# Predictive modelling

George, John & William

11/17/2021

## Data Preparation and cleaning

```r
FIHI_sub2 <- FIHI_sub2 %>%
  mutate( permanent_address = factor(permanent_address, levels = 1:2,
                      labels = c("Yes", "No")),
          FI_q28= factor(FI_q28, levels = 1:2,
                      labels = c("Yes", "No")),
          FI_q30= factor(FI_q30, levels = 1:2,
                      labels = c("Yes", "No")),
          FI_q31= factor(FI_q31, levels = 1:2,
                      labels = c("Yes", "No")))
```

```r
metrics <- function(model_object, response="", test_data=dat$test) {
  # response = "permanent_address"
  # model_object <- log
  #
  if(is.null(test_data)) test_data <- testing

  # make predictions
  prediction <- predict(model_object, test_data)
  prediction_A <- predict(model_object, test_data, type="prob")

  target <- test_data[, response]

  cmat <- confusionMatrix(prediction, target, mode = "prec_recall")
  ROC <- roc(target, predictor = prediction_A[,2])
  AUC_m<-round(ROC$auc, digits=4)
  misscal<- round(mean(prediction != target),digits = 2)

 # Returned outputs
 return(list(
   accuracy = (1-misscal),
   mcr = misscal,
   sens = round(cmat$byClass[1],2),
   spec = round(cmat$byClass[2],2),
   fbeta = round(cmat$byClass[7],2),
   auc = AUC_m
 ))
}
```

```r
#------- Compute performance metrics for the full models ---------------
eval_table <- function(model_list,response, resp_label='---')
{
  log.metric <- metrics(model_list$glm, response)
  lda.metric <- metrics(model_list$lda, response)
  knn.metric <- metrics(model_list$knn, response)
  nnet.metric <- metrics(model_list$nnet, response)
  svc.metric <- metrics(model_list$svmLinear, response)
  svmR.metric <- metrics(model_list$svmRadial, response)

  mod.sum <- data.frame(rbind(
                            c("Logistic", log.metric$mcr, log.metric$accuracy, log.metric$sens, log.metr
                            c("LDA",  lda.metric$mcr, lda.metric$accuracy, lda.metric$sens, lda.metric$
                            c("KNN", knn.metric$mcr, knn.metric$accuracy, knn.metric$sens, knn.metric$sp
                            c("Neural Network", nnet.metric$mcr, nnet.metric$accuracy, nnet.metric$sens
                            # c("Bagging", bag.metric$mcr, bag.metric$accuracy, bag.metric$sens, bag.me
                            c("SVM Linear",  svc.metric$mcr, svc.metric$accuracy, svc.metric$sens, svc.
                            c("SVM Radial", svmR.metric$mcr, svmR.metric$accuracy, svmR.metric$sens, svm

  names(mod.sum) <- c("Model", "Misclassification Rate", "Accuracy",
                      "Sensitivity", "Specificity", "fbeta", "AUC")

  kable(mod.sum, align = "lcccccc", caption = paste("Evaluation metrics for", resp_label, " as a respons
    kable_paper("hover", full_width = F)%>%
         kable_styling(font_size = 12, latex_options = c("HOLD_position"))

}

# display table
# eval_table(dat$model_list, "FI_q31", "Food Insecurity 3")
```

### Comparing Models

```
# ensemble_1 <- caretEnsemble(dat$model_list,
#                              metric = "ROC",
#                              trControl = trctrl)
#
# plot(ensemble_1)
```

### Variable of importance obtained from the best model

```
# Var <- varImp(dat$model_list$svmR, scale = FALSE)
# plot(Var, main ="Figure: Variable of Importance")
```

```
# var_best <- as.data.frame(Var$importance) %>%
#   filter(Yes > 0.52) %>%
#   arrange(desc(Yes)) %>%
#   dplyr::select(Yes) %>%
#   rename(Importance = Yes)
#
# plot(var_best)
```

## Housing Insecurity

```
dat <- modeling("permanent_address")
```

```
## Warning: Number of logged events: 25
```

```
## # weights:  58
## initial  value 4429.971131
## iter  10 value 3795.646761
## iter  20 value 3681.868936
## iter  30 value 3663.194375
## iter  40 value 3654.494476
## iter  50 value 3649.830524
## iter  60 value 3649.382161
## iter  70 value 3643.761990
## iter  80 value 3638.416441
## iter  90 value 3627.761237
## iter 100 value 3618.838152
## final  value 3618.838152
## stopped after 100 iterations
```

```
eval_table(dat$model_list, "permanent_address", "Permanent House Address")
```

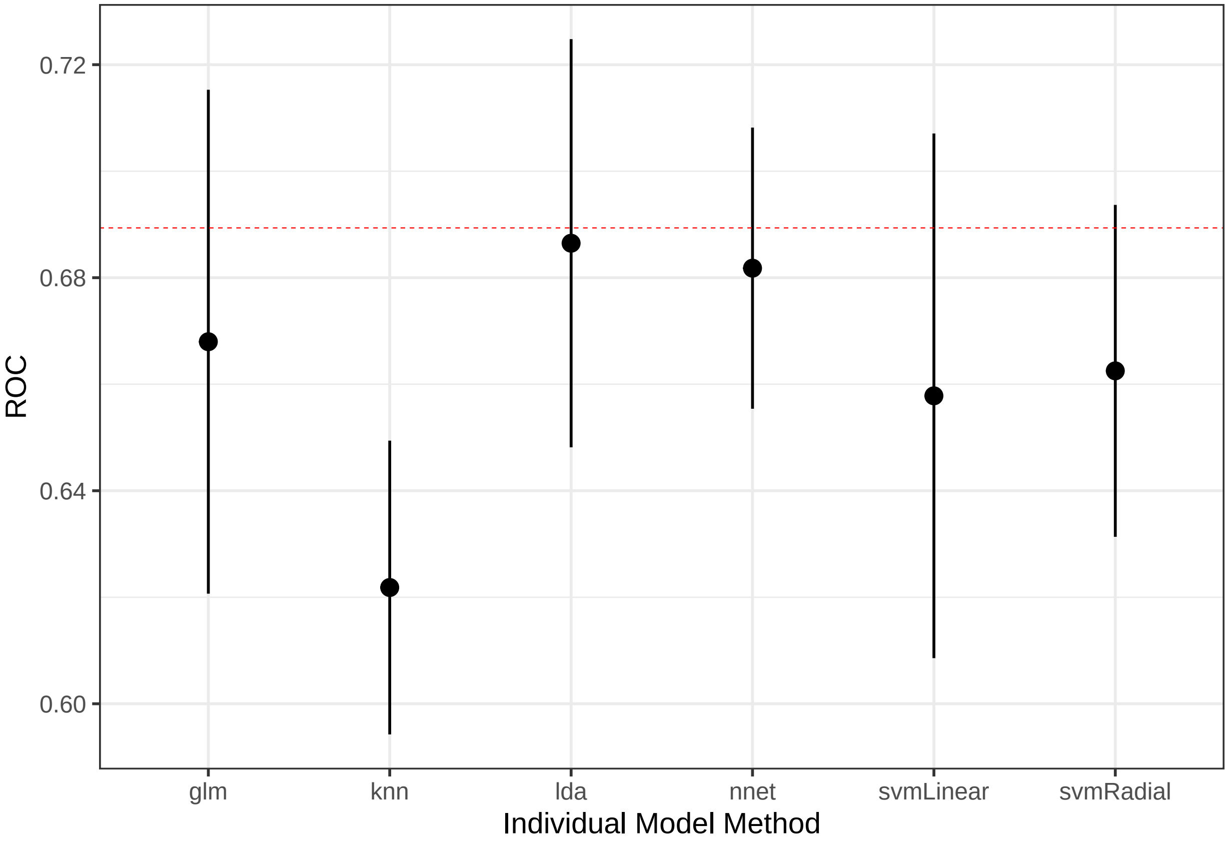
```
## Setting levels: control = Yes, case = No

## Setting direction: controls < cases

## Setting levels: control = Yes, case = No

## Setting direction: controls < cases

## Setting levels: control = Yes, case = No

## Setting direction: controls < cases

## Setting levels: control = Yes, case = No

## Setting direction: controls < cases

## Setting levels: control = Yes, case = No

## Setting direction: controls < cases

## Setting levels: control = Yes, case = No

## Setting direction: controls < cases
```

Table 1: Evaluation metrics for Permanent House Address as a response variable

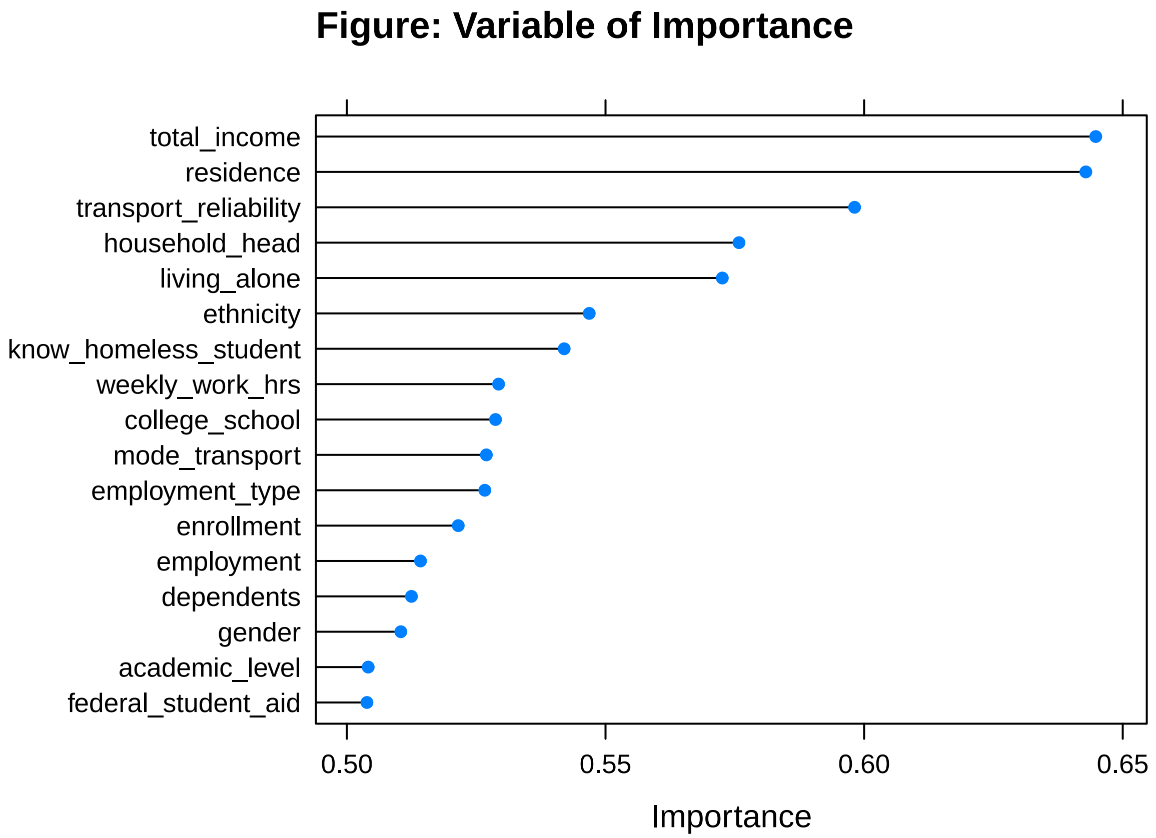
| Model | Misclassification Rate | Accuracy | Sensitivity | Specificity | fbeta | AUC |
|---|---|---|---|---|---|---|
| Logistic | 0.24 | 0.76 | 0.77 | 0.57 | 0.86 | 0.7197 |
| LDA | 0.22 | 0.78 | 0.79 | 0.59 | 0.87 | 0.7249 |
| KNN | 0.35 | 0.65 | 0.65 | 0.61 | 0.78 | 0.6806 |
| Neural Network | 0.17 | 0.83 | 0.85 | 0.41 | 0.91 | 0.6922 |
| SVM Linear | 0.23 | 0.77 | 0.78 | 0.57 | 0.87 | 0.7168 |
| SVM Radial | 0.17 | 0.83 | 0.85 | 0.47 | 0.91 | 0.7217 |

```
plot(dat$ensemble)
```

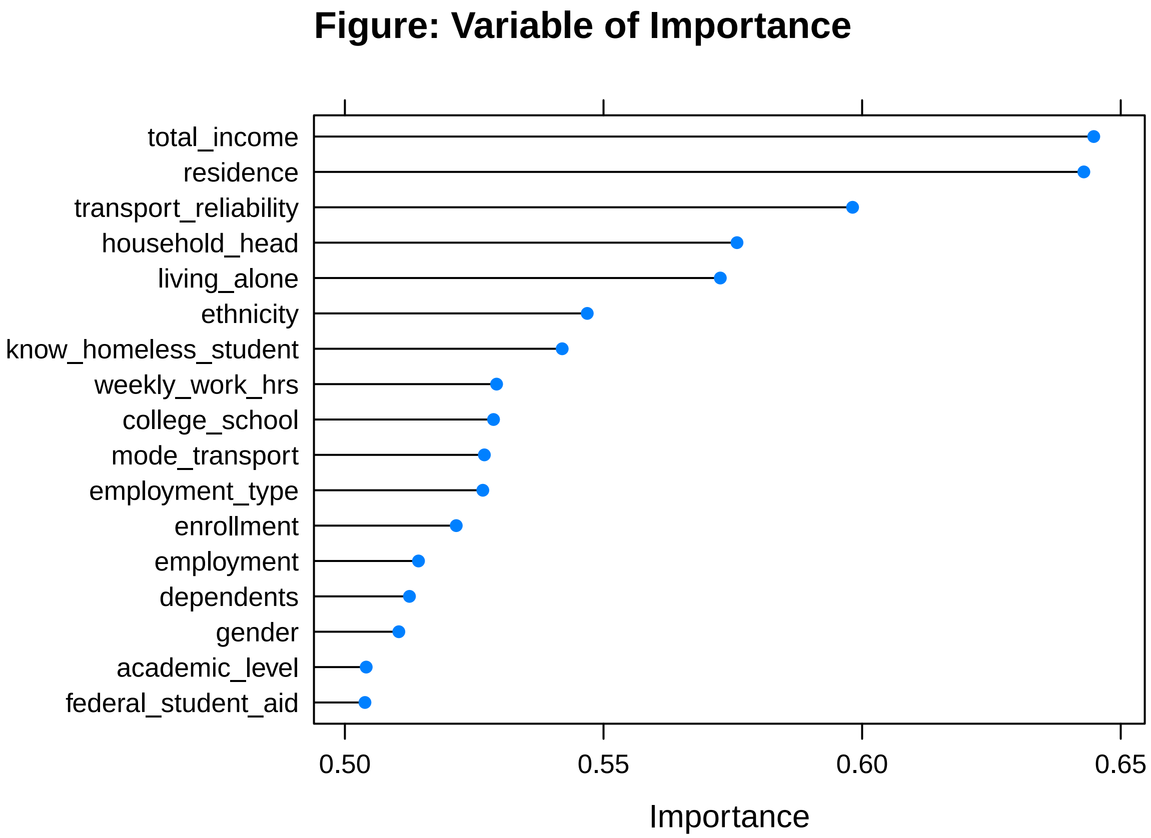
```
## Warning: Duplicated aesthetics after name standardisation: size
```



```
par(mfrow=c(1,2))
Var <- varImp(dat$model_list$svmRadial, scale = FALSE)
plot(Var, main ="Figure: Variable of Importance")
```



```r
Var <- varImp(dat$model_list$lda, scale = FALSE)
plot(Var, main ="Figure: Variable of Importance")
```



```r
dev.off()
```

```
## null device 
##           1
```