# AI for Finance: Futures Contract Pricing Forecast

Matteo Minardi - [238789]

July 2023

This document serves as a reference for all the steps computed in the project. It reports all the comments that are written in the notebook file, with the respective graphs. Tables and output predictions are not reported here, see the slides or directly the code file if those are needed.

## 1 Introduction about the project

The project consists on trying to predict the price of a future's contract based on multiple approaches.
It's common knowledge that prices of items in the market are almost impossible to predict, since the market isn't based on predefined statistical patterns, but this project shows how someone could try to solve this task with different input ingredients and draw different conclusions.

More precisely, the objective will be to predict the price of one of Apple's future's contracts.
The data is taken from the Eikon database, so it is really high quality and needs little-to-no cleaning.
Before going deeper into the solutions, this notebook also provides a general, yet still unrealistic, intuition into the problem.

Multiple regressions have been tried, with *different approaches*:
- Based on old stock prices
- Based on old percentage returns
- Incorporating the market and Sentiment Analysis of the news system

For each method, an OLS has been computed to check the R squared values and the significance of the parameters.

With each approach, two methods have been performed: one not-so-fair, trying to predict the future values relying on future data that wouldn't be available in a real life scenario, just to show and ideal case, and a second one done in the correct way but obviously on a shorter forecast window.

Each approach will be discussed in a more exhaustive way in its dedicated section.

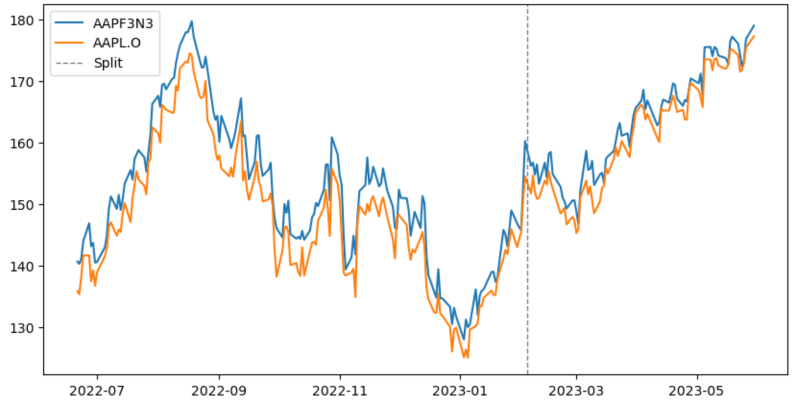
## 2 Boilerplate Code

This small section allows to connect to the Eikon database in order to retrieve the necessary data.
For this initial part, the only two necessary series are the prices of both the specific future's contract and the price of the Apple stock.

Downloading the time series about Apple stock prices and the future's contract prices, and in addition computing the SPREAD
Printing the obtained data at the end of this section.

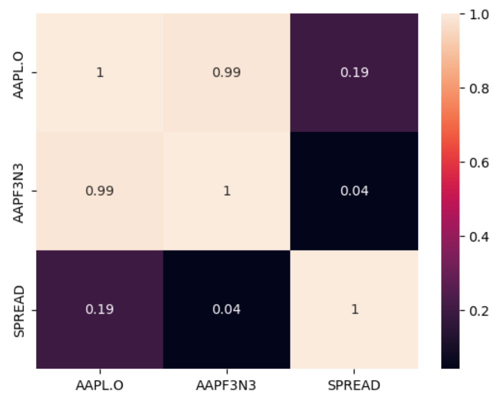
## 3 General intuition: ideal (unrealistic) case

Visualizing *ideal* results, assuming to split a hypothetical dataset right where the dotted line is placed, trying to predict the evolution of the price of the contract, training it on the price of the main stock, and seeing how they might behave in the future.



Visualizing correlation between series, it's even more clear to see how much the price of a future's contract depends on the price of the main stock, with the absolute SPEAD resulting less important than expected.

This is surely because the $R^2$ is already explanatory enough and the absolute difference in price is not that important, since we just care if the prices themselves are just going to increase or decrease.



# 4 Regression using old prices

Actual regression trying to predict the price of a future's contract given the price of the main stock, using a Random Forest estimator.

This estimator is great because it's able to generalize in a very simple way to almost any kind of data, and since this regression isn't particularly difficult it will work perfectly fine.

The analysis has been done both in a not-so-correct way, which means trying to predict the future's contract prices assuming to know the prices of the Apple stock in the future, and another, with a window of just one prediction, where it is correctly assumed to not know anything about the future, as it is in reality.

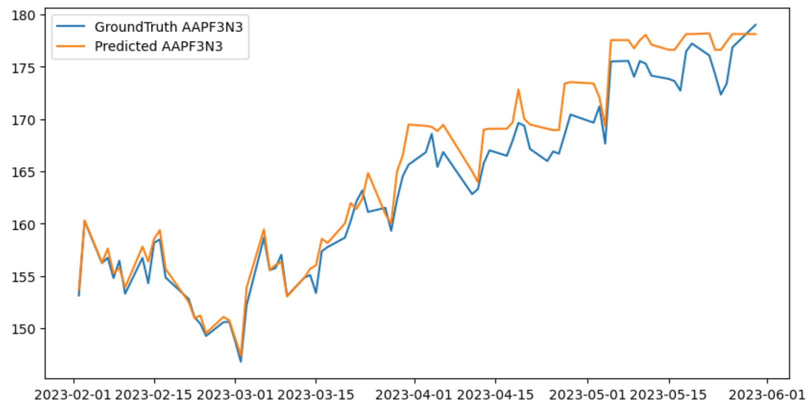
Defining what we will use for the prediction, and sub dividing the data set into a train set and a test set, that will be used to test the accuracy obtained.

## 4.1 OLS model for econometric statistics

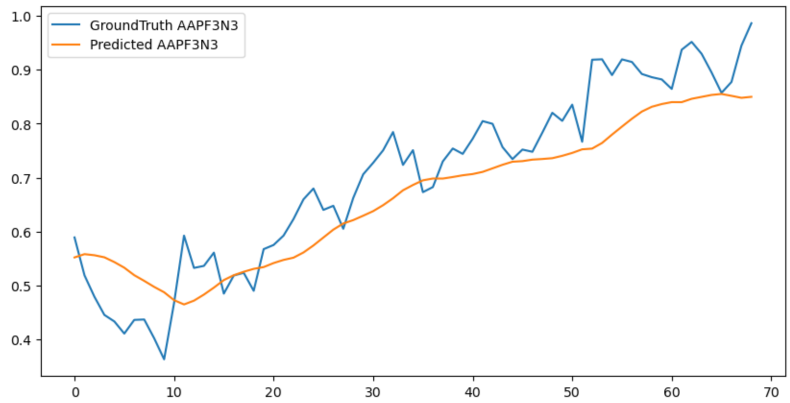
Given the fact that the price of the Apple stock and the contract have almost a correlation of 1, it was expected that also the R-squared was going to be 1, since obviously the Apple price is extremely significant for the estimate.

## 4.2 1. Not-so-fair prediction

First, the not-so-fair prediction: it's clear to see that if we could know the prices that the Apple's stock would assume in the future, it would be pretty easy to compute the related future's contract prices since they are extremely correlated. Unfortunately this is not how it works in real life.



Now also showing ad unfair training using a LSTM, that will soon later be used in a fair way, just for a quick comparison.
The model is trained to predict a single output value given the series of the 10 previous input values, so its output will tend to have a move average-like evolution compared to the actual observed value.



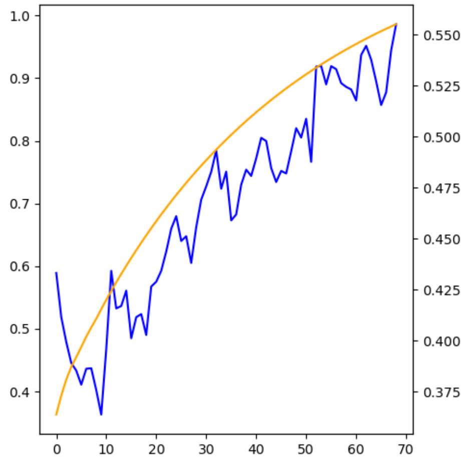
### 4.3 2. Fair prediction

Now, for a real life case approach, the price of the future's contract will be estimated using only truly available data.
This means that the future prices will be predicted only using prices available until "today", hence why only one future time step can be predicted.

In order to be able to achieve this, a lag to the output series is necessary, so that the model can learn to predict the future.
Only a single lag will be applied since it will only output a single value, that we can compare with the last ground truth of the downloaded series. And now it's possible to run a simple regression, like the preceding case.

One could also decide to try to predict even further in the future, using previous predictions to influence later ones. This approach can sometimes be meaningful, but generally it's not recommended since every time we make a prediction we are introducing some error, and using those values to predict other ones might generate even more errors along the way.
Now trying to predict future prices only given the prices of the single contract, thus trying to make that single time series predict itself.

To make this happen, window sequences are generated, so that it's possible to slide that window across the data and try to predict each following value given the 10 values before, and using predicted values as input to even further ones.
As expected, the results are not perfect but still pretty ok.



# 5 Regression using old returns

Returns are a way to analyze the performances of a stock, without the need to look at the absolute values of the prices.

This approach is definitely more general than the previous one because it allows to compare two different stocks that might be behaving in the same way, but it's not easy to notice since their absolute values might be on completely different scales, probably because the sizes of the related companies are too different. So percentage values sometimes are extremely helpful.

In this case it has been noticed to *not* be as effective as anticipated. Returns are computed taking the difference between a price at time step $t$ and at time step $t-1$, and dividing it by the value at time step $t-1$, so each value represents how much a price increased or decreased, compared to the previous time step. In short:

$$r_t = \frac{p_t - p_{t-1}}{p_{t-1}}$$

Since this operation may introduce hard-to-work-with values, the means of both the Apple returns and the SPREAD are being computed so that they can be used instead, without causing significant damage to the series.

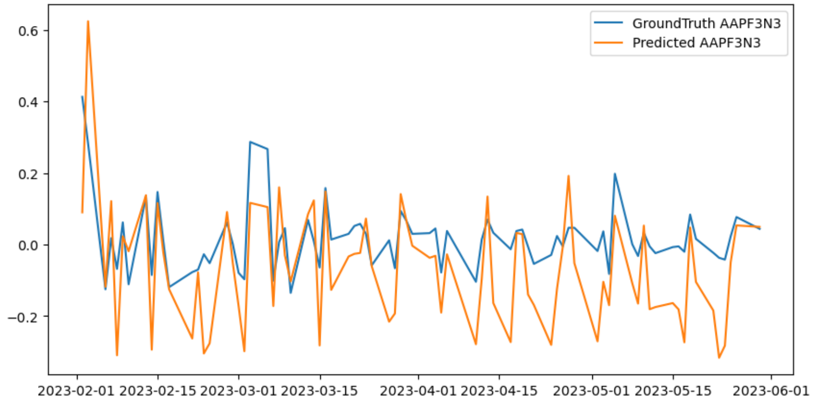
## 5.1 OLS model for econometric statistics

Now that the highest correlated parameter (Apple stock price) is not present, we can see that the R squared decreased a lot since the returns are a lot less explanatory.

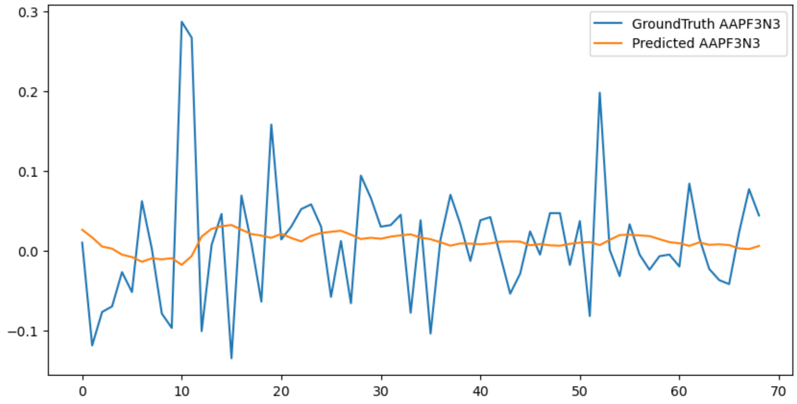
## 5.2 1. Not-so-fair prediction

Comparing the results to the previous case, we can see that the accuracy score has decreased a lot, but this is just derived by the fact that we are now working with percentages.

In fact, looking at the correlation between the predicted values and the ground truth ones, it's clear to see that the results are actually pretty good, since in reality we care only to know if the prices are either going up or down.



Again showing the behaviour of an unfairly trained LSTM model with this kind of data.
As mentioned before, since in these situations LSTM models tend to assume an average-like behaviour, it's returning values very close to zero, being able to follow some increasing or decreasing patterns but not following them to their extremes.



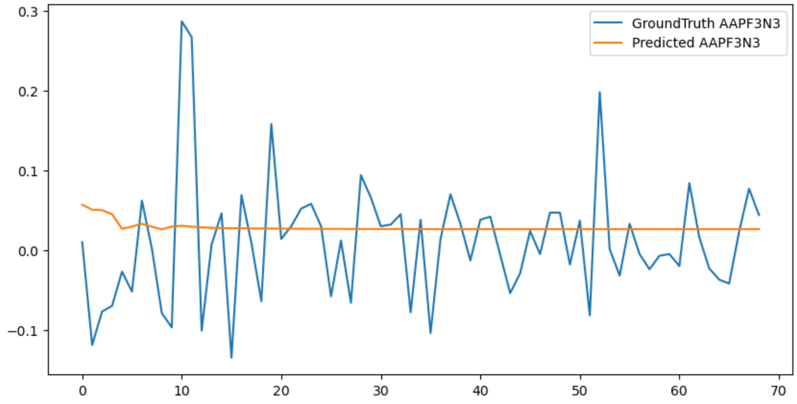
### 5.3 2. Fair prediction

As in the previous approach, it's more fair to try to predict only a single future value, taking in input only available data, like in an applicable scenario. Again, it's needed to apply a single lag to the input values for the same reason explained before.

And now it's possible to run a simple regression, like the preceding case.

To replicate the previous reasoning, it has been tried again the sliding window regression approach.

In this case, results aren't as satisfying, but this is usually what happens in real cases. The model is able to estimate the correct behaviour at the start, but right after it doesn't know what to do so it flattens.



# 6 Incorporating Sentiment Analysis and SP500

A more advanced approach would be to consider more information.

Sentiment Analysis is the process of 'computationally' determining whether a piece of writing is positive, negative or neutral. It's also known as opinion mining, deriving the opinion or attitude of a speaker.

Since it's known that the market is heavily influenced by people's feeling and fears, it would be appropriate to start considering a sentiment analysis over the news of the specific stock and use it as additional information for the final prediction, since, for example, bad news are probably very correlated with losses in the market.

In order to try to increase the performance even more, it has been decided to also consider the market performance, using the SP500 index. This could be useful to have a sort of baseline for performance comparison.

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media.
PUNKT is a tokenizer.

Downloading news form January,1 2021 to June,1 2023 about both the Apple stock and the SP500, in order to have a meaningful amount of data to work with.

One cell just shows one of Eikon's issues: it does not let users download data older than 15 months, so the biggest amount of collectable news for a given stock is 100 (for each month) times 15.

In total, we have of $100 * 15 = 1500$ news for both Apple and SP500, for a combined of $1500 * 2 = 3000$ articles to analyze.

## 6.1 Computing polarity scores

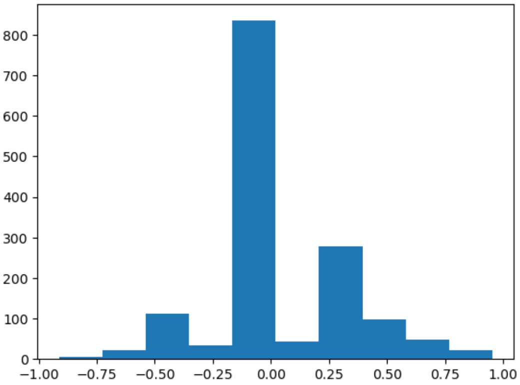
Each headline will be assigned to a value between $-1$ and 1.
For instance, a positive value refers to a positive sentiment detected. A value close to zero implies a neutral case.

For each news set, we get a table with the structure of the following one.
It's possible to see how the vast majority of scores are neutral.
The compound column basically represents an aggregation of the preceding three, being able to represent a general idea underlying a certain stock.

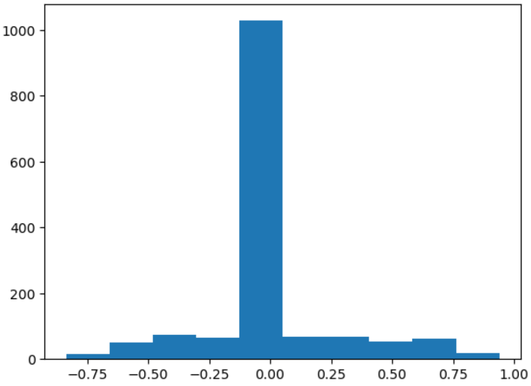
Plotting the compound distribution of both SP500 and Apple.
Again, this shows how many distinct news are labelled with the same polarity,

for each polarity score.

As anticipated, the biggest amount of news are neutral, but not in the same way.



For SP500, the values are slightly distributed with a little skew towards the positive side (+1).
This implies that for the SP500 news are typically neutral or positive, while for Apple they are very much more evenly distributed, making it more difficult to say to which side they tend to lean on.



## 6.2 Integrating cumulative sum of sentiment scores

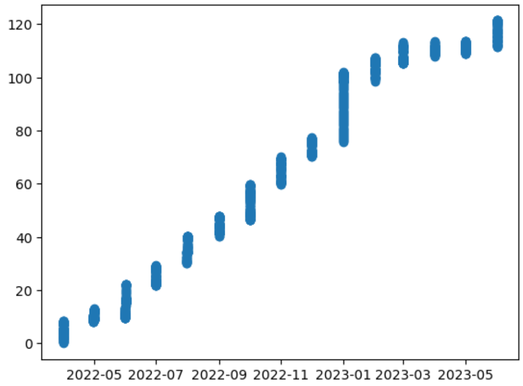
Summing up all the sentiment scores, grouping them by their date, allows to compute a general sentiment score for each time step.
This measure is used to identify particular moments in which the general public opinion changed, meaning it increased by a significant amount or decreased

surprisingly.

Its functioning is almost self explanatory: if the sum of the sentiment scores for each day is greater than zero, it means that it was a positive day, while it's the opposite for the negative case, and stable at zero for the neutral one.

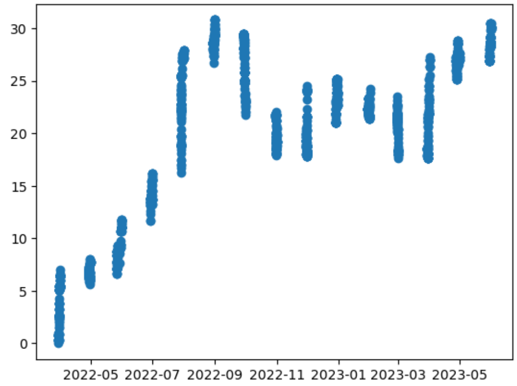
This first graph shows the almost-linearly increasing cumulative sum of the sentiment scores of the SP500.
From this graph we can notice two main dates in which the news generated extremely positive scores, which resulted in two jumps on the graph.

As expected, this graph is very stable, because it's very unlikely that something is able to shift the entire market, unless it's really groundbreaking or extremely influential in the way everyone lives.



On the other hand, the cumulative sum of Apple's sentiments is much more volatile.
This is completely normal as a single stock will always be more volatile than the entire market, since the second one is basically the mean of all the stocks.

In this case it's easy to spot both some positive trends and negative ones, as expected.

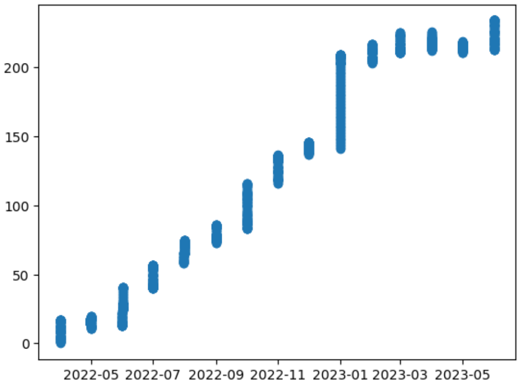


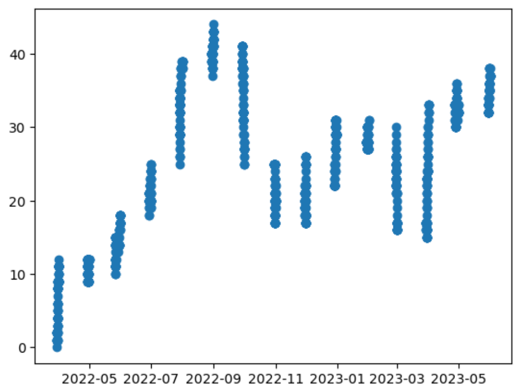
## 6.3 Quantizing sentiment scores

This operation subdivides all the different values of the sentiment scores that are originally continuously distributed between $-1$ and $-1$, into 3 bins where they assume exactly either $-1$, 0 or 1. So it discretizes the continuous domain in order to make it more simple.

The two following cells show how the values have been subdivided.
Since we are discretizing into 3 bins of the same size, it has been decided to subdivide the domain into thirds.

This is how the cumulative sum of the updated domains look like.
We can see that their behaviour is not significantly different than the original one, hence we can conclude that we did not alter the information in a bad way and that most information has been preserved correctly.

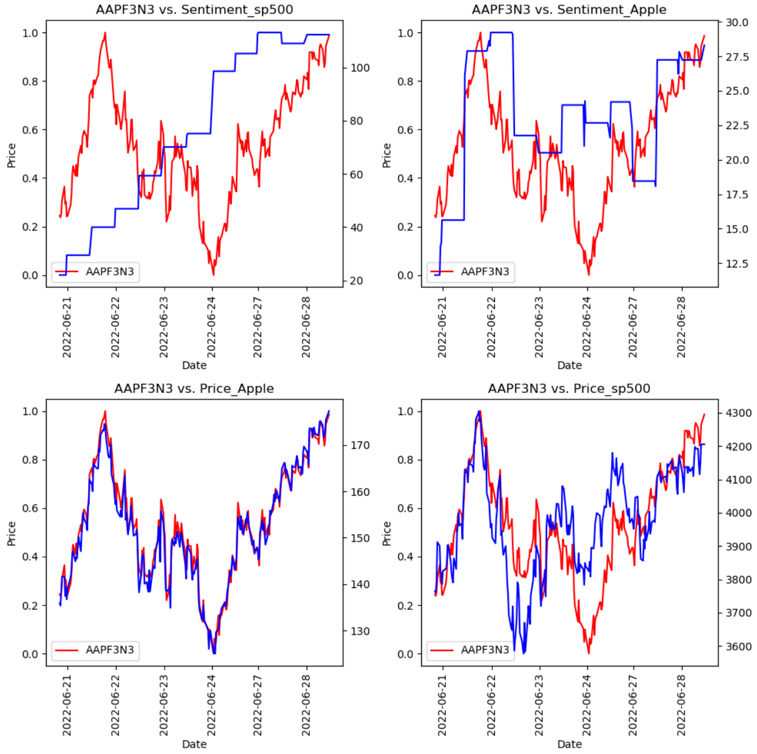




## 6.4 Confronting the patterns

Now that all the useful information is available, let's plot all the series against each other to spot any useful patterns.

First, we need to aggregate the values of the cumulative series just computed. In this case it has been used the mean to have a stable estimate.

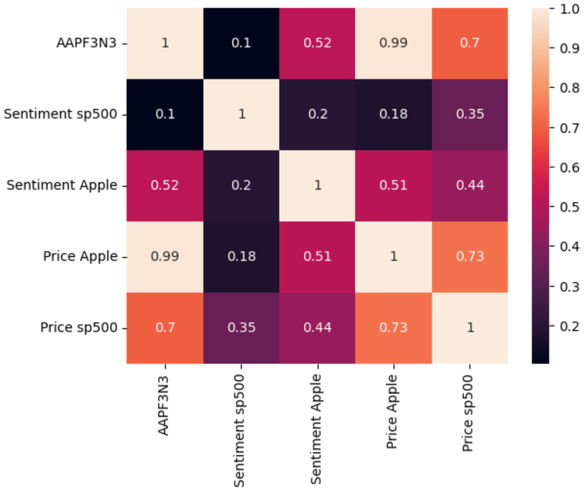


## 6.5 Organizing the data for the last predictions

Merging them together into a single data frame so that it's easier to compute the future predictions.

Also plotting the pair-wise correlations of the new series.
In general, the shared information between variables is definitely significant. Excluding the correlation between the future contract's price and Apple's price discussed earlier, it clear to see that also the sentiment analysis could be very useful.

As expected, the sentiment analysis of the Apple side is clearly more significant than the general SP500 one.



| | AAPF3N3 | Sentiment sp500 | Sentiment Apple | Price Apple | Price sp500 |
|---|---|---|---|---|---|
| AAPF3N3 | 1 | 0.1 | 0.52 | 0.99 | 0.7 |
| Sentiment sp500 | 0.1 | 1 | 0.2 | 0.18 | 0.35 |
| Sentiment Apple | 0.52 | 0.2 | 1 | 0.51 | 0.44 |
| Price Apple | 0.99 | 0.18 | 0.51 | 1 | 0.73 |
| Price sp500 | 0.7 | 0.35 | 0.44 | 0.73 | 1 |

## 6.6 OLS model for econometric statistics

Again, this model seems to have a very high R squared, probably due to the fact that it contains the Apple stock price.

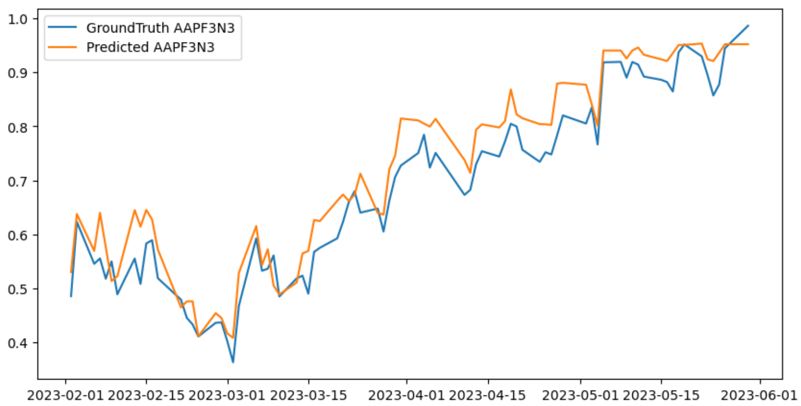
Just for the sake of the analysis, these are the results when the Apple stock price is removed: the R squared decreases a lot.

## 6.7 1. Not-so-fair prediction

For the last time, trying a simple regression using all available values.
Results are a little less accurate than the very first case because the 0.99 correlation with the Apple's stock price is diluted by the other variables.

A more correct analysis should be done removing the Apple's stock price to see the effective results of the sentiment analysis but I decided to not deepen this further since it's not worth giving much importance to an unfair prediction.

A new LSTM unfair prediction is not useful in this case since it would be the exact same as the one computed at the very first step.



### 6.8 2. Fair prediction

Ultimate single value prediction using all the additional information in both the random forest and in the LSTM.

A sliding window approach in this case would again be the same as the very first one computed so it has not been done a second time.

An approach trying to predict each time step all the explanatory variables could be a possible alternative, but not really feasible since it would mean that each time too many series would need to be predicted, and that would introduce too much error.

## 7 Conclusions

Multiple approaches have been tried, some more complicated than others, to try to predict the future price of a contract.
This project showed both good and bad results about each method. In the end, as expected, we can conclude that predictions are feasible in the short term, because there isn't a big amount of unknowns, while being definitely much harder for values further in the future.

This experiment also proved how sometimes a good sentiment analysis could help in these kind of problems, but it is still not enough to have accurate predictions because of the very random behaviour of the market.

Although it might seem less accurate than the very first method, it's still probably better because incorporating more information as the independent variables in surely better because its error is much less variance influenced, so overall the last model is surely the most robust.