ISSN: 2317-0840

Impact of the COVID-19 pandemic on IBOVESPA: A statistical analysis with machine learning models PROPHET and AUTOARIMA

Antônio Victor Alves Silva[1], Erika Fialho Morais Xavier[2], Nyedja Fialho Morais Barbosa[1], Silvio Fernando Alves Xavier Junior[1†], Jader da Silva Jale[3]

[1] *Universidade Estadual da Paraíba (UEPB)*
[2] *Centro de Integração de Dados e Conhecimentos para Saúde(Cidacs) - Fiocruz*
[3] *Departamento de estatística e informática (DEINFO) - UFRPE*

Abstract: *This work analyzed the impact of the covid-19 pandemic in the year 2020 on Brazilian stocks using the Bovespa index and identified outliers in the data. It also observed a trend of stability in the following years, indicating economic recovery. The seasonality in the regular patterns was identified and represented in a line graph, highlighting the lowest medians in June and July. Prophet and autoARIMA models were used for forecasting, and the results were evaluated using various error metrics, including RMSE, MAE, SMAPE, MAPE, MASE, and RSQ. Although the Prophet model performed better with the differentiated data, the AutoARIMA model performed better with the original and log1p transformed data. The study is relevant to understand the impact of the pandemic on Brazilian stocks and how forecasting models can be used to assist in decision-making.*

Keywords: *Statistical modeling, Time series, Forecasting.*

Impacto da pandemia de COVID-19 no IBOVESPA: Uma análise estatística com os modelos de machine learning PROPHET e AUTOARIMA.

Resumo: *Esse trabalho analisou o impacto da pandemia de covid-19 no ano de 2020 nas ações brasileiras utilizando o indice bovespa e identificou valores atípicos nos dados, também observou uma tendência de estabilidade nos anos seguintes indicando recuperação econômica. A sazonalidade nos padrões regulares foi identificada e representada em um gráfico de linha, destacando as piores medianas nos meses de junho e julho. Foram utilizados modelos Prophet e autoARIMA para previsões, e os resultados foram avaliados usando várias métricas de erro, entre eles o RMSE, MAE, SMAPE, MAPE, MASE e RSQ. Embora o modelo Prophet tenha apresentado melhor desempenho com os dados diferenciados, o modelo AutoARIMA apresentou melhor desempenho com os dados originais e transformados com log1p. O estudo é relevante para entender o impacto da pandemia nas ações brasileiras e como os modelos de previsão podem ser usados para ajudar na tomada de decisões.*

Palavras-chave: *Modelagem estatística, Séries temporais, Previsão.*

Introduction

The impact of the sars-covid-19 pandemic on Brazilian stocks can be evaluated through the Bovespa Index. According to the information on the B3 website (s.d.), the Bovespa index is an essential measure of the performance of stocks traded on B3, bringing together the most significant companies in the Brazilian capital market. It is recalculated every four months and represents about 80% of the market's financial volume. The financial indicator uses the total return of stocks as a criterion, reflecting variations and distribution of dividends from companies, and is widely used as a benchmark for equity fund profitability and to assess the performance of the Stock Exchange.

It is essential to consider using time series to analyze the Bovespa index, as Nielsen (2021) pointed out in the book "Practical Time Series Analysis." Time series are statistical tools that predict future events based on past data. Their simplicity and transparency make them intuitive and accessible for use. Morettin and Toloi (2006) note that the main objectives of time series analysis are to describe the behavior of the series, investigate the data-generating mechanism,

†Autor correspondente: silvioxj@gmail.com

identify periodicities in the data, and make predictions of future values. Among these objectives, prediction is most commonly used, especially in economic and financial series.

Using the time series of Ibovespa, it is possible to train machine learning models such as Prophet and AutoARIMA. According to IBM Cloud Education (2022), the goal of machine learning is to simulate the way humans learn, gradually improving accuracy through the use of data and algorithms. By using the time series of Ibovespa as input data for machine learning, it is possible to train models that can predict the future behavior of the stock market.

According to a study conducted by the Getúlio Vargas Foundation (FGV), there was a significant decline in the performance of the Bovespa index between January and April 2020, recording a decrease of 32.4%, the highest among the world's major stock markets. Costa, da Silva and Matos (2021) found that the Brazilian stock market was highly responsive to COVID-19 data, both locally and internationally. This response varied across different time periods and frequencies and was influenced by the severity of the outbreak in Brazil. The study highlights the significant impact of the COVID-19 pandemic on Brazilian equities. In this context, the question arises about the ability of machine learning algorithms to accurately predict the behavior of the stock market after such a sharp decline.

The Prophet was created by *Facebook (s.d.)*, and according to it, it is a tool used to forecast information in time series. It uses an additive model to adjust non-linear trends, such as daily, weekly, yearly, and holiday seasonality. It is effective in time series with solid seasonal patterns and much historical data, and it can handle missing data and changes in trends, including outliers. *Facebook (s.d.)* states that many applications use the Prophet for accurate predictions, and it outperforms other techniques with quickly performed forecasts due to model adjustments.

The forecast package by Hyndman and Khandakar (2008), written in the R programming language (R CORE TEAM, 2021), provides the "auto.arima" function for the automatic fitting of ARIMA models. This function uses an exhaustive search method that considers all possible combinations of autoregressive and moving average terms within certain limits to select the best model that minimizes a loss function. However, this method can become impractical when simultaneously dealing with several series, especially for seasonal data generating hundreds or thousands of alternative models. To solve this problem, Hyndman and Khandakar (2008) developed a stepwise search algorithm that significantly reduces the number of tested combinations.

This work aims to analyze the impact of the covid-19 pandemic in the year 2020 on Brazilian stocks, specifically on the Bovespa index, through statistical analysis techniques. Additionally, the identification of seasonality in the index's regular patterns will also be performed. A forecasting strategy will be applied using the Prophet and AutoARIMA machine learning tools, evaluating the results with error metrics such as RMSE, MAE, SMAPE, MAPE, MASE, and RSQ. Finally, an analogy will be made between the performance of the models using the original and transformed data to assess which provides better accuracy in the 15-day forecast.

Methodology

The study used a dataset with information about the Bovespa Index obtained from Yahoo! Finance. The data collection was methodical and rigorous, covering the period from January 2, 2020, to December 29, 2022, based on the daily closing data, totaling 670 observations.

According to Morettin and Toloi (2006), there are two ways to assess a stochastic process's stationarity: strong and weak. The first type implies maintaining all finite-dimensional distributions unchanged over time, which means that the mean and variance of the process should remain constant under time translations. On the other hand, weak stationarity only requires that the mean and autocovariance be constant over time.

The Box-Jenkins methodology is commonly used to build parametric models for univariate time series, involving four main steps: specification, identification, estimation, and verification. During the specification step, a general class of models is considered, and then a model is identified through the analysis of autocorrelations, partial autocorrelations, and other criteria. The model parameters are estimated in the estimation step, and the chosen model is verified through residual analysis. If it is not suitable, the process returns to the identification phase.

Several models can be estimated in a time series, and the one with the slightest mean squared prediction error is sought if the purpose of estimation is forecasting. Some of the models used in the analysis are the autoregressive model (AR), which considers an autoregressive process of order p; the moving average model (MA), which considers a moving average of order q; the autoregressive moving average model (ARMA), which combines the AR and MA models and is used to represent time series; the autoregressive integrated moving average model (ARIMA), an extension of the ARMA model that is suitable for series that do not exhibit stationarity characteristics but can be transformed into stationary series through differencing; and the seasonal autoregressive integrated moving average model (SARIMA), which represents an extension of the ARIMA model, taking into account the seasonality of the data (Morettin and Toloi, 2006; Box et al., 2016). The formulas corresponding to each model are as follows:

$$\text{AR}: Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + ... + \phi_p Z_{t-p} + \epsilon_t \tag{1}$$

where ϕ_i for all $i = 1, 2, ..., p$ are model parameters, and ϵ_t is white noise at time t.

$$\text{MA}: Z_t = x_t - \theta_1 x_{t-1} - \theta_2 x_{t-2} - ... - \theta_q x_{t-q} \tag{2}$$

where there are q lags in the moving average, and $\theta_1, \theta_2, ..., \theta_q$ (with $q \neq 0$) are the parameters.

$$\text{ARMA}: Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + ... + \phi_p Z_{t-p} + x_t - \theta_1 x_{t-1} - \theta_2 x_{t-2} - ... - \theta_q x_{t-q} \tag{3}$$

where the ϕ's are autoregressive parameters, and the θ's are moving average parameters, with $\phi \neq 0$, $\theta \neq 0$, and $\sigma_x^2 > 0$.

$$\text{ARIMA}: W_t = \phi_1 W_{t-1} + \cdots + \phi_p W_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \cdots - \theta_q \epsilon_{t-q} \tag{4}$$

The ARIMA model is represented by the notation ARIMA(p, d, q). The parameters "p", "d" and "q" refer to the autoregressive terms, differentiation and moving average terms of the model, respectively. Here, W_t represents the differenced series, ϕ and θ are the autoregressive and moving average parameters, respectively, and ε_t is the error term.

$$\text{SARIMA}: \phi(B)\Phi(B)[(1-B)^d(1-B^s)^D - \alpha]Z_t = \theta(B)\Theta(B)\epsilon_t \tag{5}$$

The SARIMA model is represented by the notation SARIMA$(p, d, q)(P, D, Q)_s$, where "s" is the seasonal period. The parameters "p", "d", and "q" refer to the autoregressive terms, differentiation, and moving average terms of the non-seasonal components of the model. The parameters "P", "D" and "Q" refer to the same terms but for the seasonal components of the model.

To simplify the specification of an ARIMA model (AutoARIMA), Morettin and Toloi (2006) suggest using a method based on a penalizing function. This method determines the values of p, q, P, and Q that minimize a model's lack of fit function, using criteria such as AICc and BIC. When dealing with multiple series simultaneously, Hyndman and Khandakar (2008) propose a stepwise search algorithm that reduces the number of models to be tested. The algorithm initially fits four models and selects the lowest AICc or BIC. Then, variations in the selected model are considered, choosing the one with the lowest AICc or BIC. The process is repeated until no model with a lower AICc or BIC is found, ensuring a valid final model.

The autocorrelation function (ACF) measures the correlation between observations in a stationary time series. According to Montgomery, Jennings, and Kulahci (2015), if a time series is stationary, the joint probability distribution is the same for observations separated by a standard interval (lag) k. The ACF is calculated using the following:

$$r_k = \frac{\sum_{t=k+1}^{n}(W_t - \bar{W})(W_{t-k} - \bar{W})}{\sum_{t=1}^{n}(W_t - \bar{W})^2}, \quad k = 1, 2, ..., \tag{6}$$

where n is the number of observations and $\bar{W}$ is the mean of W.

The probability distribution of r_k follows a standard normal distribution $N(0, \frac{1}{n})$. It is important to understand the distribution of r_k for a satisfactory analysis. LÚCIA (2000) emphasizes the distribution of r_k for creating confidence intervals and hypothesis tests to verify the null correlation hypothesis.

The partial autocorrelation function (PACF), according to Shumway and Stoffer (2017), measures the correlation between observations in a time series after removing the effects of intervening variables. The PACF is represented by ϕ_{kk} and can be estimated using the Yule-Walker equations system. The ACF and PACF are useful in identifying models for time series and checking stationarity. The ACF and PACF allow for the construction of confidence intervals and hypothesis tests to verify the correlation in a time series.

The *Prophet*, offered by *Facebook*, is capable of automating time series forecasting, working with different levels of data granularity, including minute-to-minute or hourly data and daily data. It is also capable of understanding trends that grow in a non-linear fashion. The formula for the *Prophet* is given by:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{7}$$

Where $y(t)$ is the observed value in the time series at time t, $g(t)$ is the trend component, $s(t)$ is the seasonal component, $h(t)$ is the holiday component (provided by the user), and ϵ_t is the prediction error (NIELSON, 2021).

Prophet has been used in several works to forecast financial time series. For example, Yusof et al. (2021) used Prophet to forecast six different financial time series indexes, including the Standard Poor's 500 (SP500), the Dow Jones Industrial Average (DJIA), the China Securities (CSI300), the Kuala Lumpur composite (KLCI), Hong Kong's Hang Seng 300 (HS300) and Tokyo's Nihon Keizai Shinbun (Nikkei). Their results indicated that the Prophet model is competitive in modeling actual market movement simply by adopting appropriate parameters, where the Mean Absolute Percentage Error (MAPE) measure was at most 6%.

In another study, Sharma et al. (2022) used Prophet to predict the subsequent closing price of the highest-ranked bank stock on the NSE2. They found that Prophet gives a lower error rate and generates better predictions than the ARIMA model. Furthermore, Ghosh and Dragan (2022) used Prophet to discover the inherent pattern of financial stress across several critical variables. Their findings indicate that financial stress across assets and continents can be accurately predicted over both short and long-term horizon deadlines, even at a time of great financial stress during the COVID-19 pandemic. These studies demonstrate the effectiveness of the Prophet in forecasting financial time series and highlight its usefulness in various scenarios.

The *Prophet* includes annual and weekly seasonal effects components, a list of holidays, and a linear trend curve. According to information on the B3 website, the Bovespa index does not operate on national holidays, such as the Finados and Proclamation of the Republic. Therefore, a list of holidays was not included in the *Prophet*. It is essential to stay updated on changes in the Bovespa index's operation policy regarding holidays and reassess the need to include a list of holidays in the future.

According to Taylor and Letham (2017), Prophet is a useful tool that simplifies the task of creating reasonable and accurate forecasts. It provides various forecasting techniques such as ARIMA and exponential smoothing, each with its own strengths, weaknesses, and tuning parameters. The forecasts generated by Prophet can be customized in an intuitive way, even by non-experts. One can specify smoothing parameters for seasonality and trends, as well as manually set growth curves.

Six error metrics were used to assess the quality of Ibovespa index forecasts: RMSE, MAE, MAPE, SMAPE, MASE, and RSQ. The corresponding formulas for each error metric are as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2} \tag{8}$$

where n is the number of predictions, $\hat{y}t$ is the forecast for period t, and yt is the actual value of the index in period t.

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}|\hat{y}t - yt| \tag{9}$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{\hat{y}t - yt}{y_t}\right| \tag{10}$$

$$\text{SMAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\frac{|\hat{y}t - yt|}{(|\hat{y}t| + |yt|)/2} \tag{11}$$

$$\text{MASE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|\hat{y}t - yt|}{\frac{1}{h-1}\sum_{i=2}^{h}|y_i - y_{i-1}|} \tag{12}$$

where h is the number of lags in the time series.

$$\text{RSQ} = 1 - \frac{\sum_{t=1}^{n}(\hat{y}t - \bar{y})^2}{\sum t = 1^n(y_t - \bar{y})^2} \tag{13}$$

where $\bar{y}$ is the mean of the observed values in the time series.

Results

Exploratory data analysis

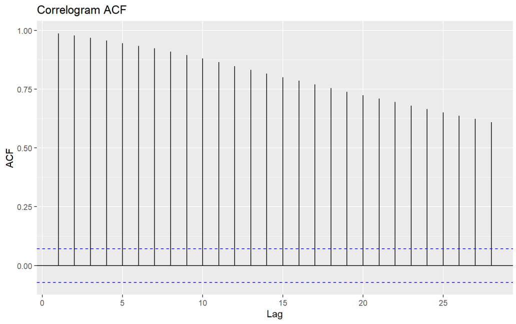
Figure 1 illustrates the behavior of the daily closing of the Bovespa Index during the analyzed period.

Figure 1: Behavior of the Daily Closing of the Bovespa Index by Line Chart.

Source: From the authors.

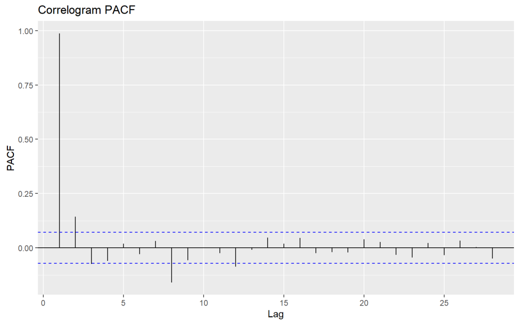
In Figure 1, it is possible to observe that at the beginning of 2020, there was a significant level change due to the pandemic (sars-covid-19). After this period, the market started to recover, showing an upward trend. From June-July 2021 onwards, the index stabilized, with variations occurring within a range of approximately 100,000 and 120,000 points.

Figure 2: ACF



Source: From the authors.

Figure 3: PACF



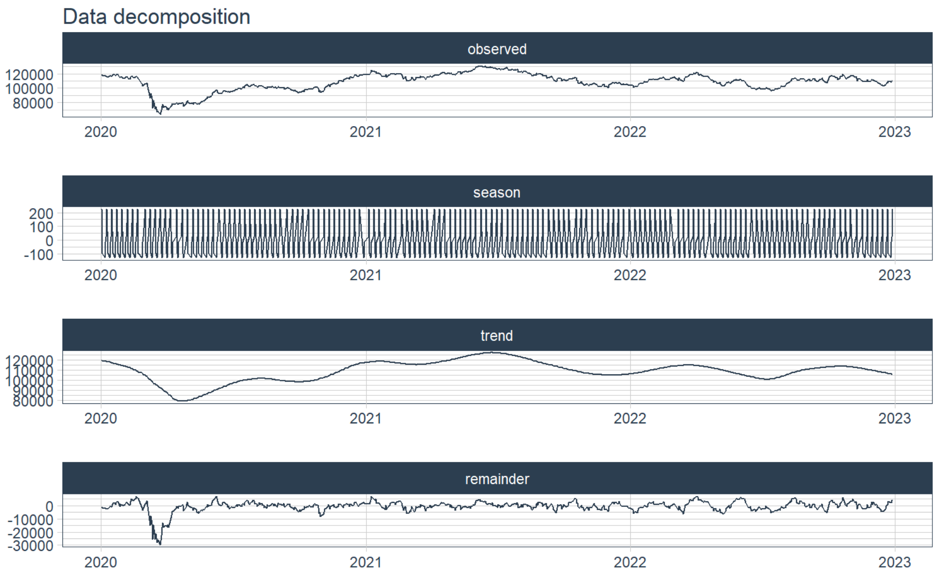
Font: From authors.

Based on the analysis of the FAC graph (Figure 2), slow decay of autocorrelation values can be observed as the lags increase (a strong indication of non-stationarity).

The PACF (Figure 3) reveals that the first two lags correlate significantly with the current point, while the remaining lags do not correlate strongly. This fact suggests a model with few parameters.

Figure 4 presents the decomposition of the data. Through this technique, it is possible to separate the series into its components, such as trend, seasonality, and residuals, thus allowing a better understanding of the patterns and behaviors exhibited by the data over time.

Figure 4: Decomposition of data

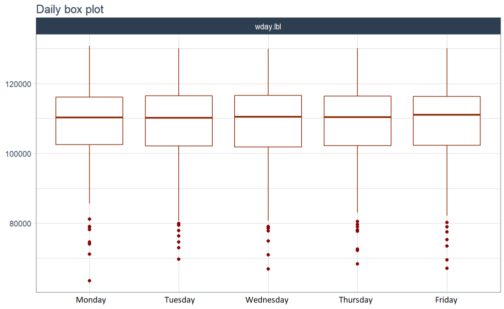


Source: From the authors.

Through decomposition (Figure 4), it was possible to identify the presence of seasonality and trend. Seasonality was observed through the regular patterns in the series, while the trend was represented by a line graph, showing how the series behaved over time. Additionally, the analysis of residuals allowed for the detection of significant variations in the time series, such as the sharp drop at the beginning of 2020 related to the pandemic, thus allowing for the assessment of the magnitude of this difference.

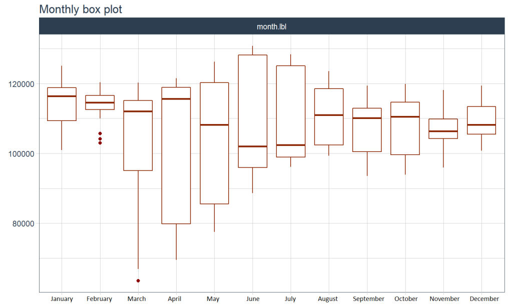
Boxplots related to different periods are presented in Figure 5, Figure 6, Figure 7, and Figure 8.

Figure 5: Daily boxplot



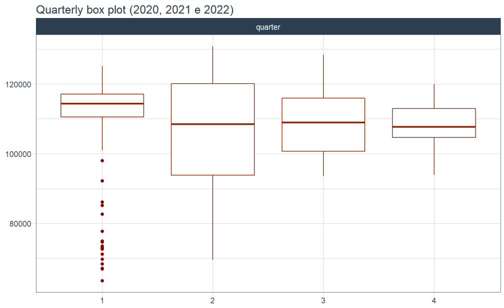
Source: From the authors.

Figure 6: Monthly boxplot



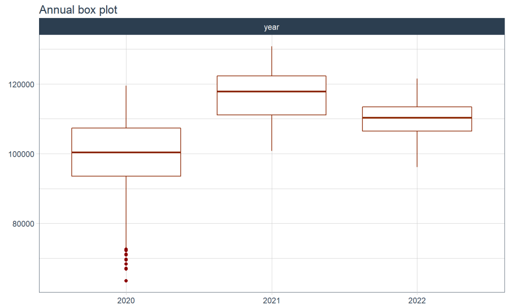
Source: From the authors.

Figure 7: Quarterly boxplot



Source: From the authors.

Figure 8: Annual boxplot



Source: From the authors.

When analyzing the daily data (Figure 5), it is possible to observe the presence of outliers and a slight variation in the medians. Despite the potential existence of seasonality in the data, it was not visually detected through the use of boxplots.

Regarding monthly data (Figure 6), it is evident that the initial months of the year, January, February, March, and April, have the highest medians among all the months. However, a decline can be observed shortly after these months. On the other hand, June and July have the lowest medians among all the analyzed months, suggesting a trend of economic decline during these periods. Conversely, a balance is observed from August to December, with prices becoming increasingly stable.

Evaluating the quarters from 2020 to 2022 (Figure 7), one can see:

In the first quarter, despite having the highest median among the analyzed quarters, there is a significant presence of outliers in the lower region of the boxplot. It can be attributed to the economic crisis caused by the pandemic in 2020. Although it presents a more dispersed boxplot in the second quarter, this quarter has the second-highest median among the quarters.

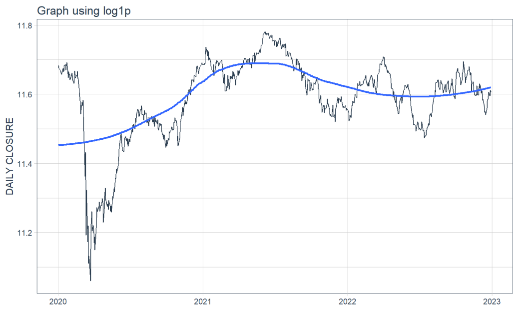
In comparison to the other quarters, the third quarter shows good behavior. However, it has the lowest median of all the quarters. On the other hand, the fourth quarter displays a more stable behavior with no significant variations. It has the third lowest median among all the quarters.

After analyzing the data from the years 2020, 2021, and 2022 (as shown in Figure 1), it can be observed that in 2020, there were more outliers in the lower region of the boxplot, possibly due to the pandemic. This indicates an unfavorable economic performance compared to the other years, with a median of around 100,000 points. In 2021, there was a recovery, and no outliers were observed. The values were higher than the previous year, with a median close to 120,000 points. Finally, in 2022, although the performance was lower than in 2021, no outliers were observed, and the median was around 110,000 points. In summary, the data shows a recovery after the sharp decline in 2020.

Data transformations

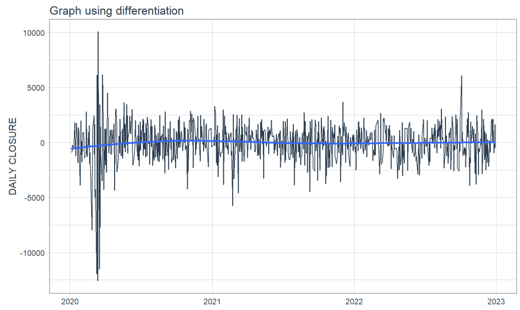
Log1p is a differentiation

Figure 9: Graph using Log1p transformation



Source: From the authors.

Figure 10: Graph using differentiation



Source: From the authors.

Due to the outliers found in the database, another transformation called log1p was also used, preserving the difference as essential but reducing it considerably (Figure 9). The logarithm of $(1 + x)$ is denoted as log1p according to the naming convention in scientific computing (LIU, 1988).

A differencing was applied to make the time series stationary (Figure 10). After this transformation, the visual analysis of the data suggests that they have become stationary, which is an essential condition for using forecasting models like ARIMA. However, it is crucial to perform other statistical tests to ensure the stability of the data, as visual analysis may not be sufficient.

However, the stationarity test (Table 1) was performed to verify the presence or absence of stationarity in the data.

Table 1: ADF Test

Type	**Lag**	**ADF**	**p-value**
	0	-31.77	0.01
	1	-19.14	0.01
	2	-14.73	0.01
1: no drift and no trend	3	-13.26	0.01
	4	-11.59	0.01
	5	-11.04	0.01
	6	-8.74	0.01
	0	-31.75	0.01
	1	-19.13	0.01
	2	-14.72	0.01
2: with drift and no trend	3	-13.25	0.01
	4	-11.58	0.01
	5	-11.03	0.01
	6	-8.73	0.01
	0	-31.74	0.01
	1	-19.12	0.01
	2	-14.72	0.01
3: with drift and trend	3	-13.25	0.01
	4	-11.58	0.01
	5	-11.03	0.01
	6	-8.73	0.01

∗ Note: actually, p-value = 0.01 means p-value <= 0.01
Source: From the authors.

The ADF tests (Table 1) show that the series becomes stationary when a differencing is applied (Figure 10), as the p-values are lower than the significance level of 0.01 for all tested hypotheses (no drift and no trend; drift and no trend; and drift and trend). In this way, rejecting the null hypothesis of non-stationarity is less likely to be an error.

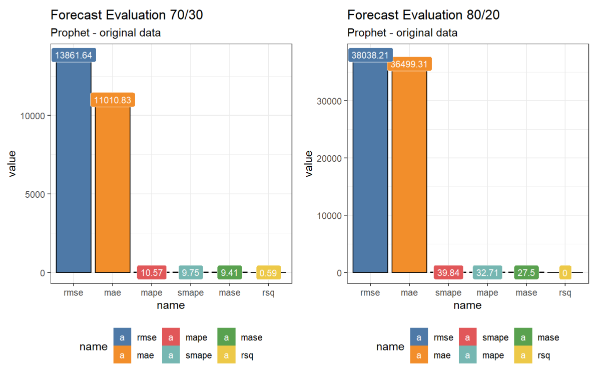
Prophet and AutoARIMA - training and testing

We employed two scenarios to create a database for training and testing purposes. The first scenario was the 70/30 proportion, and the second was the 80/20. We used these scenarios to evaluate the predictive models ability to generalize from the training data to the test data and determine which scenario performed better.

Prophet

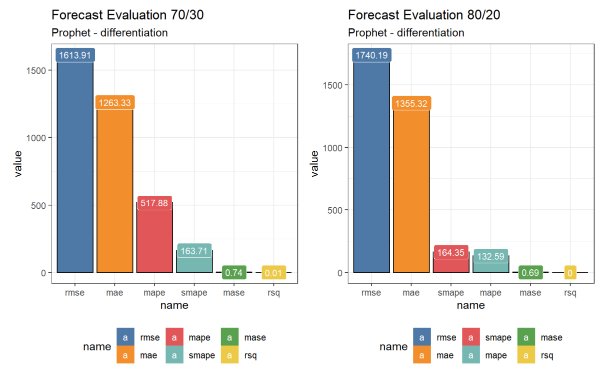
The Prophet model performance was evaluated via tests conducted using training and testing data and different transformations. The error metrics will be presented below (see Figures 11, 12, and 13).

Figure 11: Prediction errors of the original data



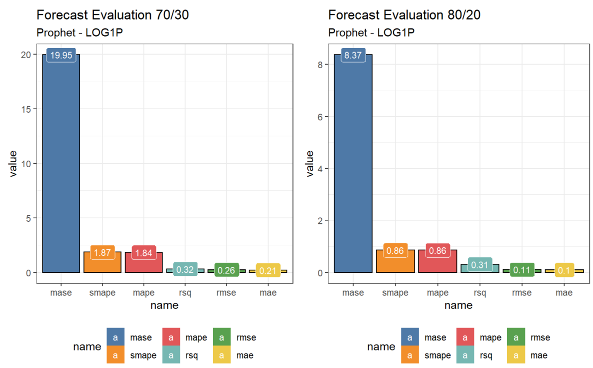
Source: From the authors.

Figure 12: Prediction errors of a differentiation



Source: From the authors.

Figure 13: Prediction errors of log1p



Source: From the authors.

After conducting tests with different training/test, it is observed that the Prophet model that presented the lowest error metrics when applied to the original data was the 70/30 proportion (Figure 11), therefore being the most recommended option for analyzing the dataset in question. The Prophet model (Figure 12), presented the lowest RMSE and MAE errors, making it the best option for time series forecasting. However, in terms of MAPE, SMAPE, MASE, and RSQ errors, the model did not yield the best results.

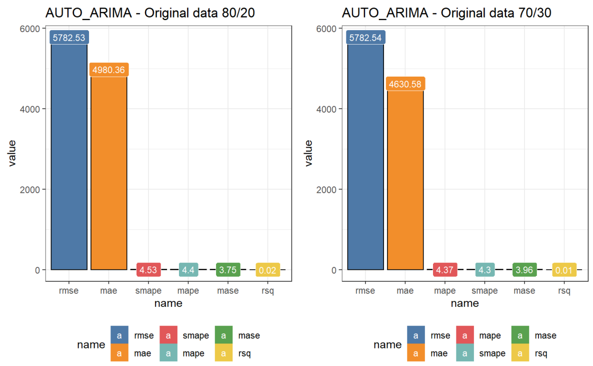
For the Prophet model with log1p transformation, it was observed that the strategy that achieved the lowest error metrics was the 80/20 proportion (Figure 13). Thus, it can be concluded that using this model with this strategy presents a more suitable data analysis option.

The conclusion that can be drawn is that, regarding the Prophet model, the configurations that presented the minor errors were: a 70/30 proportion for the original data with differentiation and an 80/20 proportion for log1p.

AutoARIMA

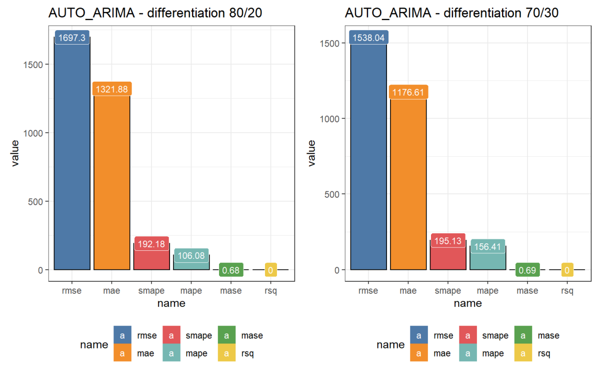
For the AutoARIMA models, we used the exact proportions of training and testing data for Prophet models and applied the same transformations. Error metrics are presented in Figures 14-16 to help select the most appropriate model.

Figure 14: Prediction errors of the original data



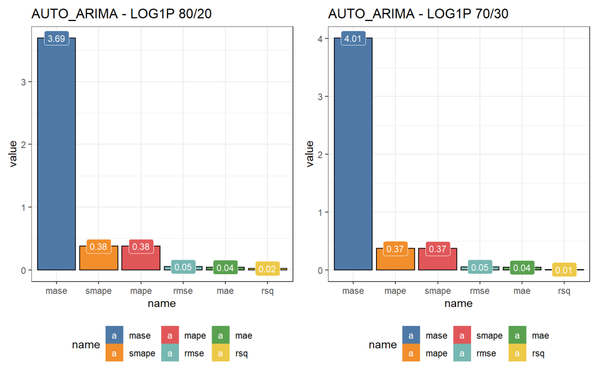
Source: From the authors.

Figure 15: Prediction errors of a differentiation



Source: From the authors.

Figure 16: Prediction errors of log1p



Source: From the authors.

An analysis of the metrics of autoARIMA forecasting models showed that the best performances were achieved for the original datasets with a train-test proportion of 70/30 (Figure 14), as well as for differenced datasets with the same proportion (Figure 15). As for the log1p transformation, the 80/20 ratio yielded the best results (Figure 16). It is essential to highlight that the results of the best performances obtained by autoARIMA were similar to those found by the Prophet model.

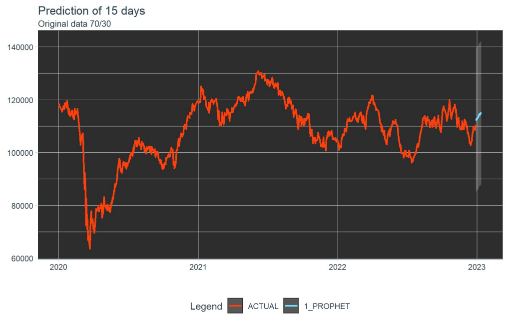
Prediction

A 15-day forecast will be performed using the data that exhibited the best error metrics. It is significant to highlight that this data has been converted back to its original form to ensure the accuracy and comprehensibility of the visualization of the final results. Additionally, the outcome will be compared to the actual values to obtain the prediction error. When making a forecast for a shorter number of days, such as 7 days, there may be an issue where the forecast is similar to the one made for 15 days but scaled down to only 7 days. Therefore, it is not advantageous to input forecasts for periods shorter than 15 days, as they present similar prediction errors.

Original data

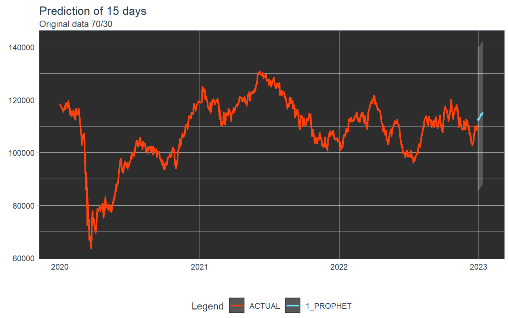
We used the Prophet and AutoARIMA models to perform a 15-day forecast based on the original data, which was divided into 70/30 proportions. The resulting predictions are shown in Figure 17 and Figure 18.

Figure 17: Prophet prediction of the original data 70/30



Source: From the authors.

Figure 18: AutoARIMA prediction of the original data 70/30



Source: From the authors.

The Table 2 and Table 3 show the percentage prediction error of two models, Prophet and AutoARIMA, for 15 days based on the original data divided into 70/30 proportions.

Table 2: Prophet (original data) - difference (%) of actual vs forecasted (15 days)

Date	Forecast	Actual	Diff(%)
30/12/2022	112,106.20	-	-
31/12/2022	112,707.60	-	-
01/01/2023	112,919.20	-	-
02/01/2023	112,754.40	106,376.00	5.99
03/01/2023	113,092.20	104,166.00	8.56
04/01/2023	113,327.60	105,334.00	7.57
05/01/2023	113,475.00	107,518.00	5.55
06/01/2023	113,664.10	108,836.00	4.42
07/01/2023	114,295.90	-	-
08/01/2023	114,523.30	-	-
09/01/2023	114,360.00	109,227.00	4.71
10/01/2023	114,685.40	110,912.00	3.39
11/01/2023	114,895.80	111,763.00	2.80
12/01/2023	115,006.60	111,877.00	2.79
13/01/2023	115,149.00	111,036.00	3.70

Source: From the authors.

Table 3: AutoARIMA (original data) - difference (%) of actual vs forecasted (15 days)

Date	Forecast	Actual	Diff(%)
30/12/2022	110,213.30	-	-
31/12/2022	110,262.80	-	-
01/01/2023	110,202.40	-	-
02/01/2023	110,325.40	106,376.00	3.72
03/01/2023	110,447.10	104,166.00	6.03
04/01/2023	110,852.50	105,334.00	5.00
05/01/2023	110,725.60	107,518.00	2.66
06/01/2023	110,540.70	108,836.00	1.56
07/01/2023	110,618.10	-	-
08/01/2023	110,656.00	-	-
09/01/2023	110,800.60	109,227.00	1.44
10/01/2023	110,785.70	110,912.00	-1.11
11/01/2023	110,669.30	111,763.00	-0.98
12/01/2023	110,613.20	111,877.00	-1.13
13/01/2023	110,673.00	111,036.00	-0.32

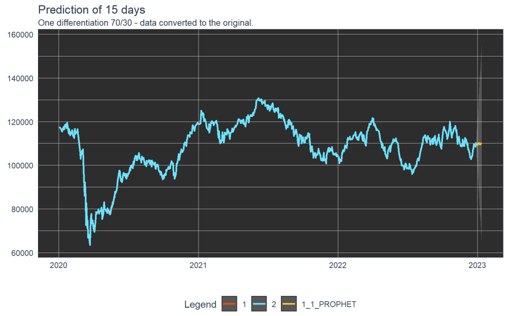
Source: From the authors.

When comparing the prediction error table of the original data in Table 2 with the one found in Table 3, it is possible to verify that the AutoARIMA model, whose performance is illustrated in Figure 18, achieved a superior predictive result compared to the Prophet model (Figure 17).

With a differentiation

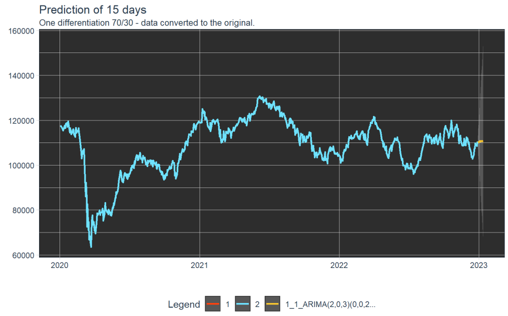
Based on the differentiated data, 15-day forecasts were made using the Prophet and AutoARIMA models, with the data divided into 70/30 proportions. The resulting predictions can be visualized in Figure 19 and Figure 20.

Figure 19: Prophet prediction of data with a 70/30 proportion



Source: From the authors.

Figure 20: AutoARIMA prediction of data with a 70/30 proportion



Source: From the authors.

The Table 4 and Table 5 present the prediction error percentage of two models, Prophet and AutoARIMA, for 15 days. This analysis was performed based on differenced data divided into 70/30 proportions.

Table 4: Prophet (one differentiation) - difference (%) of actual vs forecasted (15 days)

Date	**Forecast**	**Actual**	**Diff(%)**
30/12/2022	109,824.90	-	-
31/12/2022	109,569.90	-	-
01/01/2023	109,315.50	-	-
02/01/2023	109,397.80	106,376.00	2.84
03/01/2023	109,816.80	104,166.00	5.44
04/01/2023	110,033.70	105,334.00	4.47
05/01/2023	110,152.90	107,518.00	2.46
06/01/2023	110,217.10	108,836.00	1.27
07/01/2023	109,922.20	-	-
08/01/2023	109,614.70	-	-
09/01/2023	109,631.80	109,227.00	0.37
10/01/2023	109,975.20	110,912.00	-0.85
11/01/2023	110,107.90	111,763.00	-1.48
12/01/2023	110,136.40	111,877.00	-1.56
13/01/2023	110,105.80	111,036.00	-0.84

Source: From the authors.

Table 5: AutoARIMA (one differentiation) - difference (%) of actual vs forecasted (15 days)

Date	**Forecast**	**Actual**	**Diff(%)**
30/12/2022	110,213.3	-	-
31/12/2022	110,262.8	-	-
01/01/2023	110,202.4	-	-
02/01/2023	110,325.4	106,376.00	3.72
03/01/2023	110,447.1	104,166.00	6.05
04/01/2023	110,852.5	105,334.00	5.23
05/01/2023	110,725.6	107,518.00	2.89
06/01/2023	110,540.7	108,836.00	1.57
07/01/2023	110,618.1	-	-
08/01/2023	110,656.0	-	-
09/01/2023	110,800.6	109,227.00	1.44
10/01/2023	110,785.7	110,912.00	-1.02
11/01/2023	110,669.3	111,763.00	-0.98
12/01/2023	110,613.2	111,877.00	-1.12
13/01/2023	110,673.0	111,036.00	-3.27

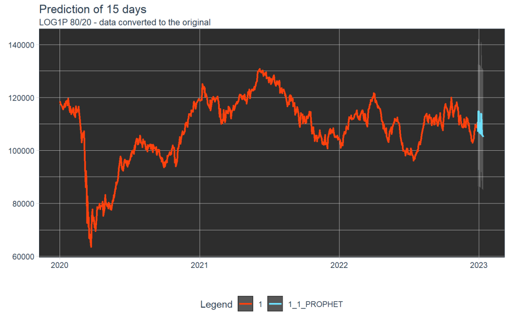
Source: From the authors.

When comparing the prediction error table with a differentiation present in Table 4 to the one found in Table 5, it can be observed that the Prophet model, whose performance is illustrated in Figure 19, presented superior predictive performance compared to the AutoARIMA model (Figure 20).

Log1p

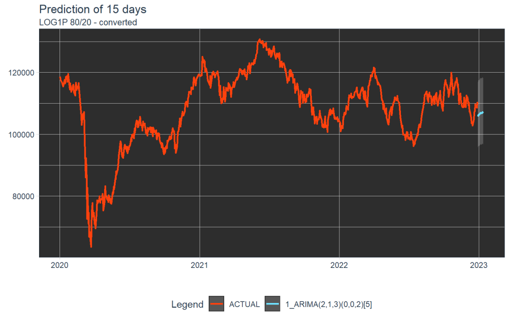
Based on the data using the log1p transformation, a 15-day forecast was performed using the Prophet and AutoARIMA models, where the data was divided into 80/20 proportions. The figures (Figure 21 and Figure 22) show the resulting predictions.

Figure 21: Prophet prediction of log1p data 80/20



Source: From the authors.

Figure 22: AutoARIMA prediction of log1p data 80/20



Source: From the authors.

The Table 6 and Table 7 display the percentage error rate of the Prophet and AutoARIMA models in predicting 15 days based on data transformed with the log1p function and divided in 80/20 proportions.

Table 6: Prophet (Log1p) - difference (%) of actual vs forecasted (15 days)

Date	Forecasted	Actual	Diff(%)
30/12/2022	107,007.40	-	-
31/12/2022	114,716.50	-	-
01/01/2023	114,629.50	-	-
02/01/2023	106,714.80	106,376.00	0.32
03/01/2023	106,810.00	104,166.00	2.54
04/01/2023	106,806.40	105,334.00	1.40
05/01/2023	106,625.30	107,518.00	-0.83
06/01/2023	106,315.20	108,836.00	-2.32
07/01/2023	113,903.00	-	-
08/01/2023	113,735.90	-	-
09/01/2023	105,799.90	109,227.00	-3.14
10/01/2023	105,804.20	110,912.00	-4.61
11/01/2023	105,705.00	111,763.00	-5.43
12/01/2023	105,425.70	111,877.00	-5.79
13/01/2023	105,016.40	111,036.00	-5.42

Source: From the authors.

Table 7: AutoARIMA (Log1p) - difference (%) of actual vs forecasted (15 days)

Date	Forecasted	Actual	Diff(%)
30/12/2022	105,783.10	-	-
31/12/2022	106,210.40	-	-
01/01/2023	106,377.30	-	-
02/01/2023	106,255.80	106,376.00	-0.11
03/01/2023	106,466.00	104,166.00	2.21
04/01/2023	106,620.40	105,334.00	1.22
05/01/2023	106,812.50	107,518.00	-0.66
06/01/2023	106,998.00	108,836.00	-1.69
07/01/2023	106,782.80	-	-
08/01/2023	106,808.90	-	-
09/01/2023	106,864.80	109,227.00	-2.16
10/01/2023	107,097.10	110,912.00	-3.44
11/01/2023	107,146.40	111,763.00	-4.12
12/01/2023	106,994.80	111,877.00	-4.34
13/01/2023	106,870.80	111,036.00	-3.77

Source: From the authors.

Based on the comparison of the log1p prediction error tables presented in Table 6 and Table 7, it was found that the AutoARIMA model, whose performance is illustrated in Figure 22, achieved superior predictive performance compared to the Prophet model (Figure 21).

Conclusion

Based on the analysis, it is concluded that the covid-19 pandemic in the year 2020 significantly impacted Brazilian stocks, generating atypical values in the data. However, the following years showed a trend of stability, indicating an economic recovery. Seasonality was identified in the regular patterns present in the series, and the trend was represented in a line graph, highlighting the lowest medians observed in June and July.

The models *Prophet* and autoARIMA were used, which showed better performance with original data and differentiation in a 70/30 ratio and log1p transformation in an 80/20 ratio. The error metrics used were RMSE, MAE, SMAPE, MAPE, MASE, and RSQ.

Upon analyzing the 15-day prediction error tables (%) of the Prophet and AutoARIMA models, it was found that the AutoARIMA model achieved superior predictive performance to the Prophet model in forecasting both the original data and the log1p-transformed data. However, when considering the differenced data, the Prophet model stood out. Despite demonstrating superior predictive performance only on the differenced data, the Prophet model was deemed the best, indicating that differencing was an effective strategy to enhance its forecasting ability for a 15-day horizon of the Bovespa index time series.

Although Prophet is designed to handle trends and seasonality, if the data has a very complex or non-linear trend or seasonality, differencing can help simplify this structure and potentially improve model performance.

References

Antonio Costa, Cristiano da Silva, Paulo Matos (2022). The Brazilian financial market reaction to COVID-19: A wavelet analysis. International Review of Economics & Finance, Volume 82, Pages 13-29. Available at: https://doi.org/10.1016/j.iref.2022.05.010. Accessed on: 20 Feb. 2024.

B3. Índice Ibovespa. Available at: Link to acess. Accessed on:21 feb. 2023.

B3. Calendário de negociação. Available at: Link to acess. Accessed on: 21 feb. 2023.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time Series Analysis: Forecasting and Control (5th ed.). Hoboken, NJ: John Wiley & Sons. ISBN 978-1118675021.

FGV. COVID-19 E MERCADO FINANCEIRO. Rio de Janeiro: FGV, 2020. Available at:Link to acess. Accessed on: 21 feb. 2023.

HYNDMAN, R. J.; KHANDAKAR, Y. Automatic Time Series Forecasting: the forecast package for R. *Journal of Statistical Software*, v. 27, n. 3, p. 1-22, 2008.

IBM Cloud Education. Machine Learning. Available at: https://www.ibm.com/br-pt/cloud/learn/machine-learning. Accessed on: 03 jan. 2023.

Liu, Z. A. Berkeley elementary function test suite. M.S. thesis, Computer Science Division, Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA, USA, December 1987.

LÚCIA, F. A. V. A. V. Manual de econometria. Vasconcelos, MAS; Alves, D. São Paulo: Editora Atlas, 2000.

MONTGOMERY, D. C.; JENNINGS, C. L.; KULAHCI, M. Introduction to time series analysis and forecasting. [S.l.]: John Wiley & Sons, 2015.

MORETTIN, P. A.; TOLOI, C. Análise de séries temporais. In: Análise de séries temporais. [S.l.: s.n.], 2006. p.

Nielsen, A. (2021). Análise Prática de Séries Temporais. São Paulo: Alta Books.

Facebook. Prophet: Forecasting at Scale. Available at: https://facebook.github.io/prophet/. Accessed on 13 feb. 2023.

Ghosh, I., Dragan, P. Can financial stress be anticipated and explained? Uncovering the hidden pattern using EEMD-LSTM, EEMD-prophet, and XAI methodologies. Complex Intell.

Syst. 9, 4169–4193 (2023). https://doi.org/10.1007/s40747-022-00947-8. Accessed on: 20 Feb. 2024

R CORE TEAM. R: a language and environment for statistical computing. R Foundation for Statistical Computing: Vienna, Austria. Available at http:www.R-project.org, 2021.

Sharma, K., Bhalla, R., & Ganesan, G. (2022). Time Series Forecasting Using FB-Prophet. Available at: https://ceur-ws.org/Vol-3445/PAPER_07.pdf. Accessed on: 20 Feb. 2024.

SHUMWAY, R.; STOFFER, D. Time series analysis using the R Statistical Package.[S.l.]: free dog publishing, 2017.

Taylor, S. J., & Letham, B. (2017). Prophet: forecasting at scale. Facebook Research. Available at:
https://research.facebook.com/blog/2017/02/prophet-forecasting-at-scale/. Accessed on: 21 feb. 2023.

Yahoo! Finanças. *IBOVESPA* (ˆBVSP). Available at:
https://br.financas.yahoo.com/quote/%5EBVSP/history/. Accessed on: 31 Dec. 2022.

Yusof, U.K., Khalid, M.N.A., Hussain, A., Shamsudin, H. (2021). Financial Time Series Forecasting Using Prophet. In: Saeed, F., Mohammed, F., Al-Nahari, A. (eds) Innovative Systems for Intelligent Health Informatics. IRICT 2020. Lecture Notes on Data Engineering and Communications Technologies, vol 72. Springer, Cham. Available at: https://doi.org/10.1007/978-3-030-70713-2_45. Accessed on: 20 Feb. 2024.