Project Part C: Classification

In [57]:
```
analyst = "Khoa Nguyen" # Replace this with your name
```

In [58]:
```
f = "setup.R"; for (i in 1:10) { if (file.exists(f)) break else f = paste0("../", f) }; source(f)
options(repr.matrix.max.rows=674)
```

1 Introduction

1.1 Objective

Build, evaluate, and tune a classifier trained on a transformed dataset about public company fundamentals. Later, use the classifier along with additional analysis to recommend a portfolio of 12 company investments that maximizes 12-month return of an overall

1.2 Approach

Retrieve a dataset ready for predictive model construction.

Build a model to predict whether stock price will grow more than 30% over 12 months, given 12 months of past company fundamentals data, using a machine learning model construction method.

Evaluate and tune the model for optimal business performance.

2 Business Model & Business Parameters

The business model is ...

$$\text{profit} = \left(\sum_{i \in \text{portfolio}} (1 + \text{growth}_i) \times \text{allocation}_i \right) - \text{budget}$$

$$\text{profit rate} = \text{profit} \div \text{budget}$$

$$\text{budget} = \sum_{i \in \text{portfolio}} \text{allocation}_i$$

Business parameters include ...

- budget is total investment to allocate across the companies in the portfolio
- portfolio size is number of companies in the portfolio
- allocation is vector of amounts to allocate to specific companies in the portfolio, must sum to budget
- threshold is growth that qualifies as lowest attractive growth

In [59]:
```
# Set the business parameters.

budget = 1000000
portfolio_size = 12
allocation = rep(budget/portfolio_size, portfolio_size)

fmtsx(fmt(budget), fmt(portfolio_size), fmt(allocation))
```

budget	portfolio_size	allocation
1,000,000	12	83,333
		83,333
		83,333
		83,333
		83,333
		83,333
		83,333
		83,333
		83,333
		83,333
		83,333
		83,333

Portfolio to be filled with companies predicted to have the highest probabilities of growing above 30%.

3 Data

In [60]:

```
# Retrieve data.
# How many observations and variables?
# Present the first few observations.

data = read.csv("My Data.csv", header=TRUE, na.strings=c("NA", ""), stringsAsFactors=FALSE)
data$big_growth = factor(data$big_growth, levels=c("YES","NO"))

fmtx(size(data))
fmtx(data[1:6,], FFO)
```

size(data)

observations	variables
4,305	9

data (first few observations)

big_growth	growth	prccq	gvkey	tic	conm	PC1	PC2	PC3
NO	0.0507	43.69	1,004	AIR	AAR CORP	1.4098	0.2125	-0.1874
NO	-0.3829	32.11	1,045	AAL	AMERICAN AIRLINES GROUP INC	-2.8093	0.2246	1.4366
YES	0.3158	6.75	1,050	CECE	CECO ENVIRONMENTAL CORP	1.5247	0.4396	-0.1679
NO	-0.2165	8.66	1,062	ASA	ASA GOLD AND PRECIOUS METALS	1.5737	0.6384	0.0123
NO	-0.1185	15.25	1,072	AVX	AVX CORP	1.2813	0.4529	0.0929
NO	0.0002	85.20	1,075	PNW	PINNACLE WEST CAPITAL CORP	0.3698	-0.4861	-0.0128

4 Build Classification Model

In [61]:
```
model = naiveBayes(big_growth ~ PC1 + PC2 + PC3, data)
model
```

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
    YES      NO
0.08362 0.91638

Conditional probabilities:
     PC1
Y       [,1]  [,2]
  YES  1.129 1.332
  NO  -0.103 5.674

     PC2
Y         [,1]  [,2]
  YES  0.24124 0.898
  NO  -0.02201 4.793

     PC3
Y          [,1]  [,2]
  YES -0.014239 0.677
  NO   0.001299 3.653
```

5 Evaluate Classification Model (5-fold cross-validation)

In [62]:
```
set.seed(0)
fold = createFolds(data$big_growth, k=5)
str(fold)
```

```
List of 5
 $ Fold1: int [1:861] 9 13 17 19 31 42 44 54 60 66 ...
 $ Fold2: int [1:861] 1 2 6 11 16 25 32 49 55 59 ...
 $ Fold3: int [1:861] 4 8 14 22 28 34 40 45 50 52 ...
 $ Fold4: int [1:861] 3 5 15 18 21 24 26 27 30 36 ...
 $ Fold5: int [1:861] 7 10 12 20 23 29 33 35 37 46 ...
```

In [63]:
```
# accuracy = c()
fold_performance = data.frame()

for (i in 1:5)
  { data.test  = data[fold[[i]],]
    data.train = data[setdiff(1:nrow(data), fold[[i]]),]
    model_train = naiveBayes(big_growth ~ PC1 + PC2 + PC3, data.train)
    prob = predict(model_train, data.test, type="raw")  # use predict arguments appropriate for your model
    class.predicted = as.class(prob, class="YES", cutoff=0.5)
    CM = confusionMatrix(class.predicted, data.test$big_growth)$table
    cm = CM/sum(CM)
    accuracy = cm[1,1]+cm[2,2]

    data.test$class.predicted = class.predicted
    data.test$prob = prob[,1]
    data.test = data.test[data.test$class.predicted == "YES",]
    data.sorted = data.test[order(data.test$prob, decreasing=TRUE),]
    company.data.growth = data.sorted[1:12, "growth"]
    profit = sum((1 + company.data.growth)*allocation) - budget
   fold_performance = rbind(fold_performance,data.frame(fold=i,accuracy=accuracy,profit = profit))}

fmtx(fold_performance,"Fold Performance")
```

Fold Performance

fold	accuracy	profit
1	0.2323	-144,476
2	0.2230	-114,764
3	0.2334	-22,672
4	0.2033	4,896
5	0.2021	-119,455

In [64]:
```
accuracy.cv = mean(fold_performance$accuracy)
profit.cv = mean(fold_performance$profit)
profit_rate.cv = profit.cv/budget
fmtx(data.frame(accuracy.cv, profit.cv, profit_rate.cv), "5-Fold Cross-Validation Estimated Performance")
```

5-Fold Cross-Validation Estimated Performance

accuracy.cv	profit.cv	profit_rate.cv
0.2188	-79,294	-0.0793

6 Tune Classification Model

In [65]:
```
tune = data.frame()
for (f in exhaustive(names(data[,c("PC1","PC2","PC3")]), keep="big_growth")) # try every combination of varia
for (q in c(0.25, 0.33, 0.50))  # try several values for cutoff
{
    nfold = 5
    set.seed(0)
    fold = createFolds(data$big_growth, k=nfold)
    accuracy = c()
    profit = c()
    for (i in 1:nfold) { data.train = data[setdiff(1:nrow(data), fold[[i]]),]
                         data.test  = data[fold[[i]],]
                         model = naiveBayes(big_growth ~ ., data.train[,f], laplace=TRUE)
                         prob = predict(model, data.test, type="raw")
                         class.predicted = as.class(prob, class="YES", cutoff=q)
                         CM = confusionMatrix(class.predicted, data.test$big_growth)$table
                         cm = CM/sum(CM)
                         accuracy[i] = cm[1,1]+cm[2,2]

                        data.test$class.predicted = class.predicted
                        data.test$prob = prob[,1]
                        data.sorted = data.test[order(data.test$prob, decreasing=TRUE),]
                        company.data.growth = data.sorted[1:12, "growth"]
                        profit[i] = sum((1 + company.data.growth)*allocation) - budget }

    accuracy.cv = mean(accuracy)
    profit.cv = mean(profit)
    profit_rate.cv = profit.cv/budget

    tune = rbind(tune, data.frame(method="naive bayes", variables=vector2string(f), cutoff = q,
                                  accuracy.cv, profit.cv, profit_rate.cv))}
best = tune[which.max(tune$profit.cv),]
fmtx(best, "best model")

fmtx(tune, "search for best model")
```

best model

method	variables	cutoff	accuracy.cv	profit.cv	profit_rate.cv
naive bayes	PC3, big_growth	0.25	0.2049	47,492	0.0475

search for best model

method	variables	cutoff	accuracy.cv	profit.cv	profit_rate.cv
naive bayes	PC1, big_growth	0.25	0.2987	-85,297	-0.0853
naive bayes	PC1, big_growth	0.33	0.8035	-85,297	-0.0853
naive bayes	PC1, big_growth	0.50	0.9134	-85,297	-0.0853
naive bayes	PC2, big_growth	0.25	0.3554	-146,897	-0.1469
naive bayes	PC2, big_growth	0.33	0.7022	-146,897	-0.1469
naive bayes	PC2, big_growth	0.50	0.9157	-146,897	-0.1469
naive bayes	PC3, big_growth	0.25	0.2049	47,492	0.0475
naive bayes	PC3, big_growth	0.33	0.7823	47,492	0.0475
naive bayes	PC3, big_growth	0.50	0.9122	47,492	0.0475
naive bayes	PC1, PC2, big_growth	0.25	0.2197	-142,451	-0.1425
naive bayes	PC1, PC2, big_growth	0.33	0.2355	-142,451	-0.1425
naive bayes	PC1, PC2, big_growth	0.50	0.3954	-142,451	-0.1425
naive bayes	PC1, PC3, big_growth	0.25	0.2007	-116,848	-0.1168
naive bayes	PC1, PC3, big_growth	0.33	0.2107	-116,848	-0.1168
naive bayes	PC1, PC3, big_growth	0.50	0.2381	-116,848	-0.1168
naive bayes	PC2, PC3, big_growth	0.25	0.1823	-112,989	-0.1130
naive bayes	PC2, PC3, big_growth	0.33	0.1954	-112,989	-0.1130
naive bayes	PC2, PC3, big_growth	0.50	0.2383	-112,989	-0.1130
naive bayes	PC1, PC2, PC3, big_growth	0.25	0.1991	-79,294	-0.0793
naive bayes	PC1, PC2, PC3, big_growth	0.33	0.2046	-79,294	-0.0793
naive bayes	PC1, PC2, PC3, big_growth	0.50	0.2188	-79,294	-0.0793

 Document revised May 6, 2023