Final Project

Vasiliy Ostapenko (774 970 8)

May 10, 2022

DATA

Load Data

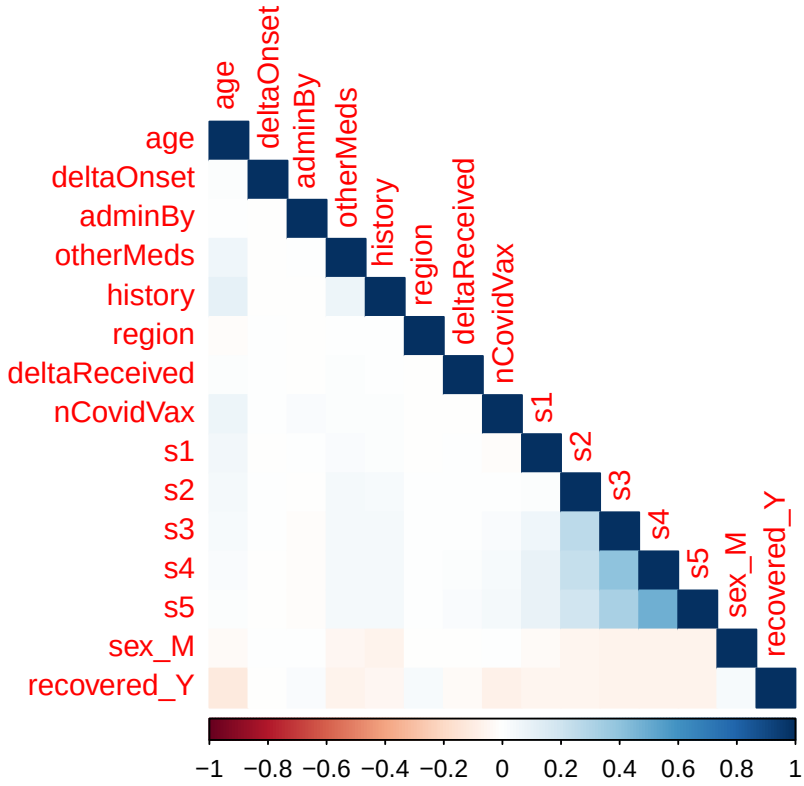
```
DATA_FOLDER = "./data"
COMBINED_FNAME = file.path(DATA_FOLDER, "combined.csv")
df = read.csv(COMBINED_FNAME) %>%
  column_to_rownames("vaersId")
```

Visualization

Categorical to Numeric Conversion

```
df = fastDummies::dummy_cols(df, remove_first_dummy=TRUE, remove_selected_columns=TRUE)
```

```
corrplot(cor(df[ , names(df) != "myocarditis"]),
         method="color", type="lower")
```




```
df = df[ , !(colnames(df) %in% c("s4", "s5"))] %>% copy()
df$myocarditis = as.factor(df$myocarditis)
```

Data Split

```
split = df %>%
  initial_split(prop=0.70, strata="myocarditis")

train = training(split)
test = testing(split)
```

```
folds = vfold_cv(train, v=3, strata="myocarditis")
```

MODELING

Recipe

```
rec = recipe(myocarditis ~ ., data=train) %>%
  step_normalize(all_predictors())
```

Models, Workflows, Parameters, CV

```
# Logistic Regression
mod_glm = logistic_reg(penalty=tune(), mixture=tune()) %>%
  set_engine("glm") %>%
  set_mode("classification")

work_glm = workflow() %>%
  add_model(mod_glm) %>%
  add_recipe(rec)

grid_glm = grid_regular(penalty(), mixture(), levels=2)

tune_glm = work_glm %>%
  tune_grid(resamples=folds, grid=grid_glm,
            metrics=metric_set(roc_auc, accuracy))

save(tune_glm, work_glm, file="./data/tune_glm.rda")
```

```
load(file="./data/tune_glm.rda")
tune_glm %>% collect_metrics() %>%
  select(-.estimator, -.config)
```

```
## # A tibble: 2 x 4
##   .metric   mean     n   std_err
##   <chr>    <dbl> <int>     <dbl>
## 1 accuracy 0.999     3 0.0000368
## 2 roc_auc  0.825     3 0.0125
```

```
# SVM
mod_svm = svm_rbf(cost=tune(), rbf_sigma=tune()) %>%
  set_engine("kernlab") %>%
  set_mode("classification")

work_svm = workflow() %>%
  add_model(mod_svm) %>%
  add_recipe(rec)

grid_svm = grid_regular(cost(), rbf_sigma(), levels=2)

tune_svm = work_svm %>%
  tune_grid(resamples=folds, grid=grid_svm,
            metrics=metric_set(roc_auc, accuracy))

save(tune_svm, work_svm, file="./data/tune_svm.rda")
```

```
load(file="./data/tune_svm.rda")
tune_svm %>% collect_metrics() %>%
  select(-.estimator, -.config)
```

```
## # A tibble: 8 x 6
```

```
##        cost    rbf_sigma .metric   mean     n   std_err
##       <dbl>        <dbl> <chr>    <dbl> <int>     <dbl>
## 1  0.000977 0.0000000001 accuracy 0.999     3 0.0000319
## 2  0.000977 0.0000000001 roc_auc  0.668     3 0.0197
## 3 32        0.0000000001 accuracy 0.999     3 0.0000319
## 4 32        0.0000000001 roc_auc  0.672     3 0.0206
## 5  0.000977 1            accuracy 0.999     3 0.0000319
## 6  0.000977 1            roc_auc  0.505     3 0.0116
## 7 32        1            accuracy 0.998     3 0.0000223
## 8 32        1            roc_auc  0.633     3 0.0252
```

```
# Random Forest
mod_rf = rand_forest(min_n=tune()) %>%
  set_engine("ranger") %>%
  set_mode("classification")

work_rf = workflow() %>%
  add_model(mod_rf) %>%
  add_recipe(rec)

grid_rf = grid_regular(min_n(), levels=2)

tune_rf = work_rf %>%
  tune_grid(resamples=folds, grid=grid_rf,
            metrics=metric_set(roc_auc, accuracy))

save(tune_rf, work_rf, file="./data/tune_rf.rda")
```

```
load(file="./data/tune_rf.rda")
tune_rf %>% collect_metrics() %>%
  select(-.estimator, -.config)
```

```
## # A tibble: 4 x 5
##   min_n .metric   mean     n   std_err
##   <int> <chr>    <dbl> <int>     <dbl>
## 1     2 accuracy 0.999     3 0.0000319
## 2     2 roc_auc  0.956     3 0.00696
## 3    40 accuracy 0.999     3 0.0000319
## 4    40 roc_auc  0.960     3 0.00720
```

```
# Boosted Trees
mod_boost = boost_tree(min_n=tune(), learn_rate=tune()) %>%
  set_engine("xgboost") %>%
  set_mode("classification")

work_boost = workflow() %>%
  add_model(mod_boost) %>%
  add_recipe(rec)

grid_boost = grid_regular(min_n(), learn_rate(), levels=2)

tune_boost = work_boost %>%
  tune_grid(resamples=folds, grid=grid_boost,
```

```
            metrics=metric_set(roc_auc, accuracy))

save(tune_boost, work_boost, file="./data/tune_boost.rda")
```

```
load(file="./data/tune_boost.rda")
tune_boost %>% collect_metrics() %>%
  select(-.estimator, -.config)
```

```
## # A tibble: 8 x 6
##   min_n   learn_rate .metric   mean     n   std_err
##   <int>        <dbl> <chr>    <dbl> <int>     <dbl>
## 1     2 0.0000000001 accuracy 0.999     3 0.0000319
## 2     2 0.0000000001 roc_auc  0.5       3 0
## 3    40 0.0000000001 accuracy 0.999     3 0.0000319
## 4    40 0.0000000001 roc_auc  0.5       3 0
## 5     2 0.1          accuracy 0.999     3 0.0000319
## 6     2 0.1          roc_auc  0.654     3 0.0763
## 7    40 0.1          accuracy 0.999     3 0.0000319
## 8    40 0.1          roc_auc  0.656     3 0.0781
```

Best Model Determination and Training

```
tune_rf_best = tune_rf %>%
  select_best("roc_auc")

work_rf_final = work_rf %>%
  finalize_workflow(tune_rf_best)

fit_rf = work_rf_final %>%
  fit(train)

save(fit_rf, file="./data/fit_rf.rda")
```

EVALUATION

Best Model Testing and Evaluation

```
load(file="./data/fit_rf.rda")
```

```
predict_rf = augment(fit_rf, test)
```

```
roc_auc(data=predict_rf, truth=myocarditis,
        estimate=.pred_1, event_level="second")
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.950
```

```
accuracy(data=predict_rf, truth=myocarditis,
         estimate=.pred_class)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.999
```

```
conf_mat(data=predict_rf, truth=myocarditis,
         estimate=.pred_class)
```

```
##            Truth
## Prediction     0     1
##          0 83862    92
##          1     0     0
```