AI Grading Weights

Brandon Shields

12/1/2021

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)

ai\_grader <- read.csv(file = "C:/Users/bshield6/OneDrive - Kent State University/Desktop/Copy of Copy of KSU-Assignment1-10-5-21.csv")  
  
#remove variables not needed for regression analysis  
  
ai\_grader <- ai\_grader[,c(3,5:9)]  
  
#Verify strucure - all variables need to be numeric or integer  
  
print(str(ai\_grader))

## 'data.frame': 228 obs. of 6 variables:  
## $ Score : int 3 4 5 4 5 4 5 3 6 3 ...  
## $ Focus : int 3 4 5 4 5 5 5 3 6 3 ...  
## $ Content : int 2 4 4 4 5 4 4 3 5 3 ...  
## $ Organization: int 3 4 4 4 5 4 4 3 5 3 ...  
## $ Language : int 3 5 4 5 5 4 5 3 6 3 ...  
## $ Mechanics : int 3 4 3 4 5 4 4 3 5 3 ...  
## NULL

ai\_grader <- as.data.frame(sapply(ai\_grader, as.numeric))

#remove missing values  
  
ai\_grader <- ai\_grader[-175,]  
  
#create data partition for testing and training  
trainIndex <- createDataPartition(ai\_grader$Score, p = .8, list = FALSE)  
data\_train <- ai\_grader[trainIndex,]  
data\_test <- ai\_grader[-trainIndex,]

#Train Model using linear regression. Method is Cross Validation with 5 folds.  
  
model\_lm <- train(Score ~ Focus + Content + Language + Organization + Language +  
 Mechanics,  
 data = data\_train,  
 trControl = trainControl(method = "CV", number = 5),  
 method = "lm")

# Review Model coeefficients and model summary to determine variables that are statistically significant,   
print(model\_lm$finalModel)

##   
## Call:  
## lm(formula = .outcome ~ ., data = dat)  
##   
## Coefficients:  
## (Intercept) Focus Content Language Organization   
## -0.1381928 0.8445057 0.0000565 0.1508499 0.0130819   
## Mechanics   
## 0.0242753

print(summary(model\_lm))

##   
## Call:  
## lm(formula = .outcome ~ ., data = dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.83739 -0.02565 0.00712 0.02025 0.85627   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.1381929 0.0652844 -2.117 0.0357 \*   
## Focus 0.8445057 0.0311413 27.119 < 2e-16 \*\*\*  
## Content 0.0000565 0.0407428 0.001 0.9989   
## Language 0.1508499 0.0333702 4.520 1.13e-05 \*\*\*  
## Organization 0.0130819 0.0392663 0.333 0.7394   
## Mechanics 0.0242753 0.0326356 0.744 0.4580   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1674 on 177 degrees of freedom  
## Multiple R-squared: 0.9712, Adjusted R-squared: 0.9704   
## F-statistic: 1193 on 5 and 177 DF, p-value: < 2.2e-16

#Review results on test data set to see how they compare  
results <- predict(model\_lm, newdata = data\_test, type = "raw")  
  
#results will need to be rounded since responses can only be integers.  
results\_rounded <- round(results, digits = 0)