Draft - Chapter 1: Meta-labeling

Author: Guillermo Creus Botella
Supervisor: Daniel P. Palomar

The Hong Kong University of Science and Technology (HKUST)

Polytechnic University of Catalonia - BarcelonaTech (UPC)

Contents

1 Stationary Time Series

What should be covered:

- Stationary financial time series are really important
- Definition of stationarity

A time series $\{x_t\}$ is said to be **strictly stationary** if the joint distribution of $(x_{t_1},\ \ldots,\ x_{t_k})$ is the same as the one of $(x_{t_1+t},\ \ldots,\ x_{t_k+t})$ $\forall t$ and $\forall(t_1,\ \ldots,\ t_k)$ where $k \in \mathbb{N}$. That is, the joint distribution is time invariant.

However, as this definition of stationarity is complicated to verify empirically, a new definition will be introduced. That being said, a time series is said to be **weakly stationary** if:

1. $\mathbb{E}[x_t] = \mu = \text{ct.}$
2. $\text{Cov}[x_t,\ x_{t-l}] = \gamma_l$, which only depends on the lag l.

With that said, let's present an Autoregressive model of order p, AR(p):

$$x_t = \sum_{i=1}^{p} \theta_i x_{t-i} + \epsilon_t$$
$$x_t = \sum_{i=1}^{p} \theta_i B^i\ x_t + \epsilon_t$$
$$\phi(B)x_t = \epsilon_t$$

Where B is the backshift operator ($Bx_t = x_{t-1}$). Additionally, the process will have a unit root (integrated of order 1 - $I(1)$) when $z = 1$ is a root of multiplicity 1 of the characteristic equation: $\phi(z) = 1 - \theta_1 \cdot z^1 - \ \ldots\ - \theta_p \cdot z^p = 0$. When $z = 1$ is a root of multiplicity r, the process is integrated of order r - $I(r)$.

In order to be weakly stationary, the roots of the polynomial ϕ must lie outside the unit circle, i.e., $|z_i| > 1$

Rewriting everything:

$$x_t - x_{t-1} = \left(\sum_{i=1}^{p} \theta_i - 1\right) x_{t-1} + \left(\sum_{i=2}^{p} \theta_i\right) \cdot (x_{t-2} - x_{t-1}) + \ \ldots + \ \theta_p \cdot (x_{t-p} - x_{t-(p-1)})$$
$$\Delta x_t = \pi \cdot x_{t-1} + c_1 \cdot \Delta x_{t-1} + \ \ldots + \ c_{p-1} \cdot \Delta x_{t-(p-1)}$$

Where $\pi = \sum_{i=1}^{p} \theta_i - 1$ and $c_j = -\sum_{i=j+1}^{p} \theta_i$

Noting that $\phi(1) = -\pi$, then $z = 1$ is a unit root if $\pi = 0$. Consequently, the Augmented Dickey-Fuller test is defined:

$$H_0 : \pi = 0$$
$$H_a : \pi < 0$$

At this point, if a time series has no unit root, then H_0 will be rejected and it will be concluded that the time series is stationary.

Regarding the order p of the AR model, the package **tseries** will be used, which sets $p = \lfloor \sqrt[3]{\mathrm{T} - 1} \rfloor$ (T = length of the time series - $x_1, \ldots, x_\mathrm{T}$) as default. This value of p corresponds to the suggested upper bound on the rate at which the number of lags should be made to grow with the sample size for the general ARMA(p,q) setup.

2 Fractional differentiation

It is often the case that the original log-prices time series y_t is non- stationary. To mend that, a differentiation of order 1 is computed and the final time series $x_t = (1 - B)y_t = y_t - y_{t-1}$ is usually stationary. However, in the process, the "memory" of the time series is erased (see figure 1).

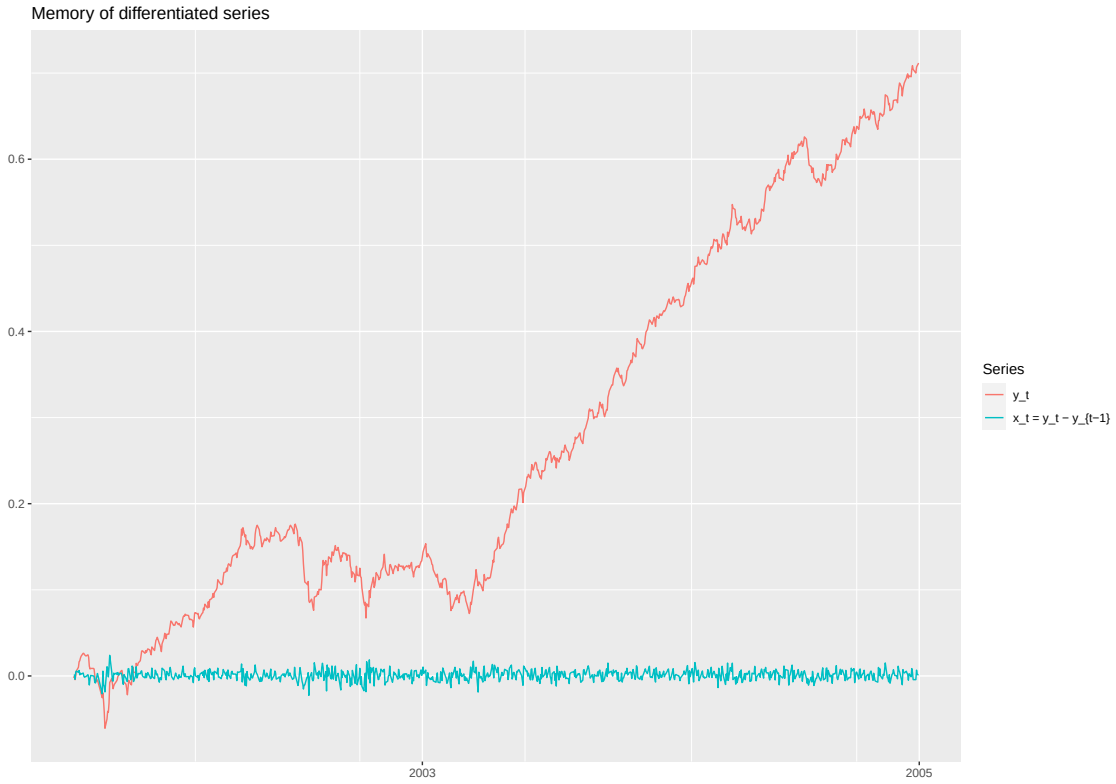

Figure 1: Memory of differentiated series

That is when fractional differentiation comes into play. It consists of defining the differentiated time series as $x_t^d = (1 - B)^d y_t$, $d \in (0, 1)$, where:

$$(1 - B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-B)^k = \sum_{k=0}^{\infty} \frac{\prod_{i=0}^{k-1} (d - i)}{k!}$$
$$= \sum_{k=0}^{\infty} B^k (-1)^k \prod_{i=0}^{k-1} \frac{d - i}{k - i} = \sum_{k=0}^{\infty} w_k B^k$$

Note that:

- $w_k = -w_{k-1} \cdot \frac{d-(k-1)}{k}$ and $w_0 = 1$. See figure 2.
- $d = 1 \Rightarrow x_t^1 = y_t$ - log-prices
- $d = 0 \Rightarrow x_t^0 = x_t$ - log-returns

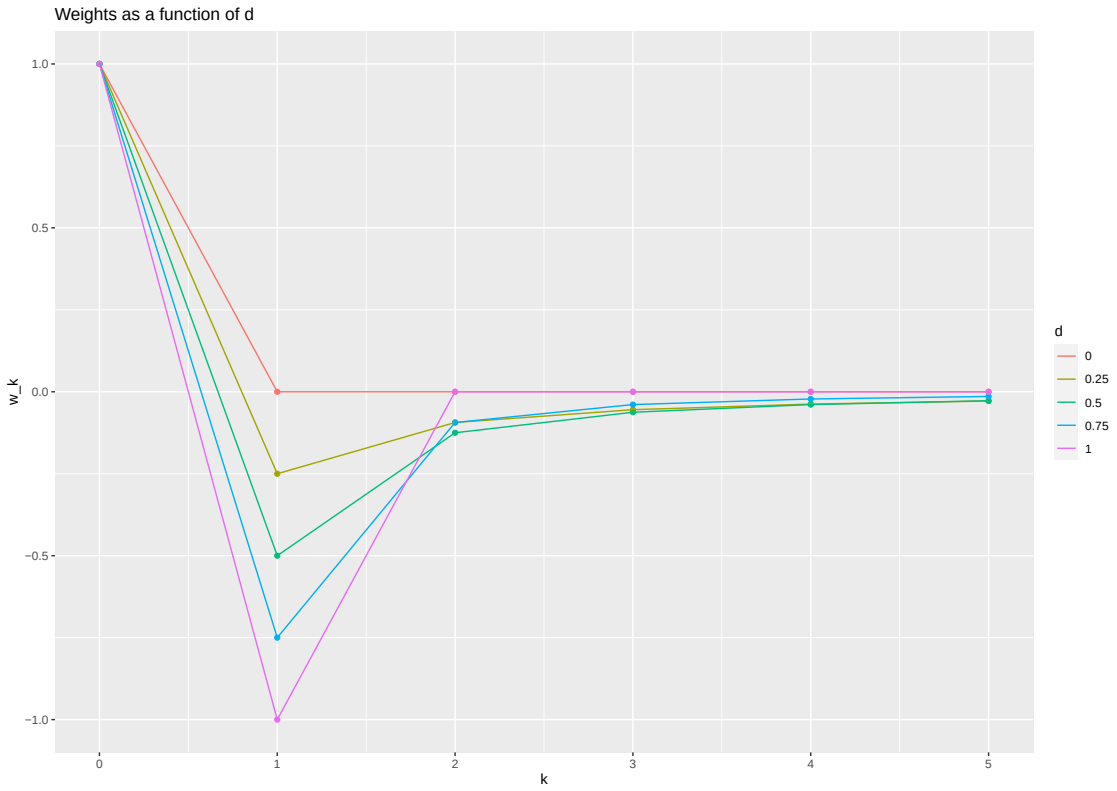

Figure 2: Weights as a function of d

Before starting to differentiate the time series, a technique called "Fixed-width window fractional differentiation" (FFD) will be defined. The purpose of this technique is to allow all the observations to have the same amount of memory. Because of data limitations, fractional differentiation can not be computed on an infinite series of weights. That is, x_T^d will use $w_0, \ldots, w_{T-1}$, while x_{T-l} will use $w_0, \ldots, w_{T-l-1}$.

Consequently, some observations will have more "memory" than others, which is not something one wants. Thus, by setting a threshold (e.g., $\tau = 10^{-4}$), the following set of weights can be defined:

$$\tilde{w}_k = \begin{cases} w_k & \text{if } k \leq l^* \\ 0 & \text{if } k > l^* \end{cases}$$

Where l^* is such that $|w_{l^*}| > \tau$ and $|w_{l^*+1}| \leq \tau$. Note that the differentiation will be valid only for $t \geq l^* + 1$ and the same set of weights, $\{w_k\}_{k=0}^{l^*}$, will be used for all estimates.

Finally, by setting a fixed τ, the FFD differentiation can be expressed as:

$$x_t^d = \sum_{k=0}^{\infty} \tilde{w}_k B^k y_t = \sum_{k=0}^{l^*} \tilde{w}_k B^k y_t \equiv \phi_d \; y_t \quad (t \geq l^* + 1) \tag{1}$$

To clarify everything, figure 3 shows x_t^d for different values of d.

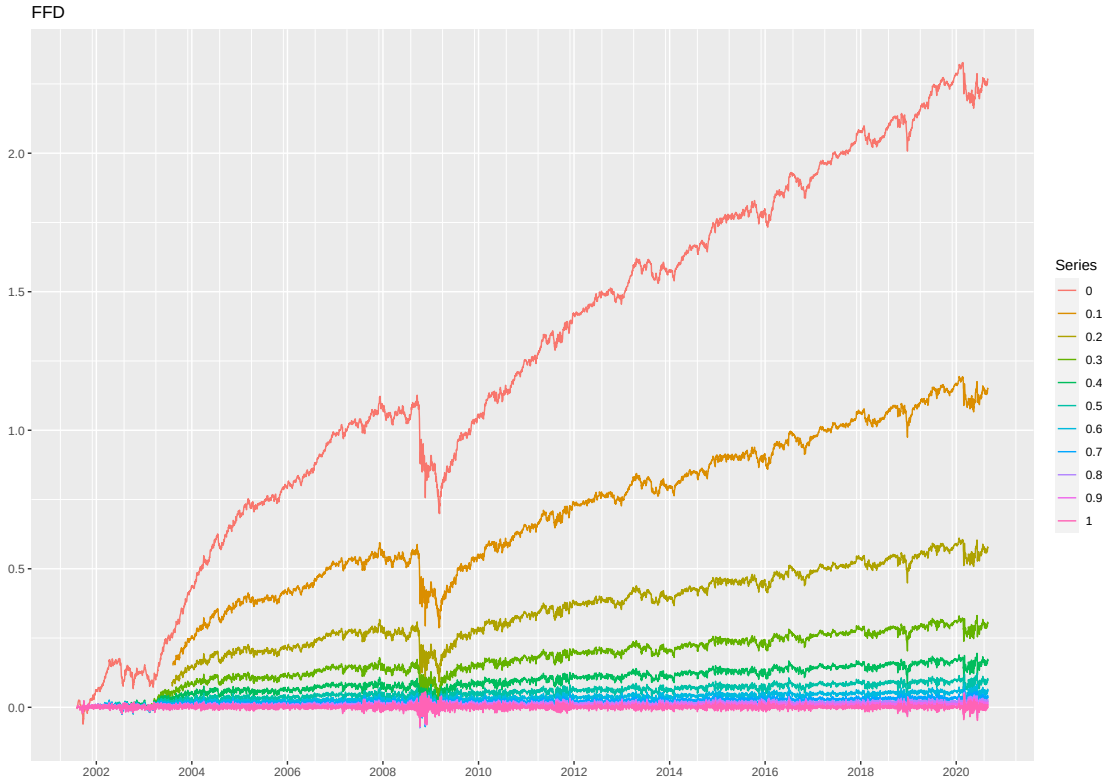

Figure 3: FFD time series - $\tau = 10^{-4}$

Now that the corresponding x_t^d time series have been computed, it is relevant to introduce again the ADF test. The objective is to find the minimum d, d^*, such that one can conclude that $x_t^{d^*}$ is stationary. In figure 4, the ADF test statistic is shown (using lag $p = \lfloor \sqrt[3]{\mathrm{T}_d - 1} \rfloor$).

Figure 4 also shows that int order to pass the ADF test, and conclude that the time series is stationary, $d = 0.2$ is sufficient.

2.1 Inverse of differentiation

In this section, the inverse of differentiation will be introduced since it is useful to go from one time series to the other. First of all, for convenience, let's define x_t^d for $1 \leq t \leq l^*$:

$$x_t^d = \begin{cases} \phi_d(y_t) & \text{if } t \geq l^* + 1 \\ \sum_{k=0}^{t-1} \tilde{w}_k \cdot B^k(y_t) & \text{if } 1 \leq t \leq l^* \end{cases} \tag{2}$$

Alternatively, if one defines $y_t = 0$ for $-l^* \leq t \leq 0$, equation 2 can be expressed as:

$$x_t^d = \phi_d(y_t) \quad (t \geq 1)$$

Now that x_t^d has been defined $\forall t \geq 1$, the inverse of differentiation comes into play this way:

$$\begin{aligned} \frac{1}{\phi_d}(x_t^d) &= y_t \\ \frac{1}{\sum_{k=0}^{l^*} \tilde{w}_k \cdot B^k}(x_t^d) = \sum_{k=0}^{\infty} c_k \cdot B^k(x_t^d) &= \sum_{k=0}^{T-1} c_k \cdot B^k(x_t^d) = y_t \end{aligned} \tag{3}$$

Equation 3 shows that to compute y_t as a function of x_t^d, only a finite number of terms are needed. That, is inherent to the definition of x_t^d $(1 \leq t \leq T)$, which for $t < 1$ is 0. Therefore, it will suffice

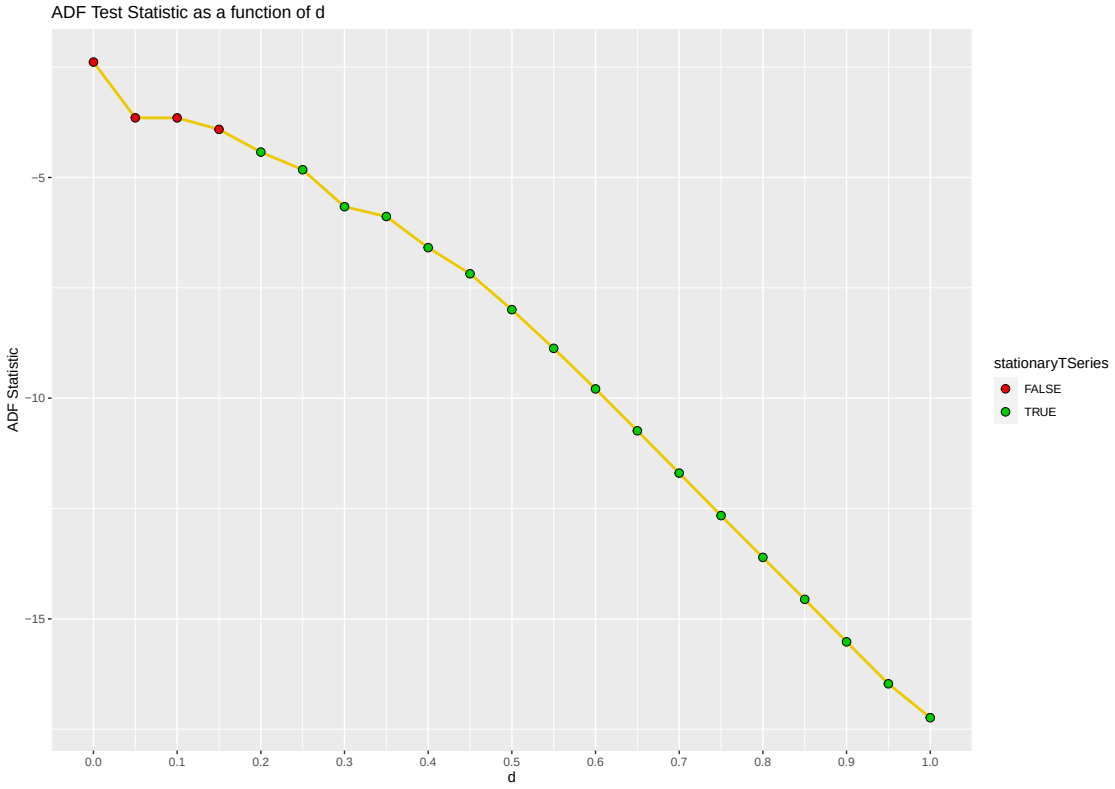

Figure 4: ADF statistic for FFD time series (p-value = 0.01)

to compute the first T coefficients of the resulting polynomial. To do that, successive polynomial long divisions (see figure 5 for the first 2 terms) will be used until the first T terms are retrieved.

$$
\begin{array}{lll}
1 & & \big| \; \tilde{w}_0 + \tilde{w}_1 B + \tilde{w}_2 B^2 + \ldots + \tilde{w}_{l^*} B^{l^*} \\
\cline{3-3}
1 \quad \tilde{w}_1/\tilde{w}_0 B \quad \tilde{w}_2/\tilde{w}_0 B^2 \ldots \quad \tilde{w}_{l^*}/\tilde{w}_0 B^{l^*} & & \frac{1}{\tilde{w}_0} - \frac{\tilde{w}_1}{\tilde{w}_0^2} B + \ldots \\
\cline{1-2}
\quad -\tilde{w}_1/\tilde{w}_0 B - \tilde{w}_2/\tilde{w}_0 B^2 \ldots \quad -\tilde{w}_{l^*}/\tilde{w}_0 B^{l^*} & & \\
\quad -\tilde{w}_1/\tilde{w}_0 B - \tilde{w}_1^2/\tilde{w}_0^2 B^2 \ldots - \tilde{w}_1 \tilde{w}_{l^*-1}/\tilde{w}_0^2 B^{l^*} - \tilde{w}_1 \tilde{w}_{l^*}/\tilde{w}_0^2 B^{l^*+1} & & \\
\quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots & &
\end{array}
$$

Figure 5: First 2 terms of polynomial long division

3 Toy Project

In the previous sections, the *memory vs. stationarity* dilemma has been presented. On top of that, through fractional differentiation, stationarity has been achieved without totally giving up memory. Also, a method has been presented to differentiate and un-differentiate in a robust way.

That being said, the problem that this toy project attempts to solve will be presented. Given a time series y_t, and Λ_t representing all the values y_t has taken up to time t, the aim of forecasting is to give a prediction $\widehat{y}_{t+1}$ given Λ_t. By sequentially giving 1-day forecasts, a new time series will be generated.

To illustrate the effect fractional differentiation can have in the predicting power of a model, two models will be designed. One will have fully differentiated features and labels, and the other will use

fractionally differentiated ones.

The last phase of the toy project will be computing prediction errors to compare the performance of the two models.

3.1 Synthetic data

Striving to create a log-prices time series with memory, the generation was the following:

$$y_t = \frac{1}{5} \cdot \sum_{i=1}^{5} y_{t-i} + \mu + \epsilon_t \quad (t \geq 6) \tag{4}$$

where $\mu = 0.005$, $\epsilon_t \sim N(0,\ \sigma = 0.01)$ and $y_{1:5} = \{0.000,\ 0.075,\ 0.150,\ 0.225,\ 0.300\}$.

As it can be seen from the equation 4, in order to determine the value of y_t, it is crucial to know the preceding 5 values of the time series. Consequently, if "memory is erased", the prediction power will decrease. That is, the fractional differentiated time series instance at time $t-1$ will contain more information from the preceding days than the fully differentiated series of log-returns.

As for the length of the time series, y_t will have 3000 observations, and 15% of them (450 obs.) will be saved for testing purposes.

3.2 Fractional differentiation of y_t

In figure 6 and 7 the differentiated time series are shown. Note that the method used is the one defined previously: fixed-width window fractional differentation method (FFD). Also, in order to determine the minimnum d^* that makes the time series stationary, figure 8 and table 1 are presented (only training data has been used).

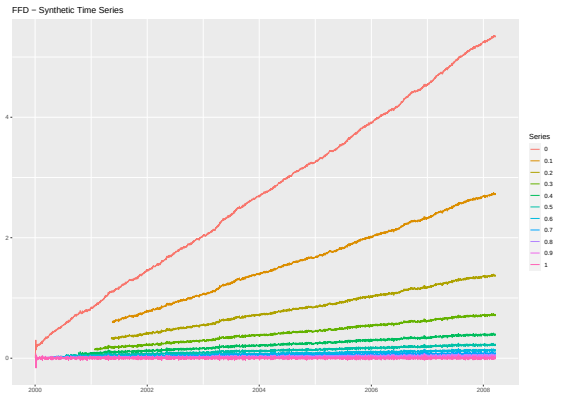

Figure 6: FFD of y_t - $\tau = 1e-4$

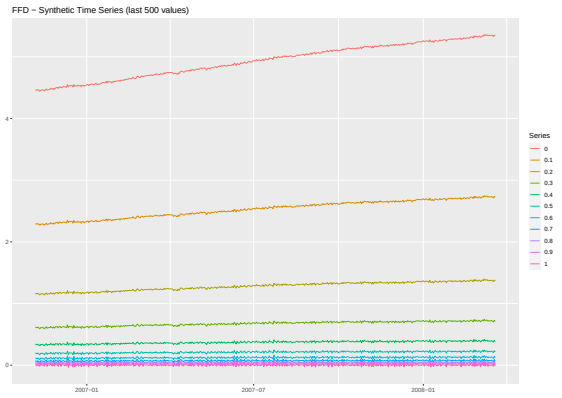

Figure 7: FFD of y_t (Zoomed in) - $\tau = 1e-4$

Thus, now that it is known that $d^* = 0.2$, whenever the FFD time series is mentioned, the order of differentiation will be d^*.

3.3 Models

With the purpose of evaluating how FFD works in this toy project, three models will be used:

1. **Naive Model** (Benchmark): Simplest model than can be built. It will predict $\widehat{y}_{t+1} = y_t$
2. **FFD Model:** It will use $y_t^{d^*}$ to predict y_t.
3. **Returns Model:** It will use $y_t^1 = y_t - y_{t-1}$ to predict y_t.

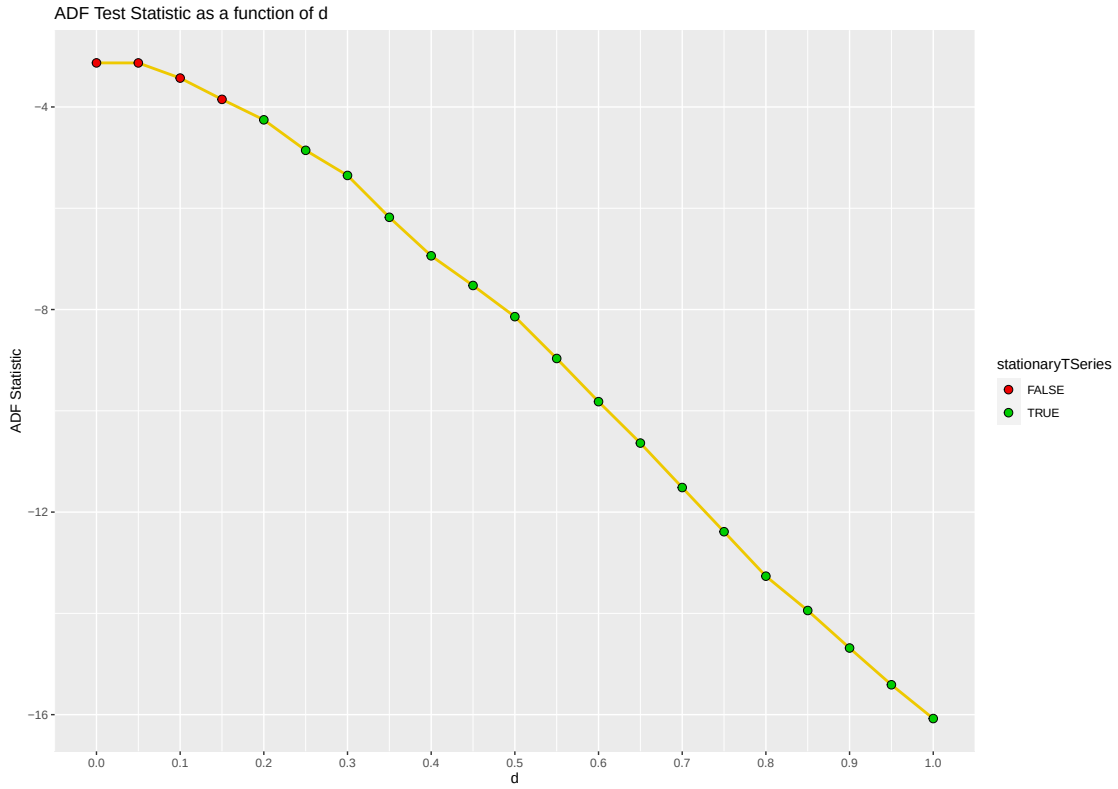

Figure 8: ADF Statistics of y_t^d - Train

Table 1: ADF Test (with trend and drift) Statistics of y_t^d - Train

d	Lag order (p)	ADF Statistic
0.00	13	-3.130
0.05	12	-3.130
0.10	12	-3.430
0.15	12	-3.850
0.20	12	-4.254
0.25	12	-4.856
0.30	12	-5.352
0.35	13	-6.179
0.40	13	-6.939
0.45	13	-7.525
0.50	13	-8.141
0.55	13	-8.967
0.60	13	-9.819
0.65	13	-10.637
0.70	13	-11.515
0.75	13	-12.388
0.80	13	-13.265
0.85	13	-13.944
0.90	13	-14.684
0.95	13	-15.410
1.00	13	-16.077

The FFD and Returns models will be similar in the sense that will use the same artificial neural network (ANN) structure and features. The former will use a fully-connected hidden layer with 4 units and an output unit (see figure 10), both with ELU (Exponential Linear Unit) as activation function.

Respecting the features, since the last 5 values have been used to determine the next one, that is what will be used. Clarifying:

- **Features:** y^d_{t-5}, y^d_{t-4}, y^d_{t-3}, y^d_{t-2} and y^d_{t-1}
- **Labels:** y^d_t

Lastly, recall that 450 observations out of 3000 will be used for testing purposes. This does not mean that 2550 observations will be used for training, since using the FFD method implies that the series will not be defined up to a point because all observations should have the same amount of memory (if this did not happen, the first observation would have zero memory while the subsequent ones would have more).

3.4 Sequential forecasts

Considering that one is attempting to predict log-prices and differentiated features are being used, an emphasis should be put in the un-differentiation, which can be expressed (see equation 3) as:

$$y_t = \sum_{k=0}^{T-1} c_k \cdot B^k(y^d_t) \qquad (T = 3000)$$

Noting n_0 as the time stamp where predictions start, the predicted time series will be defined sequentially as:

$$z_t = \sum_{k=1}^{T-1} c_k \cdot B^k(y^d_t) + c_0 \cdot \widehat{y}^d_t \qquad (t \geq n_0)$$

3.5 Results

In the previous section, it has been shown how to obtain a new time series with the predictions. In order to compare the final time series in the test data set ($t \geq n_0$), three metrics will be used: Root-mean-square error (RMSE), Mean absolute percentage error (MAPE) and Median absolute deviation (MAD). The metrics mentioned will be defined after introducing the following parameters: $e_k = y_k - \widehat{y}_k$ and $\tilde{e} = \text{median}(e_k)$.

$$\text{RMSE} = \sqrt{\frac{\sum_{k=n_0}^{T} (e_k)^2}{T - n_0 + 1}}$$

$$\text{MAPE} = \frac{1}{T - n_0 + 1} \cdot \sum_{k=n_0}^{T} \left| \frac{e_k}{y_k} \right|$$

$$\text{MAD} = \text{median}(|e_k - \tilde{e}|)$$

With the metrics defined, the results in the test data set are the following:

Table 2: Error metrics of the models considered - Test

	FFD model	Naive model	Returns model
RMSE	0.010184	0.012995	0.013126
MAPE	0.001608	0.002113	0.002148
MAD	0.006606	0.008747	0.009192

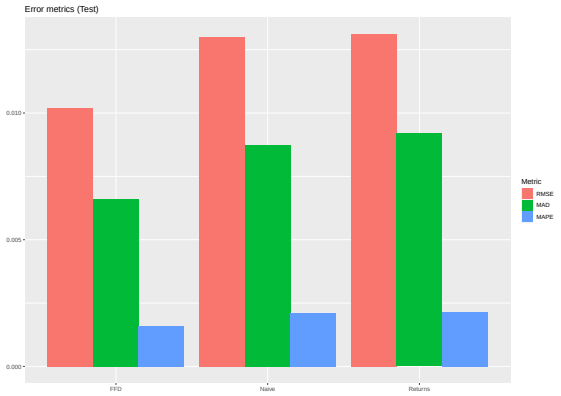

Figure 9: Error metrics of the models considered - Test

3.6 Conclusions

From the results shown in table 2 and figure 9, it can be implied that the FFD model has performed best across all error metrics. Also, the difference between the Naive and Returns Model is negligible, so it can be concluded that the Returns Model has not been able to detect the pattern behind the observations.

This goes to say that fractionally differentiating the data provides much information in a much clearer way than full differentiation. In other words, even tough the original time series can be reconstructed with both methods, the FFD method provides information with memory without having to do any transformation.

The next step is to apply the same method to a real financial time series. Although this controlled project has been successful in an FFD sense, it remains to be seen whether memory in financial time series helps with forecasts.

4 Financial data

The data used will be the **log-prices** of the S&P500 from 2005-01-01 to 2019-09-01. It will include the Open (y_t^{Open}), High (y_t^{High}), Low (y_t^{Low}) and Close (y_t^{Close}). That is, the log-price it opens/closes and the highest/lowest values during the daily trading session.

As in the project of section 3, the goal of the models will be to make a 1-day forecast. In particular, with the OHLC (Open High, Low, Close) data of day t, the models will predict the value the stock closes at time $t+1$. To do that, the features and labels will be the following:

- **Features:** y_t^{Open}, y_t^{High}, y_t^{Low} and y_t^{Close}
- **Labels:** y_{t+1}^{Close}

These features have been chosen because they provide information at different times of the day. That is, they represent the daily state of the time series, and thus, having memory could provide and advantage, since comparisons between them could be made.

4.1 Division Train/Test data sets

The data will be divided into a training and test data set. The distribution is the following:

As it can be seen in table 3, both the FFD and the Returns models have 788 observations in the test data set. However, in the train data set, the FFD has approximately 450 fewer observations. This can be explained via equation 1, since by **setting a threshold** $\tau = 1e-4$, an $l^*(\tau, d)$ exists such that the FFD is valid for $t \geq l^* + 1$. In particular, in the case of returns $d = 1$, and thus, $l^*(\tau, d = 1) = 1$ since

Table 3: Division of Train/Test

	FFD model	Returns model
Start date (Train)	2006-10-06	2005-01-04
End date (Train)	2017-07-12	2017-07-12
Start date (Test)	2017-07-14	2017-07-14
End date (Test)	2020-08-28	2020-08-28
n_{Train}	2709	3152
n_{Test}	788	788

the first observation does not have a preceding value.

Also note that n_{Test} has been kept equal because the error metrics will be evaluated in this data set. Consequently, the conditions should be the same to avoid misleading results.

5 Models

As in section 3, three models will be used: **Naive**, **FFD** and **Returns** model. The features will be the corresponding time series shown in section 4, but differentiated. To be specific, the FFD model will use the fractionally differentiated features with the minimum order ($d^* = 0.25$) that makes the 4 OHLC time series stationary (in training). Conversely, the **Returns** model will use fully differentiated features ($d = 1$).

The **FFD** and **Returns** models will use an ANN with the structure seen in figure 10. As in section 3, the activation function will be the Exponential Linear Unit (ELU).

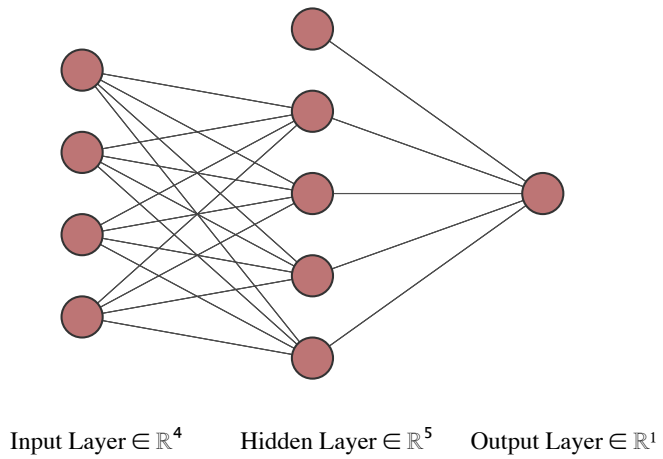

Figure 10: ANN used in the FFD and Returns Models

6 Data normalization

As a means to provide features to the neural network with the same range, all 4 features have been normalized:

$$z_t = \frac{y_t - y_{\min}}{y_{\max} - y_{\min}}$$

It should be pointed out that $y_{\max}$ and $y_{\min}$ have been computed in the **train** data set. It is extremely important to consider this, because if one were to use the maximum/minimum of the whole data set, look-ahead bias would have occurred. To be more specific, if the test observations started at time t_0 and the time stamp when the maximum is achieved is $t_0 + m > t_0$, then the forecast at time t_0 would have used information from $t_0 + m$, which is impossible.

That being said, in the train data set, $z_t \in [0, 1]$, while in the test data set, features are not guaranteed to be in $I = [0, 1]$. However, working under the assumption that the complete time series is stationary, values can not be far off.

It should be brought up that since labels are not normalized, the predicted values will not have to undergo "un-normalization".

7 Results

As in section 3.4, the sequential forecasts have been calculated and the final result is 3 time series, which are forecasts of the close of the S&P500:

$$\begin{aligned} y_t^{\text{Naive}} \quad & (n_0 + 1 \leq t \leq T) \\ y_t^{\text{FFD}} \quad & (n_0 + 1 \leq t \leq T) \\ y_t^{\text{Returns}} \quad & (n_0 + 1 \leq t \leq T) \end{aligned}$$

Where $t = n_0$ is the time stamp that marks the start of the test data set. Therefore, the prediction will be of the value of the log-price close of $t + 1 = n_0 + 1$

Table 4: Error metrics of the models considered - Test

	FFD model	Naive model	Returns model
RMSE	0.0187200	0.0185758	0.0181539
MAPE	0.0017909	0.0017573	0.0017179
MAD	0.0047790	0.0044728	0.0044403

In table 4 and figure 11 the error metrics for the three models considered are shown. In addition, in figures 12, 13 and 14 the forecasted time series are presented for different periods of time.

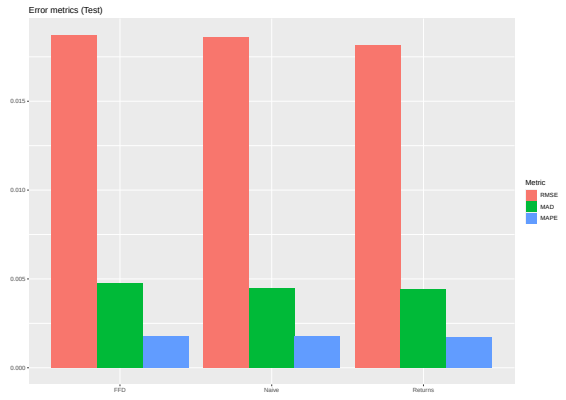

Figure 11: Error metrics of the models considered - Test

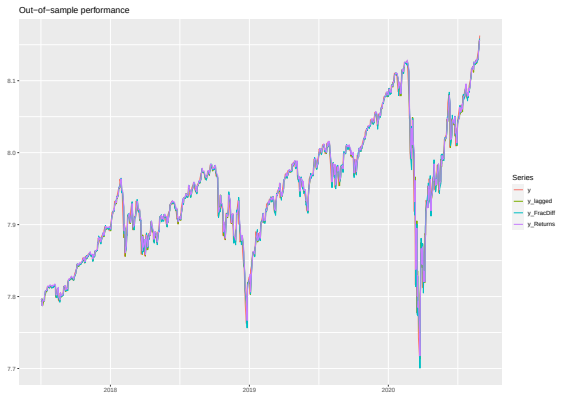

Figure 12: Forecasted time series - Test

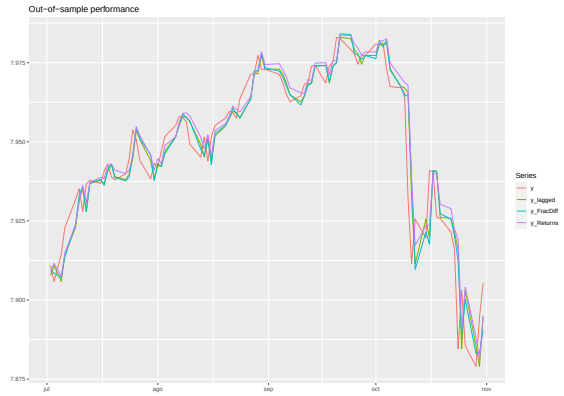

Figure 13: Results 2018-07/2018-10

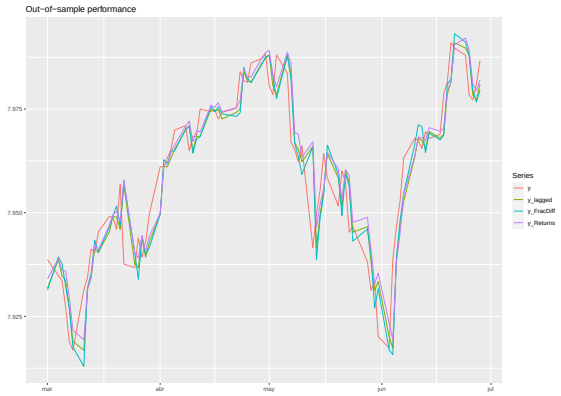

Figure 14: Results 2019-03/2019-06

8 Conclusions

As it was shown in section 7,