Ⅰ Data Preperation

1. Fetch all data
2. Sample by Volume
3. Filter by Sharpe Ratio
4. Make Indicators

Ⅱ Modeling

2. Choose Indicators
3. Make Models (for each security)
4. Choose Best Model (in each security)
5. Choose Best Models (among securities)

Ⅲ Portfolio set & test

1. Make Portfolios
2. Make signals
3. Back Testing

I Data Preperation

1. Fetch all data
2. Sample by Volume
3. Filter by Sharpe Ratio
4. Make Indicators

1. Fetch all Securities' data

```r
tickers <- read.csv("SP500Constituent.csv")$Ticker.symbol
tickers <- as.character(tickers)

all.dat <- sapply(tickers, function(x) {
  tryCatch(getSymbols(x,auto.assign = FALSE,from="2013-01-01", to="2016-12-31"),
           error=function(e) conditionMessage(e))} )
```

all.dat	list [503]	List of length 503
MMM	double [1008 x 6] (S3: xts, zoo)	9.42e+01 9.43e+01 9.48e+01 9.50e+01 9.52e+01 9.59e+01 9.48e+01 9.49e+01 9.55e+01 ...
ABT	double [1008 x 6] (S3: xts, zoo)	3.23e+01 3.29e+01 3.33e+01 3.30e+01 3.33e+01 3.37e+01 3.25e+01 3.34e+01 3.34e+01 ...
ABBV	double [1008 x 6] (S3: xts, zoo)	3.49e+01 3.50e+01 3.46e+01 3.42e+01 3.43e+01 3.36e+01 3.54e+01 3.50e+01 3.49e+01 ...
ACN	double [1008 x 6] (S3: xts, zoo)	6.76e+01 6.88e+01 6.89e+01 6.86e+01 6.88e+01 6.93e+01 6.91e+01 6.94e+01 6.96e+01 ...
ATVI	double [1008 x 6] (S3: xts, zoo)	1.08e+01 1.09e+01 1.10e+01 1.12e+01 1.11e+01 1.10e+01 1.09e+01 1.10e+01 1.12e+01 ...
AYI	double [1008 x 6] (S3: xts, zoo)	6.95e+01 6.96e+01 6.95e+01 6.87e+01 6.41e+01 6.59e+01 7.01e+01 7.16e+01 6.99e+01 ...
ADBE	double [1008 x 6] (S3: xts, zoo)	3.79e+01 3.81e+01 3.79e+01 3.78e+01 3.78e+01 3.82e+01 3.87e+01 3.83e+01 3.82e+01 ...
AMD	double [1008 x 6] (S3: xts, zoo)	2.55e+00 2.52e+00 2.51e+00 2.61e+00 2.72e+00 2.70e+00 2.57e+00 2.59e+00 2.59e+00 ...

2. Order Securities according to its total Volume

a. Order by Average Volume

```
all.vol <- lapply(all.dat, function(x) mean(x[,5], na.rm = T))
all.vol.mean <- t(as.data.frame(all.vol)); ascend.vol <- order(all.vol.mean)
```

	MMM	ABT	ABBV	ACN	ATVI	AYI	ADBE	AMD	AAP	AES	AET	AMG	AFL	A
1	2423354	7238386	8464911	2734085	7913029	413742.3	3143793	23288866	917251.7	5470951	2746560	463350.8	2220432	29

b. Sample 50 each from different range of size

```
target.all <- c(all.dat[ascend.vol[40:90]],   # Small-cap
                all.dat[ascend.vol[140:190]],
                all.dat[ascend.vol[240:290]],
                all.dat[ascend.vol[340:390]],
                all.dat[ascend.vol[440:490]])  # Large-cap
```

target.all	list [255]	List of length 255
MLM	double [1008 x 6] (S3: xts, zoo)	98.1 97.4 97.2 100.0 97.8 98.1 98.1 97.6 97.9 ...
MHK	double [1008 x 6] (S3: xts, zoo)	92.5 93.0 93.3 93.4 92.7 93.7 93.6 94.2 94.1 ...
ORLY	double [1008 x 6] (S3: xts, zoo)	9.07e+01 9.01e+01 9.05e+01 8.94e+01 8.96e+01 8.94e+01 9.07e+01 9.15e+01 9.19e+01 ...
SNPS	double [1008 x 6] (S3: xts, zoo)	3.24e+01 3.22e+01 3.19e+01 3.14e+01 3.15e+01 3.16e+01 3.25e+01 3.24e+01 3.19e+01 ...
SHW	double [1008 x 6] (S3: xts, zoo)	156 157 156 158 159 161 157 158 162 159 ...

3. Filter by Sharpe Ratio

Order by absolute Sharpe Ratio within 1 year, then choose top 50 securities

```
# recent 1 year Sharpe Ratios
all.sharpes <- apply(all.dailyReturn, 2,
                   function(x) SharpeRatio.annualized(x[length(x)-251:length(x)],Rf = 0))

# Top 50  absolute sharpe securities
toUse <- rank(abs(all.sharpes)) > 205
targets <- target.all[toUse]
```

targets	list [50]	List of length 50
SJM	double [1008 x 6] (S3: xts, zoo)	86.5 89.7 89.2 89.8 90.2 89.7 89.7 89.7 90.2 ...
VAR	double [1008 x 6] (S3: xts, zoo)	6.25e+01 6.37e+01 6.46e+01 6.33e+01 6.58e+01 6.58e+01 6.36e+01 6.47e+01 6.48e+01 ...
HRS	double [1008 x 6] (S3: xts, zoo)	4.99e+01 5.02e+01 4.96e+01 4.96e+01 4.94e+01 4.91e+01 5.05e+01 5.05e+01 5.02e+01 ...
VRSK	double [1008 x 6] (S3: xts, zoo)	5.17e+01 5.29e+01 5.34e+01 5.32e+01 5.34e+01 5.38e+01 5.30e+01 5.36e+01 5.40e+01 ...
EXR	double [1008 x 6] (S3: xts, zoo)	36.8 36.9 37.2 37.0 37.3 37.3 37.1 37.4 37.4 ...
FLIR	double [1008 x 6] (S3: xts, zoo)	2.30e+01 2.35e+01 2.39e+01 2.36e+01 2.34e+01 2.33e+01 2.36e+01 2.40e+01 2.39e+01 ...

4. Make Indicators

```
makeIndics <- function(target){

  #Daily Return    : daily
  #SMA(20days)  : sma
  #LMA(50days)  : lma
  #EMA : Exponential moving average. (20 days)  : ema
  #DEMA : Double-exponential moving average (20 days)  : dema
  #EVWMA : Elastic, volume-weighted moving average. (20 days) : evwma
  #ZLEMA : Zero lag exponential moving average. (20 days) : zlema
  #ALMA : Arnaud Legoux moving average. (20 days) : alma
  #HMA : Hull moving average. (20 days) : hma

  #RSI = AU / (AU + AD)  : rsi
  #CMO : The Chande Momentum Oscillator (CMO) is a modified RSI : cmo

  #ADX  : adx
  #CCI : The Commodity Channel Index (CCI) : cci

  # OBV : On Balance Volume  :obv
  # chaikinAD : The Chaikin Accumulation / Distribution (AD)   : cAD
```

Moving Averages:
smoothed average returns

Strength Indicators:
magnitude of variation

Movement Indicators:
people are buying or selling?

Money flowing into security?

```
indics <- data.frame(rsi, sma, lma, adx, alma, cci, cmo, dema,
                     ema, evwma, hma,zlema, obv, cAD)
 [truncated...]

 dat <- cbind(indics, daily[start:length(daily)])
 return(dat)
}
```

	rsi	sma	lma	adx	alma	cci	cmo	dema	ema	evwma	hma	zlema	obv	cAD	daily
2013-03-15	87.9	95.0	91.6	49.7	97.47195	86.5	56.6	97.9	95.2	92.6	98.48629	98.9	7302500	1020407.6	-0.004
2013-03-18	74.7	95.2	91.7	50.5	97.55341	75.0	78.2	98.0	95.4	92.9	98.46102	98.9	6488800	541760.5	-0.010
2013-03-19	46.4	95.4	91.9	49.1	97.45185	36.8	46.7	97.9	95.5	93.1	98.26118	98.7	5766300	331993.1	0.000
2013-03-20	41.5	95.7	92.0	46.5	97.17920	15.2	25.8	97.9	95.6	93.3	97.95414	98.4	6880500	1294604.4	0.004
2013-03-21	46.9	95.9	92.1	44.6	96.87751	35.6	28.4	97.9	95.7	93.5	97.63948	98.2	7642400	1534847.3	-0.008
2013-03-22	32.4	96.1	92.3	42.7	96.60936	8.8	-5.5	97.7	95.7	93.6	97.27331	97.9	7232800	1138462.8	0.005
2013-03-25	40.6	96.3	92.4	41.0	96.46385	9.2	-4.0	97.6	95.8	93.7	96.95586	97.7	7729100	1397403.3	-0.001
2013-03-26	38.7	96.5	92.5	40.2	96.38438	15.1	-15.7	97.6	95.9	93.8	96.68150	97.5	7341500	1155563.1	0.011

14 indicators ← Return →

Ⅱ Modeling

■ Data Partition

1. Class Labeling

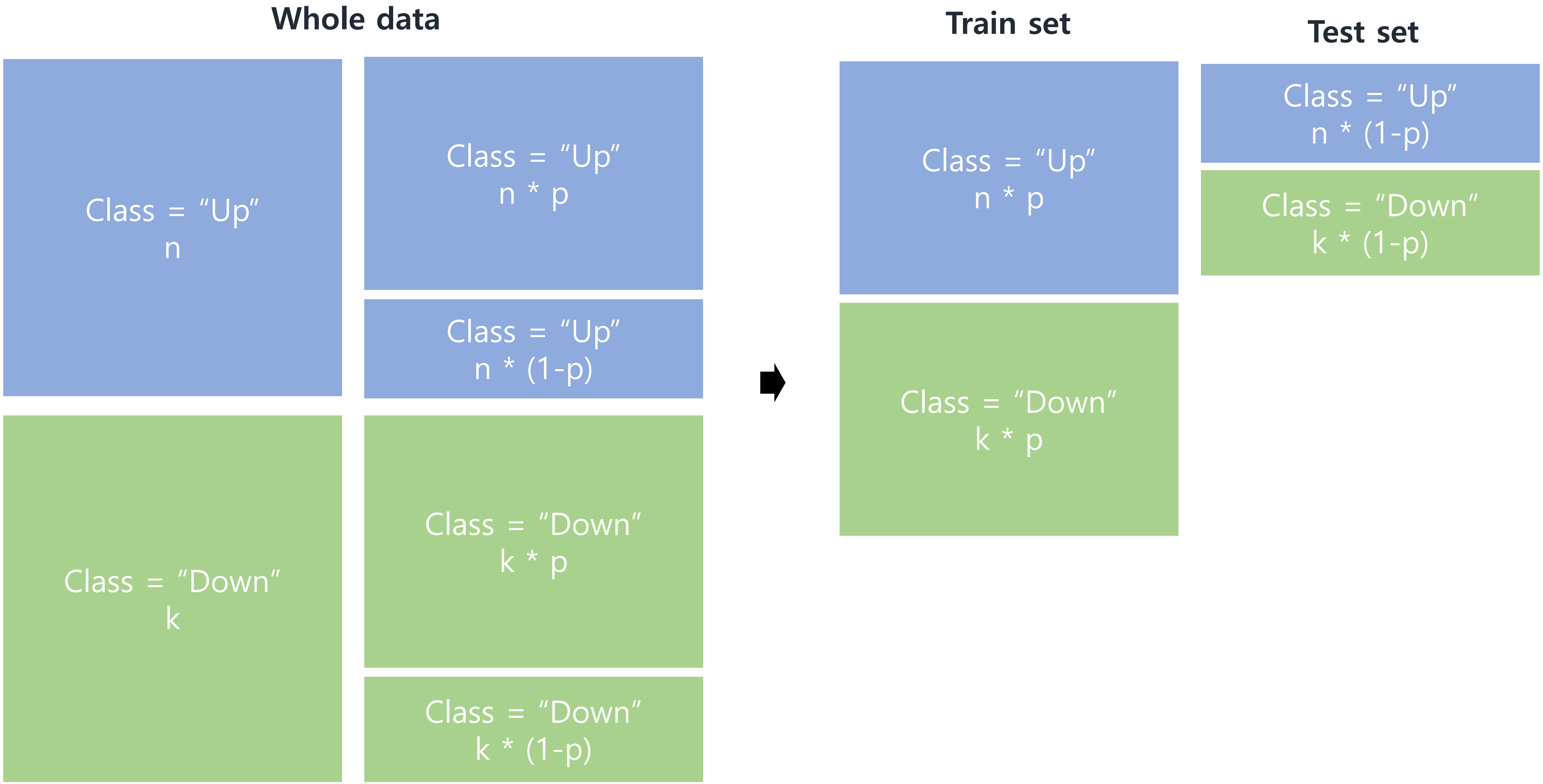
2. Choose Indicators

3. Make Models (for each security)

4. Choose Best Model (in each security)

5. Choose Best Models (among securities)

Data Partition



1. Class Labeling

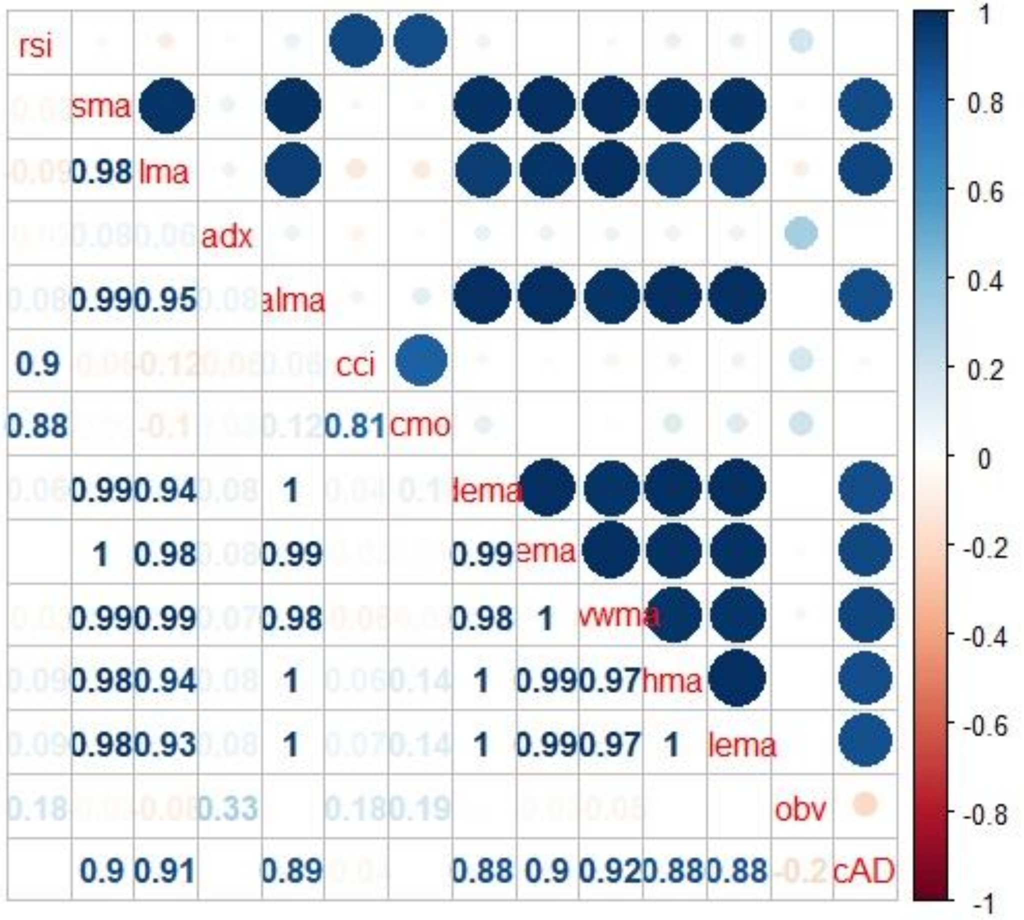
```r
doModel <- function(target, pRat){
  dat <- makeIndics(target)
  indics <- dat[,-15]
  daily <- dat[,15]

  # Remove no return change (zero)
  dat <- subset(dat, daily != 0)

  dat$daily <- ifelse(dat$daily > 0, "Up", "Down")
  dat$daily <- as.factor(x = dat$daily)
  names(dat)[15] <- "Class"
```

	rsi	sma	lma	adx	alma	cci	cmo	dema	ema	evwma	hma	zlema	obv	cAD	Class
2013-10-21	66.3	106.0	107.4	16.0	107.27851	183.1	29.9	106.6	106.6	106.8	107.20554	106.9	15071500	7514605	Down
2014-02-14	47.5	96.0	99.5	41.1	93.91004	-49.5	-22.2	93.5	95.9	97.4	92.64117	92.6	10631900	7381039	Down
2016-10-07	30.4	135.9	143.6	42.5	133.98661	-153.1	-30.0	132.9	136.4	139.8	133.75412	133.0	7946800	47823975	Down
2016-03-14	64.9	127.5	125.1	10.8	127.47284	163.2	15.0	128.4	127.4	126.0	127.65130	128.2	3327000	35048186	Down
2014-09-29	35.6	100.7	101.9	29.1	99.09438	-131.7	-36.0	99.1	100.3	101.0	98.80287	98.6	7414100	17678086	Down
2013-06-24	44.4	101.9	102.4	21.6	102.05783	-82.0	-8.8	101.6	101.9	101.3	102.14278	101.5	8663400	3722396	Down

2. Choose Indicators for the modeling

a. Make correlated indicator list

mylist	list [14]	List of length 14
rsi	character [3]	'rsi' 'cci' 'cmo'
sma	character [9]	'sma' 'lma' 'alma' 'dema' 'ema' 'ev
lma	character [9]	'sma' 'lma' 'alma' 'dema' 'ema' 'ev
adx	character [2]	'adx' 'obv'
alma	character [9]	'sma' 'lma' 'alma' 'dema' 'ema' 'ev
cci	character [3]	'rsi' 'cci' 'cmo'
cmo	character [3]	'rsi' 'cci' 'cmo'
dema	character [9]	'sma' 'lma' 'alma' 'dema' 'ema' 'ev
ema	character [9]	'sma' 'lma' 'alma' 'dema' 'ema' 'ev
evwma	character [9]	'sma' 'lma' 'alma' 'dema' 'ema' 'ev
hma	character [9]	'sma' 'lma' 'alma' 'dema' 'ema' 'ev
zlema	character [9]	'sma' 'lma' 'alma' 'dema' 'ema' 'ev



2. Choose Indicators for the modeling

b. Make a table with 3- non correlated indicators

```
## ---------------------------- ##
## Choose 3 indicators Set    ##
## ---------------------------- ##

corMat <- cor(indics)
corrplot.mixed(corMat)

# True if abs(correlation) over 0.2
corred <- abs(corMat) > 0.2

# All combinations of 3 variables
indicNames <- rownames(corMat)
varSet <- t(combn(indicNames,m = 3))
colnames(varSet) <- c("v1","v2","v3")
```

	v1	v2	v3
1	rsi	sma	adx
2	rsi	sma	obv
3	rsi	lma	adx
4	rsi	lma	obv
5	rsi	adx	alma
6	rsi	adx	dema
7	rsi	adx	ema
8	rsi	adx	evwma
9	rsi	adx	hma
10	rsi	adx	zlema
11	rsi	adx	cAD

3. Make Models

Make n models out of n sets of indicators

Train Data

Model 1

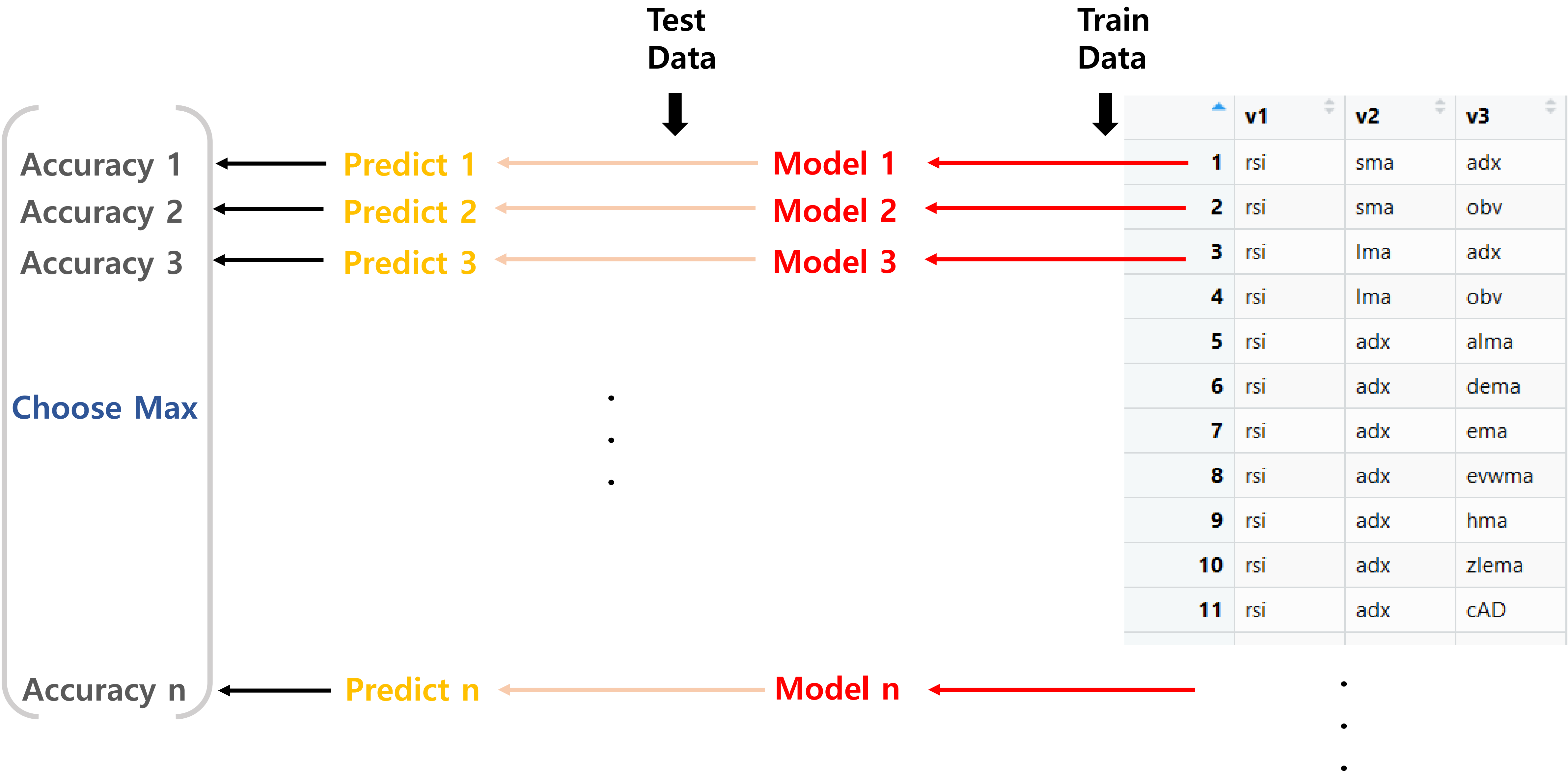
Model 2

Model 3

Model n

	v1	v2	v3
1	rsi	sma	adx
2	rsi	sma	obv
3	rsi	lma	adx
4	rsi	lma	obv
5	rsi	adx	alma
6	rsi	adx	dema
7	rsi	adx	ema
8	rsi	adx	evwma
9	rsi	adx	hma
10	rsi	adx	zlema
11	rsi	adx	cAD

4. Test Models and Choose the Best



	v1	v2	v3
1	rsi	sma	adx
2	rsi	sma	obv
3	rsi	lma	adx
4	rsi	lma	obv
5	rsi	adx	alma
6	rsi	adx	dema
7	rsi	adx	ema
8	rsi	adx	evwma
9	rsi	adx	hma
10	rsi	adx	zlema
11	rsi	adx	cAD

4. Test Models and Choose the Best

	rsi	sma	lma	adx	alma	cci	cmo	dema	ema	evwma	hma	zlema	obv	cAD	Class	pred.test[[1]]
2013-10-21	66.3	106.0	107.4	16.0	107.27851	183.1	29.9	106.6	106.6	106.8	107.20554	106.9	15071500	7514605	Down	Up
2014-02-14	47.5	96.0	99.5	41.1	93.91004	-49.5	-22.2	93.5	95.9	97.4	92.64117	92.6	10631900	7381039	Down	Up
2016-10-07	30.4	135.9	143.6	42.5	133.98661	-153.1	-30.0	132.9	136.4	139.8	133.75412	133.0	7946800	47823975	Down	Up
2016-03-14	64.9	127.5	125.1	10.8	127.47284	163.2	15.0	128.4	127.4	126.0	127.65130	128.2	3327000	35048186	Down	Up
2014-09-29	35.6	100.7	101.9	29.1	99.09438	-131.7	-36.0	99.1	100.3	101.0	98.80287	98.6	7414100	17678086	Down	Up
2013-06-24	44.4	101.9	102.4	21.6	102.05783	-82.0	-8.8	101.6	101.9	101.3	102.14278	101.5	8663400	3722396	Down	Down

Accuracy = 1/6

Test Data

Predict 1 ← Model 1
Predict 2 ← Model 2
Predict 3 ← Model 3

.
.
.

Predict n ← Model n

```
targets.model <- lapply(targets, function(x) {
  tryCatch(doModel(x, 0.7), error=function(e) conditionMessage(e) )
})
...
```

targets.model	list [43]	List of length 43
SJM	list [3]	List of length 3
model	list [14] (S3: rpart)	List of length 14
data	list [2]	List of length 2
train	list [639 x 15] (S3: data.frame)	A data.frame with 639 rows and 15 columns
test	list [275 x 15] (S3: data.frame)	A data.frame with 275 rows and 15 columns
acc.test	double [1]	0.5818182
VAR	list [3]	List of length 3
HRS	list [3]	List of length 3

Ⅲ Portfolio set & test

1. Make Portfolios
2. Make signals
3. Back Testing

1. Make Portfolios

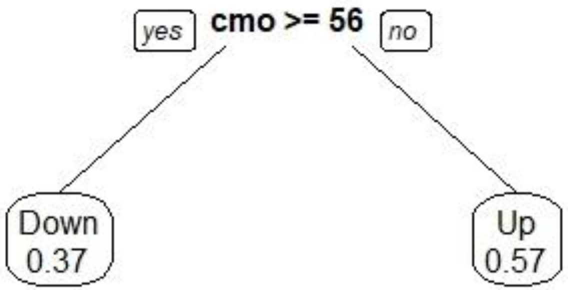
```
# Use models with over (lambda=0.57) accuracy
modelUse <- unlist(lapply(targets.model, function(x) x$acc.test > 0.57 ))

finalModel <- targets.model[modelUse]
```

finalModel	list [15]	List of length 15
SJM	list [3]	List of length 3
model	list [14] (S3: rpart)	List of length 14
data	list [2]	List of length 2
acc.test	double [1]	0.5745455
VAR	list [3]	List of length 3
model	list [14] (S3: rpart)	List of length 14
data	list [2]	List of length 2
acc.test	double [1]	0.6
FLIR	list [3]	List of length 3
model	list [14] (S3: rpart)	List of length 14
data	list [2]	List of length 2
acc.test	double [1]	0.5827338
LNT	list [3]	List of length 3

2. Generate Signals

SJM	list [3]	List of length 3
model	list [14] (S3: rpart)	List of length 14
data	list [2]	List of length 2
acc.test	double [1]	0.5745455

```
# Draw Trees
lapply(finalModel, function(x) prp(x$model, type = 0, extra=6))
```



CMO >= 56 : return will be negative 0.37 probability

CMO < 56 : return will be positive 0.57 probability

3. Back Testing

```
reports <- lapply(assess, backtest)
names(reports) <- assess
```

reports	list [15]	List of length 15
SJM	list [8]	List of length 8
AR.port	list [3 x 1] (S3: data.frame)	A data.frame with 3 rows and 1 columns
AR.null	list [3 x 1] (S3: data.frame)	A data.frame with 3 rows and 1 columns
table.Drawdown	list [5 x 7] (S3: data.frame)	A data.frame with 5 rows and 7 columns
VaR	double [1 x 1]	-0.01440313
Return.cum	double [1 x 1]	6.748106
hitRatio	double [1]	0.611691
chart.Summary.port	function	function() { ... }
chart.Summary.null	function	function() { ... }
VAR	list [8]	List of length 8
FLIR	list [8]	List of length 8

3. Back Testing - Result

Annualized Return table of Portfolio

	SJM	VAR	FLIR	LNT	HRL	XL	DXC	DLTR	DRE	AEP	ESRX	ATVI	GLW	ARNC	NFLX
Annualized Return	0.7136	0.8305	0.5374	0.1954	0.7226	0.5294	0.9996	1.0785	0.9524	0.4043	1.0711	0.7076	0.5340	2.6705	2.7460
Annualized Std Dev	0.1799	0.1947	0.2459	0.1749	0.1961	0.1764	0.3283	0.2356	0.1990	0.1752	0.2033	0.2939	0.2396	0.3558	0.4574
Annualized Sharpe (Rf=0%)	3.9674	4.2649	2.1857	1.1170	3.6844	3.0015	3.0446	4.5785	4.7860	2.3071	5.2677	2.4072	2.2293	7.5049	6.0031

Annualized Return table of Null (no investment rule)

	SJM	VAR	FLIR	LNT	HRL	XL	DXC	DLTR	DRE	AEP	ESRX	ATVI	GLW	ARNC	NFLX
Annualized Return	0.0734	0.0539	0.0906	0.1222	0.1640	0.0555	0.3148	0.1553	0.1331	0.0732	0.0394	0.2565	0.1809	-0.0108	0.4931
Annualized Std Dev	0.1831	0.1986	0.2475	0.1752	0.1990	0.1785	0.3310	0.2401	0.2035	0.1766	0.2087	0.2957	0.2410	0.3661	0.4651
Annualized Sharpe (Rf=0%)	0.4007	0.2716	0.3662	0.6979	0.8242	0.3108	0.9510	0.6465	0.6542	0.4146	0.1887	0.8676	0.7506	-0.0295	1.0602

Annualized Return (Portfolio – Null)

	SJM	VAR	FLIR	LNT	HRL	XL	DXC	DLTR	DRE	AEP	ESRX	ATVI	GLW	ARNC	NFLX
Annualized Return	0.6402	0.7766	0.4468	0.0732	0.5586	0.4739	0.6848	0.9232	0.8193	0.3311	1.0317	0.4511	0.3531	2.6813	2.2529

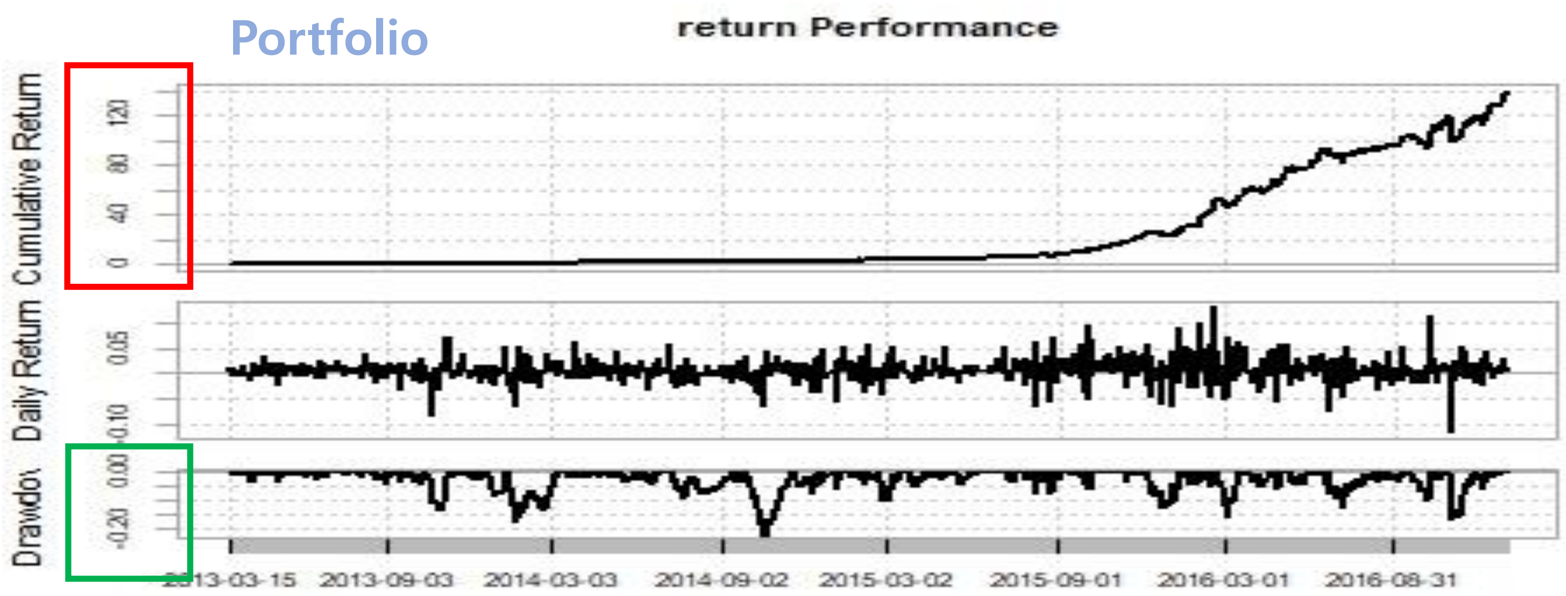
ARNC: Arconic Inc

	return
Cumulative Return	139.2379

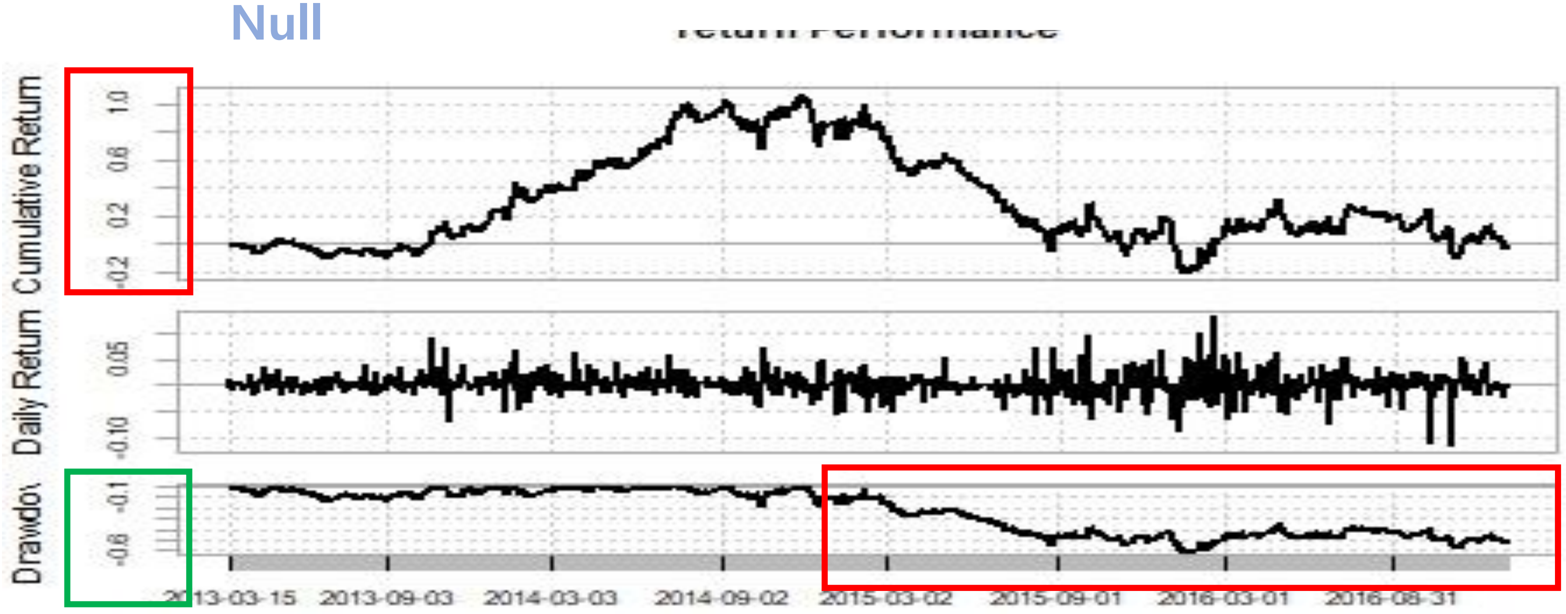
	From	Trough	To	Depth	Length	To Trough	Recovery
1	2014-09-29	2014-10-17	2014-11-18	-0.2267	37	15	22
2	2014-01-14	2014-01-22	2014-03-11	-0.1799	39	6	33
3	2016-11-01	2016-11-02	2016-11-28	-0.1667	19	2	17
4	2016-02-24	2016-03-07	2016-03-16	-0.1570	16	9	7
5	2013-10-16	2013-10-28	2013-11-07	-0.1324	17	9	8

Hit Ratio : 0.6304

Portfolio



Null



NFLX: Netflix, Inc.

	return
Cumulative Return	150.5131

	From	Trough	To	Depth	Length	To Trough	Recovery
1	2013-04-09	2013-05-07	2013-08-28	-0.3269	100	21	79
2	2016-10-05	2016-10-26	NA	-0.2261	62	16	NA
3	2015-07-16	2015-07-20	2015-08-21	-0.2174	27	3	24
4	2014-01-22	2014-01-23	2014-02-24	-0.1775	23	2	21
5	2015-10-02	2015-10-16	2015-11-10	-0.1420	28	11	17

Hit Ratio : 0.6409

Portfolio

Null