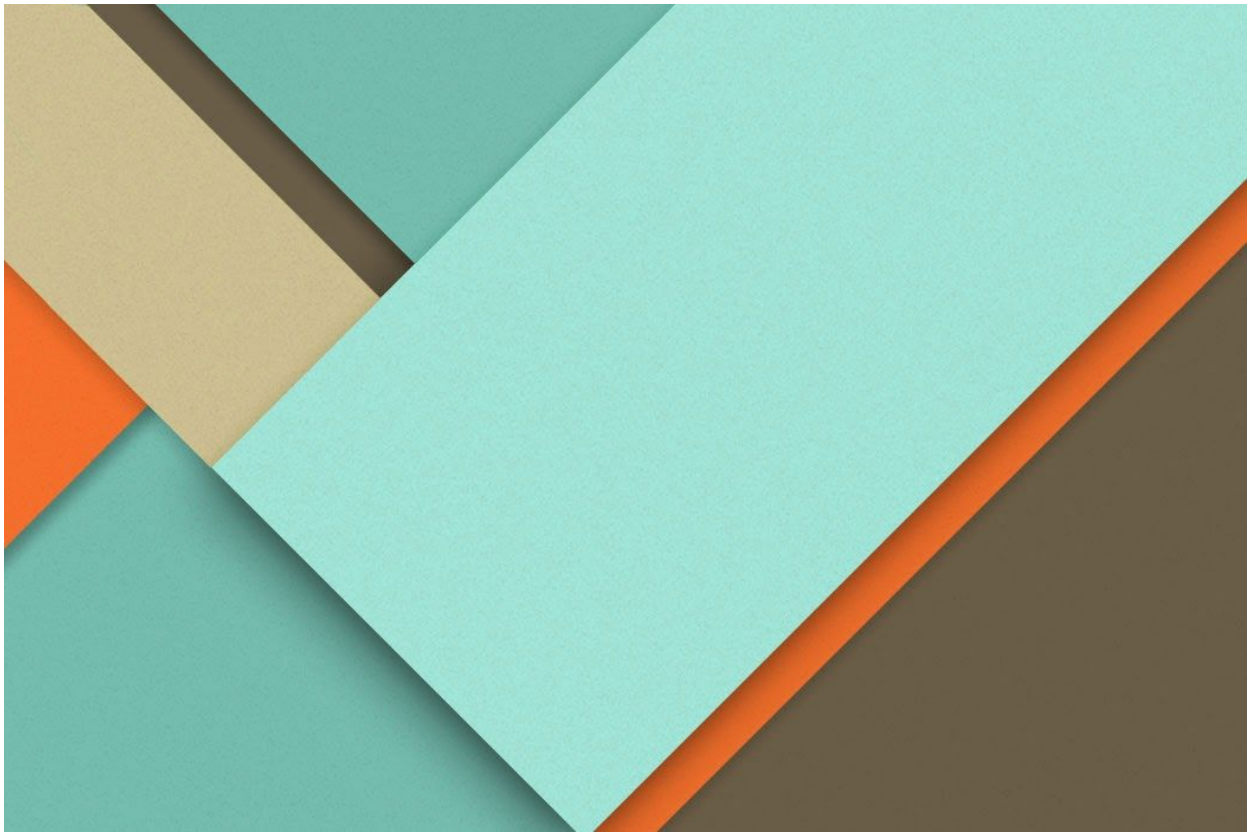


Trading Strategy Based on Unsupervised Learning

16.08.2020

Yuval Weinberger
TCDS 13

Table Of Content

1) Introduction

As part of the final project at the Data Science program, I chose to use unsupervised learning, as a basis as a trading strategy.

1.1) The Goal:

The Goal of the project is to show a significance in the connection between specific clusters and “up days”. Or in other words, creating a strategy based on clusters that is more stable, reliable and with better a risk/reward ratio than the S&P 500 index.

1.2) The Data:

I use the SPY ETF daily data, and the VIX index daily data, from 1993 to 2020. The source of the data is finance.yahoo.com Website, that provides free stock, ETF, and index historical data.

The SPY is an exchange-traded fund, which means that it is a tradable financial instrument.

The SPY is an ETF that follows the S&P 500 index, and its behavior is almost the same as the S&P 500. I will use the open, high, low, close, and volume quotes of daily candles and extract features based on this data.

The VIX is a popular measure of the stock market's expectation of volatility based on S&P 500 index options. It is calculated and disseminated on a real-time basis by the CBOE, and is often referred to as the ”fear“index or fear gauge.

I will use the VIX close price as another feature to add a layer of volatility measure to the model.

1.3) The Algorithm:

The algorithm that I will use is the K-Means algorithm.

The K-Means is an unsupervised learning algorithm that clusters data to K clusters. The K-Means algorithm identifies the K number of centroids, and then allocates every data point to the nearest cluster, by calculating euclidean distance.

K will be one of the model’s hyper-parameters.

2) Descriptive Statistics

Below are Descriptive Statistics of the data that the model will use: SPY, and VIX quotes.

2.1) SPY:

2.1.1) Raw Data:

Date	Open	High	Low	Close	Volume
1993-01-29	43.96875	43.96875	43.75000	43.93750	1003200
1993-02-01	43.96875	44.25000	43.96875	44.25000	480500
1993-02-02	44.21875	44.37500	44.12500	44.34375	201300
1993-02-03	44.40625	44.84375	44.37500	44.81250	529400
1993-02-04	44.96875	45.09375	44.46875	45.00000	531500

The raw data is the Open, High, Low, Close, and Volume quotes from 1993.

2.1.1) Data Trend:

The general trend is an up trend.

We can split our data into 3 different market behaviors:

I) from 1993 to 2010 – market consolidation.

II) from 2010 to 2018 – strong up trend.

III) from 2018 to 2020 – extremely high volatility market.

I) SPY data from 1993 to 2010:

During this period of time there were two big financial crises that led to “bear” markets. The first is the “dot-com” crisis from 2000, and the second is the “sub-prime” crisis.

II) SPY from 2010 to 2018:

During this period of time the markets had a “bull” market with almost 200 points gain in eight years.

III) SPY from 2018 to 2020:

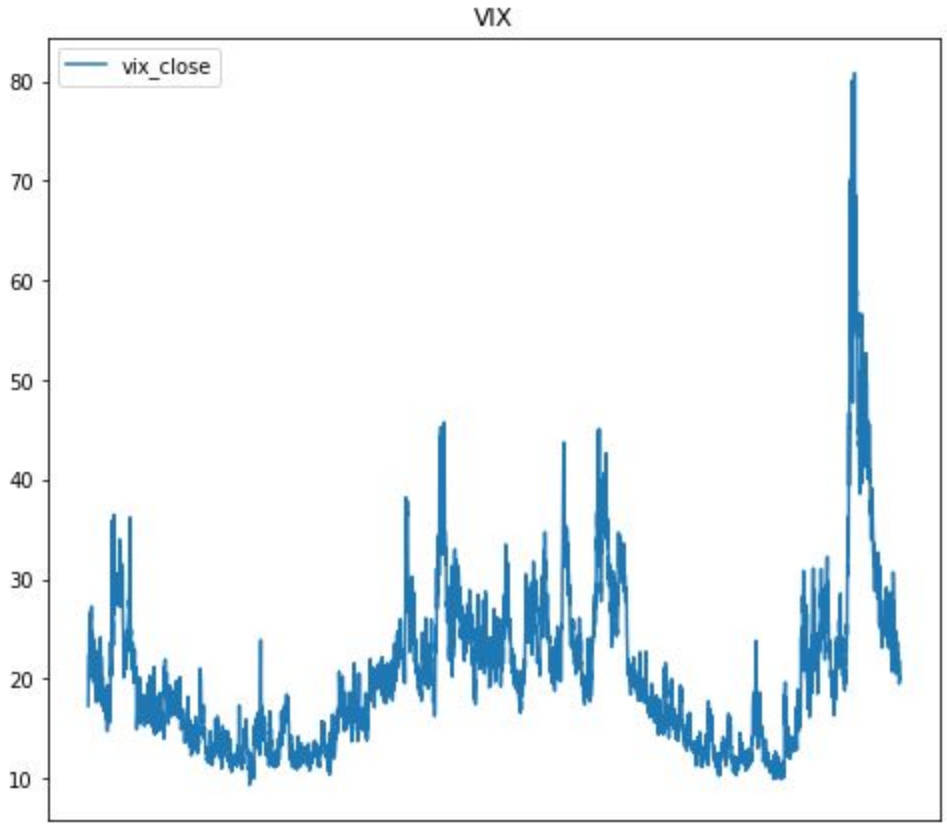
Since 2018 the markets high levels of volatility have increased. We can identify at least 3 V-shape recovery periods that happened in a very short time.

This piece of information is very important, because if we train the K-Means algorithm on a specific market behavior, we can't expect it to work on different market behaviors. To let the model identify these market behaviors , the VIX data will be added.

2.2) VIX:

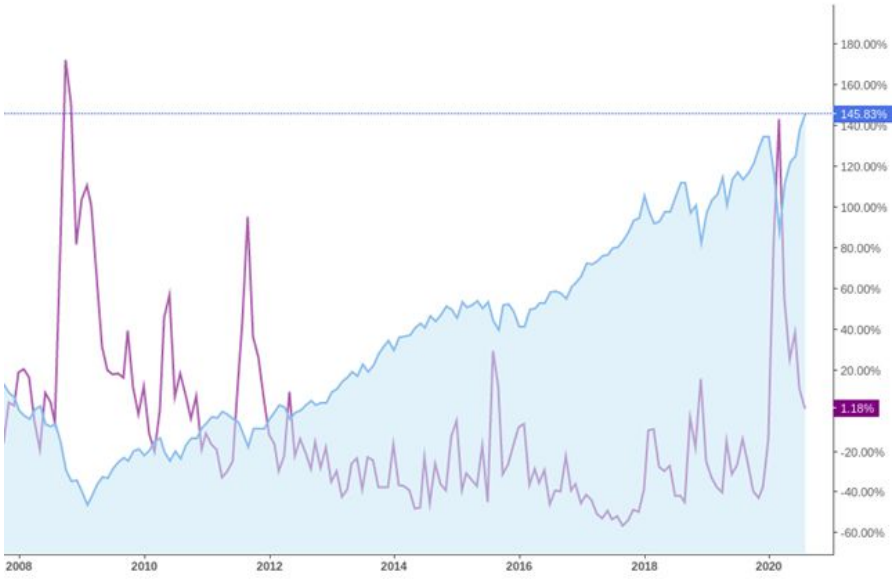
The VIX data will be used to identify and filter specific market dips, and avoid them.

2.2.1) VIX Behavior:



We can see that the VIX is a spiky index. Every time the fear in the market is strong, the VIX will rise at a remarkable rate.

2.2.2) VIX and SPY Correlation:



We can see that when there is panic in the markets, the VIX rises. For example, in the 2008 subprime crisis the VIX was high.

3) Feature Extraction

Before training and testing the model, features extraction will be done from the raw data, to train the model to more specific data. For the raw data it is possible to extract various features, including momentum, and trend analyzers.

We will use as features:

1) The VIX close price divided by the moving average of the last 20 days.

2) Simple difference features: high – low, open – close, etc.

```
df["hl_range"] = (df["High"] - df["Low"])
df["oc_range"] = (df["Open"] - df["Close"])
df["oh_range"] = (df["Open"] - df["High"])
df["ol_range"] = (df["Open"] - df["Low"])
df["cl_range"] = (df["Close"] - df["Low"])
df["ch_range"] = (df["Close"] - df["High"])

df["vix_close"] = df["vix_close"] / talib.SMA(df.vix_close, 20)

```

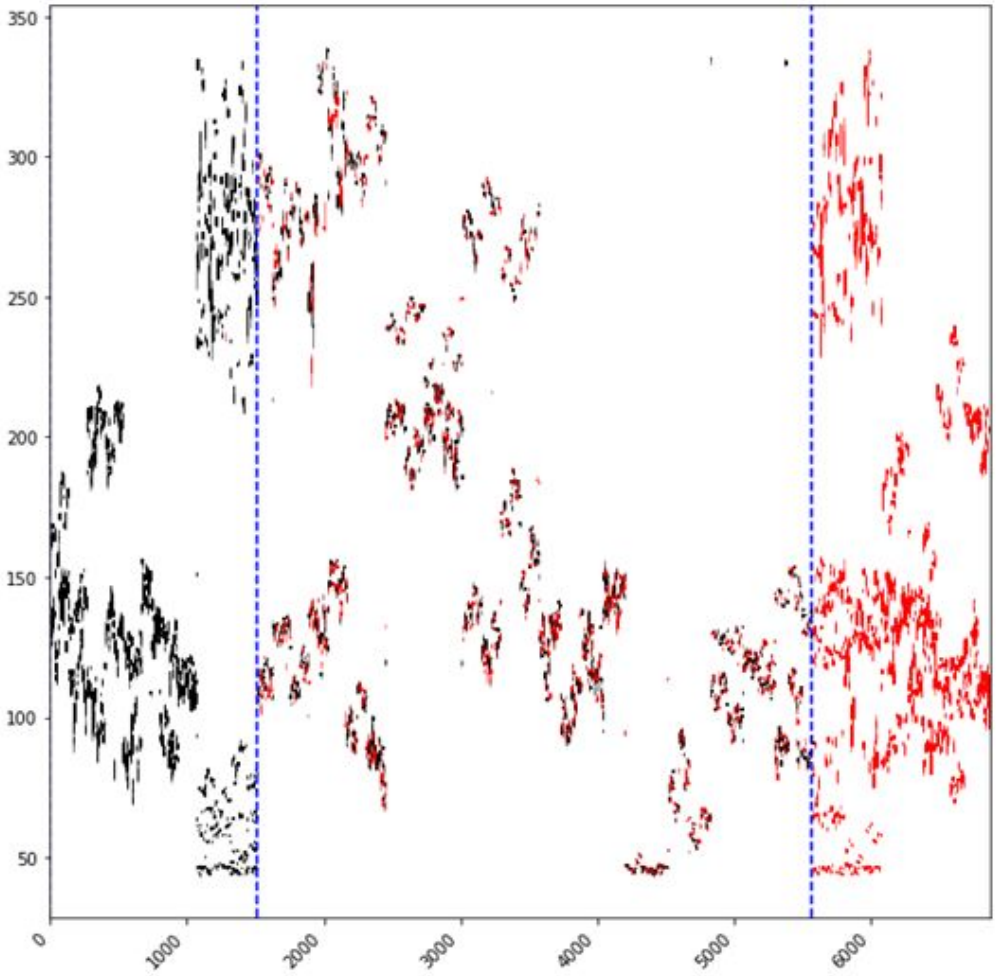
4) Model

4.1) K-Means and The Features:

As mentioned in the introduction, I will use the K-Means algorithm.

The K-Means is an algorithm to identify similarities in data.

The current features magnify day candle different sizes, therefore, the algorithm will classify “up days” and “down days”, to different clusters. To choose K I will use the help of the Silhouette score.



In the example above, we can see three clusters. two of them are clearly separated as “up days” and “down days”, by the K-Means algorithm.

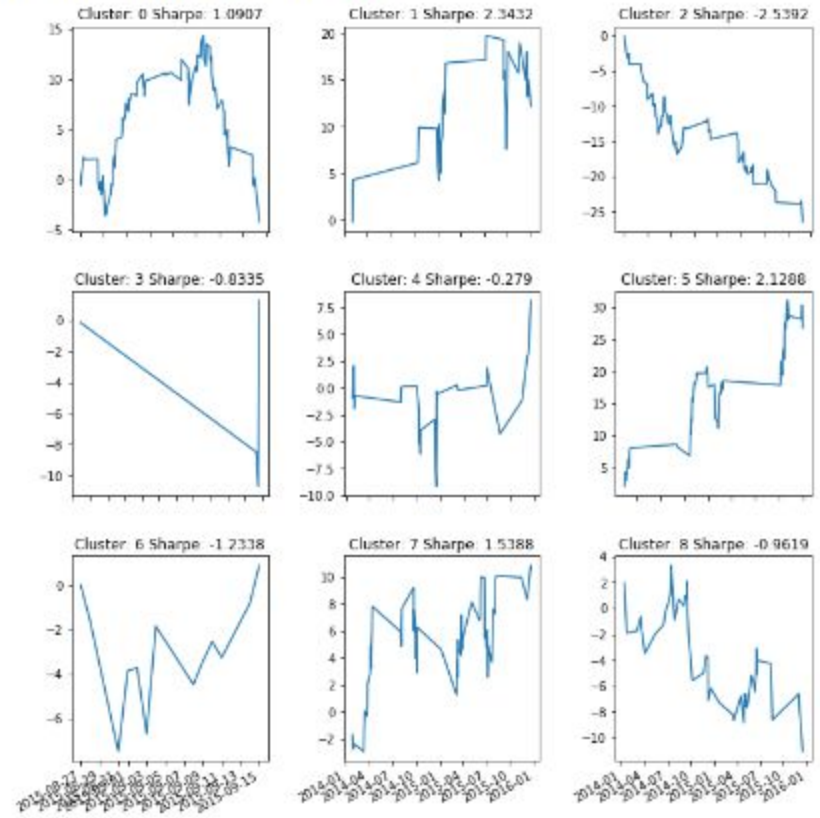
4.2) Train-Test Process:

Stock and ETF quotes are time-series data, therefore we can't randomly shuffle the data.

I will use the Walk-Forward train test process in order to fit and test the model every time on small pieces of data. Afterwards, the tested data will be added to the train set, and repeat the process.

As part of the Walk-forward process the user will choose the clusters to be used as trading strategy on the testing phase.

```
start date training:  2014-01-01 00:00:00
end date training:  2016-01-01 00:00:00
```




```
<Figure size 5040x5040 with 0 Axes>
```

```
Enter number of cluster models :
```

At the end of the testing date range, the user receives a cumulative gain curve, to see how much the strategy gained on the testing periods.

5) Model Evaluation and Results

5.1) Evaluation Factors:

To evaluate the model the point gain will be used as a meter of profit. Every time a day is classified in a chosen cluster, the gain/loss in points will be added to the general gain curve.

```
self.gain_df["gain"] = self.gain_df["Close"].shift(-1) - self.gain_df["Close"]
```

Gain is not the only measure of a successful trading strategy. I will also use the mean / standard deviation of the gain curve to decide if the gain curve is volatile. Another measure will be the number of trades that the strategy identified. Less trading days means more free liquidity to other trading strategies.

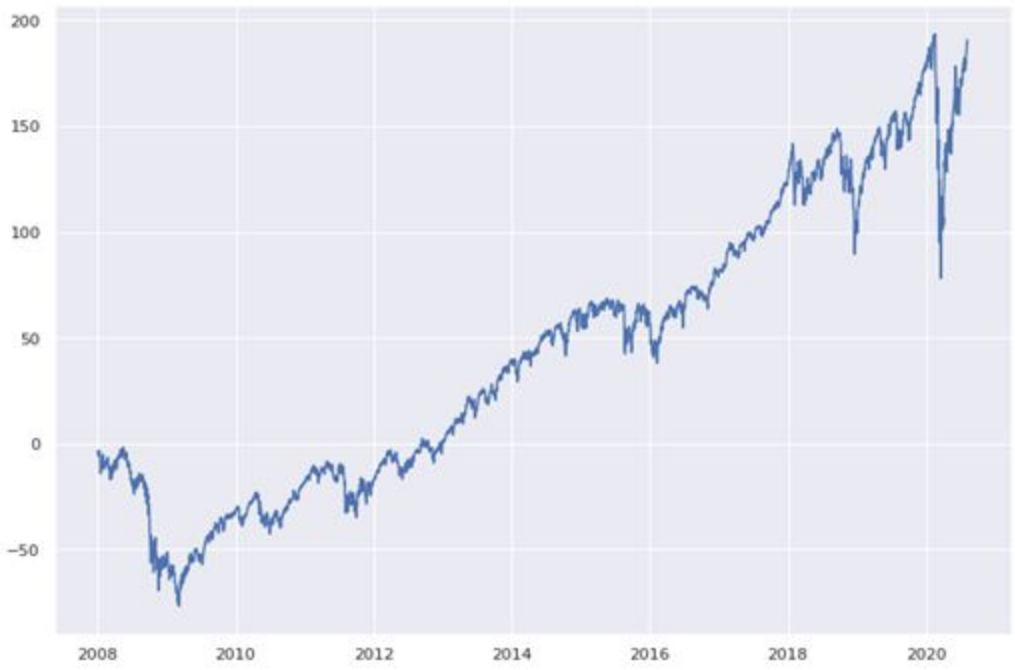
5.2) Results:

As we saw in the previous sections, we can change a lot of parameters.

The features calculation, number of days in history to train on, number of days to test until training again, and the number of clusters. Therefore we have endless different trading strategy possibilities.

One of the trading strategies that was created using this method yields the following cumulative gain curve:

While the market during period of time had the following gain curve:



As we can see, until 2018 this strategy out-performed the market in almost every aspect: better risk/reward, smaller draw-downs, more free liquidity and higher returns (adding the compound Interest will create higher returns than the classic buy and hold strategy).

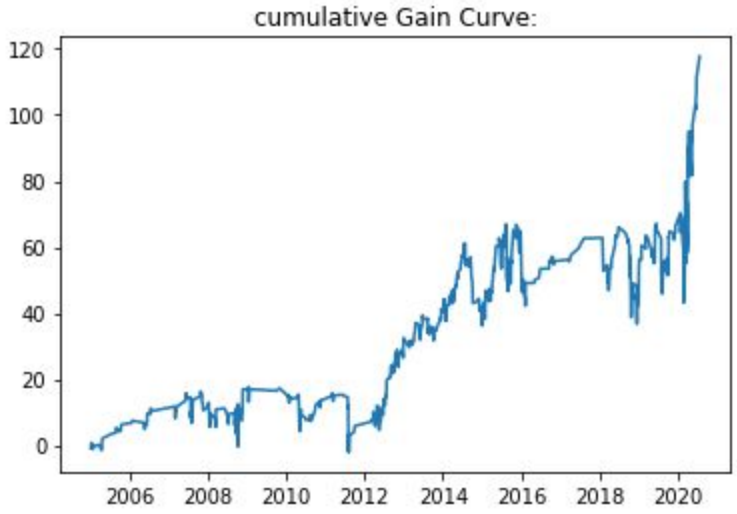
However,since 2018, the markets became more volatile, and this strategy stopped performing well in the new market behavior.

6) Model Weaknesses and Future Work

6.1) Model Weaknesses:

As seen in the previous sections, the model needs to be trained on the same market behaviour data in order to get more precise results regarding winning days on different market behaviors.

Another example is the following strategy:




```
number of trades: 703
mean to std ratio: 1.6954900371257449
```

We can see that during the “bull” market of 2010 – 2018 this strategy performed poorly compared to the market, but when there was the big volatile market of 2020 the strategy performed significantly better than the market buy and hold strategy.

Another disadvantage of the strategy is the fixed exit.

The strategy only exits at the close of the next trading day. This means that when there is a strong sell day, the strategy will exit the long position only at the end of the day, instead of at a stop-loss point.

6.2) Future Work:

The strategies discussed were just the beginning of a full functioning trading strategy based on K-Means.

First of all, short positions can be added when there is a strong “bear” cluster.

This will make the strategy much more robust to different market conditions,

and also create profit possibilities on “bear” markets.

In addition, add stop-loss and take-profit points can be added , to lock profit, and cut losses fast, as mentioned in the previous section.

Another idea is to input a LSTM neural network with the clusters, and more features, and let the network classify the up vs down days.

One of the simplest possible improvements is to add an indicator threshold as another decision condition. For example, specific market momentum.

7) Conclusion

The trading strategy idea presented in this project,is based on unsupervised learning.

Using the simplest features and position entering conditions, it has been shown to be a strategy that outperformed the market on set periods, however,still unstable on others.

This work is a basis for future research and development on trading strategies that can combine unsupervised machine learning and technical analysis.

8) Bibliography

A framework for trading system development based on k-means clustering

K Means Clustering and Creating a Simple Trading Rule for Smoother Returns

Stock Picks using K-Means Clustering | by Timothy Ong | uptick-blog

K-Means Clustering of Daily OHLC Bar Data

k-means clustering

VIX