Final project code

Team AI

2022-06-24

import

```
library(tidyverse)
library(caTools)
library(randomForest)
library(caret)
library(e1071)
library(rpart)
library(tidymodels)
library(schrute)
library(lubridate)
library(quantreg)
library(pROC)
library(klaR)
library(psych)
library(MASS)
library(devtools)
library(ROCR)
```

Data:

```
dataset = read.csv("C:\\Tal\\Data Engineer\\BSc\\semester 4\\Advanced programming\\project\\children_at
datasetA <- dataset %>%
  filter(Test == 'A') %>%
  mutate(Hope_A = Q1+Q2+Q3+Q4+Q5+Q6+Q7+(5-Q8)+Q9+(5-Q10)+(5-Q11)) %>%
  dplyr::select(Id,Hope_A)

datasetB <- dataset %>%
  filter(Test == 'B') %>%
  mutate(Hope_B = Q1+Q2+Q3+Q4+Q5+Q6+Q7+(5-Q8)+Q9+(5-Q10)+(5-Q11)) %>%
  dplyr::select(Id,Hope_B)

datasetC <- dataset %>%
  filter(Test == 'C') %>%
  mutate(Hope_C = Q1+Q2+Q3+Q4+Q5+Q6+Q7+(5-Q8)+Q9+(5-Q10)+(5-Q11)) %>%
  dplyr::select(Id,Hope_C)

dataset = left_join(dataset,left_join(datasetA,left_join(datasetB,datasetC)))
```

```
## Joining, by = "Id"
## Joining, by = "Id"
## Joining, by = "Id"
```

```
dataset <- dataset %>%
  mutate(n_of_t = T1+T2+T3+T4+T5+T6+T7+T8+T9+T10,Hope = Q1+Q2+Q3+Q4+Q5+Q6+Q7+(5-Q8)+Q9+(5-Q10)+(5-Q11))
```

Inital hypothesis tests:

- test 1: Compare the Hope at first survey and third

```
Hope_1 <- dataset %>%
  filter(Test == 'A') %>%
  dplyr::select(Hope)

Hope_3 <- dataset %>%
  filter(Test == 'C') %>%
  dplyr::select(Hope)

t.test(Hope_3$Hope,Hope_1$Hope,paired = TRUE ,alternative = 'greater')
```

```
##
##  Paired t-test
##
## data:  Hope_3$Hope and Hope_1$Hope
## t = 4.1231, df = 205, p-value = 2.714e-05
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.7912484       Inf
## sample estimates:
## mean of the differences
##                1.320388
```

- test 2: Compare the change of Hope from second to first survey among children who served and doesn't served

```
army_1 <- dataset %>%
  filter(Test == 'A' ,(A4 == 1 | A5 == 1)) %>%
  dplyr::select(Hope)

army_3 <- dataset %>%
  filter(Test == 'B' ,(A4 == 1 | A5 == 1)) %>%
  dplyr::select(Hope)

no_army_1 <- dataset %>%
  filter(Test == 'A', A4 == 0,A5 == 0) %>%
  dplyr::select(Hope)

no_army_3 <- dataset %>%
  filter(Test == 'B', A4 == 0,A5 == 0) %>%
  dplyr::select(Hope)
```

```r
vec1 = army_3$Hope - army_1$Hope
vec2 = no_army_3$Hope - no_army_1$Hope


t.test(vec1,vec2,paired = FALSE ,alternative = 'greater')
```

```
##
##  Welch Two Sample t-test
##
## data:  vec1 and vec2
## t = 0.58233, df = 46.582, p-value = 0.2816
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -0.9431012        Inf
## sample estimates:
## mean of x mean of y
##  2.289017  1.787879
```

Regression models:

Predicting the Hope at third survey

```r
# Selecting the features
reg_data = dataset %>%
  filter(Test == 'A') %>%
  dplyr::select(Religiousness,Permanency,A4,n_of_t,Hope_A,Hope_B,Hope_C)

# Setting seed
set.seed(1234)

# Splitting the data train-test split
split = sample.split(reg_data$Hope_C, SplitRatio = 0.8)
training_set = subset(reg_data, split == TRUE)
test_set = subset(reg_data, split == FALSE)
```

- Model 1 Random forest:

```r
# Create regressor
RF_regressor = randomForest(x = training_set[-7],
                            y = training_set$Hope_C,
                            ntree = 100)
# Predict test result
y_pred = predict(RF_regressor, test_set[-7])

# Calculate MSE
e = test_set[7] - y_pred
e2 = e*e
mse = mean(e2$Hope_C)

# Calculate RSQR
res <- caret::postResample(test_set[7],y_pred)
```

```
rsq <- res[2]

# Print results
print(paste0("Random forest- MSE: ",mse))
```

```
## [1] "Random forest- MSE: 10.5957326288136"
```

```
print(paste0("Random forest- RSQR: ",rsq))
```

```
## [1] "Random forest- RSQR: 0.168024135104851"
```

- Model 2 SVR:

```
# Create regressor
SVR_regressor = svm(formula = Hope_C ~ .,
                    data = training_set,
                    type = 'eps-regression',
                    kernel = 'radial')

# Predict test result
y_pred = predict(SVR_regressor, test_set[-7])

# Calculate MSE
e = test_set[7] - y_pred
e2 = e*e
mse = mean(e2$Hope_C)

# Calculate RSQR
res <- caret::postResample(test_set[7],y_pred)
rsq <- res[2]

# Print results
print(paste0("SVR- MSE: ",mse))
```

```
## [1] "SVR- MSE: 8.77153611098419"
```

```
print(paste0("SVR- RSQR: ",rsq))
```

```
## [1] "SVR- RSQR: 0.210414933211146"
```

- Model 3 Decision tree:

```
# Create regressor
DT_regressor = rpart(formula = Hope_C ~ .,
                     data = training_set,
                     control = rpart.control(minsplit = 15))

# Predict test result
y_pred = predict(DT_regressor, test_set[-7])
```

```
# Calculate MSE
e = test_set[7] - y_pred
e2 = e*e
mse = mean(e2$Hope_C)

# Calculate RSQR
res <- caret::postResample(test_set[7],y_pred)
rsq <- res[2]

# Print results
print(paste0("Decision tree- MSE: ",mse))
```

```
## [1] "Decision tree- MSE: 13.551963276182"
```

```
print(paste0("Decision tree- RSQR: ",rsq))
```

```
## [1] "Decision tree- RSQR: 0.104216387112809"
```

- Model 4 Linear Regression:

```
# Create regressor
LM_regressor = lm(formula = Hope_C ~ .,
                  data = training_set)

# Predict test result
y_pred = predict(LM_regressor, test_set[-7])

# Calculate MSE
e = test_set[7] - y_pred
e2 = e*e
mse = mean(e2$Hope_C)

# Calculate RSQR
res <- caret::postResample(test_set[7],y_pred)
rsq <- res[2]

# Print results
print(paste0("Random forest- MSE: ",mse))
```

```
## [1] "Random forest- MSE: 10.8147509851727"
```

```
print(paste0("Random forest- RSQR: ",rsq))
```

```
## [1] "Random forest- RSQR: 0.141511780134415"
```

- Model 5 Quantile regression:

```r
# Create regressor
Quan_regressor <- rq(Hope_C ~ ., data = training_set)

# Predict test result
y_pred = predict(Quan_regressor, test_set[-7])

# Calculate MSE
e = test_set[7] - y_pred
e2 = e*e
mse = mean(e2$Hope_C)

# Calculate RSQR
res <- caret::postResample(test_set[7],y_pred)
rsq <- res[2]

# Print results
print(paste0("Quantile regression- MSE: ",mse))
```

```
## [1] "Quantile regression- MSE: 11.7101654404599"
```

```r
print(paste0("Quantile regression- RSQR: ",rsq))
```

```
## [1] "Quantile regression- RSQR: 0.0931746232816757"
```

Additinal hypothesis test

```r
H_data = dataset %>%
  filter(Test == 'A') %>%
  mutate(served = ifelse(A4==1|A5==1,1,0)) %>%
  mutate(Hope_A = ifelse(Hope_A>median(dataset$Hope_B),1,0),Hope_B = ifelse(Hope_B>median(dataset$Hope_
  dplyr::select(Religiousness,Permanency,n_of_t,served,Hope_A,Hope_B)

Hope_1 <- H_data %>%
  filter(served == 1) %>%
  dplyr::select(Hope_B)

Hope_3 <- H_data %>%
  filter(served == 0) %>%
  dplyr::select(Hope_B)

t.test(Hope_1$Hope_B,Hope_3$Hope_B,paired = FALSE ,alternative = 'greater')
```

```
##
##  Welch Two Sample t-test
##
## data:  Hope_1$Hope_B and Hope_3$Hope_B
## t = 3.0653, df = 51.429, p-value = 0.00173
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.1135279       Inf
```

```
## sample estimates:
## mean of x mean of y
## 0.4624277 0.2121212
```

Classification models:

Predicting whether the hope of the child will increase or decrease in the second survey

```
# Selecting the features
CL_data = dataset %>%
  filter(Test == 'A') %>%
  mutate(served = ifelse(A4==1|A5==1,1,0)) %>%
  mutate(Hope_A = ifelse(Hope_A>median(dataset$Hope_B),1,0),Hope_B = ifelse(Hope_B>median(dataset$Hope_
  dplyr::select(Religiousness,Permanency,n_of_t,served,Hope_A,Hope_B)

# Factor the binary feature
CL_data$Hope_B = as.factor(CL_data$Hope_B)

# Setting seed
set.seed(123)

# Splitting the data train-test split
training.samples <- CL_data$served %>%
  createDataPartition(p = 0.7, list = FALSE)
train.data  <- CL_data[training.samples, ]
test.data <- CL_data[-training.samples,]

# Create classifier
# Set CV with 5 folds
trainC = trainControl(method = "cv",number = 5,savePredictions = T )
model <- train(Hope_B ~. , data = train.data,method="glm", family = "binomial",trControl = trainC)
summary(model)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6927  -0.9166  -0.6061   1.1057   2.1720
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -5.374511   1.762191  -3.050 0.002289 **
## Religiousness  0.001209   0.167499   0.007 0.994240
## Permanency     0.475337   0.369900   1.285 0.198777
## n_of_t         0.329658   0.168881   1.952 0.050937 .
## served         1.278820   0.615473   2.078 0.037729 *
## Hope_A         1.480076   0.441279   3.354 0.000796 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 195.96  on 144  degrees of freedom
## Residual deviance: 171.99  on 139  degrees of freedom
## AIC: 183.99
##
## Number of Fisher Scoring iterations: 4
```

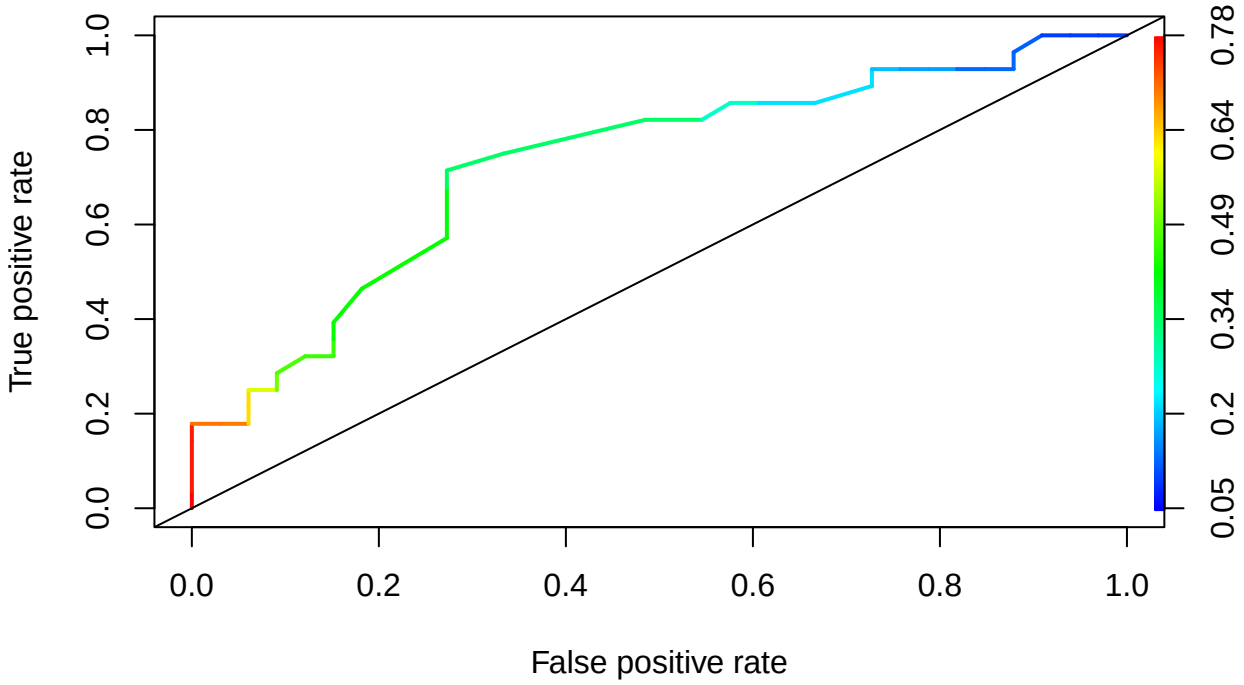
```
model
```

```
## Generalized Linear Model
##
## 145 samples
##   5 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 116, 115, 117, 116, 116
## Resampling results:
##
##   Accuracy   Kappa
##   0.6276519  0.185486
```

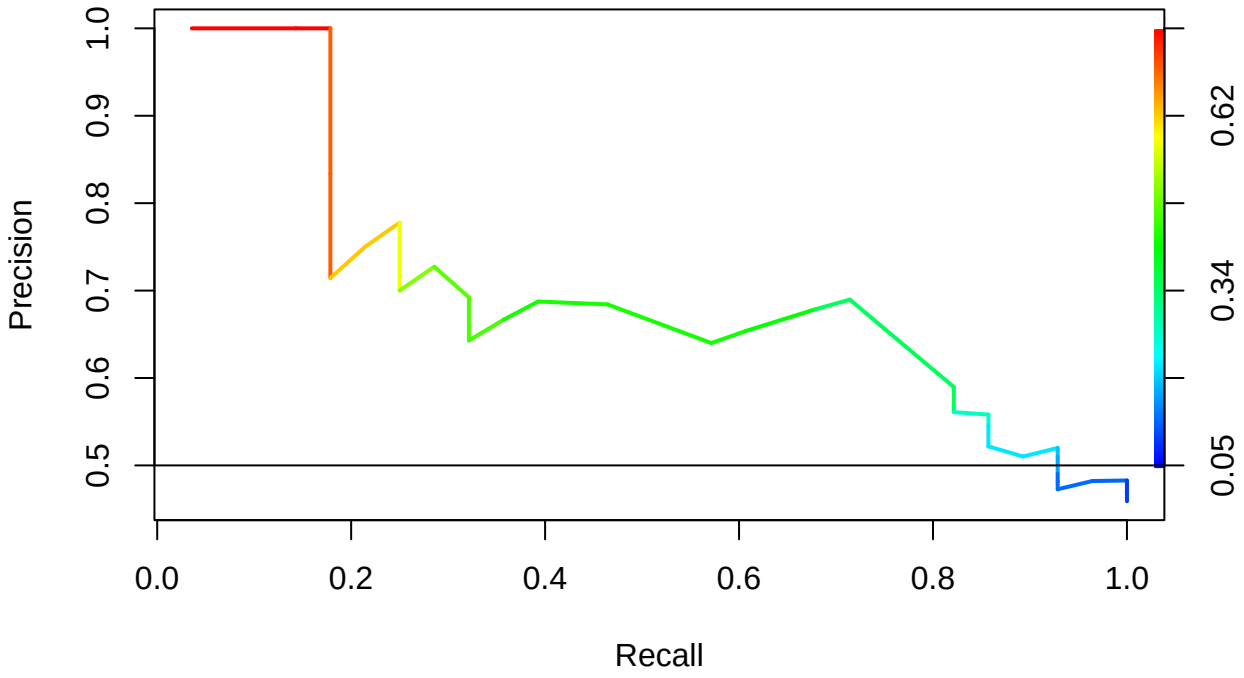
```
probabilities <- model %>% predict(test.data, type = "prob")
```

```
# Predict test result
pred = prediction(probabilities$'1', test.data$Hope_B)

# Create ROC curve
roc = performance(pred,"tpr","fpr")
plot(roc, colorize = T, lwd = 2)
abline(a = 0, b = 1)
```




```
# Create PR curve
pr = performance(pred,"prec","rec")
plot(pr, colorize = T, lwd = 2)
abline(a = 0.5, b=0)
```




```
# Find the best threshold
prediction(probabilities[2], test.data$Hope_B) %>%
  performance(measure = "tpr", x.measure = "fpr") -> result

plotdata <- data.frame(x = result@x.values[[1]],
                       y = result@y.values[[1]],
                       p = result@alpha.values[[1]])

dist_vec <- plotdata$x^2 + (1 - plotdata$y)^2
opt_pos <- which.min(dist_vec)

print(paste0("Best threshold is: p = ",round(plotdata[opt_pos, ]$p, 3)))
```

```
## [1] "Best threshold is: p = 0.343"
```

```
# Create confusion matrix
cm = table(test.data$Hope_B, probabilities$`1` > 0.343)
cm
```

```
##
##     FALSE TRUE
##   0    24    9
##   1     8   20
```