Lecture 4 (Feb 19, 2019)

Task `1’: Add naïve strategy:

1. If return today is >0, assume the same tomorrow
2. If return today <0, it will be the same tomorrow

Task 2: generate and plot your label data set

--------------------------------------------

X(1), X(2), ………..X(n), X(n+1), ….

MA(1, n) = (X(1) + ….. + X(n))/n

MA(2, n+1) = (X(2) + … + X(n+1))/n

Q: is there a recursive formula to compute moving averages?

A:

MA(2, n+1) = MA(1,n) – X(1)/n + X(n+1)/n

n \* MA(1, n) = X(1) + ….. + X(n)

suppose you have

X = (X1, ………., Xn)

m = Mean(X)

Y = X – m = (X1-m, X2-m, …., Xn – m)

Mean(Y) = mean(X) – mean(m) = 0

Effect of shifts

X = (X1, X2,…, Xn)

Y = X+b = (X1+b, X2+b, …., Xn+b)

Mean(Y) = ((x1+b)+…+ (Xn+b))/n =

(X1+…+Xn)/n + (b + ….+ b)/n

= mean(X) + b

X, Y, Z, ….

Mean(X+Y+Z) = mean(X)+mean(Y) + mean(Z)

How to use cumulative product:

D1: r1 0.07

D2 r2 -0.05

D3 r3 0.005

How to compute return for the first n days

After 1 year: x + x\*r1 = x(

1+R1)(1+R2)…(1+Rn)

1. Suppose after 10 years, your return is 50%. What is your annualized return?
2. If you put $100 in a CD (1% annual). In how many years will you double your money?

Pandas in detail next time

Homework #4

Task 1: Add naïve strategy:

1. If return today is >0, assume the same tomorrow
2. If return today <0, it will be the same tomorrow

Invest until there is a “bad” day

Task 2: generate and plot your label data set

For 2018 you labeled your dataset (green and red) for each week

52 labelled weeks for 2018

1. Add more labels (at least for 2017)
2. Plot your labels as follows:

For each week, compute the weekly return and weekly standard deviation

15 ----------------🡪 (R15, std\_dev(week))

Fri mo tu wed thu fri

Price 100 105

Therefore: R15 = 5%

Given r1, …., r5 – daily returns for the week

R1 – from prev Friday to mon

R2 - mon –tue

…..

R5 from thu to Fri

Std\_dev = np.std(return\_vector for 1 week)

For each week, compute (x, y)

X – mean of 5 daily returns

Y – st dev of 5 daily returns

A = np.array([r1, r2, r3,r4,r5])

X = np.mean(A)

Y = np.std(A)

For each week in 2018, compute weekly return (Fri-Fri) and plot it together with your label on a graph:

Color (x, y) using the color of your labels

Size of (x, y) reflecting weekly return

(I will put a simple script)

Nearest Neighbor Classification