Project Part D: Regression

In [1]:
```
analyst = "Khoa Nguyen" # Replace this with your name
```

In [2]:
```
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 regressor trained on a transformed dataset about public company fundamentals. Later, use the regressor along with additional analysis to recommend a portfolio of 12 company investments that maximizes 12-month return of an overall $1,000,000 investment.

1.2 Approach

Retrieve a dataset ready for predictive model construction.

Build a model to predict how much stock price will grow 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 [3]:
```
# 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 growths.

3 Data

In [5]:
```
# 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 Regression Model

In [6]:
```
# Construct a linear regression model to predict growth given PC1, PC2, and PC3.
# Present a brief summary of the model parameters.
model = lm(growth ~ PC1 + PC2 + PC3, data)
model
```

```
Call:
lm(formula = growth ~ PC1 + PC2 + PC3, data = data)

Coefficients:
(Intercept)          PC1          PC2          PC3
   -0.11859      0.00109     -0.00169     -0.00179
```

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

In [7]:
```
# Partition the data into 5 folds (use set.seed(0) and createFolds(...) based on growth).
# Present the first few observation numbers for each fold.
#
# You can use str(...)
set.seed(0)
fold = createFolds(data$growth, k=5)
str(fold)
```

```
List of 5
 $ Fold1: int [1:862] 8 11 16 22 30 32 38 40 41 44 ...
 $ Fold2: int [1:860] 3 9 10 23 26 27 34 39 52 64 ...
 $ Fold3: int [1:862] 2 7 19 29 35 42 53 57 61 62 ...
 $ Fold4: int [1:861] 1 4 5 6 15 17 28 33 36 43 ...
 $ Fold5: int [1:860] 12 13 14 18 20 21 24 25 31 37 ...
```

In [17]:
```
# Present the model's estimated RMSE and profit for each fold.
# Note that profit is calculated based on actual growth of the 12 companies with predicted highest growth. b
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 = lm(growth ~ PC1 + PC2 + PC3, data.train)
    growth.predicted = predict(model_train, data.test)
    error = data.test$growth - growth.predicted
    rmse = sqrt(mean(error^2))

    data.test = cbind(data.test,growth.predicted)
    data.test = data.test[order(-data.test$growth.predicted),]
    company.data.growth = data.test[1:12, "growth"]
    profit = sum((1 + company.data.growth)*allocation) - budget
   fold_performance = rbind(fold_performance,data.frame(fold=i,rmse=rmse,profit = profit))}

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

Fold Performance

fold	rmse	profit
1	0.4445	-112,168
2	0.4359	-159,110
3	0.5040	-68,571
4	0.3991	-81,949
5	0.5459	-14,433

In [19]:
```
# Present the model's 5-fold cross-validation estimated RMSE, profit, and profit rate.
rmse.cv = mean(fold_performance$rmse)
profit.cv = mean(fold_performance$profit)
profit_rate.cv = profit.cv/budget
fmtx(data.frame(rmse.cv, profit.cv, profit_rate.cv), "5-Fold Cross-Validation Estimated Performance")
```

5-Fold Cross-Validation Estimated Performance

rmse.cv	profit.cv	profit_rate.cv
0.4659	-87,246	-0.0872

6 Tune Regression Model

In [27]:
```
# Partition the data into 5 folds (use set.seed(0) and createFolds(...) based on growth).

# Build several linear regression models to predict growth.
# Iterate through unique combinations of predictor variables, chosen from PC1, PC2, PC3.

# Estimate each model's RMSE and profit, using 5-fold cross validation.

# Present the best model: chosen variables, RMSE, profit, and profit rate.
# Present all the models: chosen variables, RMSE, profit, and profit rate.

tune = data.frame()
for (f in exhaustive(names(data[,c("PC1","PC2","PC3")]), keep="growth")) # try every combination of variables
{
    nfold = 5
    set.seed(0)
    fold = createFolds(data$growth, k=nfold)
    rmse = c()
    profit = c()
    for (i in 1:nfold) {
            data.test  = data[fold[[i]],]
            data.train = data[setdiff(1:nrow(data), fold[[i]]),]
            model_train = lm(growth ~ ., data.train[,f])
            growth.predicted = predict(model_train, data.test)
            error = data.test$growth - growth.predicted
            rmse[i] = sqrt(mean(error^2))

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

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

    tune = rbind(tune, data.frame(method="linear regression", variables=vector2string(f),
                                  rmse.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	rmse.cv	profit.cv	profit_rate.cv
linear regression	PC1, PC2, growth	0.4659	-51,470	-0.0515

search for best model

method	variables	rmse.cv	profit.cv	profit_rate.cv
linear regression	PC1, growth	0.4659	-288,146	-0.2881
linear regression	PC2, growth	0.4659	-70,483	-0.0705
linear regression	PC3, growth	0.4660	-111,428	-0.1114
linear regression	PC1, PC2, growth	0.4659	-51,470	-0.0515
linear regression	PC1, PC3, growth	0.4659	-75,214	-0.0752
linear regression	PC2, PC3, growth	0.4659	-93,628	-0.0936
linear regression	PC1, PC2, PC3, growth	0.4659	-87,246	-0.0872

Document revised May 6, 2023