CIND119 - Group Project

10/04/2021

## Memebers

Patrick Litte, [patrick.little@ryerson.ca](mailto:patrick.little@ryerson.ca)

Manjola Chiappetta, [m1chiappetta@ryerson.ca](mailto:m1chiappetta@ryerson.ca)

## Summary

Our client for this project is a Portuguese bank that is looking for improvements to their telemarketing strategy. This bank is marketing long-term deposit accounts such as bonds and savings account to a large variety of existing clients with a wide range of attributes. Our role in the project was to create a variety of predictive models which could be used to predict whether a client will subscribe to a term deposit or not, based on the bank’s past experience in marketing these products. This predictive model would enable the client to target their marketing activities more efficiently to clients with a greater likelihood of subscribing to a term deposit.

In this project we compared two different predictive models, decision tree and Naïve Bayes and compared their accuracy as well as their Area under the Receiver Operator Curve. For our extra credit, we compared these two more simple models against a more advanced Gradient Boosting model (XGBoost) to determine if a more advanced model could out preform the simpler models.

### Tools

This project will leverage R for the exploratory data analysis, data preprocessing, predictive modeling, and model evaluation. We will use the tidymodels framework to implement our predictive modelling, and model evaluation, and we will use the DataExplorer package to aid in exploratory data analysis.

## Workload Distribution

|  |  |
| --- | --- |
| Member Name | List of Tasks Preformed |
| Patrick Little | - some tasks |
| Manjola Chiappetta | - some tasks |

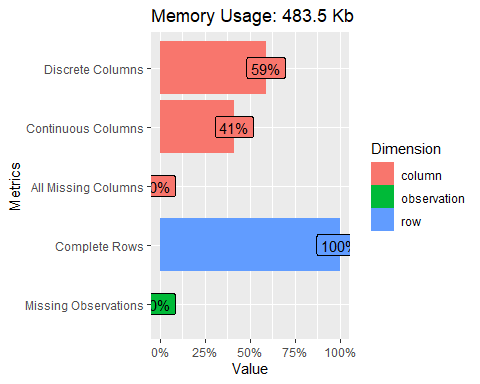
## Exploratory Data Analysis

In this section we will:   
- Look at the attribute types in the dataset   
- Find and missing values   
- Find max,min,mean and standard deviation of the atttributes   
- Determine any outlier values for the attributes under consideration   
- Analyze the distribution of numeric attributes

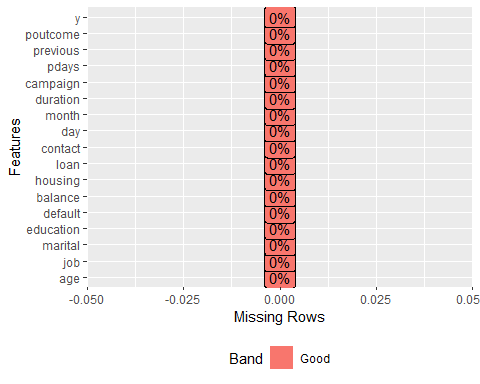
bank<-read.csv("https://raw.githubusercontent.com/PatLittle/CIND119-group-project/main/bank\_marketing/bank.csv")  
  
introduce(bank)

## rows columns discrete\_columns continuous\_columns all\_missing\_columns  
## 1 4521 17 10 7 0  
## total\_missing\_values complete\_rows total\_observations memory\_usage  
## 1 0 4521 76857 495152

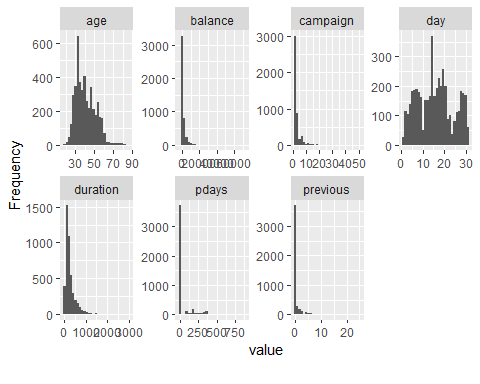
plot\_intro(bank)



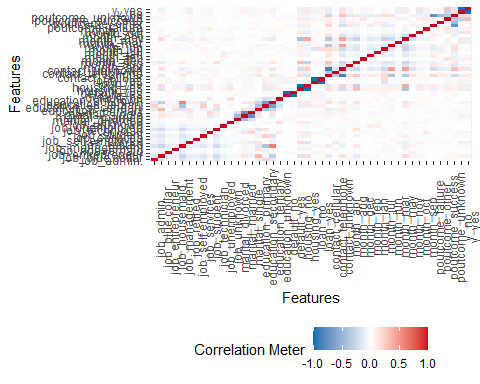
plot\_missing(bank)



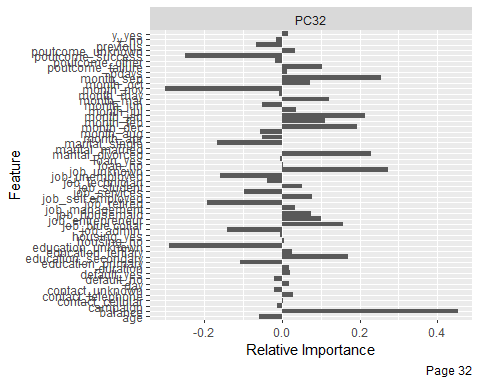
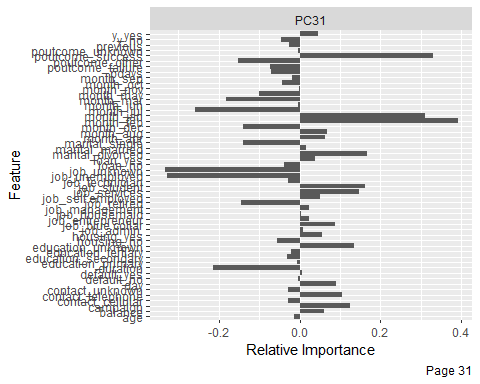
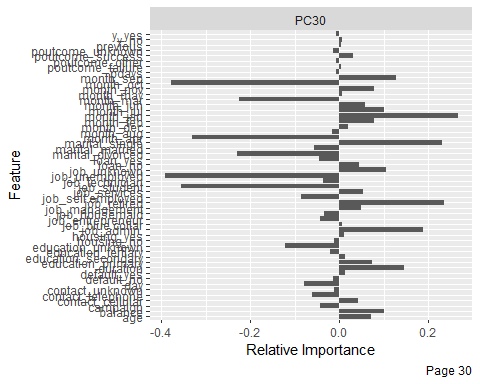
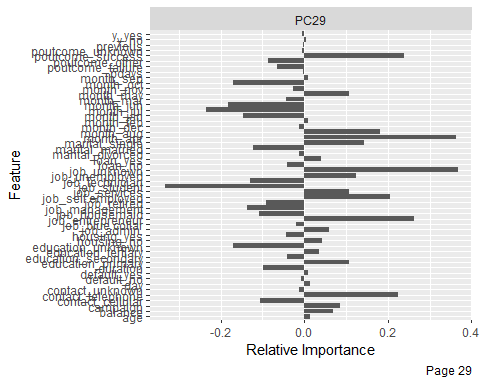
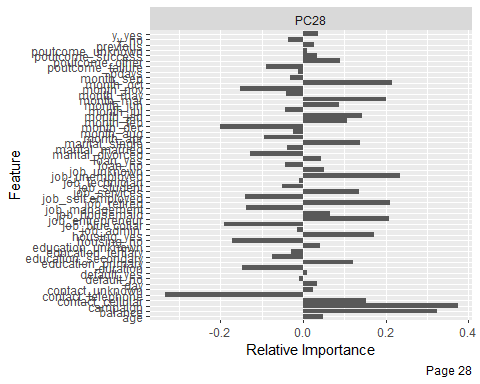
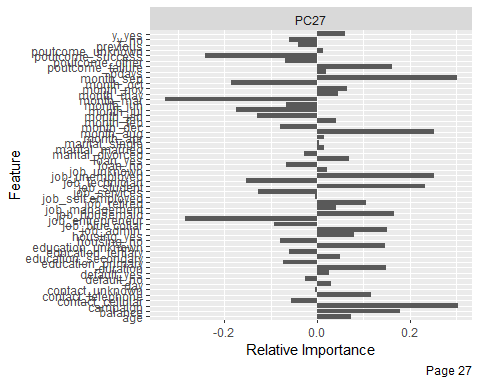
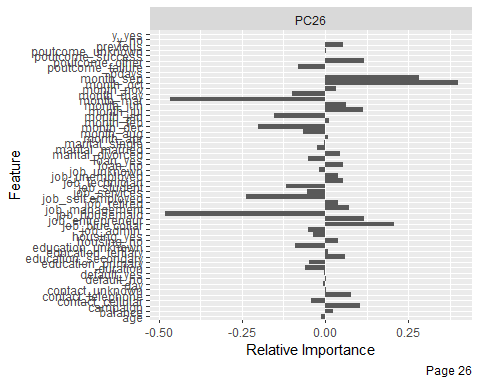
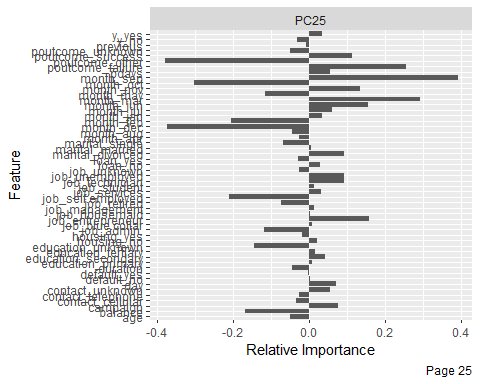
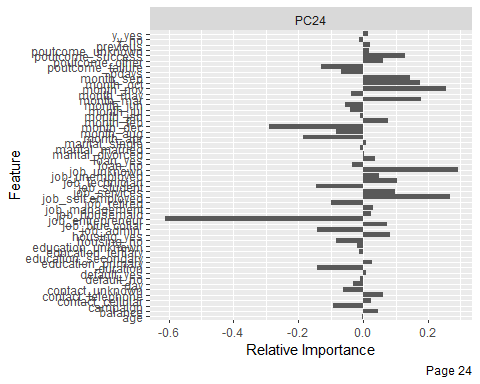
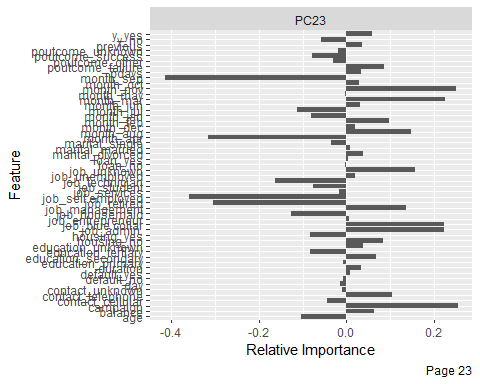
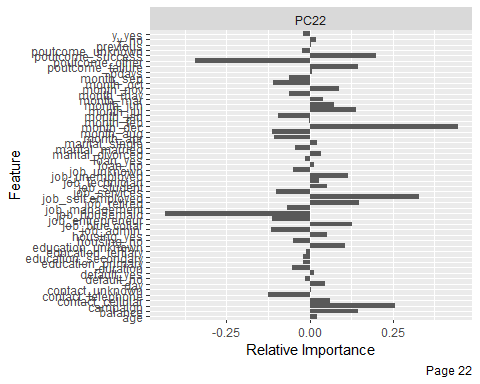
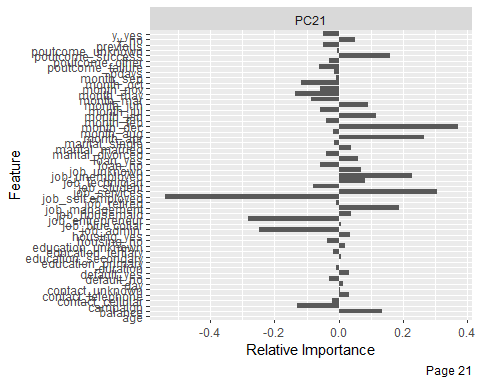
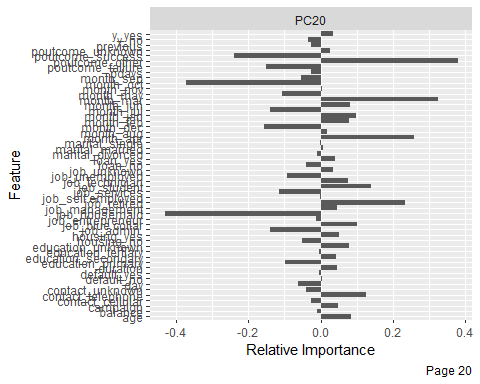
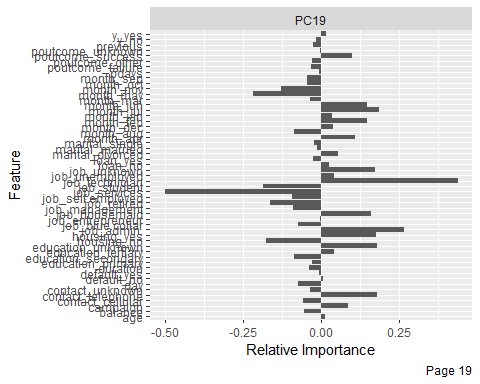
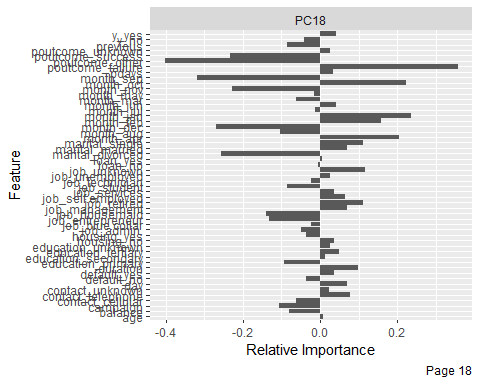
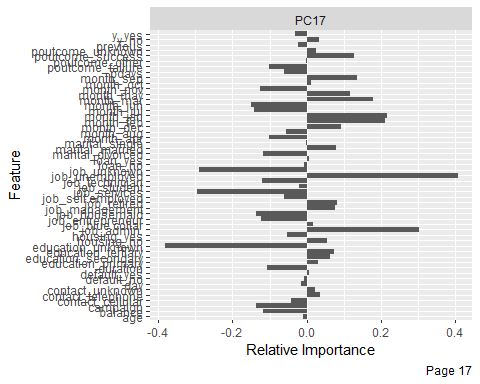
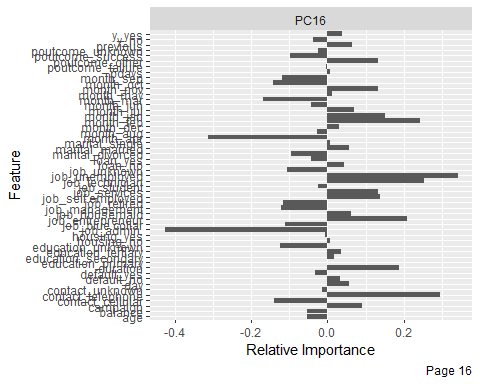
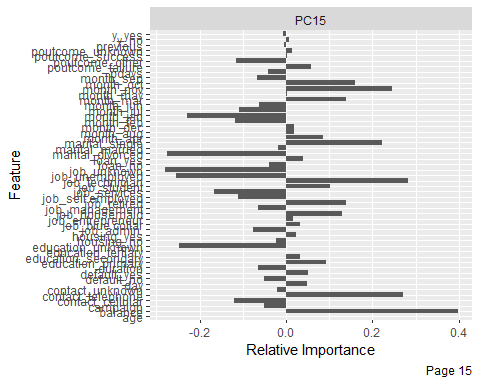
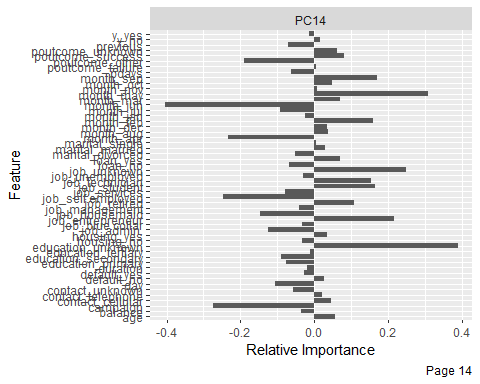
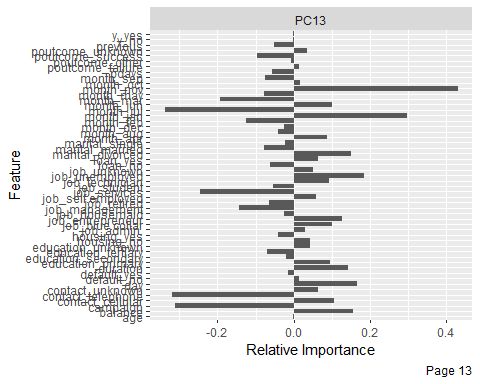
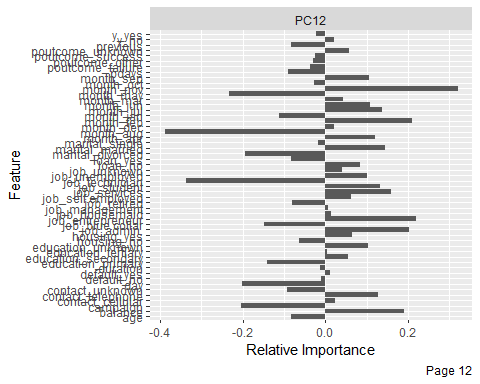
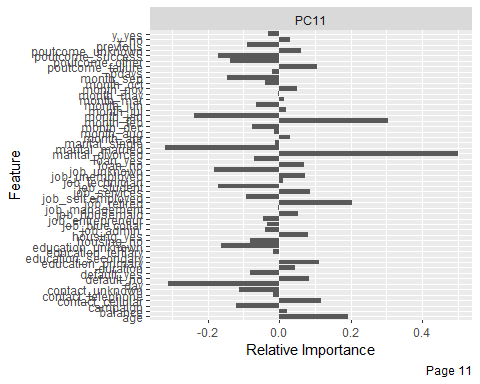
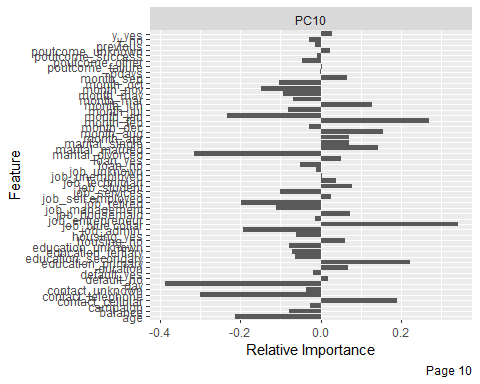
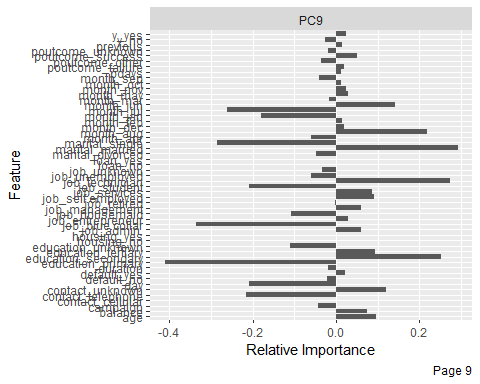
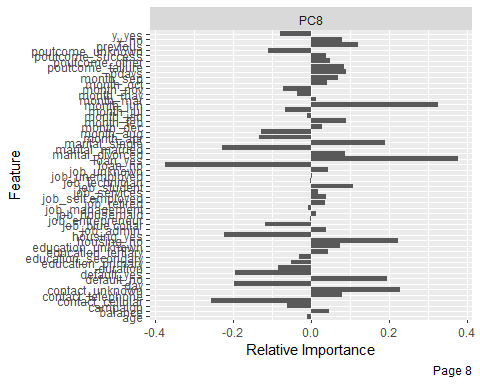
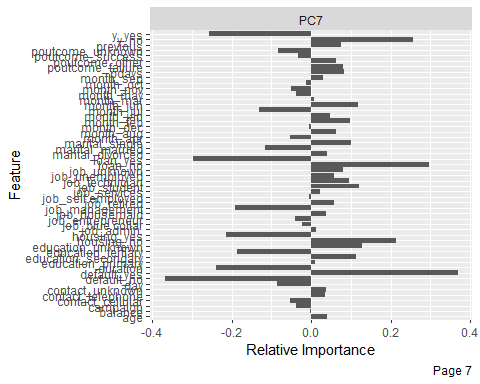
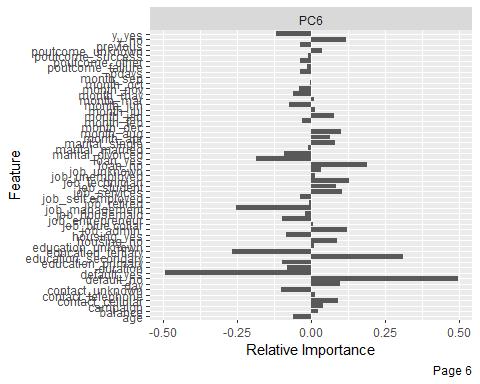
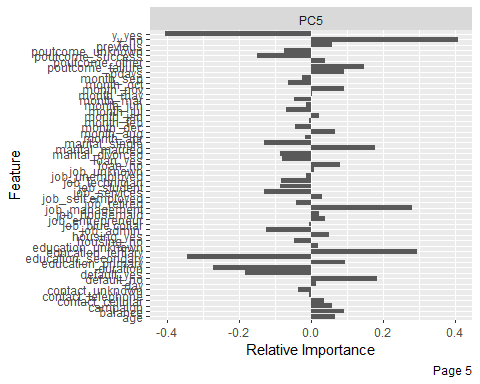
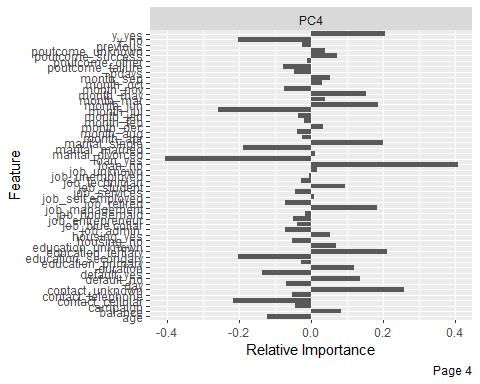
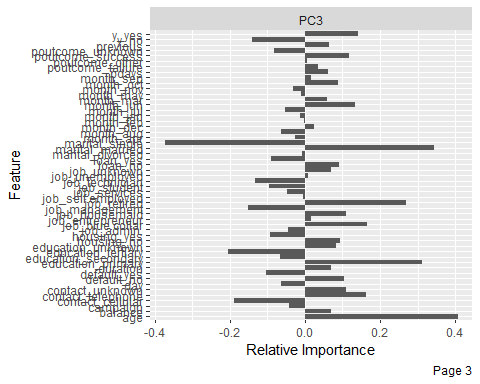
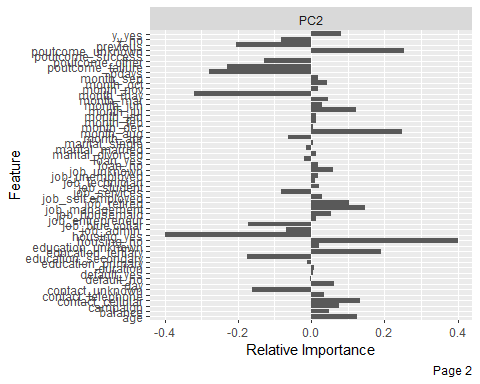
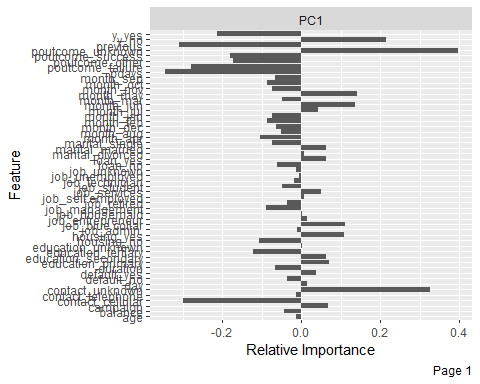
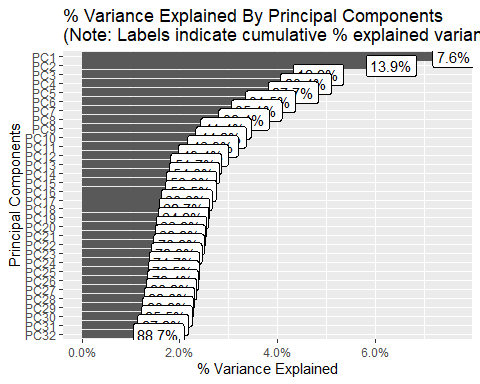
#plot\_bar(bank, by = "y")  
plot\_histogram(bank)



plot\_correlation(na.omit(bank), type = "d")



plot\_prcomp(bank, variance\_cap = 0.9, ncol =1L, nrow=1L)



str(bank)

## 'data.frame': 4521 obs. of 17 variables:  
## $ age : int 30 33 35 30 59 35 36 39 41 43 ...  
## $ job : chr "unemployed" "services" "management" "management" ...  
## $ marital : chr "married" "married" "single" "married" ...  
## $ education: chr "primary" "secondary" "tertiary" "tertiary" ...  
## $ default : chr "no" "no" "no" "no" ...  
## $ balance : int 1787 4789 1350 1476 0 747 307 147 221 -88 ...  
## $ housing : chr "no" "yes" "yes" "yes" ...  
## $ loan : chr "no" "yes" "no" "yes" ...  
## $ contact : chr "cellular" "cellular" "cellular" "unknown" ...  
## $ day : int 19 11 16 3 5 23 14 6 14 17 ...  
## $ month : chr "oct" "may" "apr" "jun" ...  
## $ duration : int 79 220 185 199 226 141 341 151 57 313 ...  
## $ campaign : int 1 1 1 4 1 2 1 2 2 1 ...  
## $ pdays : int -1 339 330 -1 -1 176 330 -1 -1 147 ...  
## $ previous : int 0 4 1 0 0 3 2 0 0 2 ...  
## $ poutcome : chr "unknown" "failure" "failure" "unknown" ...  
## $ y : chr "no" "no" "no" "no" ...

colSums(is.na(bank))

## age job marital education default balance housing loan   
## 0 0 0 0 0 0 0 0   
## contact day month duration campaign pdays previous poutcome   
## 0 0 0 0 0 0 0 0   
## y   
## 0

## Predictive Modeling / Classification

### Decision Tree

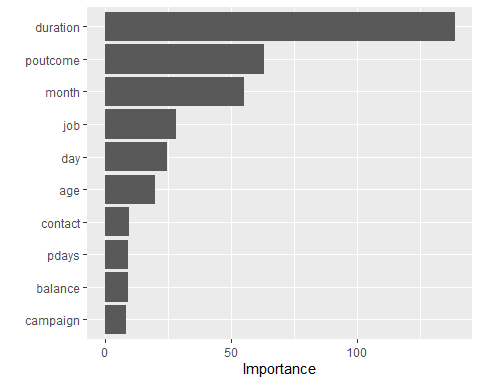
###Decision Tree  
  
  
  
bank\_clean<- bank %>% mutate\_if(is.character, factor)  
  
set.seed(888)  
bank\_split <- initial\_split(bank\_clean, prop = 0.75,   
 strata = y)  
  
bank\_training <- bank\_split %>% training()  
bank\_test <- bank\_split %>% testing()  
bank\_folds <- vfold\_cv(bank\_training, v = 10)  
  
  
  
bank\_recipe <- recipe(y ~ ., data = bank\_training)   
   
  
  
bank\_clean\_baked<-bank\_recipe %>%   
 prep() %>%   
 bake(new\_data = bank\_training)  
  
tree\_model <- decision\_tree(cost\_complexity = tune(),  
 tree\_depth = tune(),  
 min\_n = tune()) %>%   
 set\_engine('rpart') %>%   
 set\_mode('classification')  
  
tree\_workflow <- workflow() %>%   
 add\_model(tree\_model) %>%   
 add\_recipe(bank\_recipe)  
  
tree\_grid <- grid\_latin\_hypercube(cost\_complexity(),  
 tree\_depth(),  
 min\_n(),   
 size = 60)  
  
set.seed(888)  
  
tree\_tuning <- tree\_workflow %>%   
 tune\_grid(resamples = bank\_folds,  
 grid = tree\_grid)

## Warning: package 'vctrs' was built under R version 4.0.5

tree\_tuning %>% show\_best('roc\_auc')

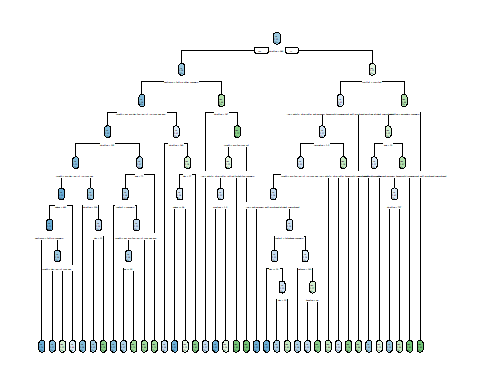
## # A tibble: 5 x 9  
## cost\_complexity tree\_depth min\_n .metric .estimator mean n std\_err  
## <dbl> <int> <int> <chr> <chr> <dbl> <int> <dbl>  
## 1 0.000000000688 10 20 roc\_auc binary 0.863 10 0.00653  
## 2 0.00000448 9 28 roc\_auc binary 0.862 10 0.00615  
## 3 0.00000322 9 27 roc\_auc binary 0.862 10 0.00623  
## 4 0.00000000641 11 24 roc\_auc binary 0.860 10 0.00646  
## 5 0.00000874 11 23 roc\_auc binary 0.860 10 0.00644  
## # ... with 1 more variable: .config <chr>

best\_tree <- tree\_tuning %>%   
 select\_best(metric = 'roc\_auc')  
  
  
final\_tree\_workflow <- tree\_workflow %>%   
 finalize\_workflow(best\_tree)  
  
  
tree\_wf\_fit <- final\_tree\_workflow %>%   
 fit(data = bank\_training)  
  
tree\_fit <- tree\_wf\_fit %>%   
 pull\_workflow\_fit()  
  
vip(tree\_fit)



rpart.plot(tree\_fit$fit, roundint = FALSE)

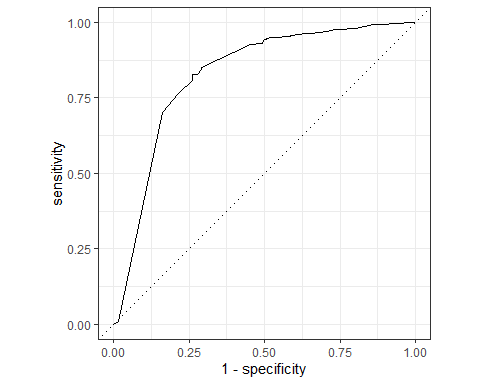
## Warning: labs do not fit even at cex 0.15, there may be some overplotting



tree\_last\_fit <- final\_tree\_workflow %>%   
 last\_fit(bank\_split)  
  
tree\_last\_fit %>% collect\_metrics()

## # A tibble: 2 x 4  
## .metric .estimator .estimate .config   
## <chr> <chr> <dbl> <chr>   
## 1 accuracy binary 0.892 Preprocessor1\_Model1  
## 2 roc\_auc binary 0.827 Preprocessor1\_Model1

tree\_last\_fit %>% collect\_predictions() %>%   
 roc\_curve(truth = y, estimate = .pred\_no) %>%   
 autoplot()



tree\_predictions <- tree\_last\_fit %>% collect\_predictions()  
  
conf\_mat(tree\_predictions, truth = y, estimate = .pred\_class)

## Truth  
## Prediction no yes  
## no 949 71  
## yes 51 59

predict(tree\_last\_fit$.workflow[[1]],bank\_test[15,])

## # A tibble: 1 x 1  
## .pred\_class  
## <fct>   
## 1 no

saveRDS(tree\_last\_fit$.workflow[[1]],"./saved\_model.Rds")  
  
trained\_model<-readRDS("saved\_model.Rds")

### Naive Bayes

set.seed(888)  
nb\_split <- initial\_split(bank\_clean, prop = 0.75,   
 strata = y)  
  
nb\_training <- nb\_split %>% training()  
nb\_test <- nb\_split %>% testing()  
nb\_folds <- vfold\_cv(nb\_training, v = 10)  
  
nb\_recipe <- recipe(y ~ ., data = nb\_training)  
   
  
  
nb\_wf <- workflow() %>%  
 add\_recipe(nb\_recipe)  
  
library(discrim)

##   
## Attaching package: 'discrim'

## The following object is masked from 'package:dials':  
##   
## smoothness

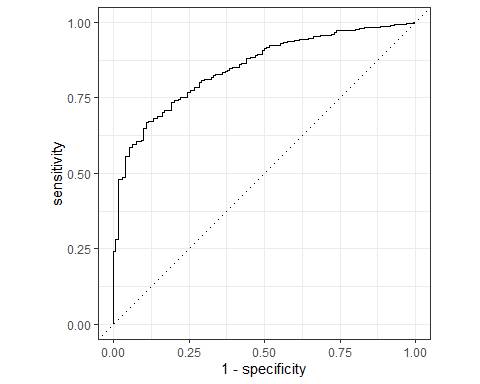
nb\_spec <- naive\_Bayes() %>%  
 set\_mode("classification") %>%  
 set\_engine("naivebayes")  
  
nb\_spec

## Naive Bayes Model Specification (classification)  
##   
## Computational engine: naivebayes

nb\_fit <- nb\_wf %>%  
 add\_model(nb\_spec) %>%  
 fit(data = nb\_training)  
  
nb\_wf\_final <- workflow() %>%  
 add\_recipe(nb\_recipe) %>%  
 add\_model(nb\_spec)  
  
nb\_rs <- fit\_resamples(  
 nb\_wf\_final,  
 nb\_folds,  
 control = control\_resamples(save\_pred = TRUE)  
)  
  
  
nb\_last\_fit <- nb\_wf\_final %>%   
 last\_fit(nb\_split)  
  
nb\_last\_fit %>% collect\_metrics()

## # A tibble: 2 x 4  
## .metric .estimator .estimate .config   
## <chr> <chr> <dbl> <chr>   
## 1 accuracy binary 0.881 Preprocessor1\_Model1  
## 2 roc\_auc binary 0.849 Preprocessor1\_Model1

nb\_last\_fit %>% collect\_predictions() %>%   
 roc\_curve(truth = y, estimate = .pred\_no) %>%   
 autoplot()



nb\_predictions <- nb\_last\_fit %>% collect\_predictions()  
conf\_mat(nb\_predictions, truth = y, estimate = .pred\_class)

## Truth  
## Prediction no yes  
## no 956 91  
## yes 44 39

## Conclusions and Recommendations

Some text wrapping up the report