Project origin

As a project manager I used to deal with a lot of investment process and regulatory topics. Part of our funds are under the UCITS directive which describes actors (fund manager, accounting, ManCo) behaviour to allow EU-wide marketing.

There are a lot of rules and this project focus on the cash management Cash invested by the final investor must be invested on the market regarding the fund strategy. The fund manager cannot keep high lever of cash position. In the opposite way, when investor claims for redeem, the funds must assess the request with a few days delay.

So, cash management can be struggling, having too much in flows force the manager to invest (maybe it's not the right moment to do) and getting redeem request might force the fund manager to sell some securities (maybe not at the right time).

The target of my project is to propose a model to predict a probability of inflows or out flows (or neutral) relative to a set of observations.

Time Series to Supervised Learning problem

It was one option to deal with the project as a Time Series forecasting model, but I decided to apply a method of Time Series transformation to a Supervised Learning problem and make classification modelling project.

One of the main pieces of my project is a Custom Transform that can be applied to the Time Series to obtain a new DataFrame. Method such as shifting value, computing rolling window, calculating expanding value are part of the Custom Transform which receives a configuration set and return a new dataset.

The project

I started the project with different interviews to get insight about the data I can collect. Pictet Human's fund managers helped me to scope the project. I also met colleagues from performances teams, marketing, sales, and economic analysis.

Build the dataset.

[notebook-01, notebook-02, notebook-03]

I collected data from several systems (Datawarehouse, operational databases, Excel document). The pivot key for the time series dataset was the funds, the year and month of flows. All data have been concatenated in the full Time Series file with or without transformation.

- Economic analysis gave me an indicator per year/month per region, so I have to compute the country exposition of the funds for a specific month, took the most relevant and apply the score.
- From our CRM, I got data relative to events (sometimes linked to a specific funds or not). Some event is linked to a strategy or some dedicated to a speaker. I mapped the score to the relevant observation.

The dataset has anonymized (only print of the run will be supply in the git folder) to comply with the bank rules. The final notebook of this phase is runnable.

Exploratory Data Analysis

EDA is big part of a ML project. I created 2 notebooks [notebook-04a, notebook-04b] to explore the Time Series dataset. In the 1st notebook I managed missing value, make a descriptive statistic, and import part I created the target value by computing the ratio of net flows Vs. the AUM (asset under management). Small in flows or out flows have no impact on the cash management. I set the threshold to +/- 1% (by tuning this parameter +/- 1.2% I can get a balanced target).

In the 2nd notebook, I tried to highlight patterns such as trends, seasonality, and other component of a Time Series. This notebook contains multiple plots to highlights Time Series observation.

[notebook-05]

In this 5th notebook a made part of features engineering. I encoded some categorical features such as portfolio risk level (from low risk to high risk). All attributes related to fund classification (strategy, name, country or asset class) have been encoded as well.

I computed some feature base on dates such as the fund age at the moment of the observation or the fund manager experience (in month).

I started to explore Time Series Transformation to Supervised Learning [TS-2-SL] method. The final version is held in a library [mytransformer.py] and will be loaded in the modelling notebooks.

The transformer can:

- Shift data
- Provide a rolling window with an average computation.
- Get an expanding transformation with the mean aggregate function.
- Provide an ewm (exponential weight function) by applying the mean.

This custom transform is seeing as a feature engineering process and I created it as a class (Custom Transformer). Possibility to run it in a pipeline.

Evaluation of the Custom Transformer

[notebook-06]

In the 6th notebook, I explored several combinations of transformation by applying method to different set of columns and plot results. This analysis helped me to decide which type of transformation I would work with in the modelling notebook. For example, rolling transformation seems to provide better result than shifting the data. I did not explore too much the expanding and ewm method.

In this notebook, I calculated for each observation (row) a linear regression with the NAV and benchmark volatility and performance. I took points (1y, 6m, 3m and 1m) and calculated the coefficient with a polyfit of degree 1. The result is stored as new feature.

Finally, I manage to deal with outliers and remove them before saving the final. I worked with Sklearn Isolation Forest.

I used to work with sweetviz library which provides in a few lines of code an complete html files with data visualization.

Modelling & Results

[notebook-07]

I started by importing my custom transformer and test it. I apply the rule to fit the transformer with the training and validation data (<2020) and then transform separately the data to avoid data leakage in the test set.

Then, I had a look on my target variable to validate that the scope of observation is balanced (after outlier's management) and plot distribution.

I consider data before 2020 as training and validation set and data after 2019 as test data.

Modelling will be done on the training set and score and evaluated on the validation set. Finally, when the model will be ready and saved, I will run it on the test set.

Modelling strategy

I decided to evaluate 4 types of custom transformation. For each type, I ran a cross_val_score to score the classifier with the transformer.

Transformer types are:

1. Let Sklearn select features and then apply a rolling strategy ([5, 15, …, …] features)
2. Decide by myself with domain knowledge the features to add in the transformer and modelling process.
3. Apply the transformer on 1 unique column.
4. Apply the transformer on a lot of features to develop a high dimension dataset and then apply a PCA to get orthogonal dimension (2)

I evaluated these transformations with different classifiers and plotted the results. On the top of that I decided to keep the Logistic Regression as classifier and the 2nd type of transformation, based on my experience of the area.

Grid Search

The I applied a grid search strategy with parameters to fine tune the Logistic Classifier. The best estimator gave me a result close to 70% accuracy.

Finalizing the model

With the result of the grid search I finalized my modelling process. I did not use my custom transformer in the pipeline because each iteration would have called the transformer (huge time machine) but to finalize the model I reuse the transformer (slightly modified) to include it in the pipeline and run the steps.

The result is stored using pickle on the disk as a model file that I can reuse.

Cherry on the cake 😊 I can run several configurations of the finalized model by choosing columns to shift, roll, expand or compute exponential window and stored them for reuse.