CEU Department of Economics and Business
CENTRAL EUROPEAN UNIVERSITY

Capstone project – Public Summary
AY 2021/22
Spring semester

A Quantitative Approach

Analysis and Enhancement of a Simple Trading Strategy based on K-Means Clustering

Ligeti, Máté

Capstone Project Summary

The Client provided the Project with clear task description:

- Find a quantitative strategy idea on public ‘quant’ forums or websites (e.g.: Quant News, Quant Net, QuantConnect, Wilmott, NuclearPhynance etc.)
- Build the strategy in Python
- Analyze (backtest) it on historical data for a set of major FX-es
- Evaluate its performance and potential for further use (including enhancement of the strategy with different configurations and ML tools)

The article we chose was *K-Means Clustering and Creating a Simple Trading Rule for Smoother Returns* (Bergstrom 2018) published on QuantNews. It has described a simple strategy in which we take the volume and volatility data of an asset (SP500 eMini futures in the article), and cluster the trading days with K-Means clustering into volatility/volume groups. We assume that middle volatility/volume clusters indicate superior returns for the following day. We always invest 1$, hold it for exactly 1 day, then close the position. We enter Long, whenever our select cluster “gives signal”.

As a first step, we replicated the strategy to validate the published results and see if it still holds for FX instruments.

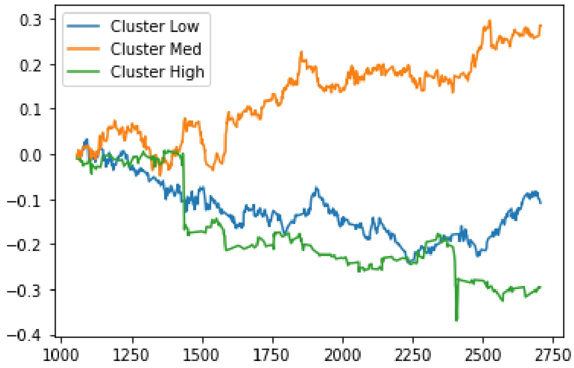

Figure 1 - GBP/USD equity curve for the original strategy

We saw that for certain instruments this ‘goldilocks’ volatility cluster may present itself, but the results were inconsistent across assets – we needed further research.

Our initial approaches involved sophisticating the clusters and making a comparison of the K-Means (automated) clusters with manually set statistical quartile clusters (later referred to as ‘expanding’ method). In this method we had classified all our historical trading days into 4 volume and 4 volatility groups by comparing the days’ said metrics to expanding quartile limits of prior trading days.

The same results occurred – we had good performing clusters, but not consistently in the same volatility range across assets. Noticing that the original strategy always went long on signal, we tried to further sophisticate the model by adding sign (+/-) predictions for the model. Now we did not only enter market, when we were in the select volatility/volume range, but also only when we got either positive (entering Long) or negative (entering Short) predictions. This resulted in ambiguous results again – for Long position the filter albeit limiting the best performing cluster, it also seemed to improve more clusters’ performance in similar volume/volatility ranges. However it didn’t work vice versa. Short signals produced best results in unexpected clusters and great losses in others.

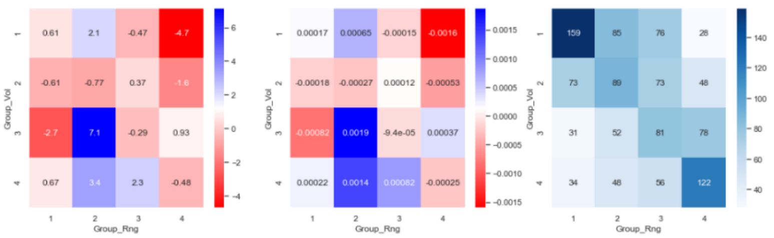

Figure 2 - Quartile cluster performance with the manual cluster method.. From left to right: (1) Simple returns (2) Risk -adjusted returns (3) Numerosity

Together with the Client we decided to:

1. **Examine the methods** (manual clustering based on expanding quartiles, K-Means, random forest, linear regression and boosting) **separately** and not in combinations (see ML +/- filtering for manual clusters)
2. Instead of looking at results in a cluster level, we used **all methods for prediction** and **let them go L/S based upon their predictions**
3. Use **single PnL metric to compare each methods**, instead of looking at cluster performances – because we use clusters' historical performances to decide on a L/S entry, this single PnL metric should hold all the clusters' individual performances

For our final analysis we have built 5 different models: besides the original K-Means, manual clustering ('expanding'), 3 Machine Learning models – Elastic Net, Random Forest and Boosting.

To produce predictions for the non-ML/half-ML methods (K-Means was only used for clustering, not to predict returns) we have set historic performances of clusters as an expected value – thus retrieving predictions. The **PnLs were then calculated** as:

$$mathematical\ sign\ (+/-)\ of\ our\ predictions\ \times\ actual\ price\ change\ =\ PnL$$

We compared results on equity (PnL) curves and calculated model performances by **3 PnL metrics**:

1. Simple (non-adjusted) annualized returns
2. Risk-adjusted returns (Sharpe-ratio)
3. Drawdown-adjusted returns (Martin ratio or Ulcer Performance Index)

After our enhancements so far – increasing clusters, adding squares and cross product to features, implementing 5 different models (including 3 Machine Learning tools), differentiating Long/Short entries – we have attempted to improve results further by trying to **engineer more volatility features** (see 'boost2' model).

After these analyses were automatized we have **run the simulation across 7 major pairs** of the US dollar to see if we can recognize some patterns across assets.

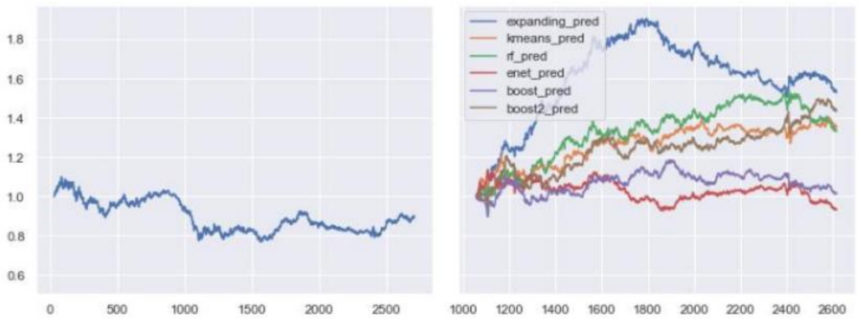

Figure 3 - Model returns on the EUR/USD. From left to right: (1) Benchmark curve of market price changes (2) Model PnLs

The findings showed that our overall returns have fallen behind benchmark returns on a risk-adjusted and non-adjusted basis too and were inconsistent across models and instruments. In the final comparison on an annual base:

- Highest simple return was 7.3% (S&P benchmark return between 2010-2020 was 12-14%[1])
- Lowest simple return was -6.1%
- Highest risk-adjusted return was 0.83 (hedge fund benchmark 2-3[2])

[1] https://www.sofi.com/learn/content/average-stock-market-return/ (Retrieved on June 10th, 2022)

[2] Client information

- Lowest risk-adjusted return was -0.79
- Highest drawdown-adjusted return was: 3.4
- Lowest drawdown-adjusted return was: -3.7

Even though we did receive some better results, never consistently. Superior returns presented themselves:

- On occasional simulation runs. As all our models – besides the expanding method – consisted of a degree of randomness in model building, yielding slightly different results on each runs.
- On different models. We couldn't recognize consistent patterns in comparative model efficiency.
- On different assets. We couldn't recognize consistently better performances on particular assets.

Finally, we concluded this to be an overall poor strategy, not worth considering for investment. In our opinion this can be attributed to 2 major things:

- **FX is** on hand **a zero sum game** without value creation, so it is very hard to pick up some behavioral or macroeconomic patterns even for Machine Learning algorithms, which to an extent might also underpin that FX markets are efficient
- **The strategy was too simple** – Predicting market movement from only volatility and trading volume date might contain insufficient information for our learners to make judgement on consistent patterns. This we could see from model optimization – it was really hard to configure the models in a way to have even a positive test score (that the predictions are significantly accurate), while receiving differentiated predictions instead of a constant.

Our recommendation for further research:

- In terms of randomness, try to **further investigate model configuration** – this research implemented model optimization on a select instrument (EUR/USD), which was then applied to other instruments as well. Albeit it means excessive work, one could rigorously go over the control parameters of the models across multiple instruments, take note of the optimal configurations and reconcile the results. This would be a cross-asset "regularization" to prevent overfitting the models on assets (instead of train data in its original meaning).
- Focus on **1 asset** instead of optimizing for an asset class - the underlying macro factors might have too much influence to have universally good strategy for FX-s
- Focus on **1 model**
- Focus on **1 cluster** – Even though in the process we wanted to have a better understanding of model performances across assets and ditched the idea of selecting cherry-picked clusters, but running the same analysis on many assets to detect these 'goldilocks' and investigating them separately might not be a bad idea. Some of these outlier performances might hide to underlying behavioral / macroeconomic patterns which can be picked up by a learner and exploited for an alpha