Machine Learning Engineer Nanodegree

Capstone Project

Louis Wan
Jan 8th, 2019

I. Definition

Project Overview

Algorithmic trading begins in the early 1970s introduced by New York Stock exchange. Many traders and engineers develop technical indicators to identify the market status. After 10 years, algorithmic trading became widely used in trading S&P 500 equity and futures markets. As the development of electronic communication networks goes on and the US decreased the minimum tick size, algo-trading was strongly encouraged. Till 2001, the concept of machine learning based trading agent is built by a team of IBM. [#1] IBM owns MGD and Hewlett-Packard owns ZIP. Both of the agents out-perform human traders.

The financial market is dynamic and full of uncertainty. Lots of financial industry like hedge funds have to works well with data, like time series data like the price of stocks and futures and multidimensional data like fundamental factors of a company. Investment bank hires many financial experts to building strategies for trading but people work emotionally. Algorithmic trading can help the expert to do judgment but the signals and indicators are all defined by a human. Algo-trading just provides a systematic way to trade according to a human logic. Can we build a reinforcement agent that can recognize signal by itself? This is the propose of this project.

Problem Statement

Over hundreds of technical and statistical indicators are used for machine learning based trading agent. To tackle a classification problem, decision tree and neural network will be used for trial. All data will be normalized before the model received to guarantee there is not over/under-weight the variables. After training, the agent can turn the indicators into trading decisions(buy or sell with different amount). The agent at first will take data and theirs TAs as input. Then calculate the 'score' of each fund. It is considered as a classification problem. In the second part, we may directly correct the score to the nearest integer percentage(MPF must have an integer percentage i.e. the smallest unit is 1%) and assign it to the portfolio weight.

Metrics

We will use the CAR/MDD for assessing trend-trading algorithm. The higher the CAR/MDD mean the higher ratio of compound annual return(reward) to maximum drawdown(risk). The definition is stated in Table 1.1.

Input

```
import numpy as np
import pandas as pd
from datetime import datetime


def car(_date, X):
    start = datetime.strptime(_date[0], '%d/%m/%Y')
    end = datetime.strptime(_date[-1], '%d/%m/%Y')
    delta = end - start
    delta_days = delta.days
    delta_years = float(delta_days) / 365
    print(delta_years)
    return_rate = (X[-1] - X[0])/X[0]
    return np.power(1 + return_rate, 1.0/delta_years) - 1


def max_drawdown(X):
    mdd = 0
    peak = X[0]
    for x in X:
        if x > peak:
            peak = x
        dd = (x - peak) / peak
        if dd < mdd:
            mdd = dd
    return mdd


file_name = 'dataset/HK_Equity_Fund_B_testing.csv'
df = pd.read_csv(file_name)
date_list = df.Date.values.tolist()
price_list = df.Price.values.tolist()
_car = car(date_list, price_list) * 100
_mdd = max_drawdown(price_list) * 100
print('Compound Annuel Return = %.4f percent' % _car)
print('Maximum DrawDown = %.4f percent' % _mdd)
print('CAR/MDD = %.4f' % (_car/_mdd))
```

Output

```
Compound Annuel Return = 9.4378 percent
Maximum DrawDown = -28.0587 percent
CAR/MDD = -0.3364
```

Table 1.1

You will obtain the performance of holding Sun Life MPF Hong Kong Equity Fund (Class B) by running the block above. The fund with the best performance is Sun Life MPF Hong Kong Equity Fund (Class B) in the testing period. The compound annual return is 9.4378 percent and the maximum drawdown of the fund is 28.0587 percent. So, the CAR/MDD is 0.3364. If the learning agent has both high CAR/MDD and the CAR, that's means the learning agent is better than the benchmark. The CAR/MDD (another name calmar ratio) will be calculated by the class pyalgotrade.stratanalyzer in the pyalgotrade library. Code is stated in Table 1.2.

```
From pyalgotrade.stratanalyzer import returns as rets
from pyalgotrade.stratanalyzer import sharpe, drawdown,
trades


retAnalyzer = rets.Returns()
strategy.attachAnalyzer(retAnalyzer)
sharpeRatioAnalyzer = sharpe.SharpeRatio()
strategy.attachAnalyzer(sharpeRatioAnalyzer)
drawDownAnalyzer = drawdown.DrawDown()
strategy.attachAnalyzer(drawDownAnalyzer)
strategy.run()

car = retAnalyzer.getCumulativeReturns()[-1]
mdd = drawDownAnalyzer.getMaxDrawDown()
if mdd > 0:
    calmar_ratio = retAnalyzer.getCumulativeReturns()[-1]
/ mdd
else:
    calmar_ratio = None
```

Table 1.2

II. Analysis

Data Exploration

The project will use Sun Life Rainbow mandatory provident fund(MPF) Scheme for financial trading products. The reason why using Sun Life MPF is that of these funds include a different kind of global instrument such as bonds, stocks, and foreign exchanges. Less noise has to tackle with. Secondly, Sun Life MPF service charge is counted and reflected inside the product itself with little entry barrier. Usually, the bond trading is requiring 1.5M cash in security account which is not feasible for the majority of people. The twelve funds are provided by Sun Life MPF.

Exploratory Visualization

The project will use Sun Life Rainbow mandatory provident fund(MPF) Scheme for financial trading products. [#2] The reason why using Sun Life MPF is that of these funds include a different kind of global instrument such as bonds, stocks, and foreign exchanges. Less noise has to tackle with. Secondly, Sun Life MPF service charge is counted and reflected inside the product itself with little entry barrier. Usually, the bond trading is requiring 1.5M cash in security account which is not feasible for the majority of people. The twelve funds are provided by Sun Life MPF.

Sun Life Rainbow MPF Scheme

Included for trading

1. Sun Life MPF Conservative Fund (Class B), Launch Date: 01 Dec 2000
2. Sun Life MPF Hong Kong Dollar Bond Fund (Class B), Launch Date: 01 Dec 2000
3. Sun Life MPF Stable Fund (Class B), Launch Date: 01 Dec 2000
4. Sun Life MPF Balanced Fund (Class B), Launch Date: 01 Dec 2000
5. Sun Life MPF Growth Fund (Class B), Launch Date: 01 Dec 2000
6. Sun Life MPF Hong Kong Equity Fund (Class B), Launch Date: 01 Dec 2000

Excluded from trading

7. Sun Life MPF Global Equity Fund (Class B), Launch Date: 01 Mar 2008
8. Sun Life MPF Asian Equity Fund (Class B), Launch Date: 01 Mar 2008
9. Sun Life MPF Greater China Equity Fund (Class B), Launch Date: 01 Mar 2008
10. Sun Life MPF Global Bond Fund (Class B), Launch Date: 01 Jan 2010
11. Sun Life MPF RMB and HKD Fund (Class B), Launch Date: 30 Jun 2012
12. Sun Life FTSE MPF Hong Kong Index Fund (Class B), Launch Date: 10 Dec 2013

Fund number 1 to 6 will be used for trading only due to the length of data of fund No. 7 to 12 is not enough. For technical indicators, I will use the python package named TA-lib for generating indicators. Since the only close price of the fund can be used, some of the indicators will not be available for use. I will use data from 01 Dec 2000 to 31 Dec 2017, total 6240 days. The data will be split into training and testing data in proportion 80/20.

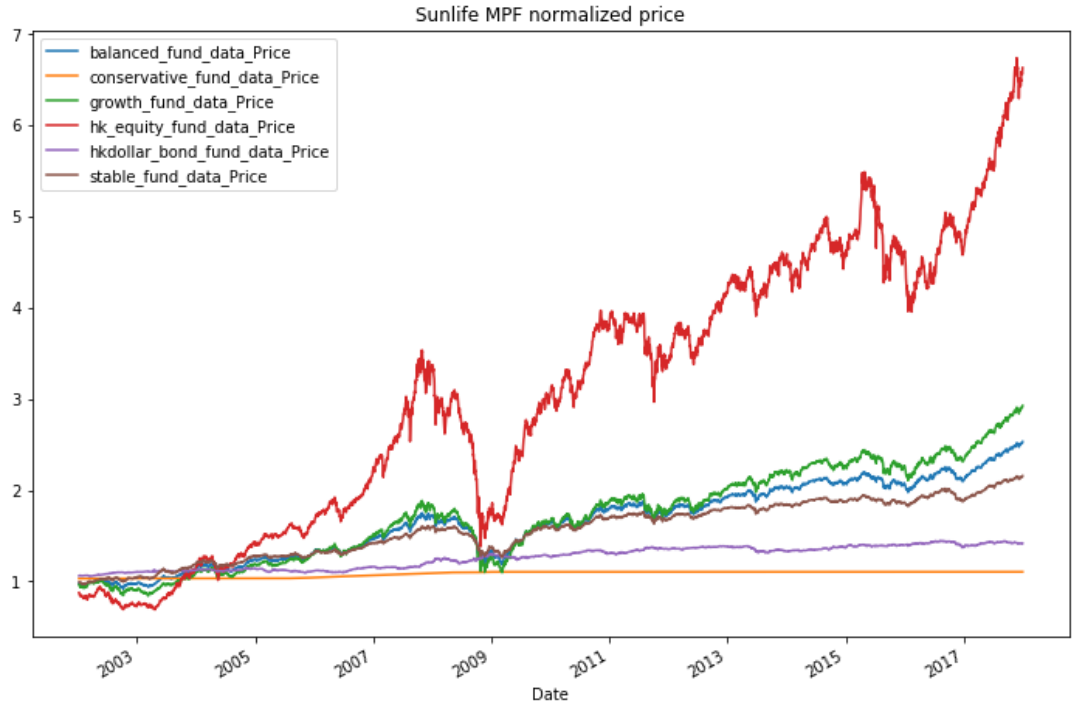
The training data is 4992 days(from 01 Dec 2000 to 02 Aug 2014) and the testing data is 1428 days(from 02 Aug 2014 to 31 Dec 2017). The financial data available in the world (index of the major market, fx rate, interest rate) will also include in the dataset. The data will grab from Yahoo! Finance. [#3]

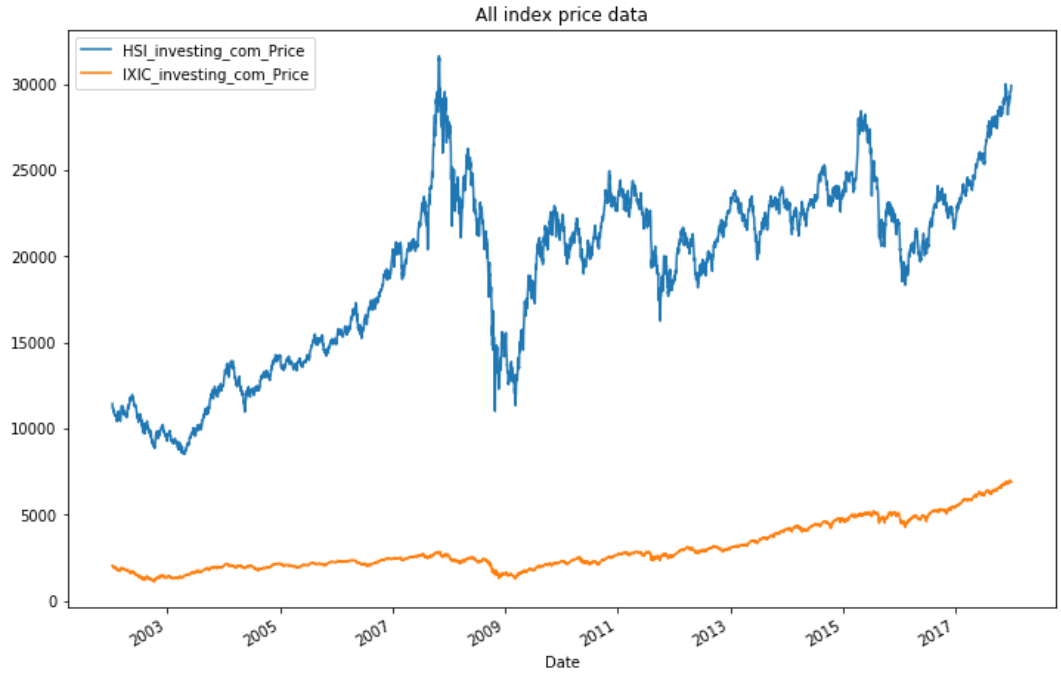
Input

```
from data_preprocess import DataMerger
import matplotlib.pyplot as plt

fund_fname_list = [
    'dataset/balanced_fund_data.csv',
    'dataset/conservative_fund_data.csv',
    'dataset/growth_fund_data.csv',
    'dataset/hk_equity_fund_data.csv',
    'dataset/hkdollar_bond_fund_data.csv',
    'dataset/stable_fund_data.csv',
    'dataset/HSI_investing_com.csv',
    'dataset/IXIC_investing_com.csv',
    'dataset/us2yrbondyield_investing_com.csv',
    'dataset/us10yrbondyield_investing_com.csv'
]

plot_config = {
    'fund': {
        'config': {
            'title': 'Sunlife MPF normalized price',
```

```
            'figsize': (12, 8)
        },
        'colnames':[
            'balanced_fund_data_Price',
            'conservative_fund_data_Price',
            'growth_fund_data_Price',
            'hk_equity_fund_data_Price',
            'hkdollar_bond_fund_data_Price',
            'stable_fund_data_Price',
        ]
    },
    'index': {
        'config': {
            'title': 'All index price data',
            'figsize': (12, 8)
        },
        'colnames':[
            'HSI_investing_com_Price',
            'IXIC_investing_com_Price'
        ]
    },
    'bond': {
        'config': {
            'title': 'All bond yield data',
            'figsize': (12, 8)
        },
        'colnames':[
            'us2yrbondyield_investing_com_Price',
            'us10yrbondyield_investing_com_Price'
        ]
    }
}

data_merger = DataMerger(fund_fname_list)
merged_df = data_merger.run(save_csv=False)
merged_df[plot_config['fund']['colnames']].plot(**plot_c
onfig['fund']['config'])
merged_df[plot_config['index']['colnames']].plot(**plot_
config['index']['config'])
merged_df[plot_config['bond']['colnames']].plot(**plot_c
onfig['bond']['config'])
```

Output


Sunlife MPF normalized price
balanced_fund_data_Price
conservative_fund_data_Price
growth_fund_data_Price
hk_equity_fund_data_Price
hkdollar_bond_fund_data_Price
stable_fund_data_Price
7
6
5
4
3
2
1
2003
2005
2007
2009
2011
2013
2015
2017
Date


All index price data
HSI_investing_com_Price
IXIC_investing_com_Price
30000
25000
20000
15000
10000
5000
0
2003
2005
2007
2009
2011
2013
2015
2017
Date

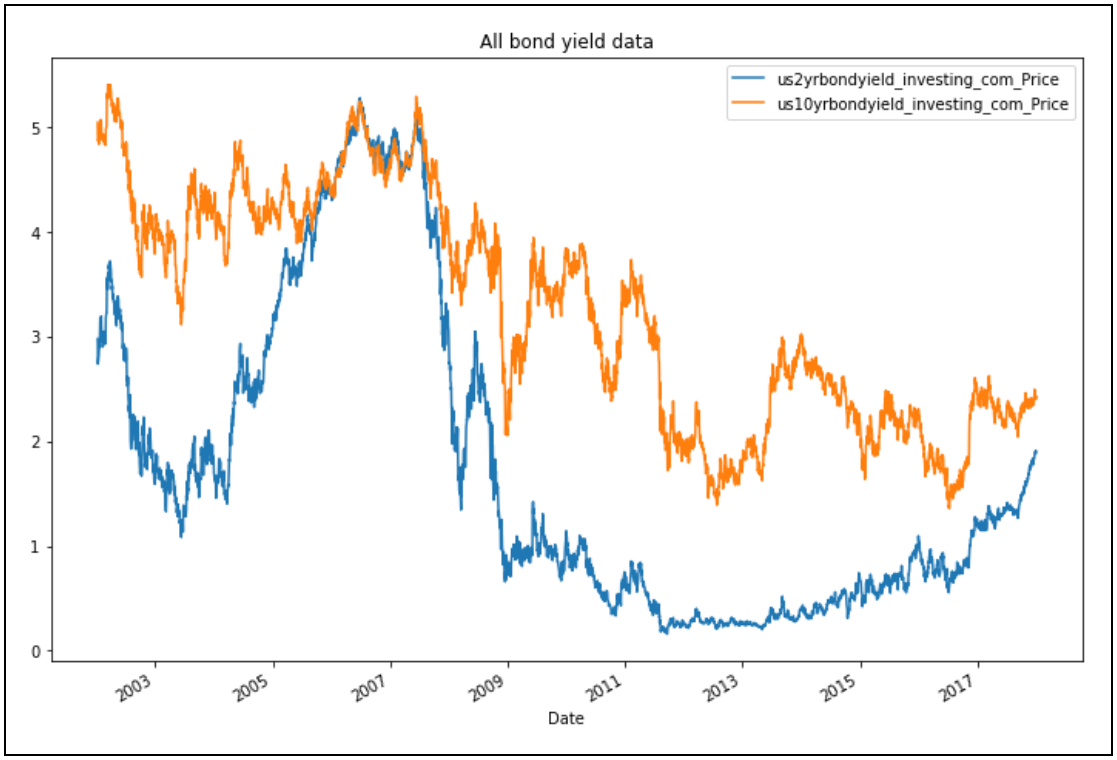


Table 2.1

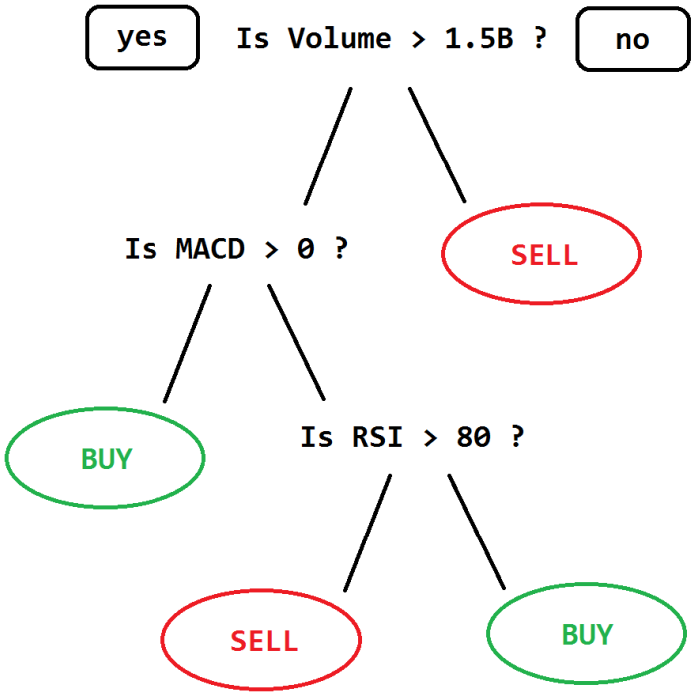
Here is the summary of basic statistics of the dataset

instument	min	max	median	mean	skew	kurtosis
hkdollar_bond_fund	1.0568	1.4449	1.2895	1.2629	-0.18444	-1.53403
IXIC	1114.11	7505.77	2499.29	3023.7	1.028874	0.161653
us10yrby	1.358	6.79	3.45	3.4414	0.294826	-0.92201
hk_equity_fund	0.691	6.9707	3.1383	3.0647	0.089122	-1.02756
conservative_fund	1.031	1.1052	1.1047	1.0822	-0.80231	-1.20682
HSI	8409.01	33154.12	20387.13	18938.	-0.18298	-0.90475
us2yrby	0.157	6.936	1.3755	2.0288	0.93318	-0.2293
stable_fund	0.9686	2.1956	1.5922	1.5671	-0.2014	-1.08174
growth_fund	0.8409	3.0204	1.7203	1.7168	0.140831	-0.96174
balanced_fund	0.9212	2.5968	1.6693	1.6491	-0.01935	-1.06319

Algorithms and Techniques

Decision tree regreessor and Multiple layer perceptron are common machine learning algorithms for classification problem.

Decision tree classifier (classification trees) is suggested to use. It takes a discrete set of values into tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. It finally calculates the scores of each class.

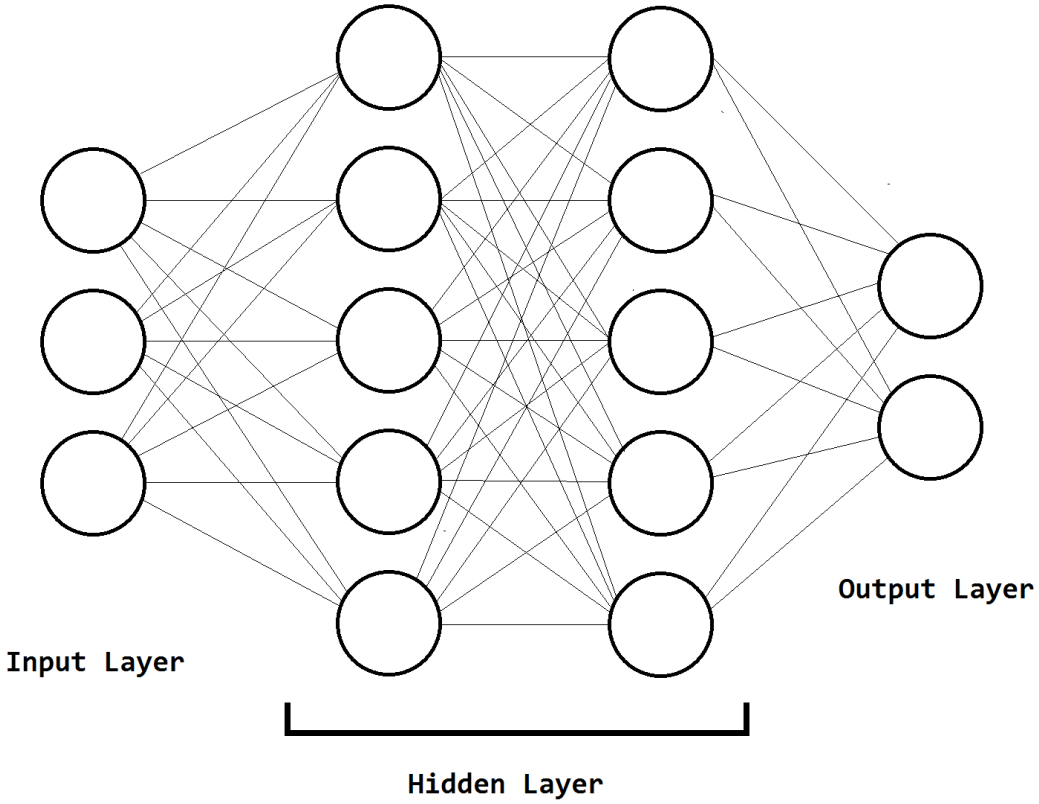


In the tree structure above, every interior node contain a question corresponds to a input variable. A threshold is optimized through a training algorithm, which is minimizing the information entropy S, for a better classification quality.

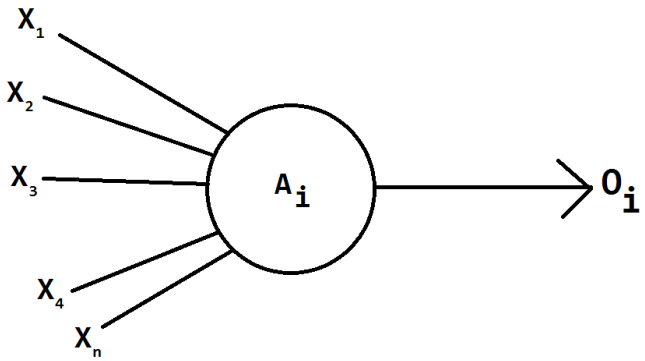
$$S = -\sum_i P_i \log P_i$$

Finally, thee leaf of the tree is labeled with a class or a probability distribution. We always have a decision that any kind of data given. The simplicity and training efficiency of decision tree makes it becomes a main machine learning model today.

Multiple layer perceptron have similar structure but function differently. It can split into three parts, input layer, hidden layer and output layer. Raw data vector will be put inside the input layer for every node a value.



For node operations, each node get values from the previous layer. It does a linear transform and pass through the activation function. Finally, the values obtained will pass to next layer.



$$O_i = f_{A_i}(\sum_{i=1}^{n} w_i X_i)$$

The network will keep on passing values until it reach the output layer. After the soft-max transformation, we can obtain the probability of belongs to the given class.

For the variable used, all the daily close price of each fund will be excluded and it prevents from autocorrelation and overfitting. I would like to focus on the fundamental factors (HSI index, IXIC index, US 2-year treasury bond yield and US 10-year treasury bond yield) and technical factors (MACD, RSI). The following features combination will be tested.

1. ['HSI_close','IXIC_close']
2. ['us2yrby_close','us10yrby_close']
3. ['HSI_close','IXIC_close','us2yrby_close','us10yrby_close']
4. ['hk_equity_fund_rsi',growth_fund_rsi','balanced_fund_rsi','conservative_fund_rsi','hkdollar_bond_fund_rsi','stable_fund_rsi']
5. ['hk_equity_fund_macd_histogram','growth_fund_macd_histogram','balanced_fund_macd_histogram','conservative_fund_macd_histogram','hkdollar_bond_fund_macd_histogram','stable_fund_macd_histogram']
6. ['hk_equity_fund_rsi','growth_fund_rsi','balanced_fund_rsi','conservative_fund_rsi','hkdollar_bond_fund_rsi','stable_fund_rsi','hk_equity_fund_macd_histogram','growth_fund_macd_histogram','balanced_fund_macd_histogram','conservative_fund_macd_histogram','hkdollar_bond_fund_macd_histogram','stable_fund_macd_histogram']

Table 2.2

Benchmark

The fund with the best performance is Sun Life MPF Hong Kong Equity Fund (Class B) in the testing period. The benchmark will set to hold 100% asset of Hong Kong Equity Fund until the end of testing period.

Input

```
python /path-to-the-directory/benchmark_strategy.py
```

Output

```
……
Importing data growth_fund from file path
dataset/growth_fund_data.csv
Importing data balanced_fund from file path
dataset/balanced_fund_data.csv
2014-08-05 00:00:00 strategy [INFO] BUY 206453 at $4.84
************************
** Backtesting Report **
************************
Final portfolio value: $1430681.60
```

```
Cumulative returns: 43.07 %
Sharpe ratio: -0.19
Max. drawdown: 27.93 %
Longest drawdown duration: 758 days, 0:00:00
Calmar ratio: 1.5418

Total trades: 0

Profitable trades: 0

Unprofitable trades: 0
```

From the output above, If we have a machine learning based strategy with higher or equal to 43.07% of cumulative returns and higher calmar ratio than 1.5418, that is a strategy good enough to solve the problem stated.

Input

```
import pandas as pd
import matplotlib.pyplot as plt

def drawdown_curve(_price):
    dd = list()
    peak = _price[0]
    for x in _price:
        if x > peak:
            peak = x
        dd.append((x - peak) / peak)
    return dd

hk_eq_fname = 'dataset/hk_equity_fund_data.csv'
hk_equity_fund_df = pd.read_csv(hk_eq_fname)
hk_equity_fund_df['Date']=pd.to_datetime(hk_equity_fund_
df['Date'], format='%Y-%m-%d')
hk_equity_fund_df= hk_equity_fund_df.sort_values('Date')
hk_equity_fund_df.index= range(len((hk_equity_fund_df)))
drawdown = drawdown_curve(hk_equity_fund_df['Price'])
hk_equity_fund_df['Drawdown'] = drawdown
hk_equity_fund_df = hk_equity_fund_df.set_index('Date')

hk_equity_fund_df[['Price']].plot(figsize=(12, 8))
hk_equity_fund_df[['Drawdown']].plot(figsize=(12,8),
style='r-')
```

Output

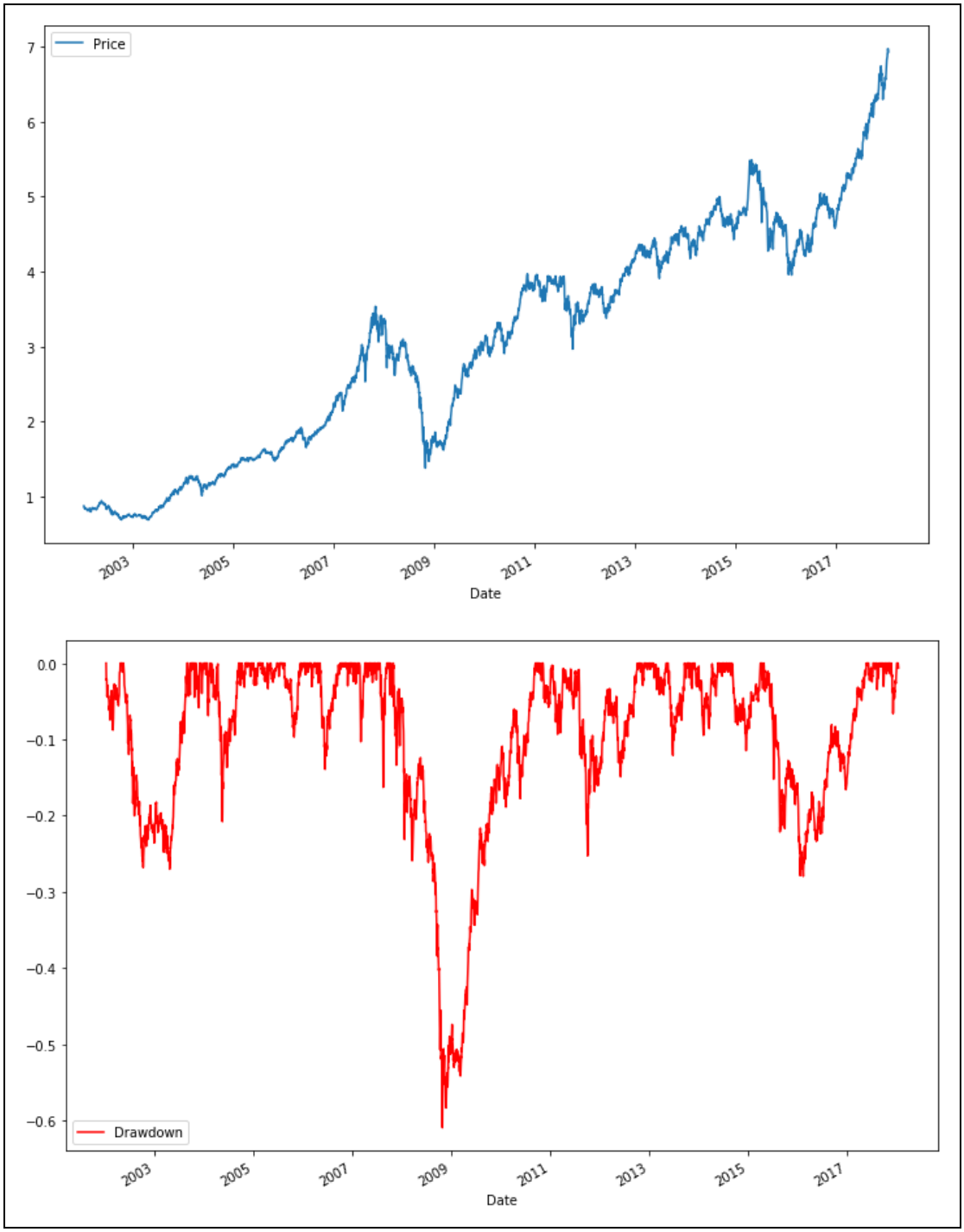

Price
7
6
5
4
3
2
1
2003
2005
2007
2009
2011
2013
2015
2017
Date
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.6
Drawdown
2003
2005
2007
2009
2011
2013
2015
2017
Date

Table 2.3

III. Methodology

Data Preprocessing

The data preprocessing job can be split into two parts, the features selection and data cleaning. First part we talk about the features selection. Heng Seng Index(^HSI), NASDAQ Index(^IXIC), US 10-year and 2-year treasury bonds yield, RSI and MACD of each MPF assets will be used as features. Close price of Heng Seng Index(^HSI) and NASDAQ Index(^IXIC) represents the stock market status. US 10-year and 2-year treasury bonds yield represents the economy status and the interest rate policy. RSI and MACD of each assets represents the trading activity of traders. Those features can be classified to fundamental factors or technical factors as follows.

Fundamental:
Heng Seng Index(^HSI), NASDAQ Index(^IXIC), US 10-year and 2-year treasury bonds yield
Technical:
RSI and MACD of each MPF assets

There are six combination of features will be backtested stated in Algorithms and Techniques. Secondly, the whole row of data will be discard if there is any missing column or wrong data like day with zero volume, open/close price greater than high and open/close price less than low, etc. A normalizer is set in the pipeline before the model. It is ensuring all the data will normalized between 0 and 1.

Implementation

The core algorithm of machine learning model is using a python library call scikit-learn. Pipeline(from class sklearn.pipeline) is used for managing the data process. Each data will pass through a StandardScalar (from class sklearn.preprocessing) ensuring it was normalized before input to the model. DecisionTreeRegressor(from class sklearn.tree) or MLPClassifier(from class sklearn.neural_network) will be used for classification.

Different to normal predictive analytics accuracy measure, we use fund performance measure, like return, risk, instead of accuracy and f1-score. We will need a backtesting system for simulating trades and orders. Pyalgotrade is used because of its user friendly with strong support of technical indicators and performance analytics. The RSI and MACD indicators(from class pyalgotrade.technical) is build in the library. It also handles the return, maximum drawdown and calmar ratio calculation(from class pyalgotrade.stratanalyzer).

For implementation difficulties, a lot of problem met but I would only like to mention the most important two. First one is about the data quality problem, the data quality from yahoo.com and investing.com is not good enough for direct use. Some of the data row have values greater than high or less than low in thee same bar. I usually get bar with 0 volume due to a non-trading day. The class pyalgotrade.csvfeed usually warns me about the data problem. I spend extra time for data cleaning and fitting into required format.

Secondly, there is a time lag in filling the order. For example, If Peter have 1000 HKD and would like to buy 100 units of 10 HKD asset A, he have to place the order now and wait for tomorrow morning fill the position. But the price of asset is always variating. If the price of asset A rise, the order cannot be filled by 100 units and error occurs. Fnally, I add an adjustment factor for cash arrangement to solve the problem.

Refinement

Run plenty of backtests is a time-consuming job. If all parameters of all technical indicators and all economic data is using, the whole process will become a monthly-counting task. So, this project is going to use main economic data and typical technical indicators for training. After running the brute force search, we have four model show better results than the benchmark. Here is the summary of result:

Model form	Cumulative returns	Maximum Drawdown	Calmar ratio
mlp_classifier	0.554726926	0.162191744	3.420192135
mlp_classifier	0.499784304	0.132149661	3.781956757
decision_tree	0.48302803	0.05988571	8.065831286
mlp_classifier	0.468409276	0.167575588	2.795211891
benchmark	0.430681603	0.279335335	1.541808532

Model form	Selected features	Predict N Days after
mlp_classifier	HSI_close, IXIC_close, us2yrby_close, us10yrby_close	10
mlp_classifier	HSI_close, IXIC_close, us2yrby_close, us10yrby_close	20
decision_tree	us2yrby_close, us10yrby_close	5
mlp_classifier	HSI_close, IXIC_close, us2yrby_close, us10yrby_close	5
benchmark	NA	NA

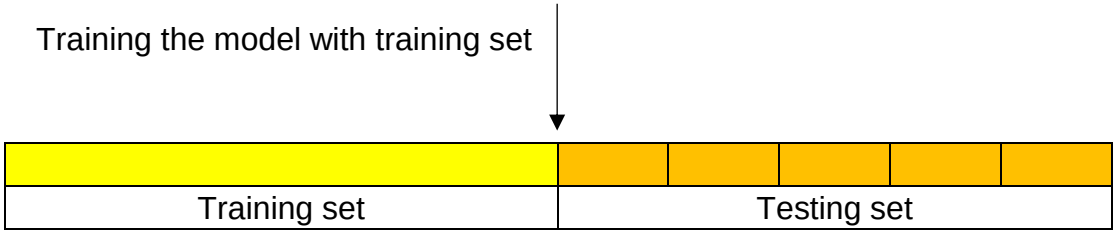
Index:
mlp_classifier - Multiple layer perceptron classifier
decision_tree - Decision tree classifier
HSI_close - Close price of Heng Seng Index
IXIC_close - Close price of NASDAQ Index
us2yrby_close - Day close of US 2-year Treasury Bond yield
us10yrby_close - Day close of US 10-year Treasury Bond yield

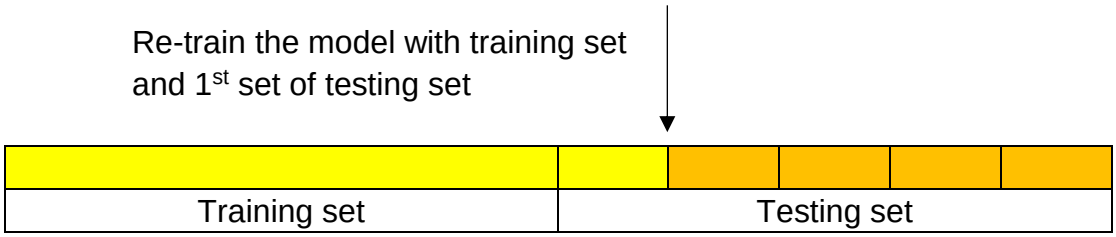
We can see there are four solutions having higher cumulative return and Calmar ratio and the decision tree model will be consider as optimal solution since it has the highest Calmar ratio. Day close of US 2-year and 10-year Treasury Bond yield will be the key features and close price of Heng Seng index and NASDAQ index will be stay in the watchlist.

IV. Results

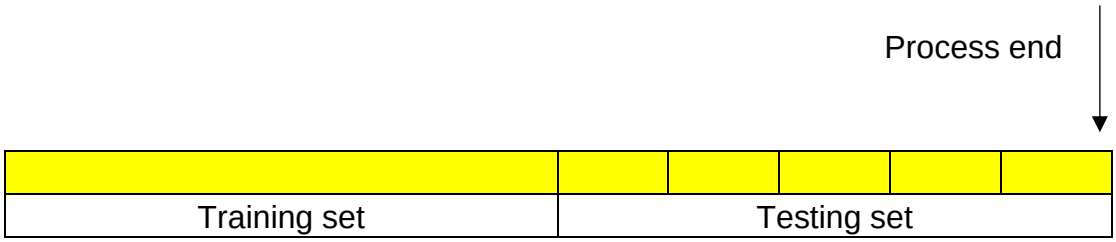
Validation and Justification

The suitable validation method for algorithmic trading will be walk-forward test. The idea of walk-forward test is re-training the model after a period of time. The testing set will be divided into uniform sets. When a part of data is fed to the model, the model is forced to re-train and update its coefficients. If the robustness of model form is strong, model will maintain the similar performance. Here is the example with testing set divided into 5 sets.





……



The process will go so on and so forth until the end of the time series is fed. Assume n is the number of uniform n-set of testing set, report of n from 2 to 24 is generated and summarized below. All cumulative returns lower than benchmark is filtered.

Model Form	Cumulative returns	Maximum Drawdown	Calmar ratio
decision_tree	0.445921457	0.081689289	5.458750606
decision_tree	0.439571546	0.037831128	11.61930843
mlp_classifier	0.470201375	0.065865527	7.138808371
mlp_classifier	0.451940884	0.065086873	6.943656408
decision_tree	0.438059876	0.037199872	11.77584352
mlp_classifier	0.497342997	0.059316926	8.384503944
mlp_classifier	0.490416044	0.065086971	7.534780598
mlp_classifier	0.475878085	0.065087009	7.311414218

Model Form	selected_features
decision_tree	HSI_close, IXIC_close
decision_tree	HSI_close, IXIC_close, us2yrby_close, us10yrby_close
mlp_classifier	us2yrby_close, us10yrby_close
mlp_classifier	us2yrby_close, us10yrby_close
decision_tree	HSI_close, IXIC_close, us2yrby_close, us10yrby_close
mlp_classifier	hk_equity_fund_macd_histogram, growth_fund_macd_histogram, balanced_fund_macd_histogram,conservative_fund_macd_histogram, hkdollar_bond_fund_macd_histogram,stable_fund_macd_histogram
mlp_classifier	us2yrby_close, us10yrby_close
mlp_classifier	us2yrby_close, us10yrby_close

Model Form	predict_n_days	n set of testing sets
decision_tree	10	3
decision_tree	5	5
mlp_classifier	10	6
mlp_classifier	10	9
decision_tree	5	10
mlp_classifier	10	10
mlp_classifier	10	18
mlp_classifier	10	24

From the table above, we can focus on two sets of results, labeled in yellow and orange. These two sets of results have the same model form and selected features.

Model Form	selected_features
decision_tree	HSI_close, IXIC_close, us2yrby_close, us10yrby_close
mlp_classifier	us2yrby_close, us10yrby_close

Both two sets of result have similar maximum drawdown, cumulative returns and calmar ratio. The set using decision tree split the testing data into 5, 10 sets and MLP classifier split the testing data into 6, 9, 18, 24 sets. The funny fact inside is the number of set to split the data have share the same factor, 5 is the factor of (5, 10) and 3 is the factor of (6, 9, 18, 24). It is because they have the same exactly moment of re-training.

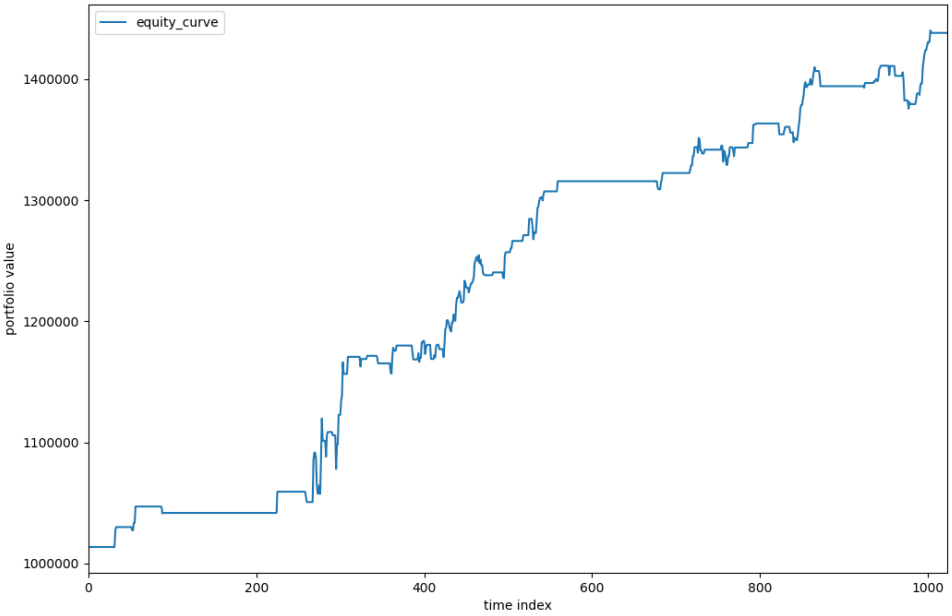
n = 5

n = 10 re-training

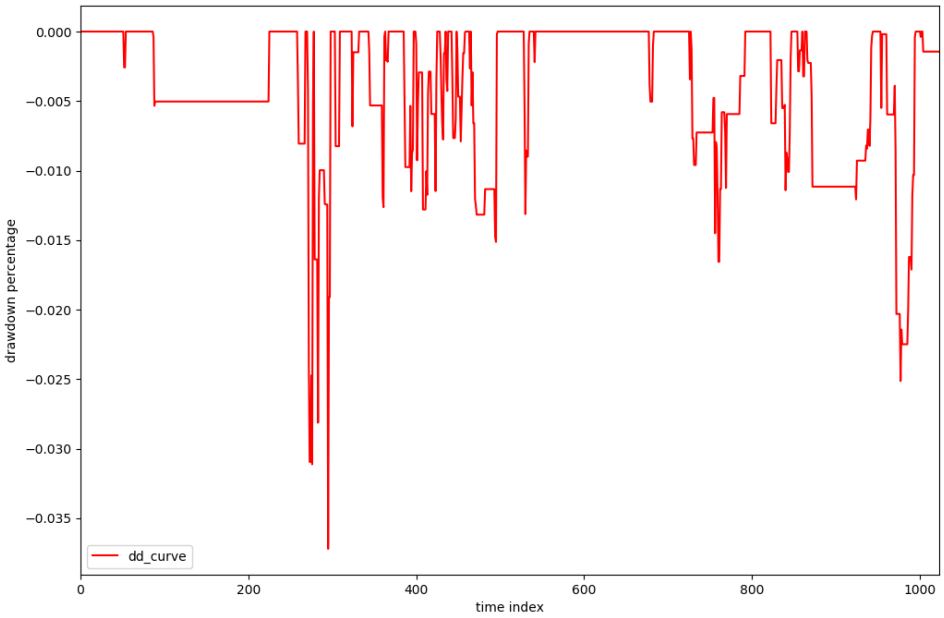
So, the result set with non-trivial highest common factor have high robustness in their model. Unfortunate, we cannot find any matched parameters with the model build in part III. But the selected features, n-step days to forecast are similar, what we need to do is keep study with more indicator, complex model form and higher number of testing set splitting. If a final solution have to be suggested, it will be stated as follows:

Model Form	decision_tree
Selected features	HSI_close, IXIC_close, us2yrby_close, us10yrby_close
predict_n_days	5
Retrain after n calendar days	143 days (n = 10, 1428 days / 10 = 143 days)

It has the highest calmar ratio 11.62 with cumulative returns 43.96% and maximum drawdown 3.78% in the result set. The reason to pick n = 10 instead of n = 5 is higher frequency of re-training is allowed have high robustness in market status identification. Here is the equity curve and the drawdown curve of the final solution.



Equity curve



Drawdown curve

V. Conclusion

Reflection

Handle time series data for building a machine learning model is a big challenge. Typical cross validation is not working because using future predict the past is not making sense. I finally pick up a new method, walk-forward test, for validating strategies for algorithmic trading. For difficulties, picking up finance and trading knowledge is a must. I have to clearify how to read economical data and technical indicators. Beside those knowledge is needed, handling data and Coding skill are very challenging task. I need a very long time to study the principle of algorithmic trading with its package pyalgotrade. Finally, the final solution proposed meet my expectation. It is a portfolio that enjoy higher return and lower risk than the benchmark.

Improvement

There are always exist improvements in machine learning models like try more different features, parameters and other complex model form to test and validate. What I would like to try are fundamental data like US/HK interest rate, inflation rate, capital flow, employment rate and purchasing managers' Index. These factor reflecting the economic status, money supply and liquidity of a state. At first I prefer using reinforcement learning agent as my primary model but I found that the package is hard to use and picking up the knowledge is very time consuming. I would like to try if further study is conducted and I believe a better solution will be produced.