PLTR VARIABLES IMPORTANCE

ZAINAB BELGADA* DAMIEN ZINSOU†

17/01/2021

Load & Prepare the Housing Dataset

```
source("./../../scripts/prepare_housing_dataset.R")

# Load The Raw Dataset
raw_housing_dataset <- readxl::read_excel("./../../data/hmeq.xls", sheet = 'hmeq')

# Impute Missing Values
# One-Hot Encode Categorical Predictors
clean_housing_dataset <- prepare_housing_dataset(raw_housing_dataset)
```

```
head(clean_housing_dataset)
```

```
## # A tibble: 6 x 19
##   BAD    LOAN MORTDUE  VALUE   YOJ DEROG DELINQ CLAGE  NINQ  CLNO DEBTINC
##   <fct> <dbl>   <dbl>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1 1      1100  25860  3.90e4 10.5  0      0      94.4  1      9      33.8
## 2 1      1300  70053  6.84e4  7    0      2     122.   0     14      33.8
## 3 1      1500  13500  1.67e4  4    0      0     149.   1     10      33.8
## 4 1      1500  73761. 1.02e5  8.92 0.255  0.449 180.   1.19  21.3    33.8
## 5 0      1700  97800  1.12e5  3    0      0      93.3  0     14      33.8
## 6 1      1700  30548  4.03e4  9    0      0     101.   1      8      37.1
## # ... with 8 more variables: JOB_Mgr <int>, JOB_Office <int>, JOB_Other <int>,
## #   JOB_ProfExe <int>, JOB_Sales <int>, JOB_Self <int>, REASON_DebtCon <int>,
## #   REASON_HomeImp <int>
```

Spliting Data Into Training & Testing Sets

50% of data will be used for training. The other half will be used for testing:

```
set.seed(8081)

n_rows <- nrow(clean_housing_dataset)
n_train   <- round(n_rows * 0.5)

i_train   <- sample(1:n_rows, size = n_train)
```

*Institut d'Economie d'Orleans, zainab.belgada@etu.univ-orleans.fr
†Institut d'Economie d'Orleans, zinsou.mezonlin@etu.univ-orleans.fr

```
train_set <- clean_housing_dataset[i_train, ]
test_set  <- clean_housing_dataset[-i_train, ]
```

```
print(dim(train_set))
```

```
## [1] 2980   19
```

```
print(dim(test_set))
```

```
## [1] 2980   19
```

Training The PLTR Model

In order to save some computation time, we will first generate predictors pairs upfront:

```
predictors_set <- clean_housing_dataset %>%
   names() %>%
   tail(n = -1)

predictors_pairs <- predictors_set %>%
   combn(m = 2) %>%
   purrr::array_branch(margin = 2)
```

```
source("./../../scripts/rules_utilities.R")
source("./../../scripts/penalized_learner.R")
source("./../../scripts/pltr_learner.R")

# Using Adaptive LASSO Penalty (penalty = 2)
results <- pltr_learner(train_set, test_set, predictors_pairs, penalty = 2)
```

The number of rules extracted by the PLTR algorithm:

```
results[["count_extracted_rules"]]
```

```
## [1] 103
```

Test Results

```
source("./../../scripts/compute_evaluation_criteria.R")

eval_metrics <- compute_evaluation_criteria(
   test_set$BAD %>% as.character() %>% as.numeric(),
   results[['Predicted_Y_Test_Prob']],
   results[['Predicted_Y_Test_Class']]
)
```

```
eval_metrics
```

```
## # A tibble: 1 x 6
##     AUC  GINI    F1   PCC     BS    KS
##   <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>
## 1 0.907 0.815 0.703 0.893 0.0813 0.681
```

Predictors Importance

Top-15 Predictors:

```
results[["coef_ranks"]] %>%
   head(n = 15) %>%
   dplyr::mutate(
      Predictor = factor(Predictor, levels = Predictor[order(Coefficient_Magnitude)])
   ) %>%
   ggplot(aes(x = Predictor, y = Coefficient_Magnitude)) +
   geom_bar(stat = "identity", fill = "#f68060", alpha = .6, width = .4) +
   coord_flip() +
   xlab("") +
   ylab("COEFFICIENT MAGNITUDE") +
   theme_bw()
```