TSA Project Report - Pierluigi Marchioro 881929

Introduction

This project makes use of two datasets that consist of the following 2 time series:

- SMH ETF price, daily, in the 2024-08-23-to-2000-06-05 date interval
- Global semiconductor sector trailing annual sales, monthly frequency, from Jan-2012 to Jan-2024

The SMH price time series has been downloaded via its aforementioned Yahoo Finance page, while the semiconductor sector sales time series has been downloaded from this page at Statista.com.

The main goals of the project are the following:

- *Task 1*: Trying to predict the price of the SMH ETF, with a relatively short forecast horizon, using simple models such as those from the ETS and SARIMA families
- *Task 2*: Analyzing the relationship between global semiconductor sales and SMH price action and trying to use the former to predict the latter.

On a final note, the SMH dataset originally proposed, downloaded from the AlphaVantage API, has been substituted by the equivalent one downloaded from yahoo finance, which was properly adjusted for two stock splits that happened from the year 2000 up until now.

Parentheses on project structure and installation

The project is divided in the following main folders:

- `data`, containing the data used for the analysis
- `notebooks`, containing the notebooks used to implement the analysis; such notebooks can be consulted for additional charts that won't be shown in this report
- `src`, containing the core of the codebase, appropriately divided in python modules referred by the notebooks on necessity

To execute the notebooks, one should first install the project dependencies:

1. Install `poetry` as shown in the poetry installation guide (pipx approach suggested)
2. From the terminal, `cd` into the project root directory
3. Run `poetry config virtualenvs.in-project true && poetry install` to install the project dependencies in a virtual environment `<project-root>/.venv` directory

Methodology

The general idea is that each task was preceded by an EDA phase, with generally the same task-independent structure, to understand the structure of the time series at hand and how to model them appropriately for the next phases, which directly tackle the goals of the project. In particular, each time series handled in the project was transformed into three additional variants:

- *differenced*: the k-differenced time series, with $k = 1$ having always proved sufficient to successfully detrend by difference
- *log*: the log-transformed time-series, especially suited in the case of stock prices given their multiplicative nature
- *log-differenced*: the log-transformed time-series, to which k-order differencing has been applied; again $k = 1$ has proven sufficient and no further differencing has been attempted

Task 1 - Price Prediction

EDA

The goal of this Exploratory Data Analysis was to understand the characteristics of the daily closing prices of the SMH ETF and to try to find the correct transformations to make such a time series (weakly) stationary.

The core points of this phase are as follows:

- A normal trading week consists of 5 days, meaning that the number of data points per year is much less than 365. Specifically, the number of trading days in a normal year seems to be ~252, for a total of 6095 data points
- STL and MA-based decomposition have been performed, with seasonal period set to 252 (annual), to identify trend and seasonality of the time series. They proved very useful in identifying successfully detrended series, as well as suggesting that the seasonal component of stock prices is not very strong, which was to be expected.
- Plots of the rolling standard deviation have been made to help to determine if a time series had been correctly transformed to an additive model, as a clue of a multiplicative model is the fact that variance/std increases as the level of the series increases.
- ACF/PACF plots were implemented to understand the configuration of a later ARIMA model
- Stationarity tests, specifically the Augmented Dickey-Fueller and the Kwiatkowski–Phillips–Schmidt–Shin test, have been implemented to identify if a series had been successfully transformed into (weakly) stationary.

Model Choice and Comparison

The models were chosen from the ETS and SARIMA families, as explained in the introduction section. In particular, since, as explained later, the chosen SMH price time series variant was the *log* series:

- various ETS models have been fitted, each with a different combination of additive components
- various ARIMA models have been fitted, either via automatic procedures, or by hand, after looking at the ACF/PACF plots of the chosen time series and the residuals of previously fitted models.
    - Generally speaking, the AR order was chosen based on the latest spike over the white noise bands of the ACF; the same goes for the MA order and PACF plots.
    - The differencing order of the ARIMA model is always set to 1, as it has been shown in the EDA phase that no more differencing is required (perhaps the series is even slightly overdifferenced, judging by the slightly negative autocorrelation at lag 1) to remove the trend component and make it weakly stationary.

- The AR and MA orders were increased if the ACF plots of the residuals, along with the Ljung-Box test, showed that there was still some autocorrelation left to capture; this is the reason why, as shown later, various hand-picked models of increasing order have been tried

Comparison between models were made in terms of AICc (corrected AIC) and cross-validation errors (MAE, MAPE, RMSE).
Cross-validation has been performed on a rolling basis, starting from 95% of the dataset (~300 samples left for validation) to keep training times manageable, with *step* = 10 and a relatively short *forecast horizon* = 10.

Task 2 - Sales-Price Transfer Function Modelling

For this task, two time series were involved: the SMH price time series and the global semiconductor sales time series.
First of all, since the SMH price and semiconductor sales series have different frequencies and date intervals, the former has been adjusted to the latter, with both series becoming monthly and spanning the date interval $[2012\text{-}01, 2024\text{-}01]$.
This task was then implemented in 4 separate phases, detailed in the below subsections.

EDA

This phase largely utilized the same tools of Task 1's EDA, and for the same reasons, i.e. making both the involved time series (weakly) stationary. An additional tool was used, that of the CCF plots, to check if there was any meaningful and understandable cross-correlation structure between time series variants; pre-whitening will still be required to make sense of them.

Pre-whitening

In this phase, the goal was to apply the pre-whitening procedure discussed during the course, to disentangle the linear association between the input and target time series from their autocorrelation. Such pre-whitening procedure is summarized as follows:

1. determine an ARIMA *time series model* for the X-variable, and store the residuals from this model;
2. fit the ARIMA X-model to the Y-variable, and keep the residuals;
3. examines the CCF between the X and Y model residuals.
   The X variable in this case is the sales time series, which allegedly influences the smh price, while the Y variable is the SMH price, which supposedly doesn't influence X.

After the two time series have been transformed with such a technique, we are able to move on to the next phase, about transfer function modelling and lagged regression.

It is noted that pre-whitening has been applied to pairs of same-variant time series. For example, pre-whitening has been applied to $X = \text{sales log-diffed}$ and $Y = \text{smh log-diffed}$.

Transfer Function Modelling Lagged-Regression

The sequential methodology described in the course was implemented to estimate the coefficients of the transfer function, which was then used to make forecasts.

Lagged-Regression

As regards the lagged-regression model, the CCF plots of the two pre-whitened time series were used to identify lags at which the correlation is meaningful, which in turn determine which terms to use in the regression model. In particular, I referred to this table to interpret CCF patterns.
It is noted that, since the CCF (see results section for more details) didn't clearly fall into any of those cases, I tried the combinations of terms suggested by what I thought were the most similar aforementioned patterns.

Results

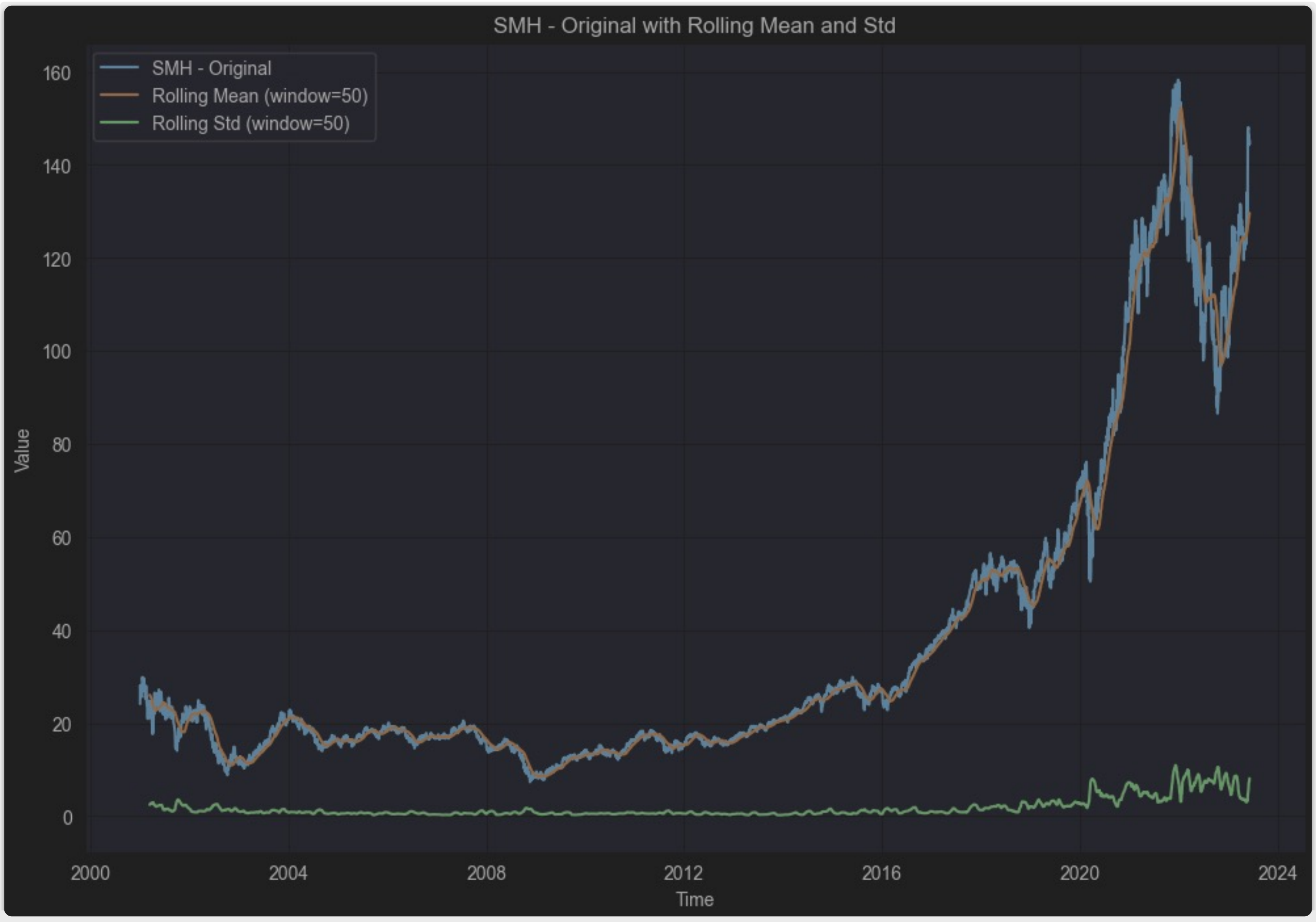
Task 1

EDA

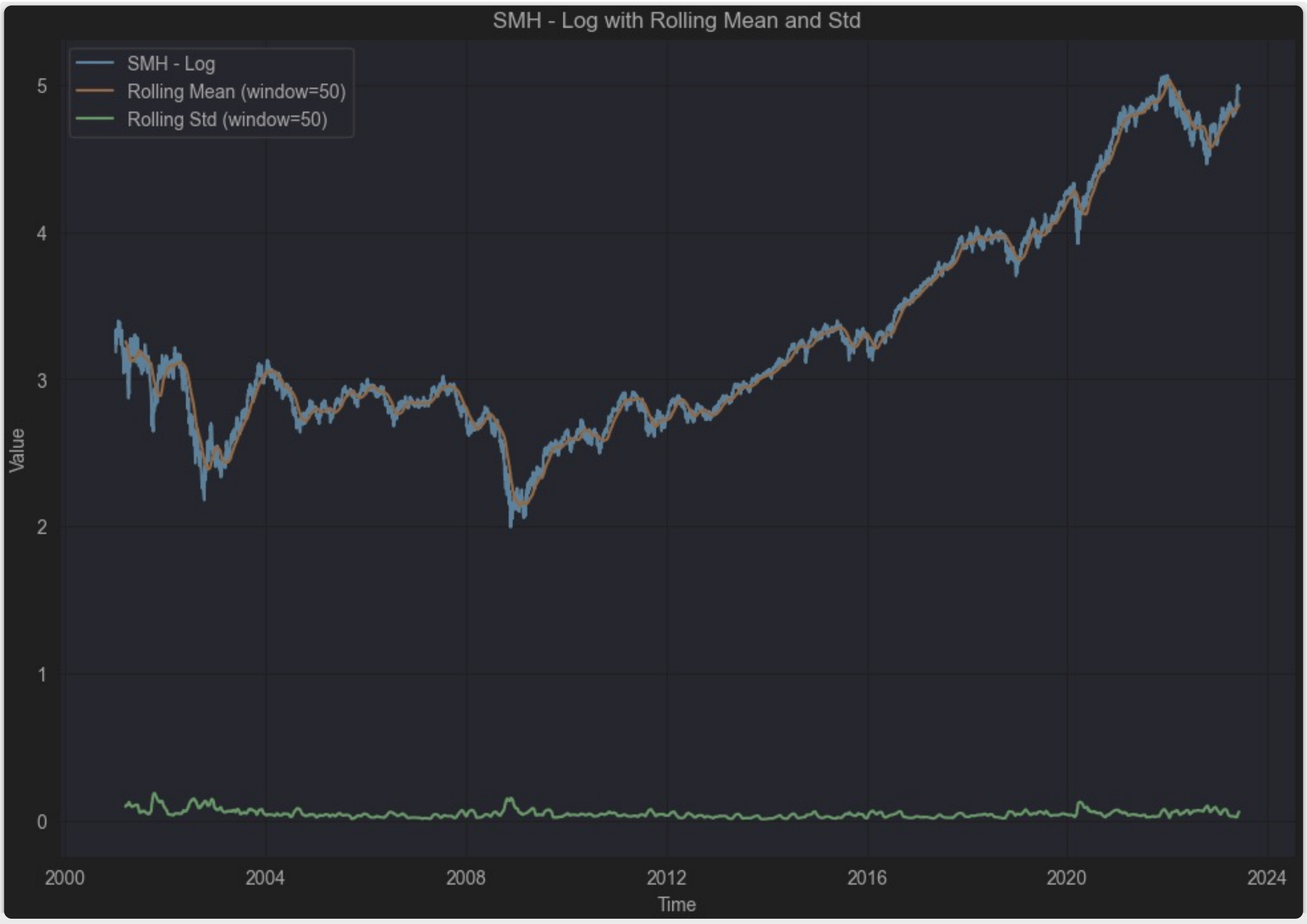
The final result of this phase was identifying the *log* time series as the one to use for the next phase, the reason being twofold:

- it's 1-order difference is stationary, which was proven via the aforementioned stationarity tests, making it suitable for ARIMA models
- it is additive in nature thanks to the log transformation, which could possibly make the modelling phase easier
- predictions are nice to interpret, since models operate directly on the log-scale

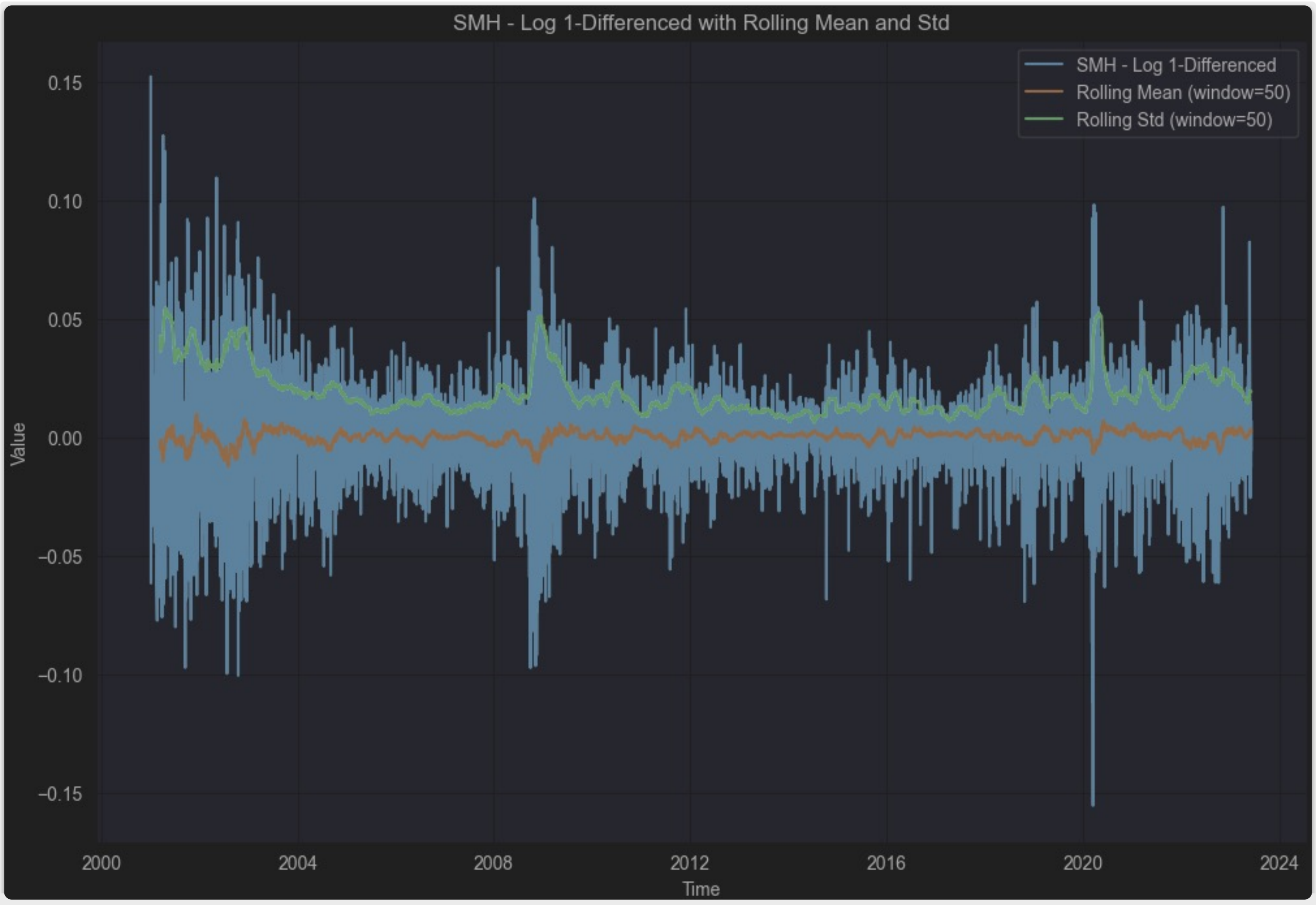
Below follow some interesting plots/tables about the SMH price time series and the aforementioned final choice.



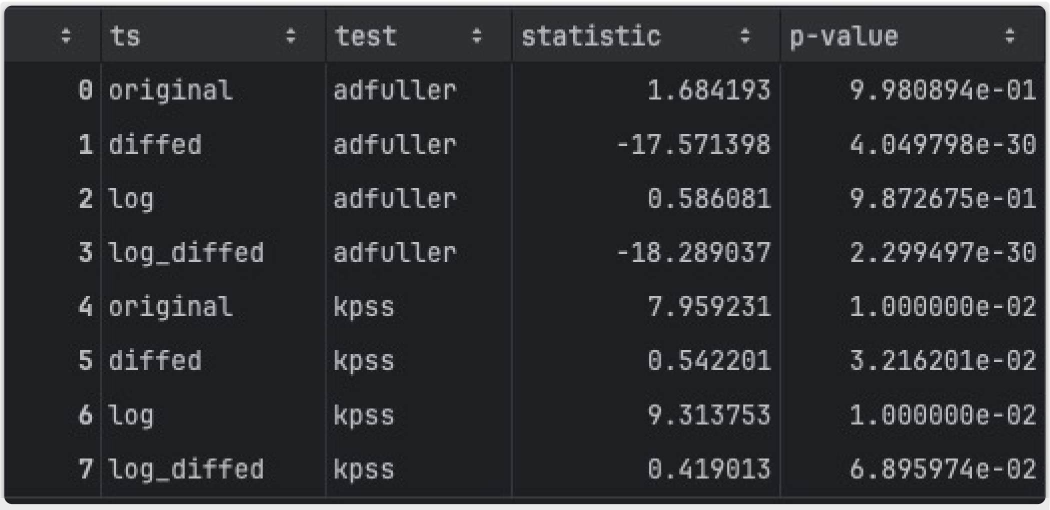
Original scale SMH



Log scale SMH



Log-Diffed SMH

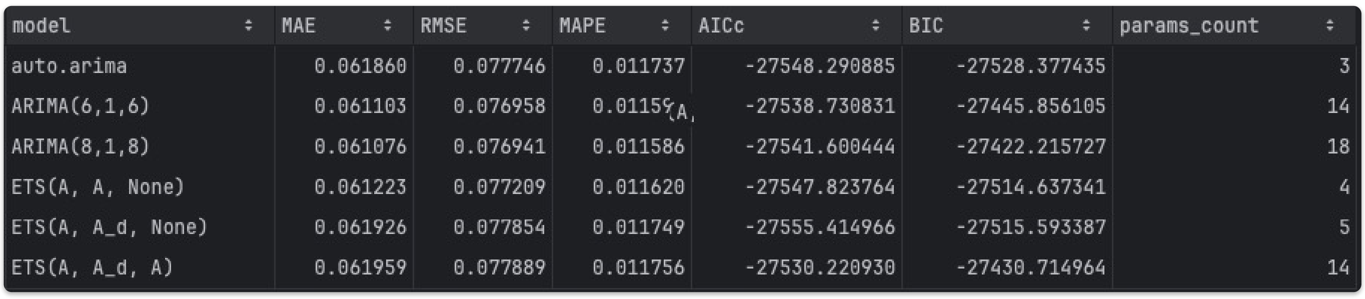


	ts	test	statistic	p-value
0	original	adfuller	1.684193	9.980894e-01
1	diffed	adfuller	-17.571398	4.049798e-30
2	log	adfuller	0.586081	9.872675e-01
3	log_diffed	adfuller	-18.289037	2.299497e-30
4	original	kpss	7.959231	1.000000e-02
5	diffed	kpss	0.542201	3.216201e-02
6	log	kpss	9.313753	1.000000e-02
7	log_diffed	kpss	0.419013	6.895974e-02

Stationarity tests results for all the SMH variants

Model Comparison

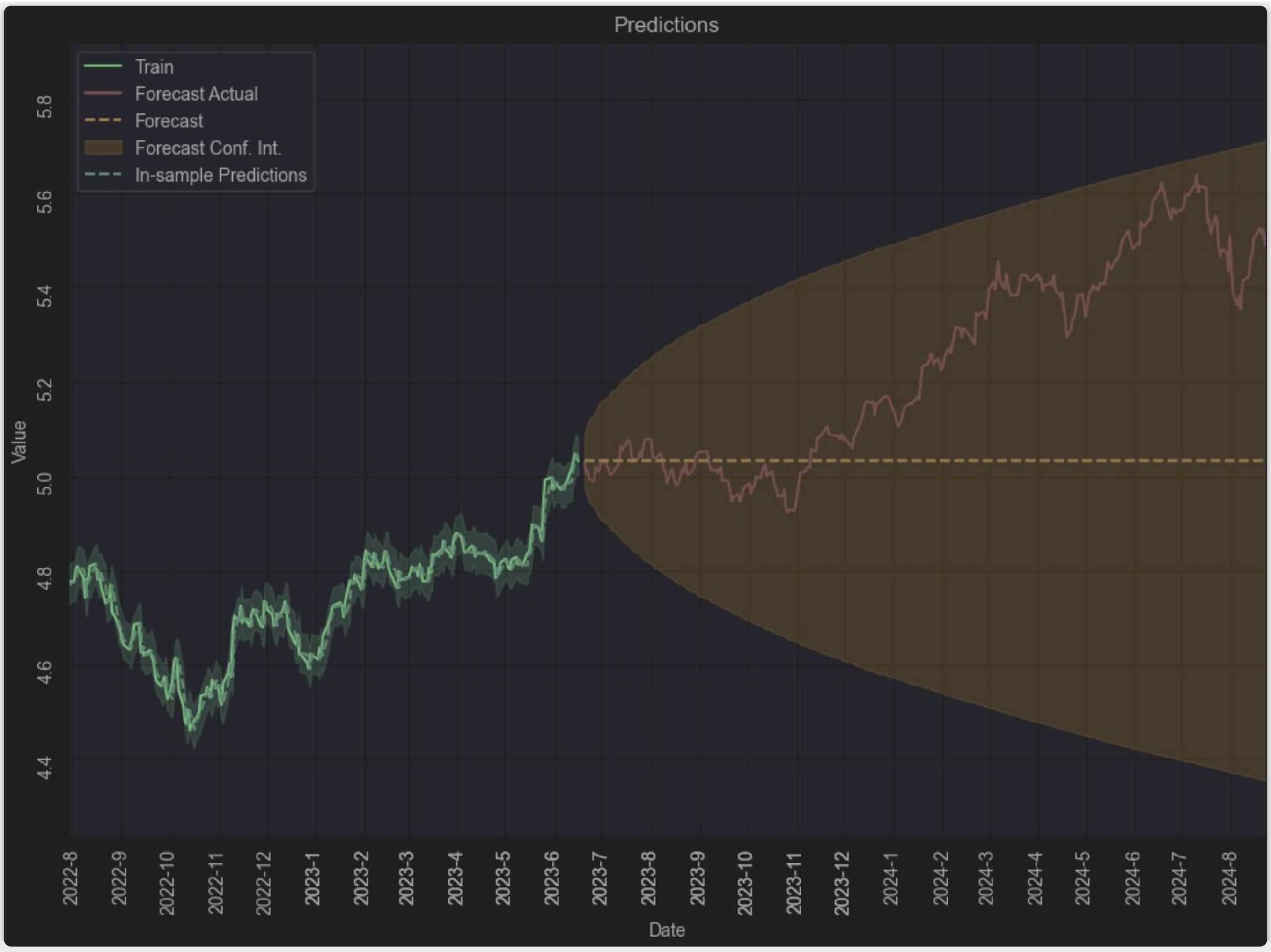
The final model statistics were the following:



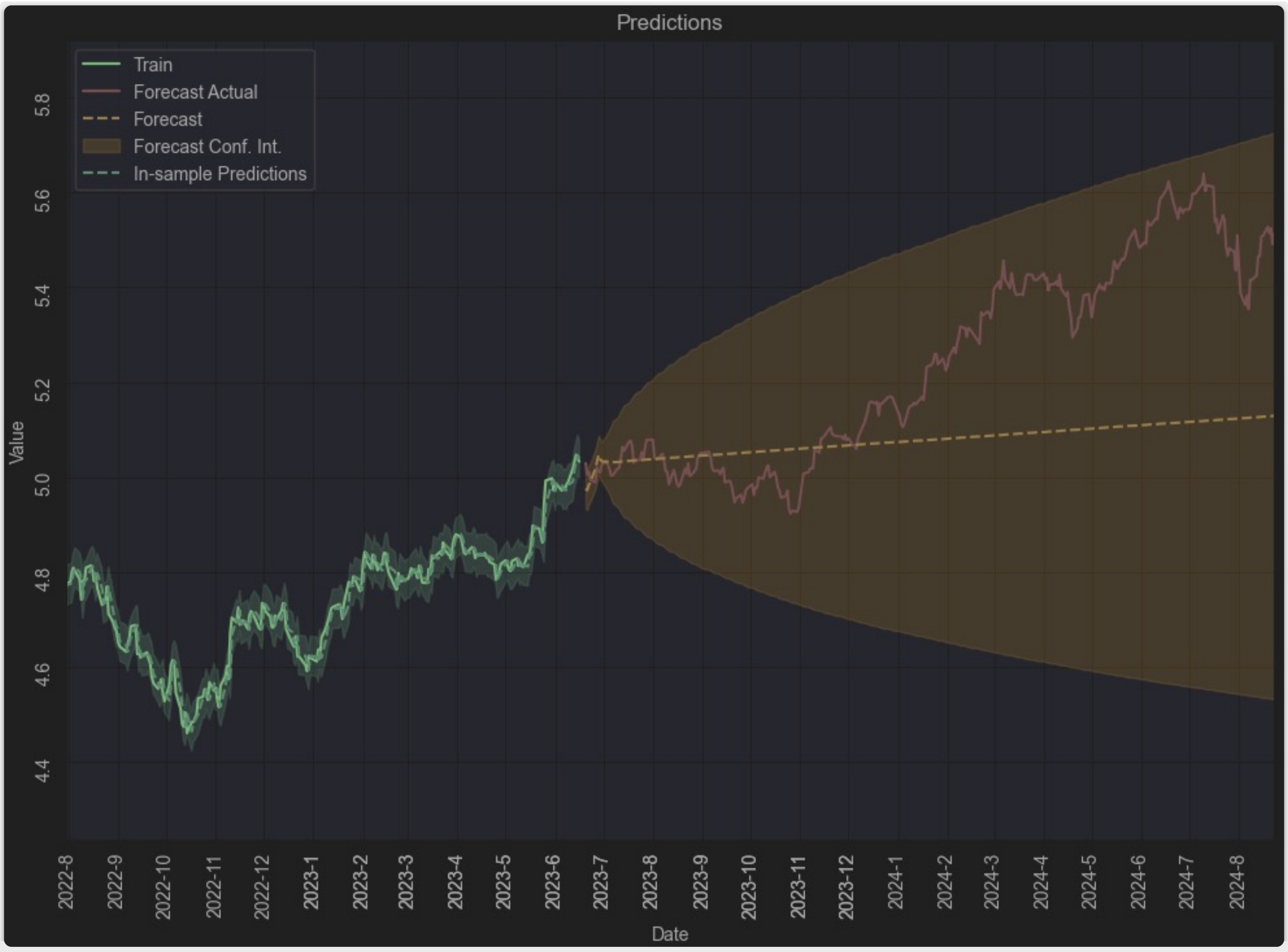
model	MAE	RMSE	MAPE	AICc	BIC	params_count
auto.arima	0.061860	0.077746	0.011737	-27548.290885	-27528.377435	3
ARIMA(6,1,6)	0.061103	0.076958	0.01159(A,	-27538.730831	-27445.856105	14
ARIMA(8,1,8)	0.061076	0.076941	0.011586	-27541.600444	-27422.215727	18
ETS(A, A, None)	0.061223	0.077209	0.011620	-27547.823764	-27514.637341	4
ETS(A, A_d, None)	0.061926	0.077854	0.011749	-27555.414966	-27515.593387	5
ETS(A, A_d, A)	0.061959	0.077889	0.011756	-27530.220930	-27430.714964	14

The best model according to AICc was the *ETS model with additive errors and additive, damped trend*.
The best model according to cross-validated accuracy measures was the *ARIMA(8, 1, 8)*.

The below plots about non-CV in-sample predictions and forecast give a general idea about the behavior of the two best models:



ETS(A, A_d, None) Model



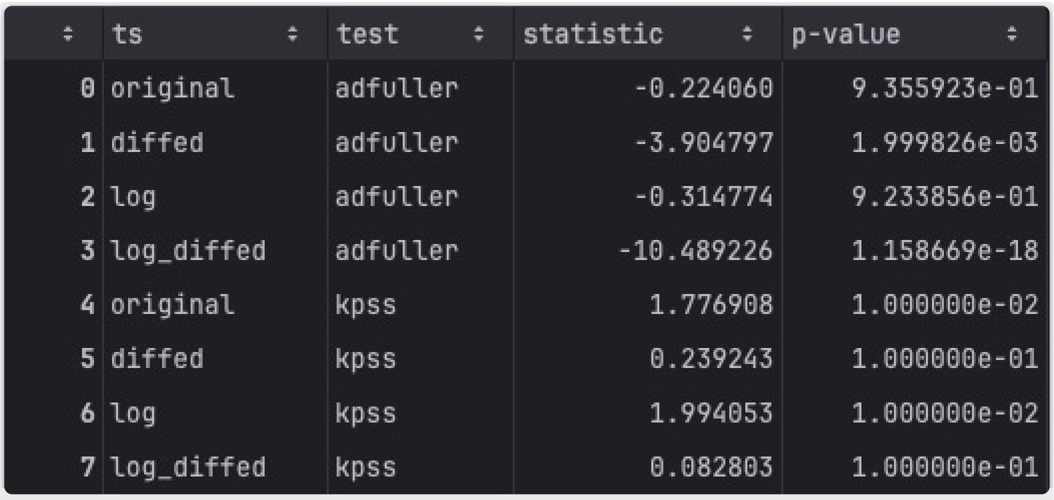
ARIMA(8, 1, 8)

The former seems to make flat predictions based on the latest seen level, while the latter seems to at least understand the general trend. Although both models display a substantial degree of uncertainty, as evidenced by the relatively large confidence bands, the latter still seem to include the actual realization of the time series in the forecast interval. Finally, the ARIMA model shows slightly tighter prediction intervals on a short forecast horizon, which, coupled with the prior considerations on the general understanding of the trend, probably make it the somewhat better model.

Task 2

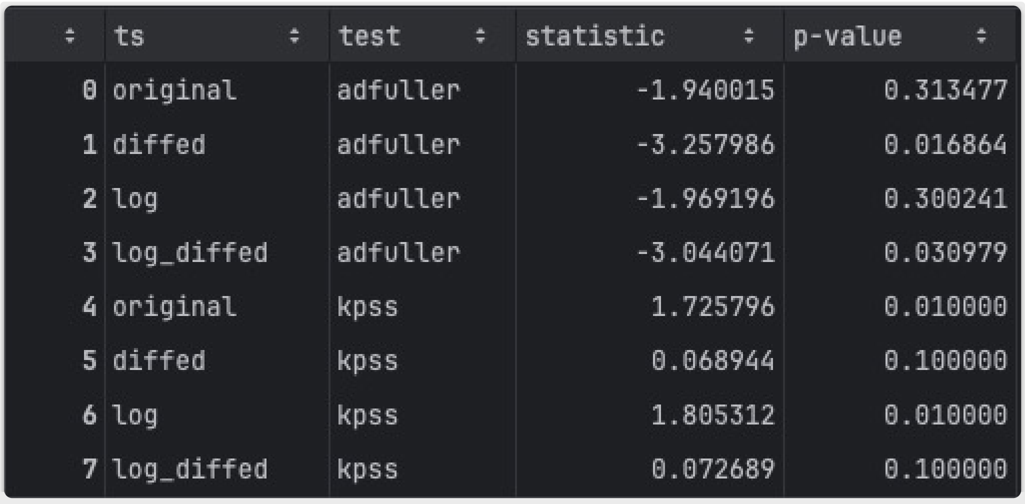
EDA

As shown by the tables below, the pairs $X = \text{sales 1-differenced}, Y = \text{smh 1-differenced}$ and $X = \text{sales log-differenced}, Y = \text{smh log-differenced}$ have been chosen as candidates for the pre-whitening step due to their stationarity.



	ts	test	statistic	p-value
0	original	adfuller	-0.224060	9.355923e-01
1	diffed	adfuller	-3.904797	1.999826e-03
2	log	adfuller	-0.314774	9.233856e-01
3	log_diffed	adfuller	-10.489226	1.158669e-18
4	original	kpss	1.776908	1.000000e-02
5	diffed	kpss	0.239243	1.000000e-01
6	log	kpss	1.994053	1.000000e-02
7	log_diffed	kpss	0.082803	1.000000e-01

SMH Monthly Closing Price - Stationarity tests



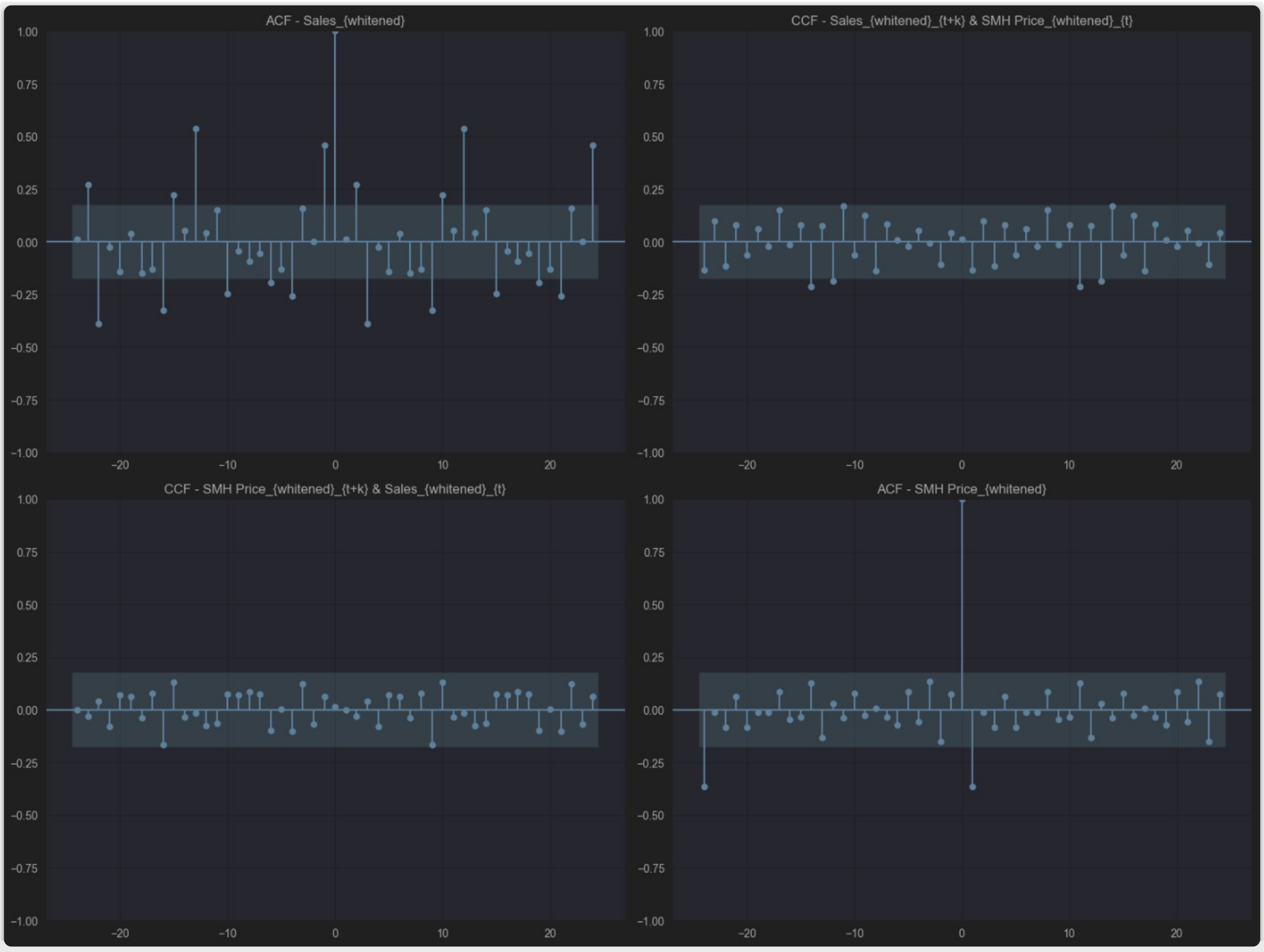
	ts	test	statistic	p-value
0	original	adfuller	-1.940015	0.313477
1	diffed	adfuller	-3.257986	0.016864
2	log	adfuller	-1.969196	0.300241
3	log_diffed	adfuller	-3.044071	0.030979
4	original	kpss	1.725796	0.010000
5	diffed	kpss	0.068944	0.100000
6	log	kpss	1.805312	0.010000
7	log_diffed	kpss	0.072689	0.100000

Global Semiconductor Sales - Stationarity tests

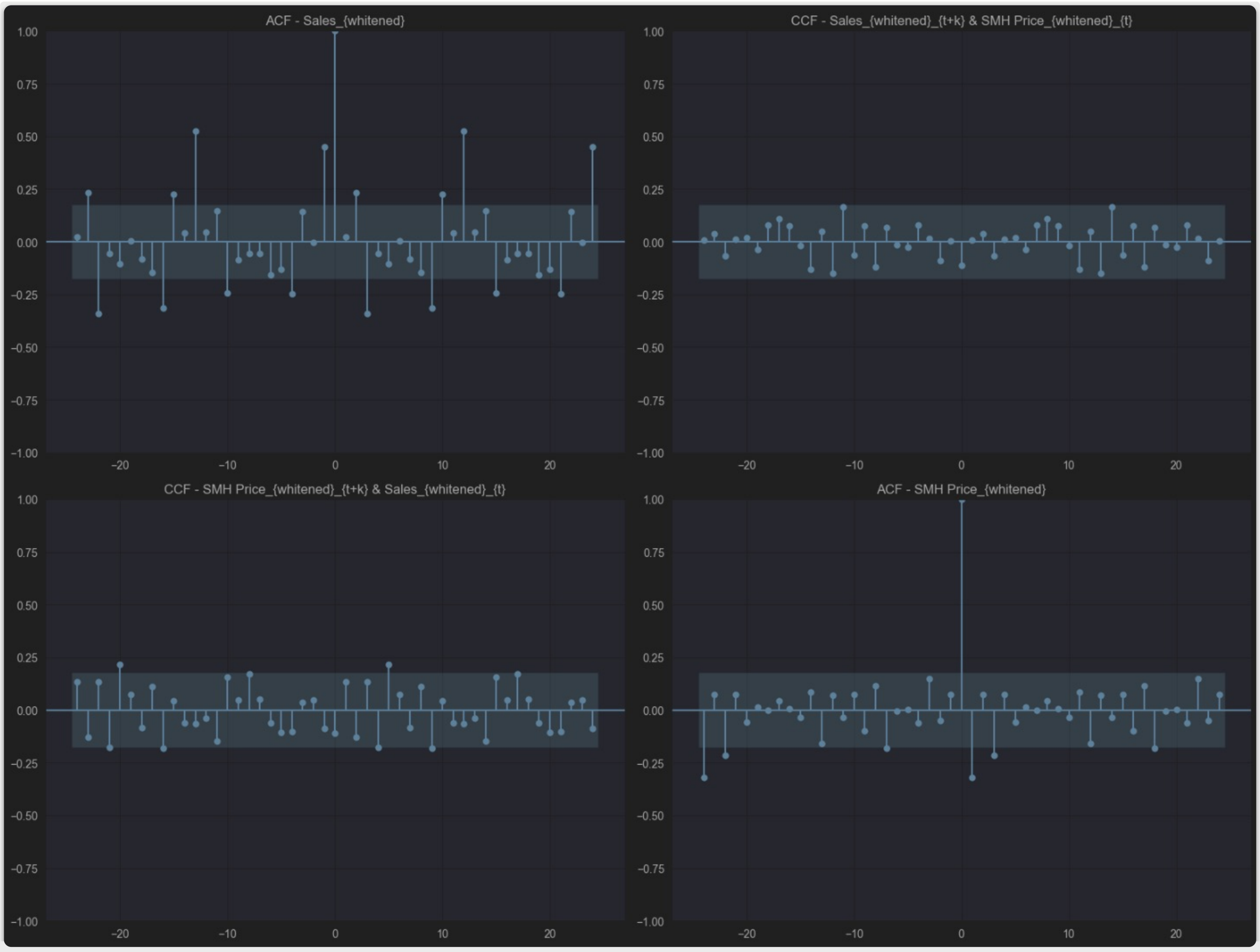
Pre-whitening

At this step, one of the two candidate time series is chosen, based on the quality of the CCF plots after the pre-whitening step.

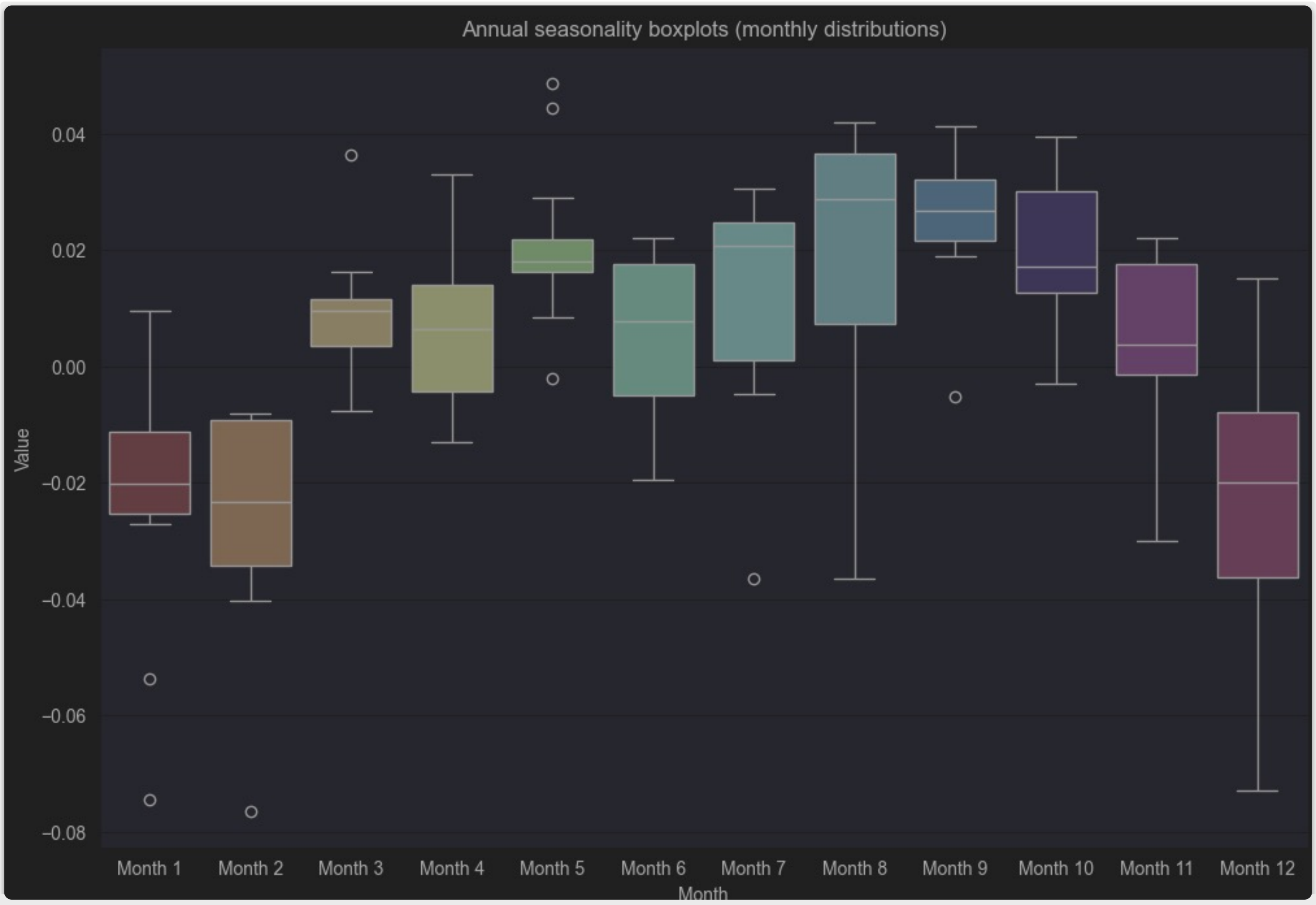
After taking a look at the below plots, the choice fell onto the pair $X = \text{sales log-differenced}, Y = \text{smh log-differenced}$, due to their CCF seemingly containing more significant lags. It is to be noted, however, that the X_{whitened} time series is not properly behaving like white-noise due to an apparent seasonal behavior, which is backed by the seasonality analysis of the Sales time series in the previous EDA phase. For this reason, in the next, phase, such a time series will be deseasonalized in order to potentially improved the cross-correlation structure of the pair.



Log-diffed pair pre-whitened CCF plots



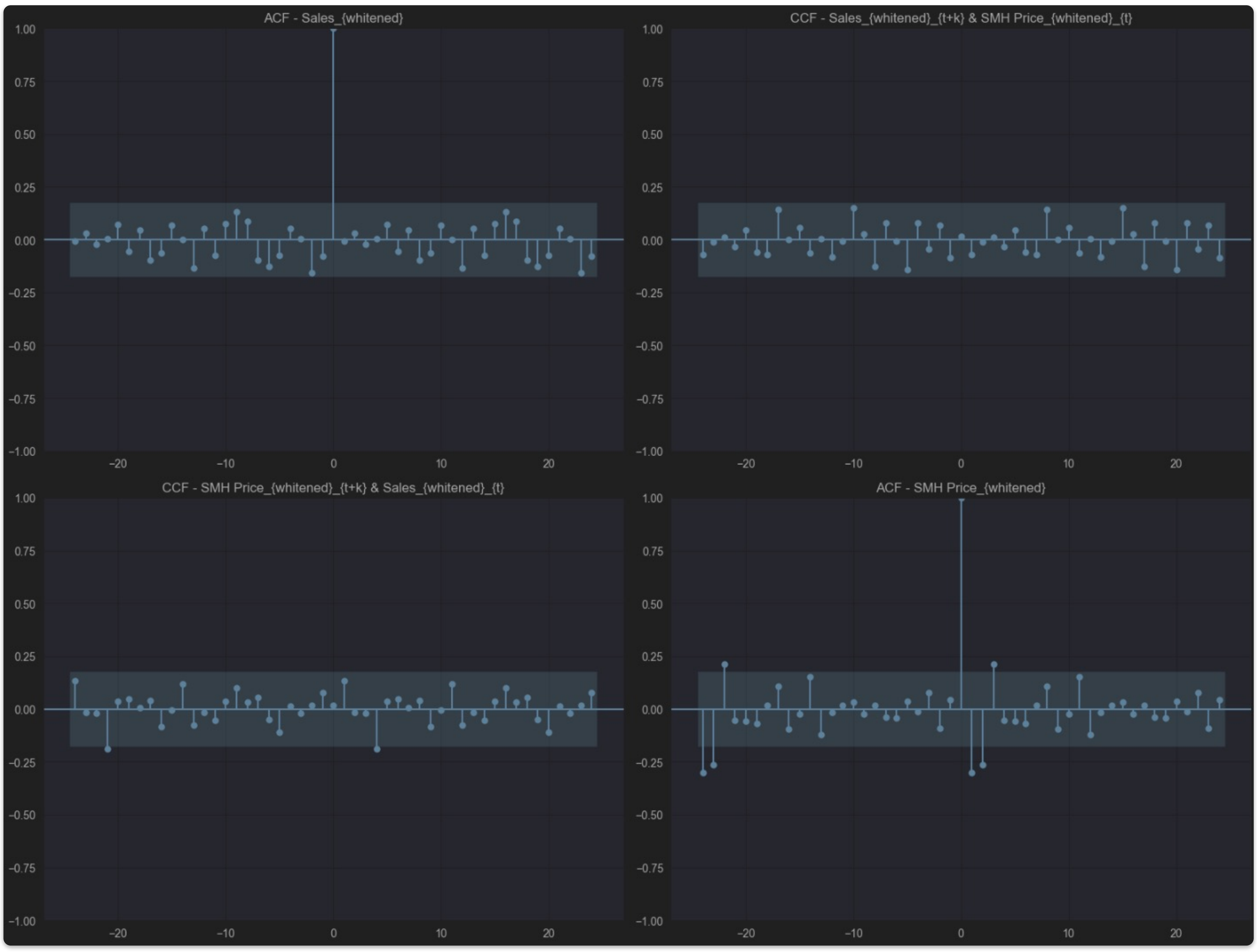
Diffed pair pre-whitened CCF plots



Boxplots at each period of the annual season that the (log-)changes in semiconductor sales exhibit

Transfer Function Modelling and Lagged Regression

As mentioned in the previous section, the chosen time-series pair has been deseasonalized via MA-based additive decomposition, $period = 12$, so that the pre-whitening model, which doesn't take into account seasonality, can transform the input value to something as close as white noise as possible. The CCF plots below illustrate what was obtained:


ACF - Sales_{whitened}

CCF - Sales_{whitened}_{t+k} & SMH Price_{whitened}_{t}

CCF - SMH Price_{whitened}_{t+k} & Sales_{whitened}_{t}

ACF - SMH Price_{whitened}

It's probably for this reason that task of trying to predict the SMH price (it's log returns) with the log change of the global semiconductor sales proved very difficult. In particular:

- Notice how the CCF plot of the pre-whitened time series doesn't show any significant correlation, meaning that the input series may not have any predictive power for the target series.
- The estimates for the coefficients of the transfer function are all extremely close to zero, possibly due to the extremely low covariance of the two time series. As a consequence, the estimated model predicts an almost constant 0 with a very wide prediction interval, resulting ineffective.
- None of the fitted lagged regression model proved effective. Most if not all coefficients weren't statistically significant with 0 being included in their 95% confidence interval. 95% Prediction intervals weren't satisfying either, being relatively large and failing to including actual values of the time series in several occasions.

Discussion and further perspectives

Task 1

Although the ARIMA(8, 1, 8) may have been identified as the slightly better model out of all the ones tried, the reality is that each of those proved equally ineffective in precisely predicting the SMH (log-)price, as shown by the just marginally different CV error rates and the general tendency to stick to the last seen level in-sample.
Stock prices are notoriously difficult to predict due to the complexity of the underlying process, meaning that underperforming with simple models such as those analyzed in this project is to be expected. Additionally, and for these reasons, the practice of directly modelling a stock price is generally avoided in a professional context, possibly making this nothing more than a fun exercise. Nonetheless, non-linear approaches that also take into account differ regimes of the time series may be a reasonable approach to improve on this work. Other improvements may derive from using a sliding-window-based CV approach, so

that only recent memory of the time series is modelled. The sliding window in fact identifies a limited set of samples to use to fit the model, and is moved until the end of the series is reached.

Task 2

Again, the task of predicting the price of a stock/etf, or its returns, proves very difficult, especially with relatively simple linear models such as the ones I used.
More complex models, possibly non-linear, may be considered, as well as a new set of predictors, since the global semiconductor sales do not seem to have meaningful predictive power for the target variable.