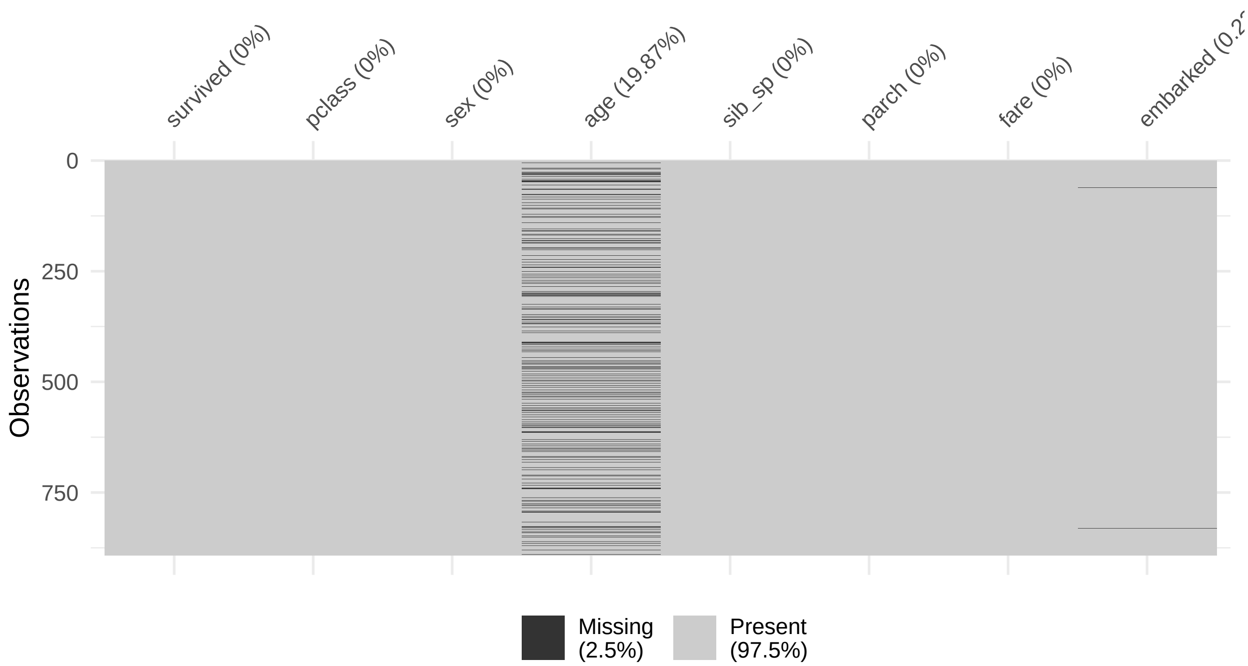
Predicting Survival on the Titanic

David Nemirovsky & Jared Klug

5/13/21

EDA

```
titanic_df =
  read_csv("./data/train.csv") %>%
  janitor::clean_names() %>%
  mutate(survived = fct_recode(as.factor(survived), yes = "1", no = "0"),
         survived = fct_relevel(survived, "yes", "no"),
         pclass = as.factor(pclass),
         sex = as.factor(sex),
         embarked = as.factor(embarked)) %>%
  select(-c(ticket, cabin, name, passenger_id))

# Missing Data EDA:
vis_miss(titanic_df)
```



```r
# Missing Survivals:
titanic_df %>%
  filter(is.na(age)) %>%
  group_by(survived) %>%
  summarise(count = n())
```

```
## # A tibble: 2 x 2
##   survived count
##   <fct>    <int>
## 1 yes         52
## 2 no         125
```

```r
# All Survivals:
titanic_df %>%
  group_by(survived) %>%
  summarise(count = n())
```

```
## # A tibble: 2 x 2
##   survived count
##   <fct>    <int>
## 1 yes        342
## 2 no         549
```

```r
# Survival percentage in NAs is similar to actual survival rate so assume missingness is not related to

# Bagging imputation will be used to fill in missing age data, embarked will be replaced by the most po

titanic_df %>%
  group_by(embarked) %>%
  summarise(count = n())
```

```
## # A tibble: 4 x 2
##   embarked count
##   <fct>    <int>
## 1 C          168
## 2 Q           77
## 3 S          644
## 4 <NA>         2
```

```r
titanic_df[is.na(titanic_df$embarked),"embarked"] = "S"

trainX = titanic_df[-1]
train_bag = preProcess(trainX, method = "bagImpute")
train_imp = predict(train_bag, trainX)
train_df = cbind(titanic_df[1], train_imp)

# Descriptive Summary Table:
eda_df =
  train_df %>%
  mutate(survived = fct_recode(as.factor(survived),
                               Survived = "yes", Died = "no"),
         survived = fct_relevel(survived, "Died", "Survived"),
```

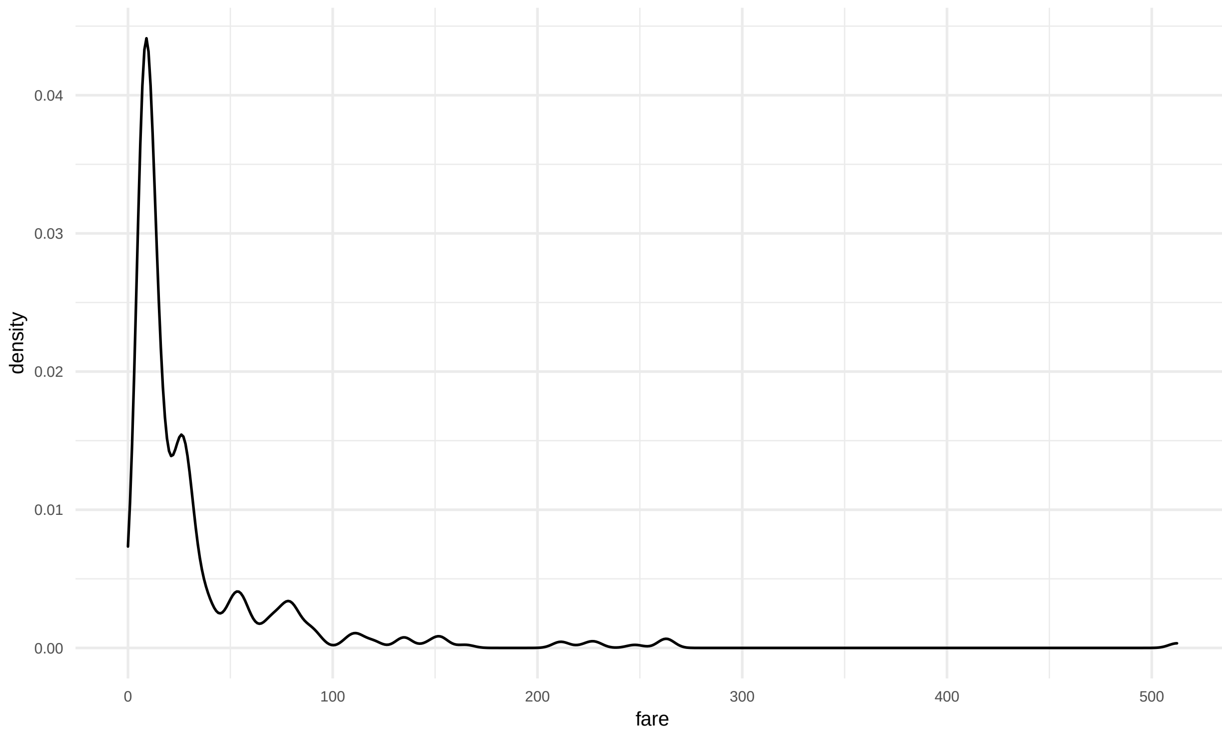
```r
         sex = fct_recode(as.factor(sex),
                          Female = "female", Male = "male"),
         embarked = fct_recode(as.factor(embarked),
                               Cherbourg = "C",
                               Queenstown = "Q",
                               Southampton = "S"),
         pclass = fct_recode(as.factor(pclass),
                             Upper = "1",
                             Middle = "2",
                             Lower = "3"))

eda_df %>%
  tbl_summary(by = survived,
              label =
                list(
                  pclass ~ "Socioeconomic Status",
                  sex ~ "Sex",
                  age ~ "Age",
                  sib_sp ~ "Number of Siblings/Spouse on Board",
                  parch ~ "Number of Parents/Children on Board",
                  fare ~ "Passenger Fare",
                  embarked ~ "Port of Embarkation")) %>%
  add_overall %>%
  as_gt() %>%
  tab_options(table.width = pct(30), table.font.size = "small")
```

Characteristic	**Overall**, N = 891[1]	**Died**, N = 549[1]	**Survived**, N = 342[1]
Socioeconomic Status			
Upper	216 (24%)	80 (15%)	136 (40%)
Middle	184 (21%)	97 (18%)	87 (25%)
Lower	491 (55%)	372 (68%)	119 (35%)
Sex			
Female	314 (35%)	81 (15%)	233 (68%)
Male	577 (65%)	468 (85%)	109 (32%)
Age	28 (22, 36)	29 (22, 36)	28 (20, 36)
Number of Siblings/Spouse on Board			
0	608 (68%)	398 (72%)	210 (61%)
1	209 (23%)	97 (18%)	112 (33%)
2	28 (3.1%)	15 (2.7%)	13 (3.8%)
3	16 (1.8%)	12 (2.2%)	4 (1.2%)
4	18 (2.0%)	15 (2.7%)	3 (0.9%)
5	5 (0.6%)	5 (0.9%)	0 (0%)
8	7 (0.8%)	7 (1.3%)	0 (0%)
Number of Parents/Children on Board			
0	678 (76%)	445 (81%)	233 (68%)
1	118 (13%)	53 (9.7%)	65 (19%)
2	80 (9.0%)	40 (7.3%)	40 (12%)
3	5 (0.6%)	2 (0.4%)	3 (0.9%)
4	4 (0.4%)	4 (0.7%)	0 (0%)
5	5 (0.6%)	4 (0.7%)	1 (0.3%)
6	1 (0.1%)	1 (0.2%)	0 (0%)
Passenger Fare	14 (8, 31)	10 (8, 26)	26 (12, 57)
Port of Embarkation			

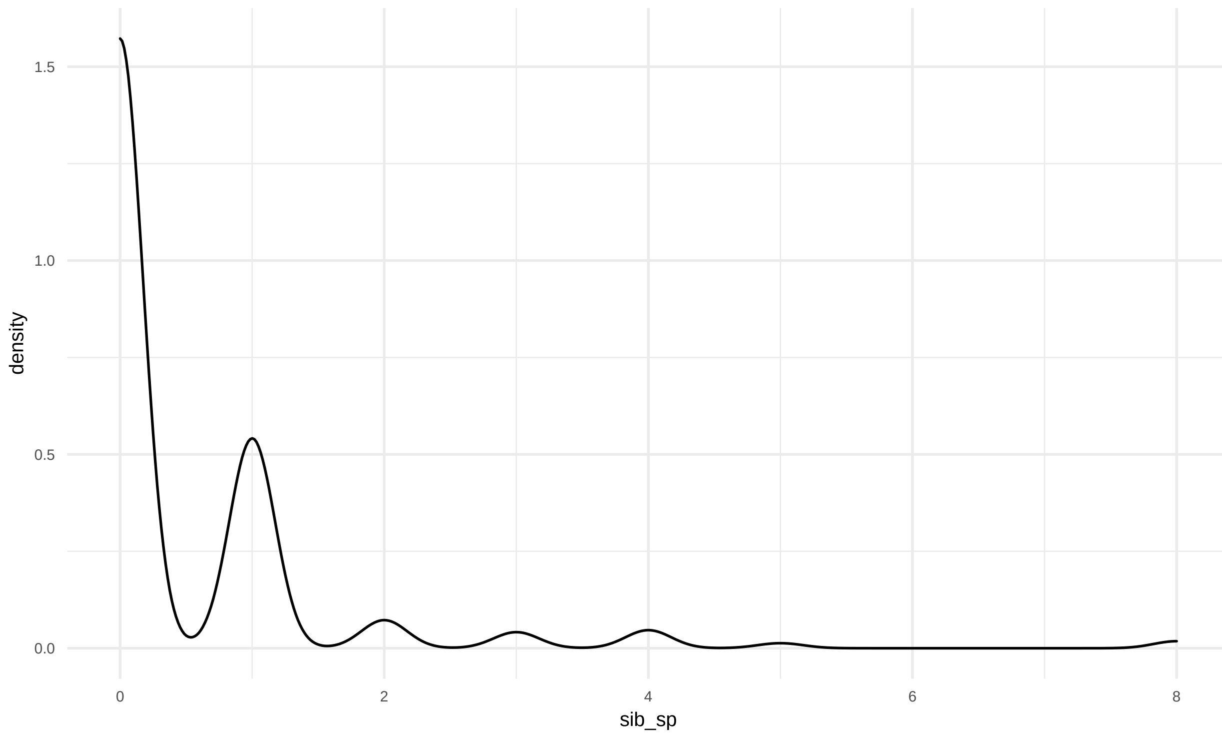
Cherbourg	168 (19%)	75 (14%)	93 (27%)
Queenstown	77 (8.6%)	47 (8.6%)	30 (8.8%)
Southampton	646 (73%)	427 (78%)	219 (64%)

[1]Statistics presented: n (%); Median (IQR)

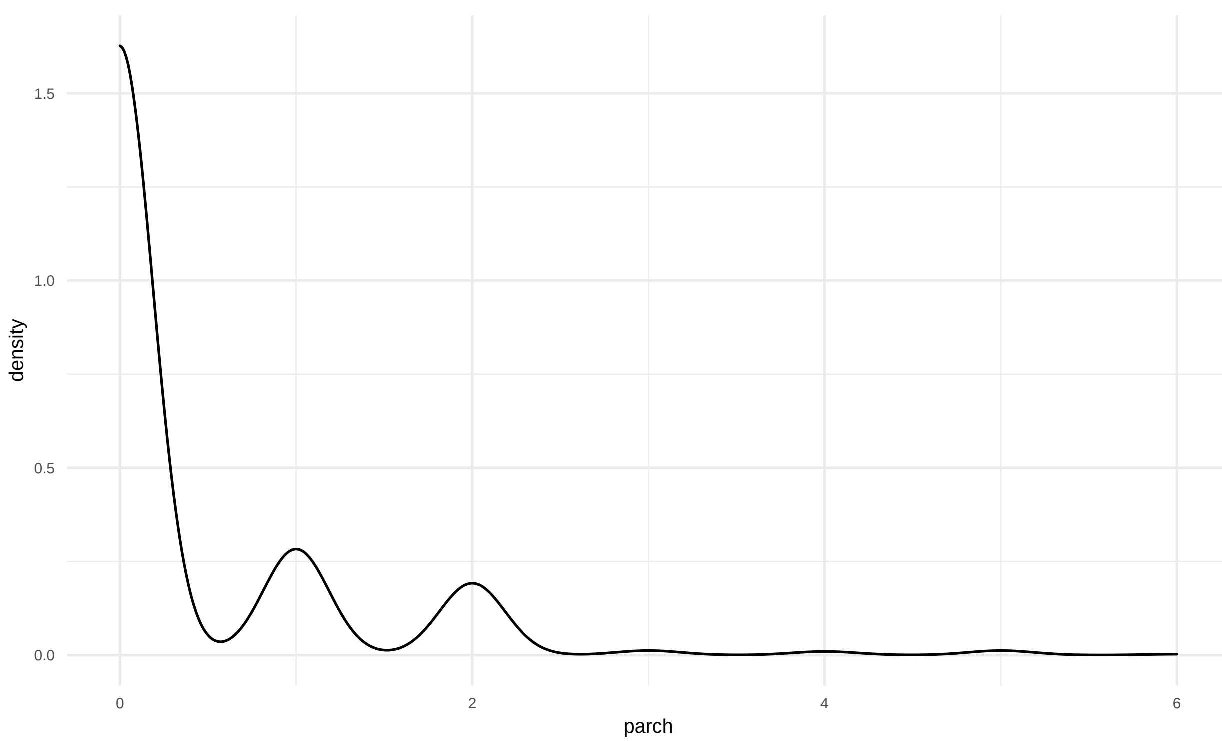
```r
# Examine skew and outliers of some predictors:
eda_df %>%
  ggplot(aes(x = fare)) +
  geom_density()
```




```r
eda_df %>%
  ggplot(aes(x = sib_sp)) +
  geom_density()
```

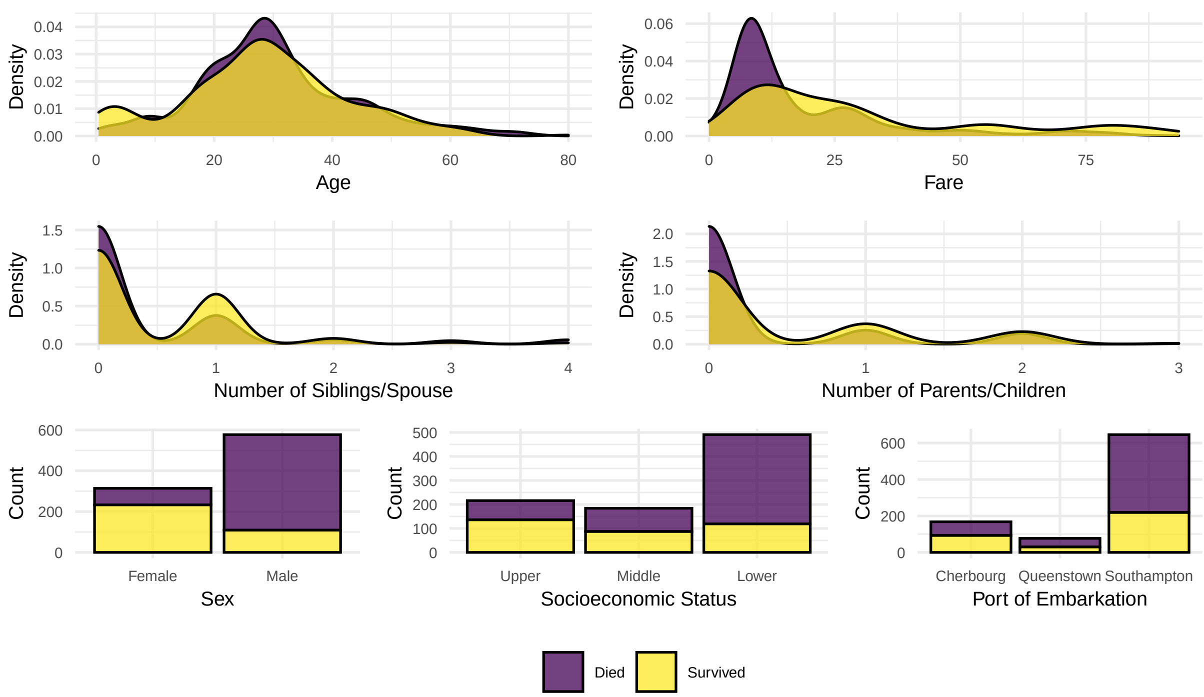



```r
eda_df %>%
  ggplot(aes(x = parch)) +
  geom_density()
```



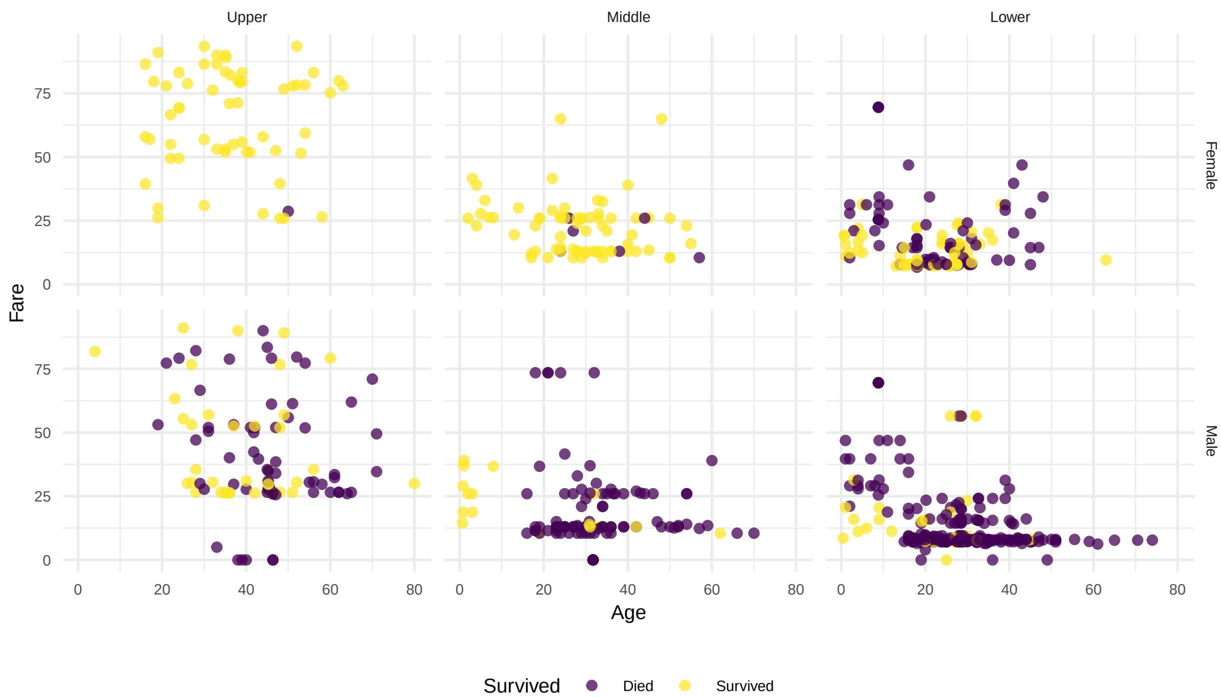
```r
# Survival by Age:
plot_age =
  eda_df %>%
  ggplot(aes(x = age, fill = survived)) +
  geom_density(alpha = 0.75) +
  labs(x = "Age", y = "Density") +
  scale_fill_discrete(name = "Survived") +
  theme(legend.title = element_blank())

# Survival by Fare:
plot_fare =
  eda_df %>%
  filter(fare < 100) %>%
  ggplot(aes(x = fare, fill = survived)) +
  geom_density(alpha = 0.75) +
  labs(x = "Fare", y = "Density") +
  scale_fill_discrete(name = "Survived") +
  theme(legend.title = element_blank())

# Survival by Number of Siblings/Spouse:
plot_sibsp =
  eda_df %>%
  filter(sib_sp < 5) %>%
  ggplot(aes(x = sib_sp, fill = survived)) +
  geom_density(alpha = 0.75) +
  labs(x = "Number of Siblings/Spouse", y = "Density") +
  scale_fill_discrete(name = "Survived") +
  theme(legend.title = element_blank())

# Survival by Number of Parents/Children:
plot_parch =
  eda_df %>%
  filter(parch < 4) %>%
  ggplot(aes(x = parch, fill = survived)) +
  geom_density(alpha = 0.75) +
  labs(x = "Number of Parents/Children", y = "Density") +
  scale_fill_discrete(name = "Survived") +
  theme(legend.title = element_blank())

# Survival by Sex:
plot_sex =
  eda_df %>%
  ggplot(aes(x = sex, fill = survived)) +
  geom_bar(color = "black", alpha = 0.75) +
  labs(x = "Sex", y = "Count") +
  scale_fill_discrete(name = "Survived") +
  theme(legend.title = element_blank())

# Survival by Socioeconomic Status:
plot_pclass =
  eda_df %>%
  ggplot(aes(x = pclass, fill = survived)) +
  geom_bar(color = "black", alpha = 0.75) +
```

```
  labs(x = "Socioeconomic Status", y = "Count") +
  scale_fill_discrete(name = "Survived") +
  theme(legend.title = element_blank())

# Survival by Port of Embarkation:
plot_emb =
  eda_df %>%
  ggplot(aes(x = embarked, fill = survived)) +
  geom_bar(color = "black", alpha = 0.75) +
  labs(x = "Port of Embarkation", y = "Count") +
  scale_fill_discrete(name = "Survived") +
  theme(legend.title = element_blank())

layout = "
AAABBB
CCCDDD
EEFFGG
"
plot_age + plot_fare + plot_sibsp + plot_parch + plot_sex +
  plot_pclass + plot_emb + plot_layout(design = layout, guides = "collect")
```




```
# Survival of Age vs Fare by SES and Sex
eda_df %>%
  filter(fare < 100) %>%
  ggplot(aes(x = age, y = fare, color = survived)) +
  geom_point(alpha = 0.75) +
  facet_grid(sex ~ pclass) +
```

```r
  labs(x = "Age", y = "Fare") +
  scale_color_discrete(name = "Survived")
```




```r
# Survival of Age vs Fare by Embarkation and Sex
eda_df %>%
  filter(fare < 100) %>%
  ggplot(aes(x = age, y = fare, color = survived)) +
  geom_point(alpha = 0.75) +
  facet_grid(sex ~ embarked) +
  labs(x = "Age", y = "Fare") +
  scale_color_discrete(name = "Survived")
```

Model Training

```
ctrl = trainControl(method = "repeatedcv", summaryFunction = twoClassSummary, classProbs = T, number = 

set.seed(37564)
mod_enet = train(survived ~ .,
                 na.action = na.exclude,
                 data = train_df,
                 method = "glmnet",
                 family = "binomial",
                 metric = "ROC",
                 tuneGrid = expand.grid(alpha = seq(0, 0.5, length = 6),
                                        lambda = exp(seq(-4, -8, length = 50))),
                 trControl = ctrl)
tuning_plot_enet =
  ggplot(mod_enet, highlight = T) +
  ggtitle("Elastic Net") +
  theme(plot.title = element_text(hjust = 0.5))
mod_enet$bestTune
```

```
##     alpha      lambda
## 165   0.3 0.001051915
```

```
set.seed(37564)
mod_mars = train(survived ~ .,
                 na.action = na.exclude,
                 data = train_df,
```

```r
                 method = "earth",
                 tuneGrid = expand.grid(degree = 1:3, nprune = 5:15),
                 metric = "ROC",
                 trControl = ctrl)
tuning_plot_mars =
  ggplot(mod_mars, highlight = T) +
  ggtitle("MARS") +
  theme(plot.title = element_text(hjust = 0.5))
mod_mars$bestTune
```

```
##    nprune degree
## 18     11      2
```

```r
set.seed(37564)
mod_knn = train(survived ~ .,
                na.action = na.exclude,
                data = train_df,
                method = "knn",
                metric = "ROC",
                preProcess = c("center","scale"),
                tuneGrid = data.frame(k = seq(1, 30, by = 1)),
                trControl = ctrl)
tuning_plot_knn =
  ggplot(mod_knn, highlight = T) +
  ggtitle("KNN") +
  theme(plot.title = element_text(hjust = 0.5))
mod_knn$bestTune
```

```
##   k
## 8 8
```

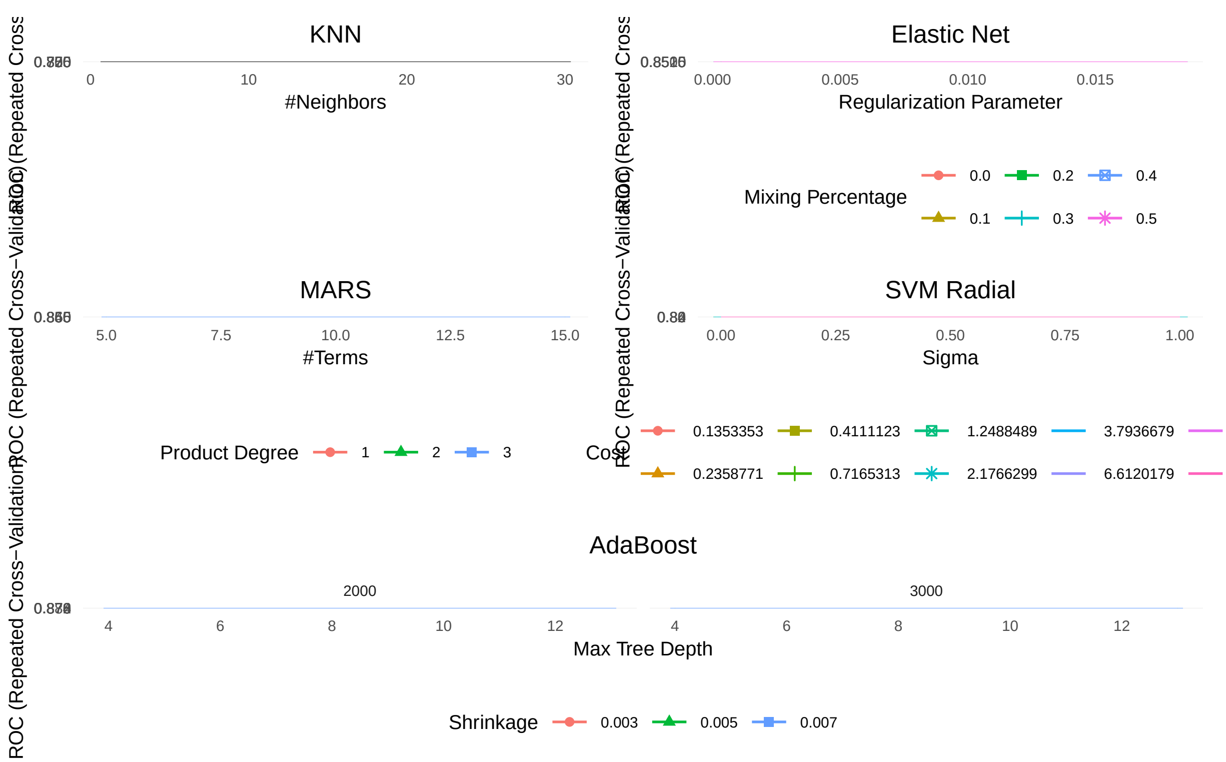
```r
set.seed(37564)
mod_boost = train(survived ~ .,
                  na.action = na.exclude,
                  data = train_df,
                  method = "gbm",
                  distribution = "adaboost",
                  tuneGrid = expand.grid(n.trees = c(2000, 3000),
                                         interaction.depth = 4:13,
                                         shrinkage = c(0.003, 0.005, 0.007),
                                         n.minobsinnode = 1),
                  metric = "ROC",
                  trControl = ctrl,
                  verbose = F)
tuning_plot_boost =
  ggplot(mod_boost, highlight = T) +
  ggtitle("AdaBoost") +
  theme(plot.title = element_text(hjust = 0.5))
mod_boost$bestTune
```

```
##    n.trees interaction.depth shrinkage n.minobsinnode
## 36    3000                11     0.005              1
```

```r
set.seed(37564)
mod_svm = train(survived ~ .,
                na.action = na.exclude,
                data = train_df,
                preProcess = c("scale", "center"),
                method = "svmRadialSigma",
                tuneGrid = expand.grid(C = exp(seq(-2,3, len = 10)),
                                       sigma = exp(seq(-8,0, len = 10))),
                metric = "ROC",
                trControl = ctrl)
tuning_plot_svm =
  ggplot(mod_svm, highlight = T) +
  ggtitle("SVM Radial") +
  theme(plot.title = element_text(hjust = 0.5))
mod_svm$bestTune
```

```
##         sigma        C
## 76 0.0285655 6.612018
```

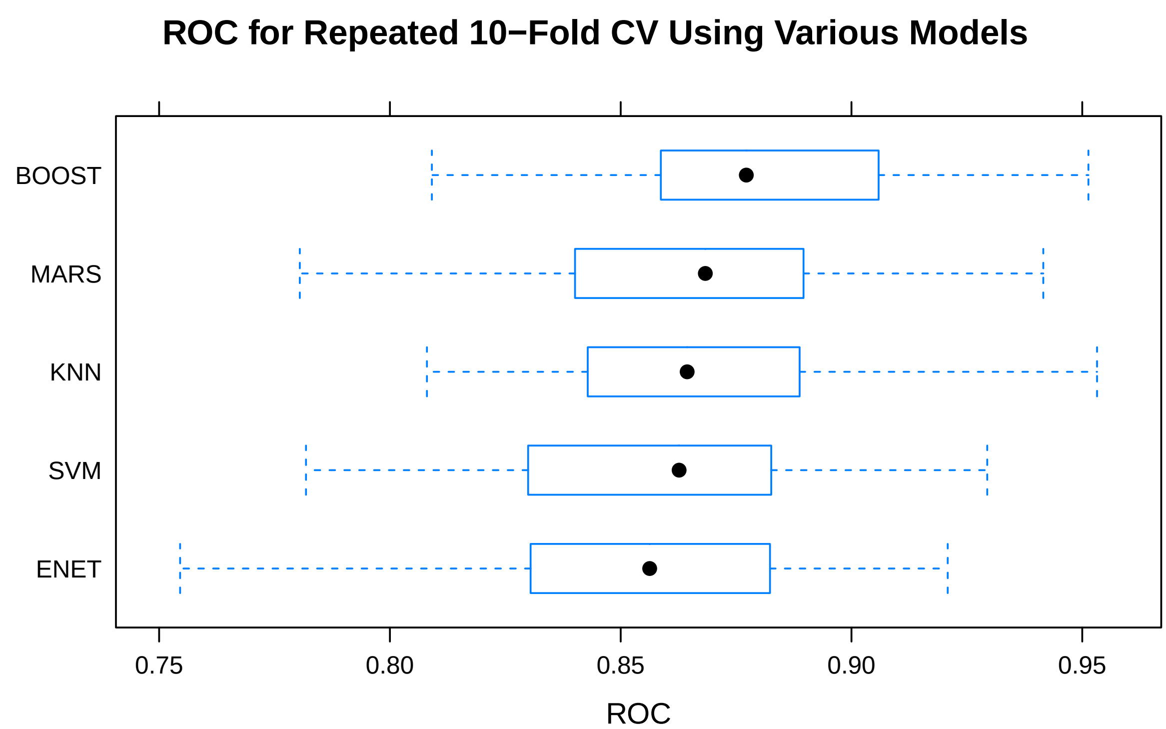
```r
layout2 = "
AABB
CCDD
EEEE
"
tuning_plot_knn + tuning_plot_enet +
  tuning_plot_mars + tuning_plot_svm +
  tuning_plot_boost + plot_layout(design = layout2)
```



```r
res = resamples(list(ENET = mod_enet, MARS = mod_mars, KNN = mod_knn, BOOST = mod_boost, SVM = mod_svm))
summary(res)
```

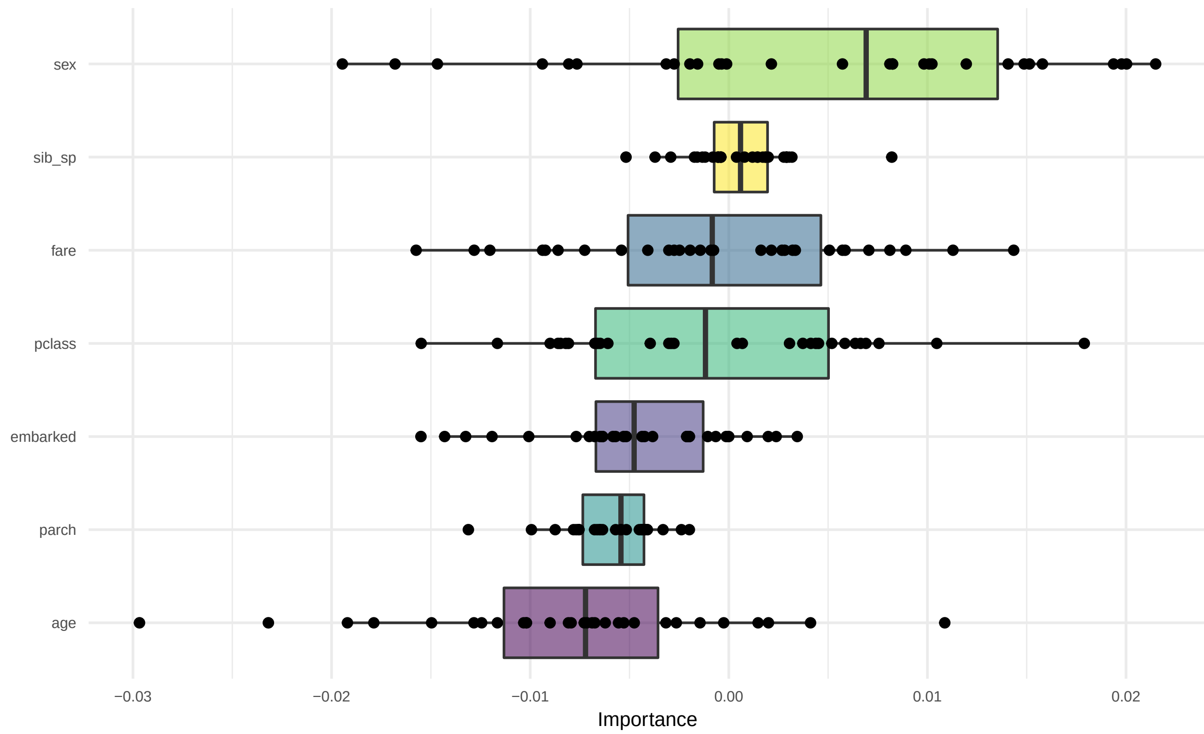
```
##
## Call:
## summary.resamples(object = res)
##
## Models: ENET, MARS, KNN, BOOST, SVM
## Number of resamples: 50
##
## ROC
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## ENET  0.7545455 0.8311497 0.8562834 0.8523355 0.8815699 0.9208556    0
## MARS  0.7804813 0.8402406 0.8683403 0.8645612 0.8894672 0.9415584    0
## KNN   0.8080214 0.8431723 0.8644038 0.8685563 0.8882659 0.9532086    0
## BOOST 0.8090909 0.8595761 0.8772193 0.8804691 0.9042170 0.9513369    0
## SVM   0.7818182 0.8322193 0.8626560 0.8603469 0.8816176 0.9294118    0
##
## Sens
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## ENET  0.5588235 0.6619748 0.7058824 0.7052437 0.7409664 0.8529412    0
## MARS  0.5882353 0.6764706 0.7058824 0.7196975 0.7714286 0.9117647    0
## KNN   0.5428571 0.6470588 0.7058824 0.7083529 0.7647059 0.8823529    0
## BOOST 0.5588235 0.6619748 0.7247899 0.7200000 0.7867647 0.9117647    0
## SVM   0.4705882 0.6176471 0.6764706 0.6625546 0.7058824 0.8529412    0
##
## Spec
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## ENET  0.8000000 0.8363636 0.8727273 0.8648418 0.8909091 0.9454545    0
## MARS  0.6909091 0.8363636 0.8727273 0.8684646 0.8909091 0.9636364    0
## KNN   0.7777778 0.8545455 0.8909091 0.8815623 0.9090909 0.9636364    0
## BOOST 0.8000000 0.8727273 0.9082492 0.9009158 0.9452020 0.9636364    0
## SVM   0.8545455 0.9090909 0.9272727 0.9242290 0.9454545 1.0000000    0
```

```r
bwplot(res, metric = "ROC", main = "ROC for Repeated 10-Fold CV Using Various Models")
```



Variable Importance

```
set.seed(37564)
vip(mod_boost,
    method = "permute",
    train = train_df,
    target = "survived",
    metric = "auc",
    reference_class = c("no", "yes"),
    nsim = 30,
    pred_wrapper = predict,
    geom = "boxplot",
    all_permutations = T,
    mapping = aes_string(fill = "Variable", alpha = 0.75))
```



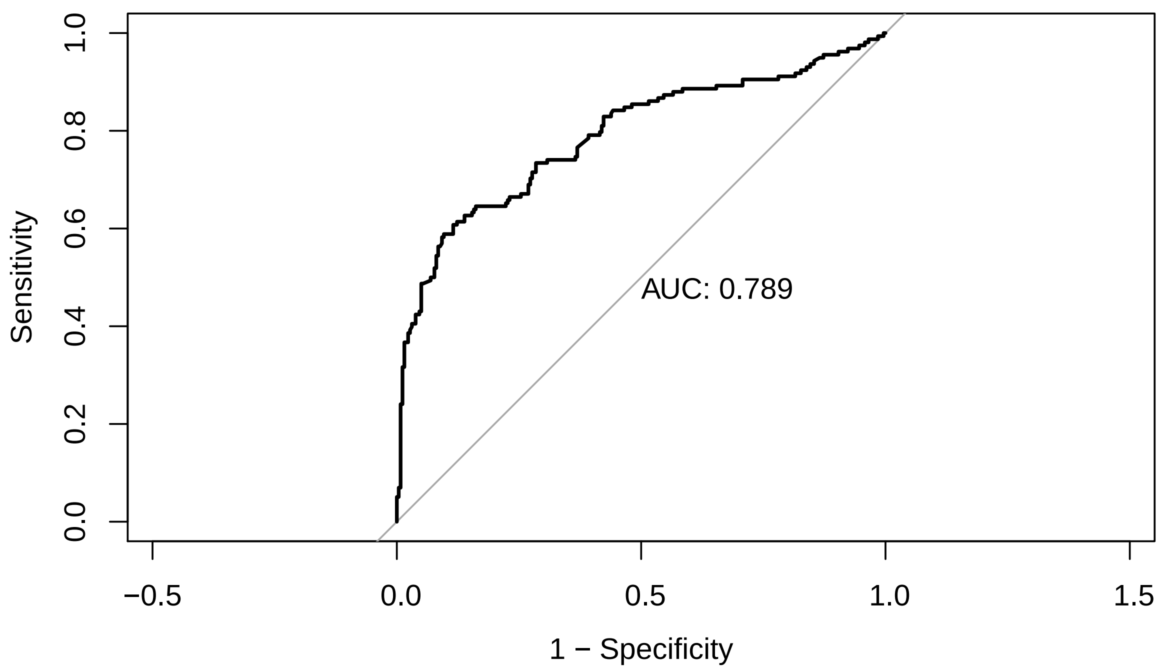

```r
# Check if ENET parameters mocks importance pattern
coef(mod_enet$finalModel, mod_enet$bestTune$lambda)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept) -4.209638353
## pclass2      1.018755246
## pclass3      2.276636897
## sexmale      2.664609147
## age          0.040746141
## sib_sp       0.358907155
## parch        0.096550464
## fare        -0.001996118
## embarkedQ   -0.001246108
## embarkedS    0.406549532
```

Predictions

```r
testna_df =
  read_csv("./data/test.csv") %>%
  janitor::clean_names() %>%
  select(-c(ticket, cabin, name)) %>%
  left_join(janitor::clean_names(read_csv("./data/titanic_results.csv"))) %>%
  mutate(pclass = as.factor(pclass),
         sex = as.factor(sex),
         embarked = as.factor(embarked))
```

```
testX = testna_df[,2:8]
test_bag = preProcess(testX, method = "bagImpute")
test_df = predict(test_bag, testX) %>%
  cbind(testna_df[1], testna_df[9])

pred_boost = predict(mod_boost, newdata = test_df, type = "prob")[,1]
roc_boost = roc(test_df$survived, pred_boost)
plot(roc_boost, legacy.axes = T, print.auc = T)
```



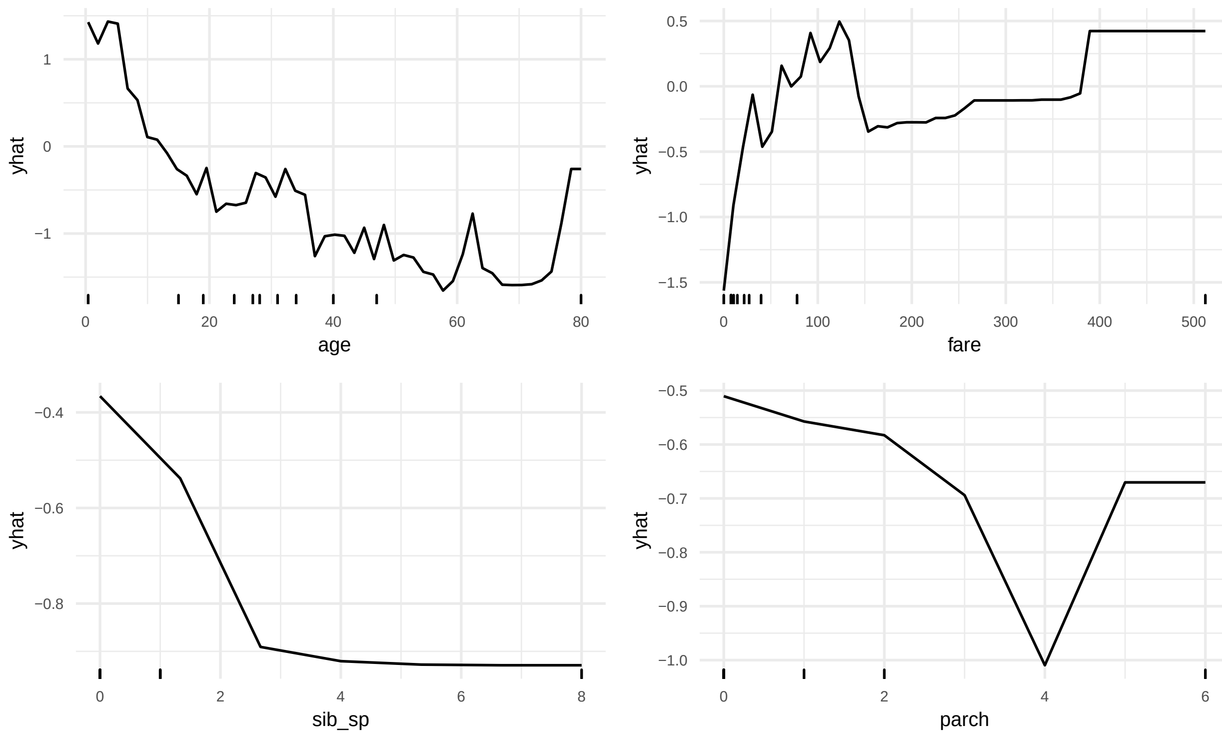
AdaBoost Model Analysis

```
cm_df = pred_boost %>%
  as.data.frame() %>%
  rename("survived" = ".") %>%
  mutate(survived = as.factor(ifelse(survived >= 0.5, 1, 0)))

confusionMatrix(data = cm_df$survived, reference = as.factor(test_df$survived))
```

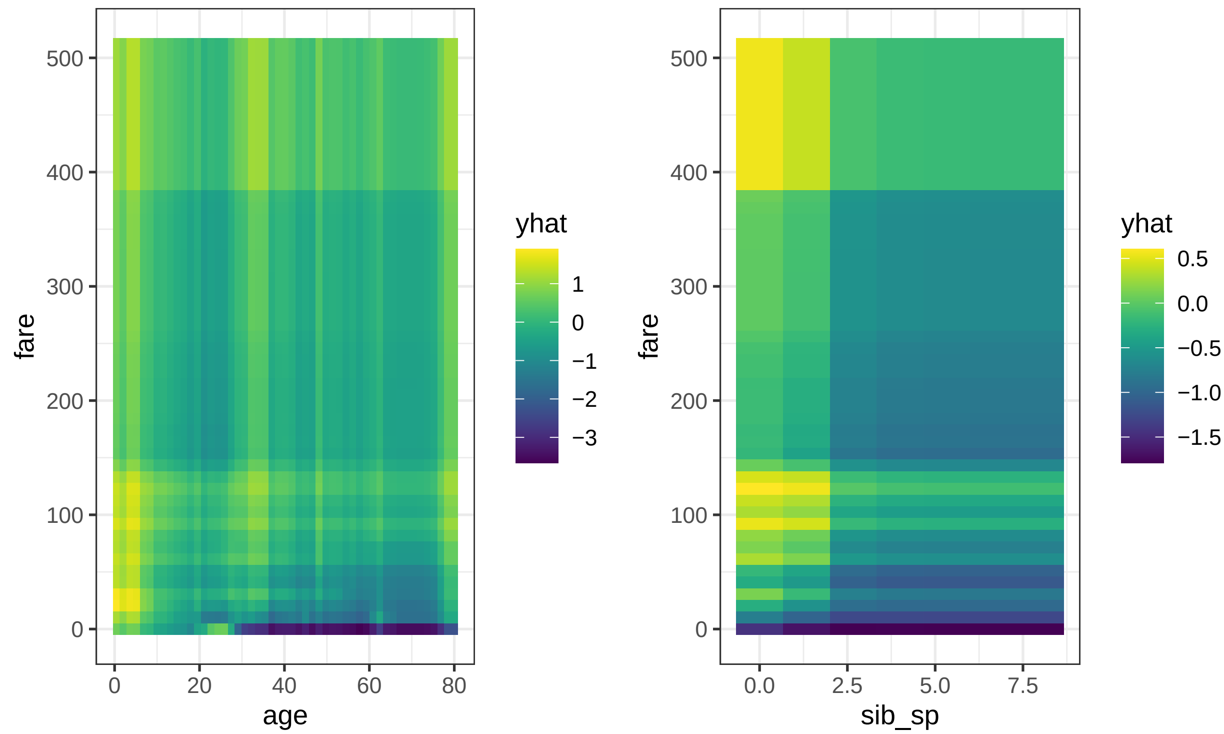
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 221  59
##          1  39  99
##
```

```
##                Accuracy : 0.7656
##                  95% CI : (0.7219, 0.8054)
##     No Information Rate : 0.622
##     P-Value [Acc > NIR] : 2.705e-10
##
##                   Kappa : 0.4887
##
##  Mcnemar's Test P-Value : 0.05495
##
##             Sensitivity : 0.8500
##             Specificity : 0.6266
##          Pos Pred Value : 0.7893
##          Neg Pred Value : 0.7174
##              Prevalence : 0.6220
##          Detection Rate : 0.5287
##    Detection Prevalence : 0.6699
##       Balanced Accuracy : 0.7383
##
##        'Positive' Class : 0
##
```

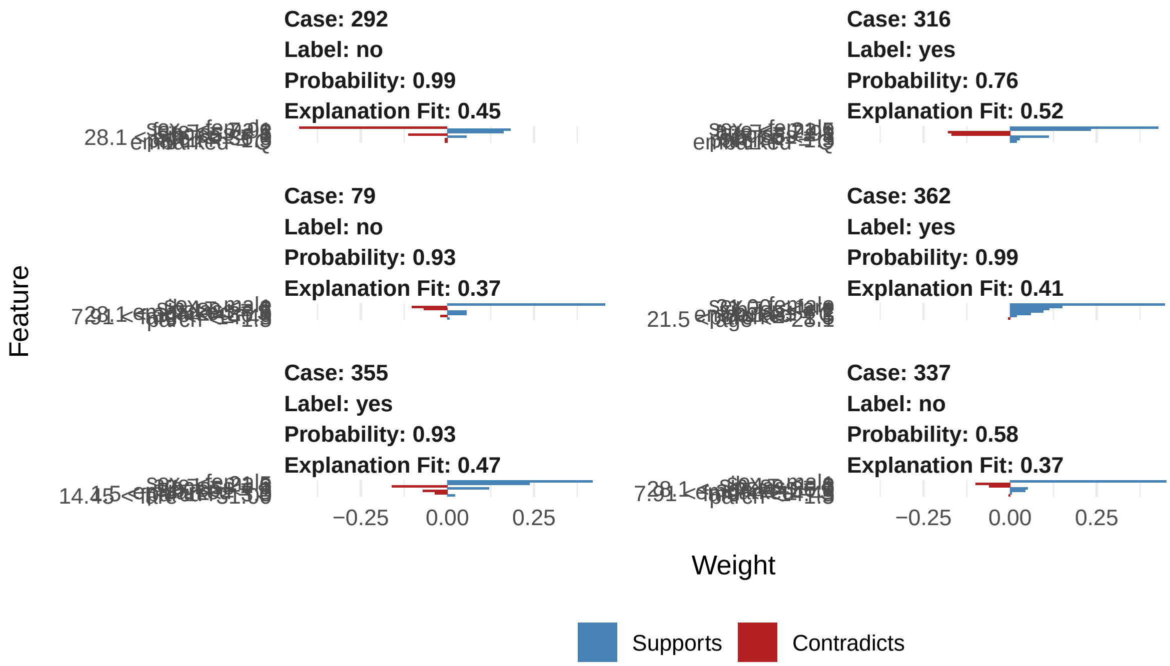
```r
pdp_age =
  mod_boost %>%
  partial(pred.var = c("age")) %>%
  autoplot(train = train_df, rug = TRUE)

pdp_fare =
  mod_boost %>%
  partial(pred.var = c("fare")) %>%
  autoplot(train = train_df, rug = TRUE)

pdp_sibsp =
  mod_boost %>%
  partial(pred.var = c("sib_sp")) %>%
  autoplot(train = train_df, rug = TRUE)

pdp_parch =
  mod_boost %>%
  partial(pred.var = c("parch")) %>%
  autoplot(train = train_df, rug = TRUE)

grid.arrange(pdp_age, pdp_fare, pdp_sibsp, pdp_parch, nrow = 2)
```



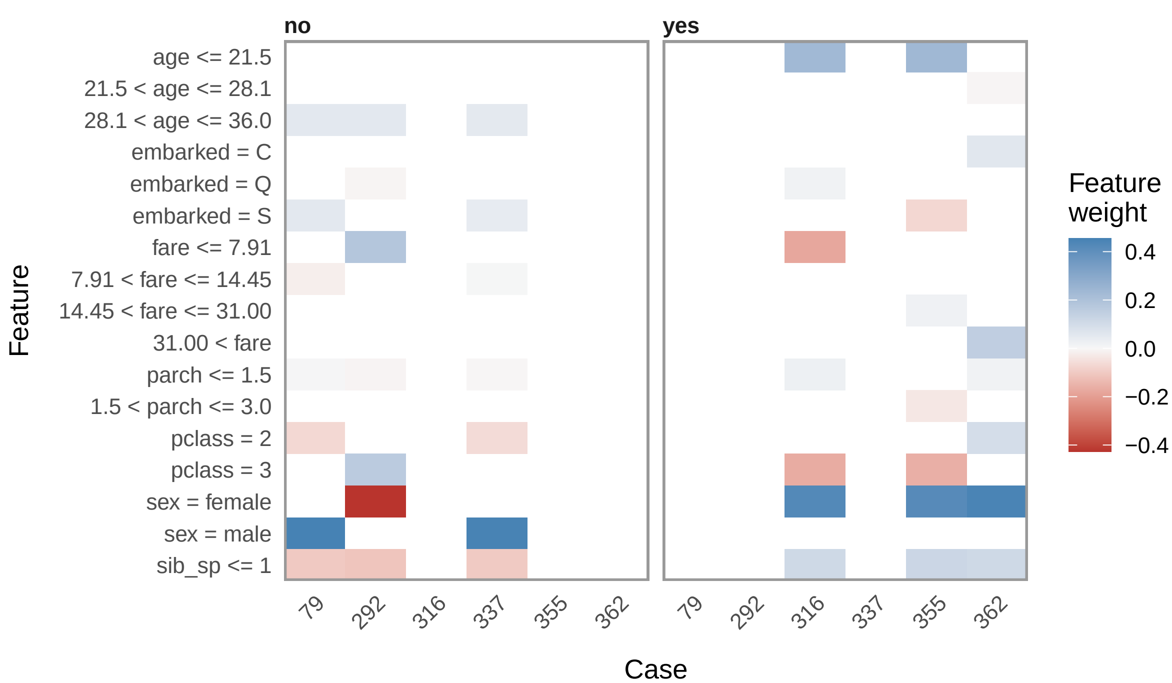

```
pdp_farexsibsp =
  mod_boost %>%
  partial(pred.var = c("sib_sp", "fare")) %>%
  autoplot(train = train_df, rug = TRUE)

pdp_farexage =
  mod_boost %>%
  partial(pred.var = c("age", "fare")) %>%
  autoplot(train = train_df, rug = TRUE)

grid.arrange(pdp_farexage, pdp_farexsibsp, nrow = 1)
```




```r
explainer = lime(train_df[, -1], mod_boost)
set.seed(15236)
new_obs = test_df[sample(418, 6), -c(8:9)]
explanation = lime::explain(new_obs,
                            explainer,
                            n_labels = 1,
                            n_features = 7)
plot_features(explanation)
```




```
plot_explanations(explanation)
```



Kaggle Competition Submission

```r
test_id = testna_df$passenger_id

test_res =
  pred_boost %>%
  as.data.frame() %>%
  mutate(survived = ifelse(pred_boost >= 0.5, 1, 0))

submission =
  cbind(test_id, test_res$survived) %>%
  as.data.frame() %>%
  rename(PassengerId = test_id,
         Survived = V2)

write_csv(submission, "submission.csv")
```