*Andres Torres Salinas*

*Title*

*“Time Series Analysis for IBM”*

Table of contents

[*1. Dataset Selection* 3](#_Toc184421352)

[*2. Data Collection and Preprocessing* 3](#_Toc184421353)

[*2.1 Brief description of the data source* 3](#_Toc184421354)

[*2.2 Preprocess of the data* 4](#_Toc184421355)

[*2.3 Exploration of the data* 6](#_Toc184421356)

[*2.4 Differentiation Transformation* 12](#_Toc184421357)

[3. Model Building 15](#_Toc184421358)

[*3.1 ACF & PACF* 15](#_Toc184421359)

[*3.2 Training and test split ratio* 16](#_Toc184421360)

[*3.3 Arima model* 16](#_Toc184421361)

[*3.2 SARIMA Model* 19](#_Toc184421362)

[*4. Model Evaluation* 20](#_Toc184421363)

[*4.1 Evaluation of best Arima model performance* 20](#_Toc184421364)

[*4.2 Evaluation of best Sarima model performance* 22](#_Toc184421365)

[*5. Conclusion (Interpretation)* 23](#_Toc184421366)

*Time Series Analysis for IBM*

# *1. Dataset Selection*

For this project I choose IBM (International Business Machines Corporation) stock price, is a multinational technology firm that provides a variety of hardware, software, and services. Its stock price, like any other technology firm, is influenced by a variety of factors, including technological breakthroughs, economic conditions, and industry trends. The technology business is noted for its volatility, with stock prices fluctuating rapidly due to factors such as new debuts, mergers and acquisitions, and changes in market sentiment. For that reason, this stock was selected for the analysis since it is challenged to assess correctly a model for predicting the stock price. I decided to use 231 data values, starting from first of January of this year to the end of the month of November.

# *2. Data Collection and Preprocessing*

## *2.1 Brief description of the data source*

Yahoo Finance is an online platform that provides financial news, stock market statistics, and investment tools. It offers real-time quotes, historical data, corporate financials, and investment tools such as stock screens and portfolio management. The site also includes market research, interactive charts, and currency exchange information, making it a popular tool for investors, traders, and analysts to track investments and keep informed about financial markets. For this project, this site was selected because for the ease usage of the data in python for the time series analysis as it serves as a library that can provide databases of the stock market in real time for any stock.



* To use this code, I import the yfinance library as yf.

A screenshot of a computer

Description automatically generated

* After this I decided to use the IBM stock price, so I set the variable of the ticker, import it from yfinance with yf.download, selecting a timelapse of ytd until the 30 of November and convert it to csv.

## *2.2 Preprocess of the data*

A screenshot of a computer screen

Description automatically generated

Since I used yfinance in Google Collab I needed to skip the first two rows of the dataset and rename each column to original column name.

A screenshot of a computer program

Description automatically generated

In the step 5 as it says I read the column names to know what features have been downloaded to check which columns I will use for the time series analysis. After checking the features, I decided to use the adjusted close column because it gives to the investors the more current and precise information of the stock’s price as it shows the close of the stock after any corporate action.

A screenshot of a computer program

Description automatically generated

The next step in the project was to check the properties of the column. The data shows that there is not missing values in the dataset, so we don’t need to handle any type of missing values. in this case, the only next step needed in the data preprocessing process that it showed the .info() method was to change the data type of the column of date to datetime. The code for this is shown below:

A screenshot of a computer

Description automatically generated

## *2.3 Exploration of the data*

In the initial part of the exploration process we look for the descriptive analytics with the .describe() method.

A screenshot of a computer

Description automatically generated

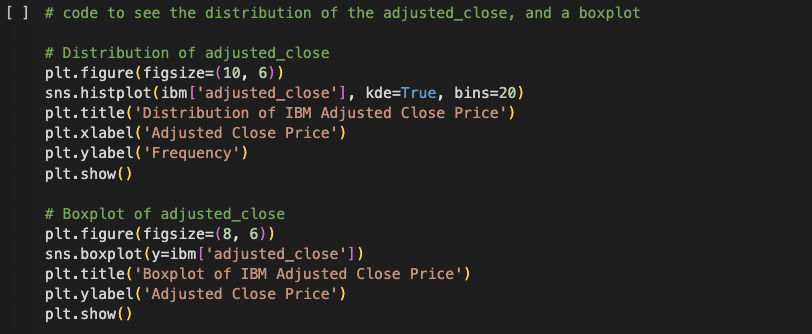
It shows some key information such a mean that is 187. As it was mentioned before, the mean was around this value indicating that the stock price was on average during this year on average on that value, another key aspect apart from the percentiles, min and max is the std that shows the volatility of the data, this std is moderate.

A screen shot of a computer code

Description automatically generated

The code calculates the percentage growth of IBM's adjusted closing stock price. It retrieves the initial and final prices, computes the growth using a formula ((final - initial) / initial) \* 100, and prints the result that it is 45.90%, since it is analyzing the growth of the stock price until November, we can interpretate that this a high grow for 11 months.

Another analytics distribution analysis, boxplot, scatterplot, timeseries and a seasonal decompose. The code below is the one that was used to create visualizations for the distribution and boxplot of the data.



I decided to do 20 bins due to the size of the data that is equivalent of 231.

A graph of a distribution of the company's stock market

Description automatically generated with medium confidence

The IBM adjusted close price distribution shows that there are two distinct peaks (bimodal) and that is right-skewed, suggesting two price trends, the most frequent prices are around 180 and 210, with a widespread indicating price variability. No outliers are observed at all although the variability is quietly high. The bimodality distribution of IBM's stock price may indicate conflicting investor perspectives on the company's long-term prospects vs short-term concerns. These perspectives may be influenced by economic variables such as inflation, interest rates, and the global outlook, as well as environmental concerns and regulatory changes. The right-skewness can indicate that mostly of the year the stock price was around the range of the largest peak and then a volatile upward trend enters into the stock price dynamic.

A diagram of a box plot

Description automatically generated

The boxplot for IBM's adjusted close price shows a symmetrical distribution with no outliers. The median price is at 185, with almost half of the values falling between 175 and 200. The whiskers extend to the minimum and maximum values, representing the data's range. This indicates a consistent price range for IBM's shares during the investigated timeframe.

A screen shot of a computer program

Description automatically generated

The code above shows how was made the scatter plot and time series plot for the IBM adjusted price.

A graph with blue dots

Description automatically generatedA graph with blue lines

Description automatically generatedThe figure displays IBM's adjusted close price in 2024, indicating an overall rising trend with volatility. The price began near $160 in January, fluctuated moderately throughout the year, and peaked above $230 in October. A dip followed, but it recovered in November, indicating resiliency. This pattern illustrates probable investor trust in IBM, with the significant increase in Q3-Q4 implying a strong market response during this time. *A screen shot of a graph

Description automatically generated*

The figure depicts IBM stock's adjusted close price, as well as its 7-day and 30-day moving averages. Moving averages reduce short-term price swings, making it easier to discern trends. The 7-day moving average responds more quickly to current price fluctuations, whereas the 30-day moving average offers a longer-term perspective. When the 7-day moving average crosses above the 30-day moving average, it is often interpreted as a positive indicator, indicating a probable upward trend. In contrast, a crossover below the 30-day moving average might be interpreted as a negative indication.

A black screen with purple and white text

Description automatically generated

A graph of a graph

Description automatically generated with medium confidence

Moving Forward, the next step that I followed was looking for the seasonal decompose, for this step I imported the seasonal\_decompose component from the statsmodels.tsa.seasonal library. The decomposition of IBM's adjusted close price reveals an upward trend, indicating long-term growth following an initial fall. Seasonal changes are consistent, demonstrating periodic influences on stock prices. Residuals are randomly distributed around zero, indicating that the data follows a trend and seasonality, with little unexplained variation on the residuals. Overall, the stock grows steadily and follows predictable seasonal patterns, providing insights into market behavior and showing the firm's long-term durability indicating that the data need to be transformed with a differentiation method or a logarithmic to make it stationary.

## *2.4 Differentiation Transformation*

A computer code with colorful text

Description automatically generatedA screenshot of a computer program

Description automatically generated

Then we tried different differentiation methods to find the best p-value, including first differentiation, second differentiation and logarithmic differentiation. In the code of the right, it calculates the adf test to see which one is stationary or not. It was extracted the p-value of this test to see if the differentiation was significant enough to reject the null hypothesis. The first p-value is for the original data is showing that it was necessary to be transformed. Afterwards, I checked the different outcomes, and I choose the first differentiation to make the transformation for my model. I decided this to do not have a overcomplexity issue in my mode.

A black screen with white text

Description automatically generated

I just did the same seasonal decompose analysis of the IBM following the same code as it shows above.

A diagram of a graph

Description automatically generated with medium confidence

The decomposition of the diff data reveals strong short-term volatility in the observed data, with only minor long-term trend changes. A strong seasonal component prevails, showing repeated, periodic trends that influence the adjusted closure price. The residuals are randomly distributed, indicating that the model successfully captures both trend and seasonal components with little unexplained variability. Overall, the research shows that seasonal impacts dominate the data, rather than long-term development or decline.

A screen shot of a computer screen

Description automatically generated

The lag plot just needed to put the data that you want to analyze and the quantity of the lags you want to analyze. The lag plot of IBM's adjusted close price with a lag of one displays a random scatter pattern. There is no strong linear relationship between the present and prior prices. The price changes appear to be independent of one another, implying that past price movements do not accurately predict future price movements.

*A screen shot of a computer program

Description automatically generated*

A diagram of a distribution of a number

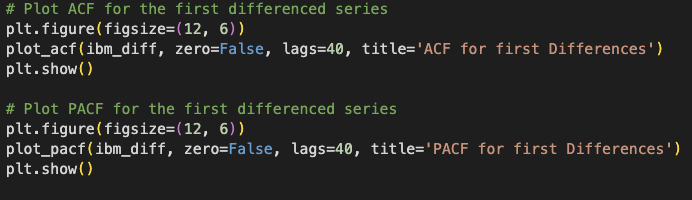
Description automatically generatedA blue line with white text

Description automatically generated

After that I checked the distribution and boxplot of the adjusted\_close price column of the diffentation to see if it now looks normally distributed, and it looks normal, except for outliers that indicates big drops or big rises between consecutive data points, so I’m not removing the outliers for the model since I don’t want to remove this from my model.

# 3. Model Building

## *3.1 ACF & PACF*



The code displays the autocorrelation (ACF) and partial autocorrelation (PACF) of the differenced IBM stock price data. These graphs are used to highlight trends and dependencies while selecting parameters for time series forecasting models.

A graph with blue dots and numbers

Description automatically generatedA graph with numbers and dots

Description automatically generated

The ACF plot for the initial differences in IBM's adjusted close price indicates that the autocorrelation coefficients are generally inside the confidence bands, implying that the differenced series is stationary. This suggests that the series' mean, and variance is constant across time. The lack of significant autocorrelation indicates that price movements are unpredictable based on previous price changes.

The PACF plot for the first differences in IBM's adjusted close price indicates that the partial autocorrelation coefficients are mainly inside the confidence bands, implying that the differenced series is stationary. This suggests that the series' mean, and variance is constant across time. The lack of considerable partial autocorrelation indicates that price movements are unpredictable based on previous price changes.

To discover the specific ARIMA model for IBM's stock price, it would need to conduct modelling and testing thorough the code or using pmdarima (pmdarima is a statistical library that serves to look for the best fit to a time series) as I did. However, using the provided ACF and PACF charts, we can make a judgment. Based on the analysis, an ARIMA (1,1,1) model could be a suitable place to start. Or Maybe a (0,1,0). Although probably a SARIMA model with a seasonal component monthly or weekly could be another good fit.

## *3.2 Training and test split ratio*

A computer screen with text and numbers

Description automatically generated

The code divides the IBM stock data into two sets: a training set (75%), and a testing set (25%), for model evaluation. It used iloc for index-based selection and displays the shapes of the resulting DataFrames (train\_data, test\_data) to confirm the split.

## *3.3 Arima model*

*A computer screen shot of text

Description automatically generated*

This code sample initially sets up the pmdarima library. The auto\_arima function is then used to automatically find the best ARIMA model parameters (p, d, and q) for the adjusted closing price of IBM stock (test\_data['adjusted\_close']). The model is then trained (fitted) on the complete dataset (ibm['adjusted\_close']), and a model summary is printed.A screenshot of a computer

Description automatically generated

The best ARIMA model employs AR (1), I (1), and MA(0). The autoregressive effect is mild and possibly inconsequential (p = 0.573). Differencing was used to attain stationarity. Model fit statistics (AIC, BIC, and HQIC) appear reasonable, indicating promise for predicting but requiring comparison with other models. Sigma represents the magnitude of the forecast mistakes.

A screenshot of a computer screen

Description automatically generated

The code builds an ARIMA model to predict IBM's adjusted closing stock price. It uses historical data, splits it into training and testing sets, and fits the ARIMA model with specific parameters. Predictions are made on the test data, and the model's accuracy is evaluated using Root Mean Squared Error (RMSE). Finally, a plot is generated to visualize the actual stock prices, training data, and predictions for comparison and model assessment. After testing this model, I found out that it wasn’t the best model, but it was a good starting point to look for the best model.

A screen shot of a computer program

Description automatically generatedA graph with a line graph

Description automatically generated with medium confidence

The Arima model prediction looks clearly that its asses well the trend of the data although it is not capturing the volatility that this data naturally it has. Although trying with different ARIMA models, these ones resulted as the one that had the best fit for this data having an order of (0, 2, 1).

## *3.2 SARIMA Model*

A screen shot of a computer program

Description automatically generatedA graph of a graph

Description automatically generated with medium confidence

The SARIMA model prediction looks clearly that its asses well the trend of the data and much better to capture the volatility that this data naturally it has due to the seasonality and other external factors. I choose a seasonal factor of m=23 to evaluate a monthly seasonality because since it captures stock price on business days it is the average of business days the months has, these ones resulted as the best model of all since it had the best RMSE of all for this data having an order of (1, 1, 0) and a seasonal order of (1, 1, 1)[23].

# *4. Model Evaluation*

## *4.1 Evaluation of best Arima model performance*

A screen shot of a computer program

Description automatically generated

This code measures the accuracy of a time series forecasting model for IBM's adjusted closing stock price using three major performance metrics: MAE, MSE, and RMSE. MAE is the average absolute difference between predicted and actual values, MSE is the average of squared differences emphasizing greater errors, and RMSE is the square root of MSE, which provides inaccuracy in the original data units. The code imports the appropriate libraries, defines a function calculate errors to compute these metrics using sklearn functions, and then runs this function with actual and predicted values, displaying the results to analyze model performance. Lower values imply greater prediction accuracy.

* MAE: On average, the model's predictions differ from real IBM adjusted closing prices by 10.83 points, demonstrating usual prediction error.
* MSE: Averaging squared errors, the MSE of 147.75 emphasizes higher forecast deviations, implying severe inaccuracies.

- RMSE: The model's forecasts are generally 12.16 points off from actual adjusted closing prices, indicating error in the data units.

A screen shot of a computer code

Description automatically generated A collage of graphs and diagrams

Description automatically generated

The code uses visualization and summary statistics to assess a statistical time series model, most likely ARIMA. Results1.plot\_diagnostics () provides four plots: residuals, distribution, normalcy (Q-Q), and autocorrelation, which are displayed using plt.show(). These help to determine model fit. print(results.summary()) returns model parameters and goodness-of-fit information, providing a numerical performance summary. The combination enables extensive model validation. results and results1 may represent separate model instances.

The diagnostic graphs show that, while the model is a decent starting point, it may not capture all the complexity of IBM's stock price. The non-normality of residuals and probable autocorrelation in some data points, the histogram and Q-Q plot suggest some deviation from normality in the residuals. The correlogram indicates potential autocorrelation at certain lags. indicate that external factors such as economic data, industry trends, and unexpected occurrences may be influencing stock prices in ways that the current model cannot account for due to the nature of the limitations of an ARIMA model that does not assess seasonality or those external factors that influence it. To increase forecasting accuracy, it is critical to evaluate and include these external aspects into the model, or to employ more modern techniques such as machine learning.

## *4.2 Evaluation of best Sarima model performance*

The code to evaluate the Sarima model performance was almost the same, I just changed “predictions” that it is the forecast variable of the Arima model to “predictions1”

*A number on a black background

Description automatically generated*

* The MAE (Mean Absolute Error) of 9.60 indicates that the model's forecasts differ by $9.60 on average from the actual IBM adjusted closing price. It's a simple measure of prediction error in the same unit as the target variables.
* MSE (Mean Squared Error): A value of 126.12 squares the mistakes prior to averaging, highlighting greater errors. This 126.12 figure is difficult to grasp on its own due to the squared units, but it is valuable for model comparison.
* RMSE (Root Mean Squared Error): RMSE of 11.23, which is the square root of MSE, is returned to the original unit ($). It is like a typical prediction error, but gives more weight to huge variances, which is useful for practical interpretation.

A collage of graphs

Description automatically generated

The same code was used for these diagnostic plots, it just changed to assess the SARIMA model.

The histogram looks mor accurate this time and Q-Q plot imply that the residuals are nearly normal. The correlogram demonstrates that there may be autocorrelation at unpredictable lag times due to the industry dynamics. These findings suggest that the model may not fully represent the data's underlying trends.

# *5. Conclusion (Interpretation)*

The project shows that the SARIMA model outperforms the ARIMA model in predicting IBM's adjusted close stock price. While both models offer useful insights, the SARIMA model better reflects the seasonality and volatility inherent in stock price movements, as indicated by lower error metrics (MAE: 9.60, RMSE: 11.23 vs. ARIMA's 10.83 and 12.16, respectively). The seasonal component of SARIMA allows it to account for periodic patterns and exogenous influences that influence stock values, resulting in more accurate predictions. Diagnostic charts validate the SARIMA model by demonstrating better residual normality and lower autocorrelation than the ARIMA model. Overall, the SARIMA model delivers a stronger fit and consistent performance for predicting IBM stock prices, especially considering the stock's seasonal and volatile character but still lacks on predicting precisely the stock price of the company due to the limitations of the models.

IBM's recent stock volatility can be linked to a variety of things. Economic uncertainties, such as future recessions, rising interest rates, and geopolitical tensions, can have a broad impact on investor sentiment and stock price movements. Investors may become more cautious, resulting in higher volatility. Furthermore, IBM's concentration on AI and cloud computing, while promising, faces a competitive environment. Any news or events involving these technologies, such as increasing competition or regulatory changes, can have a big impact on the stock price. Investors may react positively to good news, such as successful product launches or strategic alliances, but negatively to bad news, such as losing market share or increased competition. Lastly, overall market sentiment and investor behavior can affect individual stocks like IBM. Changes in risk appetite and investor preferences can lead to short-term volatility.

So probably a SARIMAX model that includes more x variables that can influence in the stock price can lead to a better outcome, or other models as Monte Carlo simulation or ml techniques.