DATA 607 Final Project: Machine Learning

David Simbandumwe

/ Overview

Applying a machine learning model to the survey data using the same variables from the linear regression models.

// Setup

```
set.seed(1234)
rm(list=ls())
```

/ Consumer Financial Protection Bureau Survey

// Download and Tidy dataset from CFPB

```
# get cfpb file
cfpb_df <- getCFPBFile()
```

```
## The following `from` values were not present in `x`: -1, 1, 2, 3, 4, 5, 6, 7, 98,
```

```
cfpb_df$cfpb_score_4cat <- cut(cfpb_df$cfpb_score, breaks = c(-10, 40, 60, 80, 100),
                               labels = c("< 40","40-60","60-80","80-100"),
                               right = FALSE,
                               include.lowest=TRUE)
cfpb_df <- cfpb_df %>% filter(cfpb_score >= 0)

# reduce cfpb data set
cfpb_df <- slice_sample(cfpb_df, weight_by=cfpb_score_4cat ,n=5000)
cfpb_df <- cfpb_df %>% select(cfpb_score, cfpb_score_4cat, econ_save_rate, house_mort
```

// prepare data

```
# Put 3/4 of the data into the training set
cfpb_split <- initial_split(cfpb_df, prop = 0.8, strata = cfpb_score_4cat)

# Create dataframes for the two sets:
cfpb_train_data <- training(cfpb_split)
cfpb_test_data <- testing(cfpb_split)


# define recipts
cfpb_rec <-
  recipe(cfpb_score_4cat ~ econ_save_rate + house_mortgage + age_8cat + econ_hh_income
         data = cfpb_train_data) %>%
  step_naomit(everything(), skip = TRUE) %>%
  step_upsample(cfpb_score_4cat, over_ratio = .5) %>%
  step_novel(all_nominal(), -all_outcomes()) %>%
  step_normalize(all_numeric(), -all_outcomes()) %>%
  step_dummy(all_nominal(), -all_outcomes()) %>%
  step_zv(all_numeric(), -all_outcomes()) %>%
  step_corr(all_predictors(), threshold = 0.7, method = "spearman")


summary(cfpb_rec)
```

```
## # A tibble: 5 × 4
##   variable        type    role      source
##   <chr>           <chr>   <chr>     <chr>
## 1 econ_save_rate  nominal predictor original
## 2 house_mortgage  nominal predictor original
## 3 age_8cat        nominal predictor original
## 4 econ_hh_income  nominal predictor original
## 5 cfpb_score_4cat nominal outcome   original
```

```
# folds and spec
cv_folds <-
 vfold_cv(cfpb_train_data,
          v = 5,
          strata = cfpb_score_4cat)


rf_spec <-
  rand_forest() %>%
  set_engine("ranger", importance = "impurity") %>%
  set_mode("classification")
```

// tune workflow

```
# workflow
rf_wflow <-
 workflow() %>%
 add_recipe(cfpb_rec) %>%
 add_model(rf_spec)


# resample
rf_res <-
  rf_wflow %>%
  fit_resamples(
    resamples = cv_folds,
    metrics = metric_set(recall, precision, f_meas,accuracy, kap,roc_auc, sens, spec)
    control = control_resamples(save_pred = TRUE)
    )
rf_res %>%  collect_metrics(summarize = TRUE)
```

```
## # A tibble: 8 × 6
##   .metric   .estimator  mean     n std_err .config
##   <chr>     <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy  multiclass 0.534     5 0.0157  Preprocessor1_Model1
## 2 f_meas    macro      0.455     5 0.0106  Preprocessor1_Model1
## 3 kap       multiclass 0.277     5 0.0218  Preprocessor1_Model1
## 4 precision macro      0.443     5 0.0116  Preprocessor1_Model1
## 5 recall    macro      0.486     5 0.00840 Preprocessor1_Model1
## 6 roc_auc   hand_till  0.779     5 0.00953 Preprocessor1_Model1
## 7 sens      macro      0.486     5 0.00840 Preprocessor1_Model1
## 8 spec      macro      0.821     5 0.00589 Preprocessor1_Model1
```

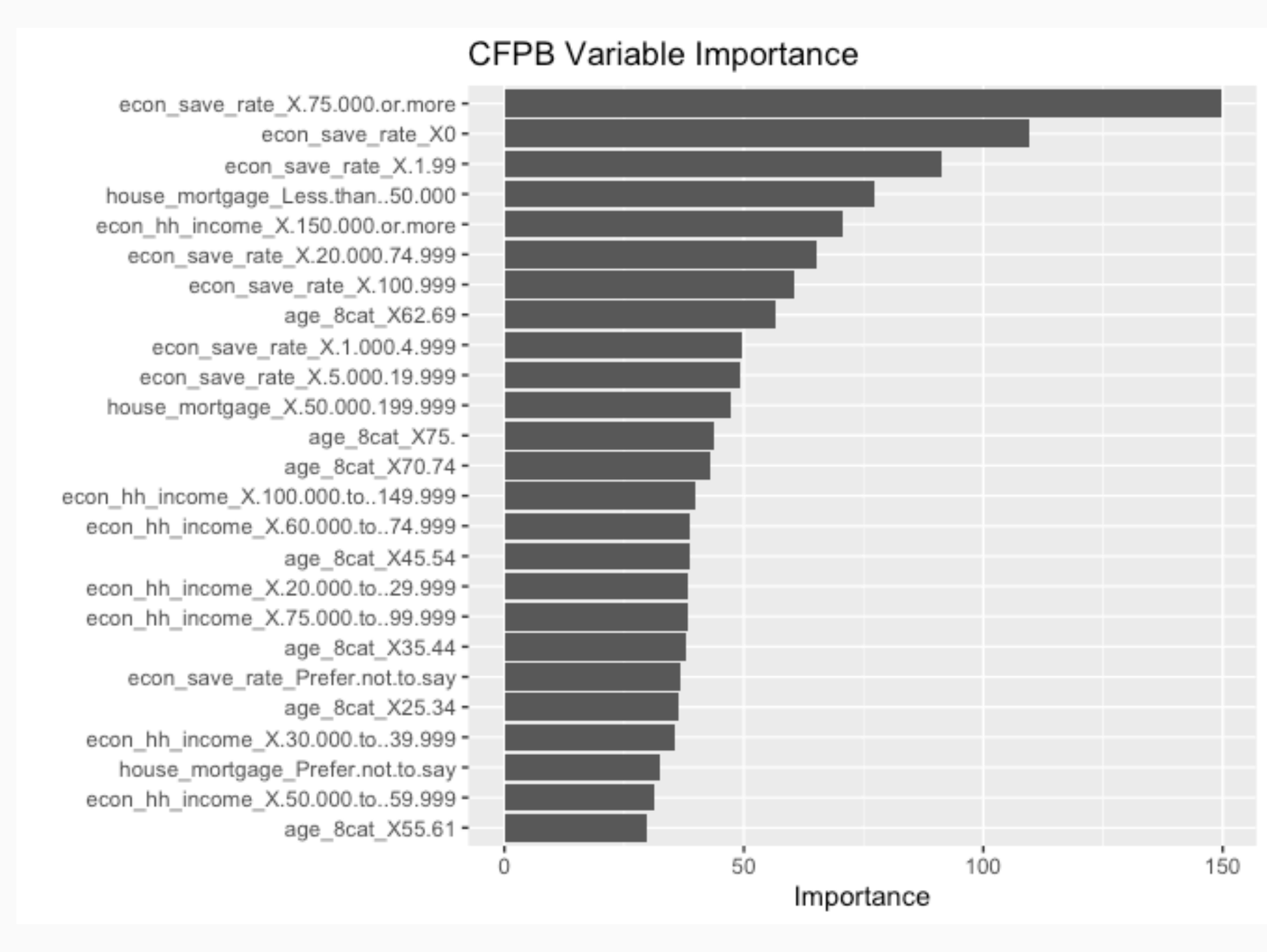
```
rf_metrics <-
  rf_res %>%
  collect_metrics(summarise = TRUE) %>%
  mutate(model = "Random Forest")
```

// Final Fit

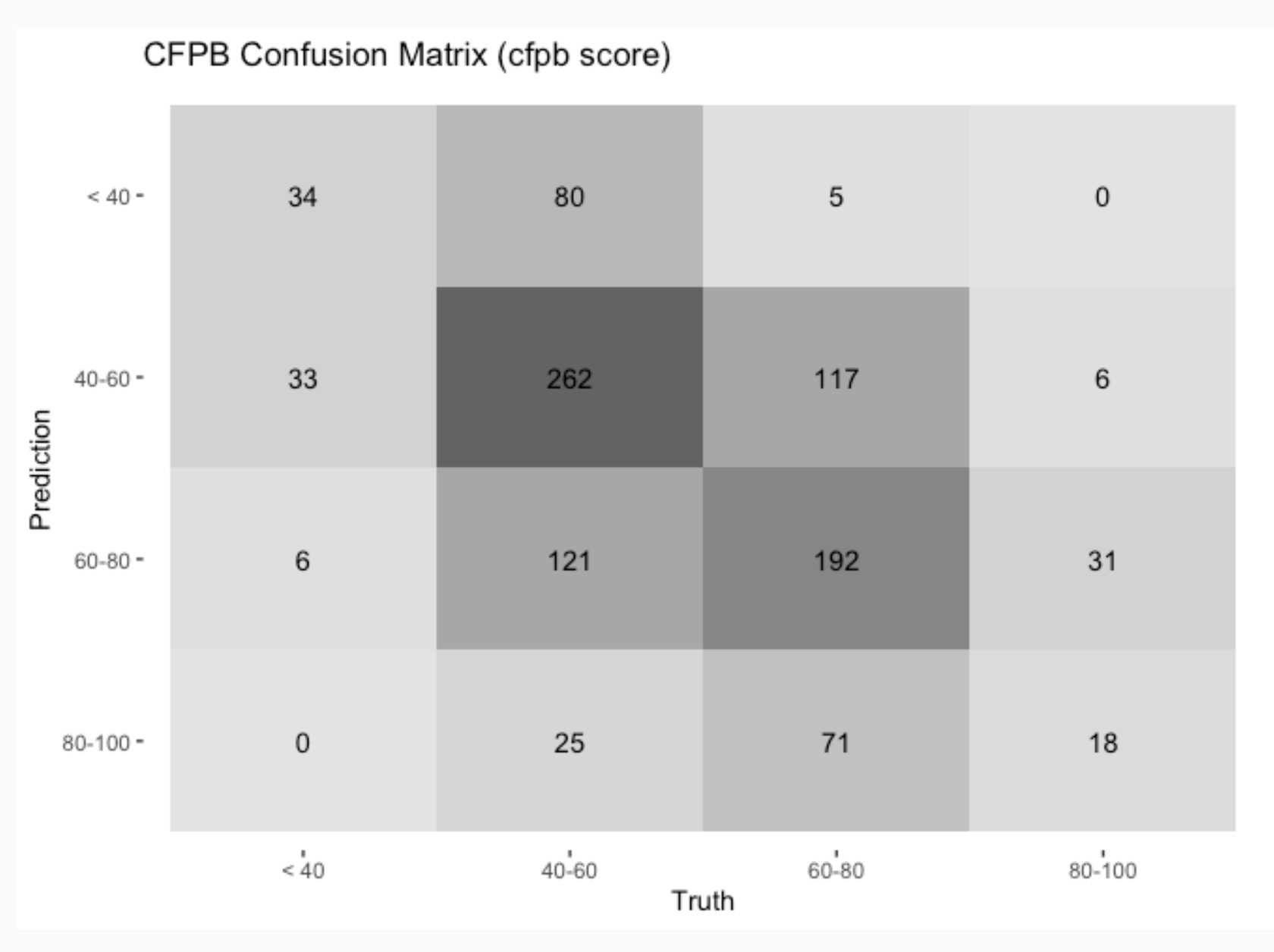
```
last_fit_rf <- last_fit(rf_wflow,
                        split = cfpb_split,
                        metrics = metric_set(recall, precision, f_meas,accuracy, kap,
                        )


last_fit_rf %>%
  collect_metrics()
```

```
## # A tibble: 8 × 4
##   .metric   .estimator .estimate .config
##   <chr>     <chr>          <dbl> <chr>
## 1 recall    macro          0.457 Preprocessor1_Model1
## 2 precision macro          0.405 Preprocessor1_Model1
## 3 f_meas    macro          0.417 Preprocessor1_Model1
## 4 accuracy  multiclass     0.505 Preprocessor1_Model1
## 5 kap       multiclass     0.236 Preprocessor1_Model1
## 6 sens      macro          0.457 Preprocessor1_Model1
## 7 spec      macro          0.812 Preprocessor1_Model1
## 8 roc_auc   hand_till      0.782 Preprocessor1_Model1
```

```
last_fit_rf %>%
  pluck(".workflow", 1) %>%
  extract_fit_parsnip() %>%
  vip(num_features = 25) +
  labs(title = "CFPB Variable Importance")
```




```
last_fit_rf %>%
  collect_predictions() %>%
  conf_mat(cfpb_score_4cat, .pred_class) %>%
  autoplot(type = "heatmap") +
  labs(title = "CFPB Confusion Matrix (cfpb score)")
```



/ Federal Reserve System Survey

// Download and Tidy dataset from FED

```
# get fed file
fed_df <- getFedFile()

#filter data
fed_df <- fed_df %>% drop_na()
fed_df <- fed_df %>% filter(!(econ_saving == "" |  credit_guess =="" | health ==""))
fed_df$credit_guess <- str_replace_all(fed_df$credit_guess, "[^[:alnum:]]", "")


fed_df$cfpb_score_4cat <- cut(fed_df$cfpb_score, breaks = c(-10, 40, 60, 80, 100),
                              labels = c("< 40","40-60","60-80","80-100"),
                              right = FALSE,
                              include.lowest=TRUE)

# reduce cfpb dataset
fed_df <- slice_sample(fed_df, weight_by=cfpb_score_4cat ,n=5000)

fed_df <- fed_df %>% select(cfpb_score, cfpb_score_4cat,age_7cat, econ_saving, econ_i
                            econ_pay_exp400, econ_skip_med)
```

// prepare data

```
# Put 3/4 of the data into the training set
fed_split <- initial_split(fed_df,
                           prop = .8,
                           strata = cfpb_score_4cat)

# Create dataframes for the two sets:
fed_train_data <- training(fed_split)
fed_test_data <- testing(fed_split)



# define recipts
fed_rec <-
  recipe(cfpb_score_4cat ~ age_7cat, econ_saving, econ_inc_4cat, econ_fin_ok, econ_pa
         data = fed_train_data) %>%
  step_naomit(everything(), skip = TRUE) %>%
  step_upsample(cfpb_score_4cat, over_ratio = .5) %>%
  step_novel(all_nominal(), -all_outcomes()) %>%
  step_normalize(all_numeric(), -all_outcomes()) %>%
  step_dummy(all_nominal(), -all_outcomes()) %>%
  step_zv(all_numeric(), -all_outcomes()) %>%
  step_corr(all_predictors(), threshold = 0.7, method = "spearman", skip = TRUE)


summary(fed_rec)
```

```
## # A tibble: 2 × 4
##   variable        type    role      source
##   <chr>           <chr>   <chr>     <chr>
## 1 age_7cat        nominal predictor original
## 2 cfpb_score_4cat nominal outcome   original
```

```
# folds and spec
cv_fed_folds <-
 vfold_cv(fed_train_data,
          v = 5,
          strata = cfpb_score_4cat)


rf_fed_spec <-
  rand_forest() %>%
  set_engine("ranger", importance = "impurity") %>%
  set_mode("classification")
```

// tune workflow

```
# workflow
rf_fed_wflow <-
 workflow() %>%
 add_recipe(fed_rec) %>%
 add_model(rf_fed_spec)


# resample
rf_fed_res <-
  rf_fed_wflow %>%
  fit_resamples(
    resamples = cv_fed_folds,
    metrics = metric_set(recall, precision, f_meas,accuracy, kap,roc_auc, sens, spec)
    control = control_resamples(save_pred = TRUE)
    )
```

```
## ! Fold1: internal: While computing multiclass `precision()`, some levels had no p.
```

```
## ! Fold2: internal: While computing multiclass `precision()`, some levels had no p.
```

```
## ! Fold3: internal: While computing multiclass `precision()`, some levels had no p.
```

```
## ! Fold4: internal: While computing multiclass `precision()`, some levels had no p.
```

```
## ! Fold5: internal: While computing multiclass `precision()`, some levels had no p.
```

```
rf_fed_res %>%  collect_metrics(summarize = TRUE)
```

```
## # A tibble: 8 × 6
##   .metric   .estimator  mean     n std_err .config
##   <chr>     <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy  multiclass 0.468     5 0.00629 Preprocessor1_Model1
## 2 f_meas    macro      0.397     5 0.00901 Preprocessor1_Model1
## 3 kap       multiclass 0.121     5 0.00818 Preprocessor1_Model1
## 4 precision macro      0.413     5 0.0105  Preprocessor1_Model1
## 5 recall    macro      0.320     5 0.00587 Preprocessor1_Model1
## 6 roc_auc   hand_till  0.641     5 0.00593 Preprocessor1_Model1
## 7 sens      macro      0.320     5 0.00587 Preprocessor1_Model1
## 8 spec      macro      0.779     5 0.00183 Preprocessor1_Model1
```

```
rf_fed_metrics <-
  rf_fed_res %>%
  collect_metrics(summarise = TRUE) %>%
  mutate(model = "Random Forest")
```

// Final Fit

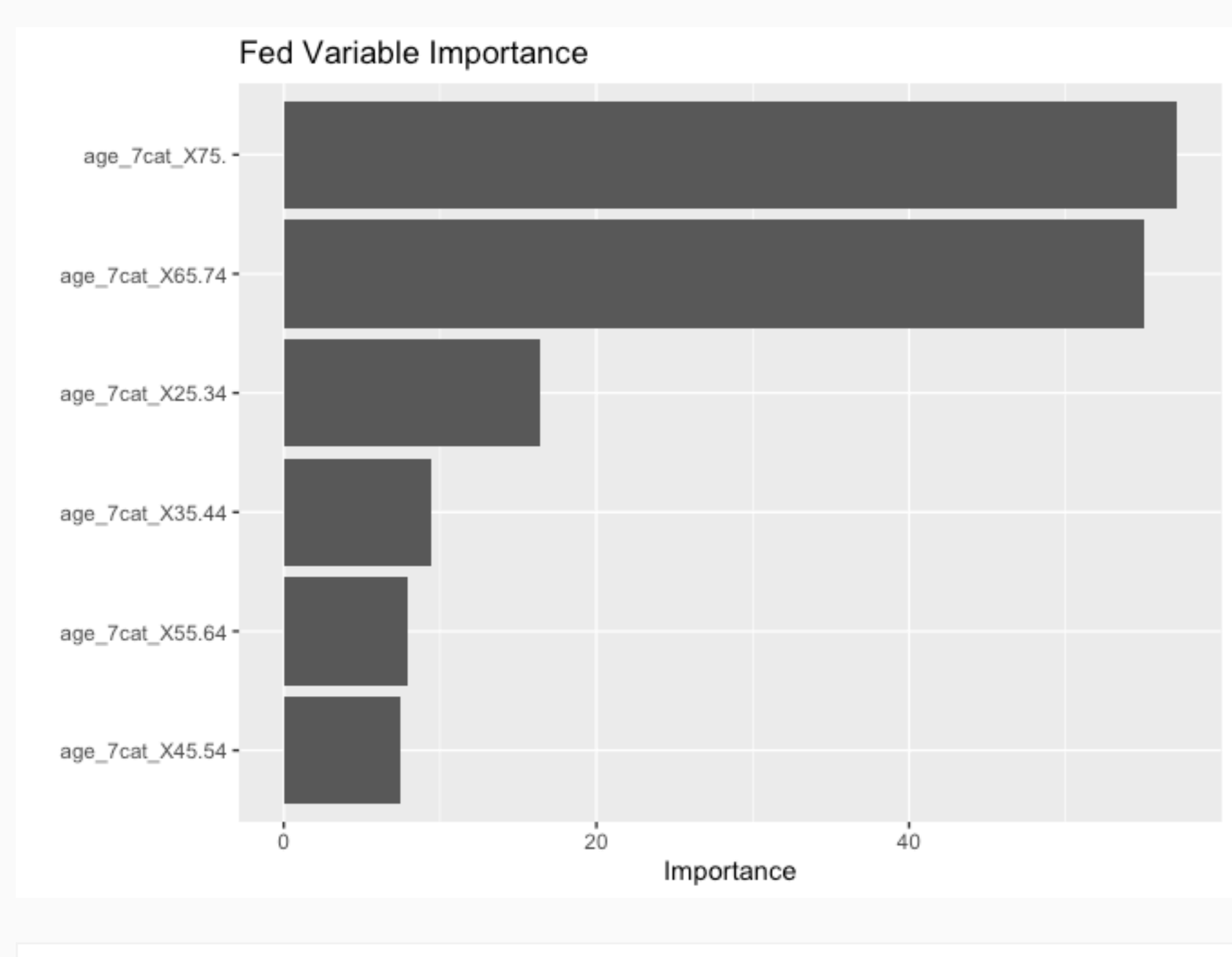
```
last_fit_fed_rf <- last_fit(rf_fed_wflow,
                        split = fed_split,
                        metrics = metric_set(recall, precision, f_meas,accuracy, kap,
                        )
```

```
## ! train/test split: internal: While computing multiclass `precision()`, some level
```

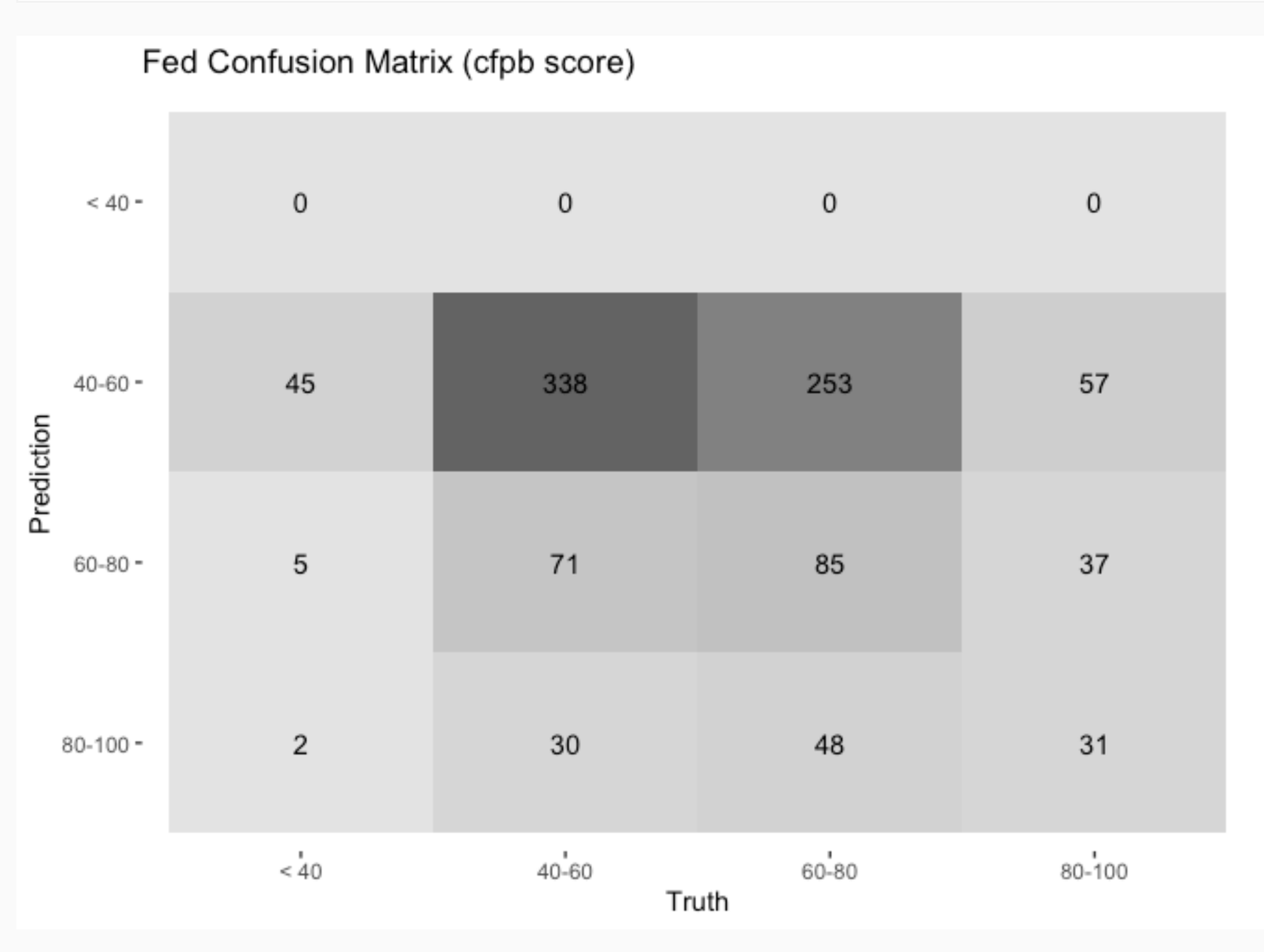
```
last_fit_fed_rf %>%
  collect_metrics()
```

```
## # A tibble: 8 × 4
##   .metric   .estimator .estimate .config
##   <chr>     <chr>          <dbl> <chr>
## 1 recall    macro         0.310  Preprocessor1_Model1
## 2 precision macro         0.399  Preprocessor1_Model1
## 3 f_meas    macro         0.384  Preprocessor1_Model1
## 4 accuracy  multiclass    0.453  Preprocessor1_Model1
## 5 kap       multiclass    0.0991 Preprocessor1_Model1
## 6 sens      macro         0.310  Preprocessor1_Model1
## 7 spec      macro         0.774  Preprocessor1_Model1
## 8 roc_auc   hand_till     0.605  Preprocessor1_Model1
```

```
last_fit_fed_rf %>%
  pluck(".workflow", 1) %>%
  extract_fit_parsnip() %>%
  vip(num_features = 10) +
  labs(title = "Fed Variable Importance")
```




```
last_fit_fed_rf %>%
  collect_predictions() %>%
  conf_mat(cfpb_score_4cat, .pred_class) %>%
  autoplot(type = "heatmap") +
  labs(title = "Fed Confusion Matrix (cfpb score)")
```



/ Conclusion

The structure of the random forest model dictates using a categorical predictions. To support this model a factor representation of the cfpb score was create with 4 categories. The resulting model has a 0.4720 precision and fails to predict any values score less than 40 or a score over 80.