Assignment: Using Machine Learning for Hedging

Welcome to the first assignment.

We will show how Machine Learning can be used in Finance to build multi-asset portfolios that have better risk/return characteristics than a portfolio consisting of a single asset.

Objectives

We will be using Linear Regression to establish the relationship between the returns of individual equities and "the market".

The purpose of the assignment is two-fold

- to get you up to speed with Machine Learning in general, and `sklearn` in particular
- to get you up to speed with the other programming tools (e.g., Pandas) that will help you in data preparation, etc.

How to report your answers

I will mix explanation of the topic with tasks that you must complete. Look for the string "**Queston**" to find a task that you must perform. Most of the tasks will require you to assign values to variables and execute a `print` statement.

Motivation

If you *do not change* the print statement then the GA (or a machine) can automatically find your answer to each part by searching for the string.

The data

The data are the daily prices of a number of individual equities and equity indices. The prices are arranged in a series in ascending date order (a timeseries).

- There is a separate `.csv` file for each equity or index in the directory `data/assignment_1`

Reading the data

You should get the price data into some sort of data structure. Pandas DataFrame is super useful so I recommend that's what you use (not required though).

Hints:

- look up the Pandas `read_csv` method
- it will be very convenient to use dates as the index of your DataFrame

Preliminary data preparation

In the rest of the assignment we will *not* be working with prices but with *returns* (percent change in prices). For example, for ticker AAPL (Apple)

$$r^{(t)}_{\mathrm{AAPL}} = \frac{p^{(t)}_{\mathrm{AAPL}}}{p^{(t-1)}_{\mathrm{AAPL}}} - 1$$

$$\text{where}$$

$$p^{(t)}_{\mathrm{AAPL}} \text{ denotes the price of ticker } \mathrm{AAPL} \text{ on date } t$$

$$r^{(t)}_{\mathrm{AAPL}} \text{ denotes the return of ticker } \mathrm{AAPL} \text{ on date } t$$

- You will want to convert the price data into return data
- We only want the returns for the year 2018; discard any other return

Question Replace the 0 values in the following cell with your answers, and execute the print statements

In [1]:
```
num_returns = 0  # Number of returns in year 2018
first_return = 0 # The return on the earliest date in 2018
last_return  = 0 # The return on the latest date in 2018
avg_return  = 0  # The average return over the  year 2018

print("There are {num:d} returns. First={first:.2%}, Last={last:.2%}, Avg={avg:.2%}".format(num=num_returns, first=first_return, last=last_return, avg=avg_return))
```

```
There are 0 returns. First=0.00%, Last=0.00%, Avg=0.00%
```

Split into Train and Test datasets

In general, you will split the data into two sets by choosing the members of each set at random.

To facilitate grading for this assignment, we will *use a specific test set*

- the training set are the returns for the months of January through September (inclusive), i.e., 9 months
- the test set are the returns for the months of October through December (inclusive), i.e., 3 months

Thus, you will be using the early part of the data for training, and the latter part of the data for testing.

Question Replace the 0 values in the following cell with your answers, and execute the print statements

In [2]:

```
train_num_returns = 0  # Number of returns in training set
train_first_return = 0 # The return on the earliest date in training set
train_last_return  = 0 # The return on the latest date in training set
train_avg_return  = 0  # The average return over the  year training set

print("Training set: There are {num:d} returns. First={first:.2%}, Last={last:.
2%}, Avg={avg:.2%}".format(num=train_num_returns,

first=train_first_return,

last=train_last_return,

avg=train_avg_return))

test_num_returns = 0  # Number of returns in test set
test_first_return = 0 # The return on the earliest date in test set
test_last_return  = 0 # The return on the latest date in test set
test_avg_return  = 0  # The average return over the  year test set

print("Test set: There are {num:d} returns. First={first:.2%}, Last={last:.2%},
 Avg={avg:.2%}".format(num=test_num_returns,

first=test_first_return,

last=test_last_return,

avg=test_avg_return))
```

```
Training set: There are 0 returns. First=0.00%, Last=0.00%, Avg=0.00%
Test set: There are 0 returns. First=0.00%, Last=0.00%, Avg=0.00%
```

$\mathbf{AAPL}$ regression

Use Linear Regression to predict the return of a ticker from the return of the SPY index. For example, for ticker AAPL

$$r^{(t)}_{\mathrm{AAPL}} = \beta_{\mathrm{AAPL,SPY}} * r^{(t)}_{\mathrm{SPY}} + \epsilon^{(t)}_{\mathrm{AAPL}}$$

That is

- each example is a pair consisting of one day's return
  - of the ticker (e.g., AAPL). This is the target (e.g, $\mathbf{y}$ in our lectures)
  - of the index SPY. This is a feature vector of length 1 (e.g., $\mathbf{x}$ in our lectures)

You will use Linear Regression to solve for parameter $\beta_{\mathrm{AAPL,SPY}}$

- In the lectures we used the symbol Θ to denote the parameter vector; here we use β
- In Finance the symbol β is often used to denote the relationship between returns.
- You may should add an "intercept" so that the feature vector is length 2 rather than length 1
  - $\mathbf{x}^{(t)} = \begin{pmatrix} 1 \\ r^{(t)}_{\mathrm{SPY}} \end{pmatrix}$

- Report the β parameter vector you obtain for AAPL
    - you will subsequently do this for another ticker in a different part of the assignment
        - so think ahead: you may want to parameterize your code
        - change the assignment to `ticker` when you report the next part

Question Replace the 0 values in the following cell with your answers, and execute the print statements

In [3]:
```
beta_0 = 0    # The regression parameter for the constant
beta_SPY = 0  # The regression parameter for the return of SPY
ticker = "AAPL"

print("{t:s}: beta_0={b0:3.2f}, beta_SPY={b1:3.2f}".format(t=ticker, b0=beta_0,
b1=beta_SPY))
```

```
AAPL: beta_0=0.00, beta_SPY=0.00
```

- Report the average of the cross validation scores, using 5 fold cross validation

Question Replace the 0 values in the following cell with your answers, and execute the print statements

In [4]:
```
cross_val_avg = 0

print("{t:s}: Avg cross val score = {sc:3.2f}".format(t=ticker, sc=cross_val_avg
) )
```

```
AAPL: Avg cross val score = 0.00
```

AAPL hedged returns

- Compute the series
$$r'^{(t)}_{\mathrm{AAPL}} = r^{(t)}_{\mathrm{AAPL}} - \beta_{\mathrm{AAPL,SPY}} * r^{(t)}_{\mathrm{SPY}}$$
for all dates t in the test set.
- Sort the dates in ascending order and plot the timeseries r'_{AAPL}

r'_{AAPL} is called the "hedged return" of AAPL

- It is the daily return you would realize if you created a portfolio that was
    - long 1 dollar of AAPL
    - short $\beta_{\mathrm{AAPL,SPY}}$ dollars of the index SPY
- It represents the outperformance of AAPL relative to the index SPY
    - SPY is the proxy for "the market" (it tracks the S&P 500 index)
    - The hedged return is the *value added* by going long AAPL rather than just going "long the market"
    - Sometimes referred to as the "alpha" (α_{AAPL})
- So **if** you are able to correctly forecast that AAPL will have positive outperformance (i.e, have $\alpha_{\mathrm{AAPL}} > 0$ most days)
    - then you can earn a positive return regardless of whether the market (SPY) goes up or down !
    - this is much lower risk than just holding AAPL long
    - people will pay you very well if you can really forecast correctly !

Question Replace the 0 values in the following cell with your answers, and execute the print statements

In [5]:
```
hedged_num_returns = 0  # Number of returns in hedged series
hedged_first_return = 0 # The return on the earliest date in hedged series
hedged_last_return  = 0 # The return on the latest date in hedged series
hedged_avg_return  = 0  # The average return over the hedged series

ticker="AAPL"
print("{t:s} hedged returns: There are {num:d} returns. First={first:.2%}, Last=
{last:.2%}, Avg={avg:.2%}".format(t=ticker,

num=hedged_num_returns,

first=hedged_first_return,

last=hedged_last_return,

avg=hedged_avg_return))
```

```
AAPL hedged returns: There are 0 returns. First=0.00%, Last=0.00%, Avg=0.00%
```

BA regression

Repeat the regression you carried out for AAPL but this time instead for the ticker BA (Boeing)

Motivation

The idea is to encourage you to build re-usable pieces of code.

So if you created some functions in solving Part 1, you may reuse these functions to easily solve part 2, particulary if you treated the ticker (e.g., AAPL or BA) as a parameter to your functions.

If you simply copy and paste the code from Part 1 you will only get partial credit.

Question Replace the 0 values in the following cell with your answers, and execute the print statements

In [6]:
```
beta_0 = 0    # The regression parameter for the constant
beta_SPY = 0  # The regression parameter for the return of SPY
ticker = "BA"

print("{t:s}: beta_0={b0:3.2f}, beta_SPY={b1:3.2f}".format(t=ticker, b0=beta_0,
b1=beta_SPY))
```

```
BA: beta_0=0.00, beta_SPY=0.00
```

Question Replace the 0 values in the following cell with your answers, and execute the print statements

In [7]:
```
cross_val_avg = 0

print("{t:s}: Avg cross val score = {sc:3.2f}".format(t=ticker, sc=cross_val_avg
) )
```

```
BA: Avg cross val score = 0.00
```

Question Replace the 0 values in the following cell with your answers, and execute the print statements

In [8]:
```
hedged_num_returns = 0  # Number of returns in hedged series
hedged_first_return = 0 # The return on the earliest date in hedged series
hedged_last_return  = 0 # The return on the latest date in hedged series
hedged_avg_return  = 0  # The average return over the hedged series

ticker="BA"
print("{t:s} hedged returns: There are {num:d} returns. First={first:.2%}, Last=
{last:.2%}, Avg={avg:.2%}".format(t=ticker,

num=hedged_num_returns,

first=hedged_first_return,

last=hedged_last_return,

avg=hedged_avg_return))
```

```
BA hedged returns: There are 0 returns. First=0.00%, Last=0.00%, Avg=0.00%
```

Returns to prices

- You have already computed the predicted returns of AAPL for each date in the test set.
- Create the predicted *price* timeseries for AAPL for the date range in the test set
- Plot (on the same graph) the actual price timeseries of AAPL and the predicted price timeseries.

There is a particular reason that we choose to perform the Linear Regression on returns rather than prices.

It is beyond the scope of this lecture to explain why, but we want to show that we can easily convert back into prices.

Question Replace the 0 values in the following cell with your answers, and execute the print statements