Meetings			
Date	**Objective**	**Outcome**	**Delegated Tasks**
06-05-2023	Plan a timeline and set a schedule. Decide company	We will be applying the principles of time series analysis to understand the revenue of **TSMC**	Read an example research paper (TSA of Jute demand) Collect the revenue data of TSMC
10-05-2023	Discuss the research paper	Built an understanding of how research is done	Make a list of TS models which could be used.
15-05-2023	Discuss potential TS models.	Classified Models in two classes- Classical and modern models. We will be comparing 1 modern model against classical models.	List down the Advantages, disadvantages, and use cases for each model. Find customers of TSMC
18-05-2023	Discuss the collected model's	Classical models finalised - **Seasonal ARIMA** and **Exponential Smoothing** models.	Collect the data regarding companies that buy from TSMC.
23-05-2023	Finalise the modern model	**LSTM** model finalized Data of customers not freely available	Find appropriate determinants for comparison between models. Read a case study for LSTM
26-05-2023	Discuss the LSTM case study. Discuss possible determinants	**RMSE**, **MAE**, **MAPE**, and **MASE** will be used to compare models. **AICc** value will be used to select parameters of Arima and ES	Test for stationarity Make **ACF** and **PACF** plots Implement the Seasonal ARIMA models and the ES models.
29-05-2023	Discuss the Classical models fit Share some plots	Time series is not stationary. It also has yearly seasonality. Log[revenue] will be analyzed	Calculate AICc values and finalize model parameters in SARIMA models and ES models.
02-06-2023	Discuss the final two models	Finalised - **ARIMA[4,1,4][1,0,0][12]** and **Holt Winters' Additive** Model	Implement LSTM Model
05-06-2023	Discuss LSTM	The LSTM model does not fit well.	Try to make a better LSTM model. Find another modern model
12-06-2023	Discuss changing the model	Discarded LSTM model Presented the **Prophet** model	Read about the prophet model Find case studies implementing this model in the field of finance.
15-06-2023	Discuss the case studies	More study required before implementing this model	List down the problems in LSTM How is the Prophet their solution
19-06-2023	Discuss more case Studies and finalise the model	Finalised the Prophet model. Could potentially use the technology change data	Implement the prophet model Read the history of tsmc company
22-06-2023	Discuss CV techniques	Cross-Validation using the Rolling Origin method will be used	Implement Cross-Validation and compare the models
26-06-2023	Share the results of the CV	The prophet model is the best Considered publishing our results	Make a report draft.

Abstract

This project report aims to analyse the monthly revenue of Taiwanese Semiconductor Manufacturing Company (TSMC) and compare the forecasting performance of three different time series models: SARIMA, Holt Winters, and FB Prophet. The evaluation metrics used for model comparison include Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Scaled Error (MASE). By selecting the best-performing model based on these metrics, we aim to provide an accurate revenue forecast for the next year.

The analysis begins with a comprehensive exploration of the monthly revenue data of TSMC, obtained from reliable sources. The dataset is examined for any trends, seasonality, or other patterns that may influence the revenue. Following this, the SARIMA, Holt Winters, and FB Prophet models are trained using the available historical revenue data.

To evaluate the performance of each model, we employ four commonly used determinants: MAE, RMSE, MAPE, and MASE. These metrics allow us to assess the accuracy and robustness of each model in capturing revenue patterns. By comparing the results, we can identify the model that provides the most accurate forecasts for TSMC's revenue.

Once the best model is identified, we use it to forecast the revenue for the next year. This forecast can assist stakeholders, such as investors, analysts, and decision-makers, in gaining insights into TSMC's future financial performance. Additionally, the report highlights the strengths and limitations of each model, providing valuable insights for future research and application in similar domains.

Overall, this project contributes to the field of time series analysis by evaluating the performance of three popular models—SARIMA, Holt Winters, and FB Prophet—for revenue forecasting. The findings of this study can aid in making informed decisions related to TSMC's financial strategies and serve as a basis for further research in the field of time series forecasting.

Literature Review

[1] The research article investigates the application of various quantitative forecasting models for predicting the demand for jute yarn. The study utilises models such as simple moving average, single exponential, double exponential (Holt's), Winters, and decomposition methods. The objective is to determine the most effective model for forecasting jute yarn demand. To evaluate the performance of the models, three error determinants are calculated: mean absolute deviation (MAD), mean absolute percentage error (MAPE), and mean square deviation (MSD). These measures provide insights into the accuracy and reliability of the forecasts generated by each model. The study focuses on a weekly basis analysis, considering a four-year period from the 1st week of 2010 to the 52nd week of 2013. The authors gather 208 weeks

of demand data for jute yarn during this time span. By comparing the forecasting results and the error determinants for each model, the researchers aim to identify the best-performing model in terms of accuracy and reliability for predicting jute yarn demand. Overall, this study provides a comprehensive analysis of various quantitative forecasting models applied to jute yarn demand, using multiple error determinants. The dataset spans four years, and the evaluation is performed on a weekly basis, allowing for a detailed examination of the models' forecasting performance.

[2] The study utilises historical data, including quantities sold and average monthly prices, spanning a 24-month time series. Forecasts using the SARIMA model are made on a monthly basis, with a forecasting horizon of 12 months. The primary objective is to assess the software tool's effectiveness in incorporating data uncertainty and its impact on the accuracy of sales forecasts. By employing probabilistic modelling techniques, the tool generates more realistic and robust forecasts that account for the inherent uncertainty in the sales data. The findings highlight the advantages of considering data uncertainty in sales forecasting. The software tool improves forecast accuracy by providing decision-makers with a range of potential outcomes. This enhances decision-making by considering uncertainties involved in the sales data. In summary, the research article presents a case study on sales forecasting using a software tool that addresses data uncertainty. By employing probabilistic modelling, the tool generates accurate forecasts and aids decision-making processes by considering the range of potential outcomes.

[3] The research paper titled "Comparative Study on Retail Sales Forecasting between Single and Combination Methods" conducts a comparative analysis of single and combination forecasting methods in predicting retail sales. The researchers begin by collecting historical sales data from various retail stores. They then apply single forecasting methods, such as time series models, regression analysis, and machine learning algorithms, to generate individual forecasts. Models like Exponential Smoothing, ARIMA, SARIMA, were employed for the study.
To select the most suitable models, the researchers employ statistical tests like the KPSS unit root test and the Canova-Hansen Test. These tests help determine the stationarity of the sales data and identify any trends or patterns present. Additionally, the research paper utilises the Akaike Information Criterion corrected (AICc) for ARIMA model selection in all the possible models. The AICc helps identify the optimal ARIMA model by balancing the goodness of fit and the complexity of the model. Furthermore, the researchers examine the residuals of the forecasting models. Residual analysis allows them to assess the model's ability to capture the underlying patterns in the sales data and identify any remaining systematic errors. By incorporating these statistical tests, model selection using AICc, and residual analysis, the research paper ensures a comprehensive evaluation of the forecasting methods. These techniques provide insights into the accuracy, reliability, and suitability of the models for predicting retail sales. These evaluation methods contribute to a thorough understanding of the strengths and limitations of both single and combination forecasting methods in the context of retail sales forecasting. To evaluate the performance of the forecasting models, the researchers utilise

appropriate error metrics. Commonly used metrics include mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), or mean absolute percentage error (MAPE). These metrics provide a quantitative measure of the accuracy of the forecasts, enabling a meaningful comparison between the single and combination methods.

[4] The research paper titled "Application of Facebook's Prophet Algorithm for Successful Sales Forecasting Based on Real-World Data" focuses on the utilisation of the Prophet algorithm, developed by Facebook, for accurate sales forecasting using real-world data. The paper begins by introducing the significance of sales forecasting in various industries, emphasising the need for accurate predictions to optimise inventory management, production planning, and business decision-making processes. The researchers employ the Prophet algorithm, which is specifically designed to handle time series forecasting with seasonality and trends. The algorithm incorporates a flexible and intuitive modelling approach, enabling users to capture different components of the sales data, including seasonality, trend, and holiday effects. In the study, the researchers collect real-world sales data from a particular industry or multiple industries. They preprocess the data by cleaning and formatting it appropriately for input into the Prophet algorithm. The Prophet algorithm automatically detects and models seasonality, trends, and holidays present in the sales data. It incorporates Bayesian methods to generate probabilistic forecasts, providing not only point estimates but also uncertainty intervals for the predictions. To evaluate the performance of the Prophet algorithm, the researchers compare the generated forecasts with the actual sales data. They utilise error metrics such as mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), or mean absolute percentage error (MAPE) to measure the accuracy of the predictions. The research paper highlights the advantages of using the Prophet algorithm for sales forecasting based on real-world data. It emphasises the algorithm's ability to handle complex time series patterns and provide reliable forecasts. The results of applying the Prophet algorithm to real-world sales data provide insights into its performance and its potential benefits for businesses in terms of improved decision-making, inventory management, and resource allocation. Overall, the research paper demonstrates the successful application of Facebook's Prophet algorithm in the field of sales forecasting. It showcases the algorithm's capabilities, accuracy, and practicality when dealing with real-world data, encouraging further adoption and exploration of this approach in various industries.

Methodology

Time Series Analysis

A time series is a collection of observations or data points measured and recorded sequentially over regular time intervals. Formally, a time series can be defined as an ordered sequence of data points represented as Y_t, where t denotes the time index

and Y_t represents the observed value or measurement at time "t". The time series data can be discrete or continuous, depending on the nature of the variable being measured.
In a time series, the observations are typically recorded at equidistant time intervals, such as hourly, daily, monthly, or yearly. The data points are often influenced by various factors, including seasonality, trends, cyclic patterns, and random fluctuations. These characteristics make time series analysis a valuable tool for understanding and predicting future behaviour based on past patterns and trends. Time series analysis involves studying the properties, patterns, and dynamics of the data, as well as developing mathematical models and statistical techniques to capture and explain its behaviour.

Components of Time Series

In time series analysis, a time series is often decomposed into several components that collectively capture the different patterns and variations present in the data. The main components of a time series are:

1. Trend: The trend component represents the long-term pattern or direction of the data. It indicates whether the variable is increasing, decreasing, or following a relatively stable pattern over time. The trend can be linear, nonlinear, or even non-existent.
2. Seasonality: Seasonality refers to the repetitive and predictable patterns that occur within a time series at regular intervals, typically within a year or a shorter period. These patterns may be influenced by seasonal factors, such as weather, holidays, or economic cycles. Seasonality can be additive or multiplicative, depending on whether the magnitude of the seasonal pattern remains constant or varies with the level of the series.
3. Cyclical: The cyclical component captures the irregular patterns or fluctuations in a time series that are not of fixed period or related to specific calendar effects. These cycles are typically longer than the seasonal cycles and can be influenced by economic, business, or geopolitical factors. Cyclical patterns are often associated with boom and bust cycles, but their duration and magnitude can vary.
4. Irregularity or Residual: The irregular component, also referred to as the residual or error component, represents the random or unpredictable fluctuations that cannot be explained by the other components. It encompasses any noise, randomness, or unexplained variation in the time series data.

Level refers to the underlying or average behaviour of the data over time, which is often incorporated into the trend component and is not typically considered as a distinct component of a time series.

Some Basic Terminology

1. Data Point: A data point refers to a single observation or measurement recorded at a specific time point in a time series.

2. Time Interval: A time interval is the duration between successive time points or observations in a time series. It represents the granularity or frequency at which the data is collected or recorded.
3. Stationarity: Stationarity is a property of a time series where the statistical properties, such as mean, variance, and covariance, remain constant over time. A stationary time series exhibits stable behaviour and is often easier to model and analyse compared to non-stationary series.
   Here are some key reasons why stationarity is desirable:
   - Simplified Modelling: Stationary time series exhibit consistent statistical properties over time, such as constant mean, constant variance, and autocovariance that depends only on the time lag. This simplifies the modelling process by allowing us to assume that the statistical properties of the data remain constant throughout the series. Modelling a stationary series is often easier and more straightforward compared to non-stationary series.
   - Predictability: Stationary time series exhibit stable and predictable behaviour. The patterns observed in the past are likely to persist in the future, making it easier to forecast future values based on historical data. Stationary series allow for reliable extrapolation and forecasting using mathematical models, such as autoregressive integrated moving average (ARIMA)
   - Interpretability: Stationarity enables the identification and interpretation of the underlying patterns and dynamics within a time series. By removing trends and seasonality through differencing or other transformations, the focus can be shifted to the cyclical or residual components, allowing for a better understanding of the true underlying behaviour of the data.

   Test For Checking Stationarity of the data ->
   1. Augmented Dickey-Fuller (ADF) Test: The ADF test is a widely used test for stationarity. It assesses whether a unit root is present in a time series, indicating non-stationarity. The null hypothesis of the test is that the series has a unit root (i.e., it is non-stationary). If the p-value associated with the ADF test is below a chosen significance level (e.g., 0.05), the null hypothesis is rejected, indicating stationarity.
   2. Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test: The KPSS test is another commonly employed test for stationarity. It examines whether a time series is trend-stationary or difference-stationary. The null hypothesis of the KPSS test is that the series is trend-stationary. If the p-value is above a specified significance level, the null hypothesis is not rejected, indicating stationarity.

   Methods to make Time Series Data Stationary ->
   To make a time series stationary, you can employ various methods and transformations. The specific approach you choose depends on the characteristics of the data and the nature of the non-stationarity. Here are some common methods for making a time series stationary:

1. Differencing: Differencing involves taking the difference between consecutive observations in the time series. This method is effective for removing a trend or linear component. If the series exhibits a constant growth rate, you can apply first-order differencing (subtracting each observation from its previous observation). For higher-order differencing, you can repeat the differencing process multiple times until the series becomes stationary.
2. Logarithmic Transformation: If the series displays exponential growth or varying variance over time, applying a logarithmic transformation can stabilise the variance and make the series more stationary. Taking the natural logarithm of the series values helps in reducing the magnitude of large values and dampening the exponential growth.
3. Seasonal Differencing: Seasonal differencing involves taking the difference between observations at a fixed seasonal interval. This method is useful for removing seasonal patterns or periodic fluctuations. By subtracting the observation from the corresponding observation in the previous season, you can eliminate the seasonal component and potentially achieve stationarity.
4. Transformation Techniques: Certain mathematical transformations can help stabilise the variance of a time series. For example, the Box-Cox transformation can be applied to handle cases where the variance of the series changes with the level. This transformation applies a power function to the series, allowing for different types of transformations depending on the chosen parameter.

It's important to note that there is no one-size-fits-all approach for achieving stationarity, and the choice of method depends on the specific characteristics and behaviour of the time series. Additionally, after applying the transformation or differencing, it is crucial to assess the stationarity of the resulting series using statistical tests, such as the Augmented Dickey-Fuller (ADF) test.

4. Lag: Lag refers to the time distance between two observations in a time series.
5. Autocorrelation: Autocorrelation measures the linear relationship between the values of a variable in a time series at different time lags. It helps identify the presence of any serial dependence or correlation between observations.

   Autocorrelation and partial autocorrelation are statistical measures used to analyse the relationship between observations in a time series data.
   - Autocorrelation (ACF): Autocorrelation measures the linear dependence between an observation in a time series and its lagged values. It quantifies how strongly the values at different lags are related to each other. The autocorrelation function (ACF) plots the autocorrelation coefficients at various lags. A high autocorrelation at a specific lag suggests that past values at that lag can help predict future values.
   - Partial Autocorrelation (PACF): Partial autocorrelation measures the direct relationship between an observation and its lagged values while accounting for the influence of intermediate lags. It provides a more

refined analysis by removing the effects of shorter lags. The partial autocorrelation function (PACF) plots the partial autocorrelation coefficients at various lags. Significant partial autocorrelation at a specific lag indicates a direct relationship between the current observation and that lag without the influence of intervening lags.

6. Forecasting: Forecasting involves predicting or estimating future values or patterns of a variable in a time series based on historical data. It aims to capture and project the underlying behaviour and trends in the data.

   Two types of Forecasting
   1. Qualitative Forecasting - If there is no data available, or if the data available are not relevant to the forecasts.
   2. Quantitative Forecasting - Can be applied when two conditions are applied:
      - Numerical Information about the past data is available.
      - It is reasonable to assume that some aspects of the past patterns will continue in the future.

   Quantitative Forecasting uses either time series data (collected at regular intervals over time) or cross-sectional data (collected at a single point of time).
   Models for Time-Series Forecasting :
   1. Explanatory Model :-
      This incorporates information about other variables(predictor variables) , rather than only historical values of the variable to be forecast.
      $ED = f(predictor\ variables, \ldots\ldots, error)$ where ED is the variable we want to forecast.
   2. Time Series Model :-
      $ED_{t+1} = f(ED_t, ED_{t-1}, ED_{t-2}, ED_{t-3}, \ldots\ldots, error)$, where t is the present hour, t+1 is the next hour , t-1 is the previous hour, …. And so on. Here, prediction of the future is based on past values of a variable, but not on external variables which may affect the system.
   3. Mixed Model :-
      $ED_{t+1} = f(ED_t, predictor\ variables, \ldots\ldots, error)$.

   However, there are several reasons we might select a time series model rather than an explanatory or mixed model.
   1. The system may not be understood, and even if it was understood it may be extremely difficult to measure the relationships that are assumed to govern its behaviour.
   2. It is necessary to know or forecast the future values of the various predictors in order to be able to forecast the variable of interest, and this may be too difficult.
   3. The main concern may be only to predict what will happen, not to know why it happens.

   Forecast Horizon -> Refers to the time duration into the future for which predictions are generated in a time series analysis. It indicates the length of the forecasted

values beyond the present or the last observed data point and helps determine the timeframe for decision-making and planning based on the forecasted information.

Residuals

Residuals are an essential component in time series analysis and modelling. They represent the differences between the observed values of a time series and the corresponding predicted values from a forecasting model. Analysing the residuals and understanding their properties is crucial for several reasons:

1. Model Assessment: Residuals allow us to assess the goodness of fit of a time series model. By examining the patterns and characteristics of the residuals, we can evaluate how well the model captures the underlying behaviour of the data. Ideally, the residuals should exhibit randomness and independence, indicating that the model adequately explains the observed variability in the data.
2. Diagnostic Checking: Residual analysis helps in diagnosing potential issues with the model. Patterns or structure in the residuals may indicate the presence of omitted variables, misspecified model assumptions, or inadequacies in the chosen modelling technique. If the residuals display non-random patterns, autocorrelation, or heteroscedasticity (varying variance), it suggests that the model may need improvement.
3. Forecast Accuracy: Residuals play a role in evaluating the accuracy of forecasts. By comparing the predicted values of the model with the actual observed values, we can calculate the residuals and assess how well the model predicts future outcomes. Smaller residuals indicate more accurate predictions, while larger residuals suggest potential forecasting errors.
4. Statistical Inference: Residuals are used in various statistical tests and inference procedures. For instance, residual analysis is important in testing the assumptions of a time series model, such as stationarity, normality, or independence. Residuals are also employed in hypothesis testing and constructing confidence intervals for parameters estimated in the model.

<u>Properties of Residuals:</u>

1. Mean Zero: The residuals should have an average value close to zero. A non-zero mean may indicate a systematic bias in the model.
2. Independence: Residuals should be independent of each other. Autocorrelation in residuals suggests that the model fails to capture the temporal dependencies in the data.
3. Homoscedasticity: Residuals should exhibit constant variance over time. Heteroscedasticity, where the variance of residuals varies with the level of the series, may indicate model inadequacy.
4. Normally Distributed: Residuals should follow a normal distribution. Departure from normality may affect the validity of statistical inference and prediction intervals. Analysing and interpreting the properties of residuals is crucial for model validation, improving forecasts, and ensuring the reliability of time series analysis.

We normally check whether the residuals of our model satisfy the above properties or not. Test for checking Residuals ->

The Ljung-Box test helps in assessing the adequacy of a time series model by examining the presence of autocorrelation in the residuals. If the test results indicate significant autocorrelation, it suggests that the model may need further refinement or consideration of additional explanatory variables.

Errors and Information Criteria ->

- Errors: Errors in time series forecasting refer to the differences between the predicted values and the actual observed values of the time series. These errors quantify the level of inaccuracy or deviation of the forecasts from the true values. Commonly used error measures include Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and others. Errors provide a direct evaluation of the forecasting accuracy and help assess the performance of different forecasting models. Lower error values indicate better accuracy and closer alignment with the actual data.
- Information Criteria: Information criteria, such as the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC), are statistical measures used to compare and select between different competing models. These criteria aim to balance the goodness of fit of the model with the complexity or number of parameters used in the model. Information criteria penalise models with a higher number of parameters, encouraging parsimony and avoiding overfitting. The lower the information criterion value, the better the trade-off between model fit and complexity. Information criteria assist in model selection by providing a quantitative measure to compare the relative performance of different models and choose the one that best balances fit and complexity.

Errors help evaluate the performance of a specific model, while information criteria aid in model selection and compare the relative performance of different models.

Cross-Validation

Cross-validation is a technique used to evaluate the performance and generalise the effectiveness of a predictive model. It involves partitioning the available dataset into subsets, typically a training set and a validation set, and iteratively training and evaluating the model on different combinations of these subsets.
The basic steps of cross-validation are as follows:

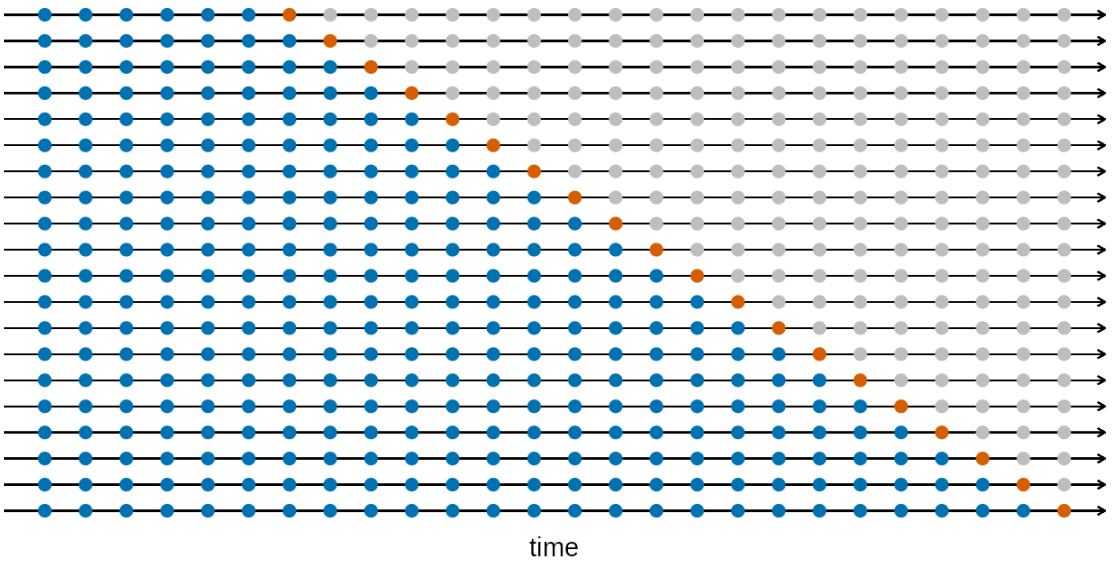
- Splitting the dataset: The original dataset is divided into k subsets or folds of approximately equal size.
- Model training and evaluation: The model is trained on k-1 folds of the data, and the remaining fold is used for validation. This process is repeated k times, with each fold serving as the validation set once.
- Performance metrics: The performance of the model is assessed on each validation set, and the evaluation results (such as accuracy, mean squared error, or other relevant metrics) are recorded.
- Aggregating results: The performance metrics from each iteration are averaged or combined to obtain an overall assessment of the model's performance.

The main purpose of cross-validation is to estimate how well a model will perform on unseen data. It helps to assess the model's ability to generalise and identifies potential issues such as overfitting or underfitting. By using different subsets of the data for training and validation, cross-validation provides a more robust evaluation of the model's performance compared to a single train-test split.
Commonly used cross-validation methods include k-fold cross-validation, stratified k-fold cross-validation, leave-one-out cross-validation, and holdout validation. The choice of the cross-validation method depends on the size of the dataset, the nature of the problem, and other considerations.

In this study, We are using a rolling origin Method for cross validation ->
Rolling origin cross-validation, also known as rolling window or sliding window cross-validation, is a variation of the traditional cross-validation technique. It is commonly used in time series analysis and forecasting to assess the performance of predictive models on sequential data.
In rolling origin cross-validation, a fixed-size training window is defined, and the model is trained on the observations within this window. The model is then tested on the next observation that immediately follows the training window, which serves as the validation or test set. This process is repeated by sliding the window one step forward and retraining the model, generating new predictions and evaluating their performance at each step.
The main steps involved in rolling origin cross-validation are as follows:

- Set the initial training window: Determine the size of the training window, which typically consists of a fixed number of sequential observations.
- Train and evaluate the model: Train the model on the data within the training window. Use the model to make predictions for the next observation in the sequence, which serves as the validation set. Evaluate the performance of the model based on the prediction accuracy or relevant metrics.
- Slide the window: Move the training window one step forward by removing the earliest observation and including the next observation in the sequence. Retrain the model using the updated training window, make predictions for the next observation, and evaluate the model's performance again.
- Repeat the process: Repeat steps 2 and 3 until the training window reaches the end of the available data, generating predictions and performance evaluations at each step.

Rolling origin cross-validation allows for the assessment of a model's performance on sequential data, which is crucial in time series analysis and forecasting. It provides a more realistic evaluation of the model's ability to generalise to future observations by simulating the real-world forecasting scenario where the model is updated and retrained as new data becomes available. This approach helps to capture changes in the underlying patterns or dynamics of the time series over time and provides a more accurate representation of the model's performance compared to traditional cross-validation methods that assume independent and identically distributed samples.



1. Univariate Time Series Model -> A univariate time series model focuses on the analysis and forecasting of a single variable over time. It assumes that the observed data points are dependent on their previous values. The model takes into account the temporal patterns, trends, and seasonality within the data to make predictions about future values of the variable. Commonly used univariate time series models include autoregressive integrated moving average (ARIMA), exponential smoothing models (such as Holt-Winters), and state space models.
2. Multivariate Time Series Model -> A multivariate time series model deals with the analysis and forecasting of multiple variables simultaneously, where the variables are interrelated and may influence each other. These models capture the dependencies and relationships among the variables to generate forecasts. Multivariate time series models are useful when there is a need to consider the joint behaviour of multiple variables and exploit the information present in their interdependencies. Machine learning techniques like recurrent neural networks (RNNs) and long short-term memory (LSTM) networks are commonly applied for multivariate time series modelling.

Models

Although there are many models which are used in time series analysis, this project mainly focuses on the most popular ones (SARIMA and Holt-Winters) and the most recent ones (FB Prophet)

Naive Model

The naive model is one of the simplest and most basic approaches for time series forecasting. It assumes that the future values of a time series will be the same as the most recent observed value. In other words, the naive model assumes that there is no trend, seasonality, or any other pattern in the data, and the future values will simply be a repetition of the last observed value.

The naive model is straightforward and quick to implement, but it is highly simplistic and assumes that the future behavior of the time series will be the same as the most recent observation. It doesn't account for any underlying patterns, trends, or seasonality in the data, so its accuracy is often limited. However, it can serve as a baseline model for comparison against more sophisticated forecasting methods.

- Equation : $Y_t = Y_{t-1}$

Arima models

1. Autoregressive (AR) Model:->
   The AR model assumes that the current value of a variable is a linear combination of its **previous values and a random error term**.
   We refer to the following equation as the **AR(p) model,** an autoregressive Model of order p, determines the number of lagged terms considered in the model.
   $y_t = c + \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + \ \dots \ + \Phi_p y_{t-p} + \varepsilon_t,\ \varepsilon_t\ is\ white\ noise.$
   Parameter Constraints for stationary data:
   For an AR(1) model: $-1 < \Phi_1 < 1$
   For an AR(2) model: $-1 < \Phi_2 < 1$,
   $\Phi_1 + \Phi_2 < 1$, $\Phi_2 - \Phi_1 < 1$. It is possible to write any stationary AR(p) model as an MA(∞) model.

2. Moving Average (MA) Model:->
   A moving average model forecasts the future values of a time series based on the weighted average of past error terms. It assumes that the current value of the series depends on the **past error terms and a random error term**. The order of the moving average model, denoted as **MA(q)**, specifies the number of lagged error terms considered in the model.
   $y_t = c + \varepsilon_t + \Theta_1 \varepsilon_{t-1} + \Theta_2 \varepsilon_{t-2} + \ \dots \Theta_q \varepsilon_{t-q},\ \varepsilon_t\ is\ white\ noise.$
   Parameter Constraints for stationary data:
   For an MA(1) model: $-1 < \Theta_1 < 1$
   For an MA(2) model: $-1 < \Theta_1 < 1$,
   $\Theta_1 + \Theta_2 < 1$, $\Theta_2 - \Theta_1 < 1$.

   It is important to note that the MA model assumes stationarity of the time series, meaning that the mean and variance of the series remain constant over time. If the time series exhibits non-stationarity, pre-processing techniques such as differencing can be applied to make it stationary before fitting the MA model.

3. Autoregressive Moving Average (ARMA) ->
   The ARMA (Autoregressive Moving Average) model is a popular time series model that combines the autoregressive (AR) and moving average (MA) components to capture **both the linear dependence on past values and the**

influence of past error terms. It is a more general form of the AR and MA models.

The full **ARMA(p,q)** can be written as:
p = order of the autoregressive component, specifying the number of lagged terms of the dependent variable included in the model.
q = order of the moving average component, specifying the number of lagged error terms included in the model.

$$y_t = c + \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + \ldots + \Phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots \theta_q \varepsilon_{t-q}.$$

4. Autoregressive Integrated Moving Average (ARIMA) ->
The ARIMA model incorporates the autoregressive component (AR) to capture the linear dependence on past values, the differencing component (I) to achieve stationarity, and the moving average component (MA) to capture the influence of past error terms. The model is widely used for time series analysis and forecasting, as it can handle data with trends, seasonality, and other patterns.

The full **ARIMA (p,d,q)** model can be written as:
p = order of the autoregressive component, specifying the number of lagged terms of the dependent variable included in the model.
d = degree of differencing, indicating the number of times the data needs to be differenced to achieve stationarity.
q = order of the moving average component, specifying the number of lagged error terms included in the model.

$$y_t = c + \Phi_1 y'_{t-1} + \Phi_2 y'_{t-2} + \ldots + \Phi_p y'_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q}.$$

y'_t is the differenced series.

5. Seasonal Autoregressive Integrated Moving Average (SARIMA) ->
The SARIMA model is an extension of the ARIMA model that incorporates seasonal patterns in time series data. It is used to capture both the non-seasonal and seasonal components of a time series.

The SARIMA model is denoted as **SARIMA(p, d, q)(P, D, Q)[s]**, where:

p represents the order of the non-seasonal autoregressive component.
d represents the degree of non-seasonal differencing.
q represents the order of the non-seasonal moving average component.
P represents the order of the seasonal autoregressive component.
D represents the degree of seasonal differencing.
Q represents the order of the seasonal moving average component.
s represents the length of the seasonal cycle.

Exponential Smoothing

Exponential smoothing is a time series forecasting method that assigns exponentially decreasing weights to past observations. It is a simple and widely used technique for generating short-term forecasts.

The basic idea behind exponential smoothing is to assign different weights to past observations, with more recent observations receiving higher weights and older observations receiving lower weights. The weights decrease exponentially as the observations become more distant in the past. This approach reflects the assumption that more recent observations are more relevant and carry more information for predicting future values.

Exponential smoothing involves three main components:

Level: The level represents the smoothed value of the time series at a particular time point. It is calculated by taking a weighted average of the current observation and the previous level. The weight assigned to the current observation is typically denoted as alpha (α), and the weight assigned to the previous level is denoted as 1 - α. The level is updated recursively as new observations become available.

Trend: If there is a trend present in the data, exponential smoothing can also incorporate a trend component. The trend represents the change in the level over time. It is calculated by taking a weighted average of the difference between the current level and the previous level (slope). Similar to the level, the trend is updated recursively using weights. The weight assigned to the current slope is denoted as beta (β), and the weight assigned to the previous trend is denoted as 1 - β.

Seasonality: In some cases, exponential smoothing can also handle seasonality, which refers to repeating patterns in the data that occur over fixed intervals, such as daily, weekly, or yearly patterns. Seasonal exponential smoothing incorporates seasonal adjustments to the forecasts by considering the seasonal indices. The seasonal indices represent the average deviation from the overall level at each seasonal period. These indices are applied to adjust the forecasts based on the seasonality observed in the historical data.

There are different variations of exponential smoothing methods, including simple exponential smoothing, double exponential smoothing (which includes a trend component), and triple exponential smoothing (which includes both trend and seasonality components). The appropriate method to use depends on the characteristics of the time series data and the patterns observed.

Exponential smoothing is a relatively simple technique to implement and can provide reasonable forecasts for short-term predictions. However, it may not capture complex

patterns or long-term trends as effectively as other more sophisticated forecasting methods.

Simple Exponential Smoothing

For simple exponential smoothing, the only component included is the level, l_t . (Other methods which are considered later in this chapter may also include a trend b_t and a seasonal component s_t .) Component form representations of exponential smoothing methods comprise a forecast equation and a smoothing equation for each of the components included in the method. The component form of simple exponential smoothing is given by:

- Forecast Equation: $Y_{t+h} = l_t$
- Level Equation: $l_t = ay_t + (1 - a)l_{t-1}$

Here Y_t denotes the forecasted value at time t, and y_t denotes the observed value at time t.

Holts' Method

Holt (1957) extended simple exponential smoothing to allow the forecasting of data with a trend. This method involves a forecast equation and two smoothing equations (one for the level and one for the trend):

- Forecast Equation: $Y_{t+h} = l_t + hb_t$
- Level Equation: $l_t = ay_t + (1 - a)(l_{t-1} + b_{t-1})$
- Trend Equation: $b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}$

where l_t denotes an estimate of the level of the series at a time t, b_t denotes an estimate of the trend (slope) of the series at a time t, a is the smoothing parameter for the level, and β is the smoothing parameter for the trend.

Holt-Winters Model

Holt (1957) and Winters (1960) extended Holt's method to capture seasonality. The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations — one for the level l_t , one for the trend b_t , and one for the seasonal component s_t , with corresponding smoothing parameters α, β, and γ. Let m denote the frequency of the seasonality, i.e., the number of seasons in a year. For example, for quarterly data m = 4, and for monthly data m=12.

There are two variations to this method that differ in the nature of the seasonal component. The additive method is preferred when the seasonal variations are roughly constant through the series, while the multiplicative method is preferred when the seasonal variations are changing proportionally to the level of the series. With the additive method, the seasonal component is expressed in absolute terms in the scale of the observed series, and in the level equation, the series is seasonally adjusted by subtracting the seasonal component. Within each year, the seasonal component will add up to approximately zero. With the multiplicative method, the seasonal component is expressed in relative terms (percentages), and the series is seasonally

adjusted by dividing through by the seasonal component. Within each year, the seasonal component will sum up to approximately m.

- **Additive Method**
  - Forecast Equation: $Y_{t+h} = l_t + hb_t + s_{t+h-m(k+1)}$
  - Level Equation: $l_t = a(y_t - s_{t-m}) + (1 - a)(l_{t-1} - b_{t-1})$
  - Trend Equation: $b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}$
  - Seasonality Equation: $s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-12}$

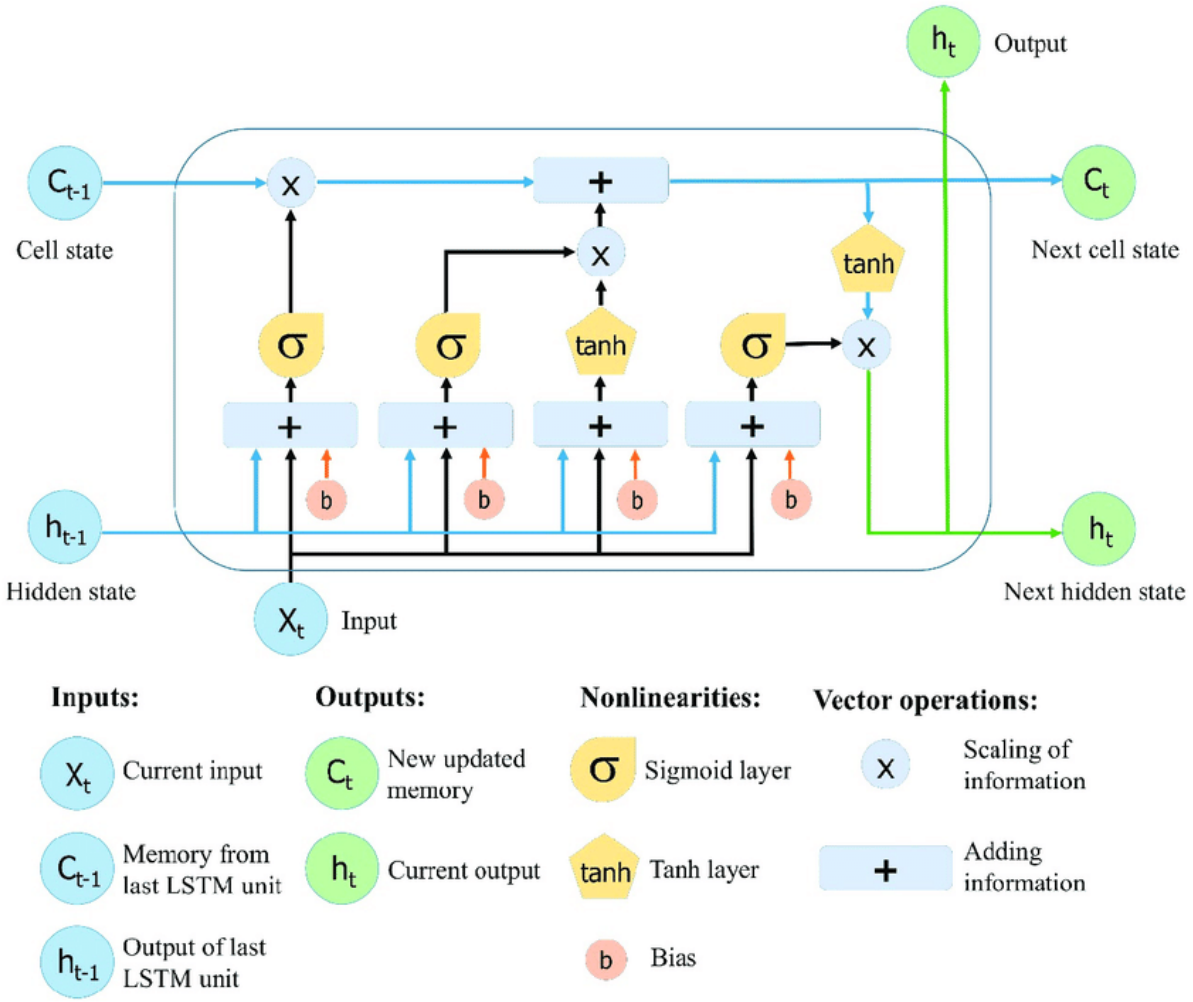
    where k is the integer part of $\frac{h-1}{m}$.
- **Multiplicative Method**
  - Forecast Equation:
    $Y_{t+h} = (l_t + hb_t)s_{t+h-m(k+1)}$
  - Level Equation:
    $l_t = a\frac{y_t}{s_{t-m}} + (1 - a)(l_{t-1} + b_{t-1})$
  - Trend Equation:
    $b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}$
  - Seasonality Equation: $s_t = \gamma\frac{y_t}{l_{t-1}+b_{t-1}} + (1 - \gamma)s_{t-m}$

LSTM

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) architecture that is commonly used for time series analysis, including the analysis of univariate data.

In the context of time series analysis, an LSTM model is designed to capture and analyse patterns and dependencies in sequential data over time. It is particularly effective when dealing with long-term dependencies in the data, as it can remember information over longer periods compared to traditional RNNs.

The key idea behind LSTM is the use of memory cells that allow the model to selectively retain or forget information at each time step. These memory cells are responsible for capturing and preserving relevant information from the past and propagating it forward in the sequence. The LSTM model achieves this through the use of three main components:



Cell State: The cell state is the long-term memory of the LSTM. It runs straight through the entire sequence, and the LSTM can selectively add or remove information from it using gates. The cell state acts as a conveyor belt, allowing information to flow while preserving relevant information.

Forget Gate: The forget gate determines which information from the previous time step should be discarded from the cell state. It takes as input the current input and the previous hidden state and outputs a value between 0 and 1 for each element of the cell state. A value of 0 indicates that the corresponding element should be forgotten, while a value of 1 indicates that it should be retained.

Input Gate and **Output Gate**: The input gate and output gate control the flow of information into and out of the memory cell. The input gate determines which values from the current time step should be updated in the cell state, while the output gate controls which values from the cell state should be used to compute the output of the LSTM.

Using these gates and memory cells, an LSTM model can effectively capture long-term dependencies in time series data. During training, the model learns the optimal parameters for the gates and the memory cells by minimizing a loss function that measures the model's performance in predicting the target values.

In the context of univariate time series analysis, an LSTM model takes the past values of a single variable as input and predicts the future values of that variable. It

can be trained on a historical sequence of data and then used to forecast future values based on the learned patterns and dependencies in the data.
Overall, LSTM models have proven to be powerful tools for time series analysis, enabling accurate predictions and capturing complex temporal patterns in univariate data.

Prophet

Prophet is a forecasting model developed by Facebook's Core Data Science team. It is designed to handle time series data with various components such as trends, seasonality, and holiday effects. Prophet utilises an **additive model** that decomposes the time series into its constituent parts and models them separately.
The Prophet model incorporates the following components:

1. Trend: Prophet captures both short-term and long-term trends in the data. It employs a piecewise linear or logistic growth curve to model non-linear changes in the trend.
   Linear - A linear trend refers to a pattern in the data where the values increase or decrease steadily and consistently over time or across a range of observations.
   Logistic - A logistic trend refers to a pattern in the data where the values initially increase or decrease rapidly, but eventually level off or reach a plateau. It is commonly observed in situations where there are limiting factors or constraints on the growth or decline of a variable. The logistic trend is characterised by an initial exponential-like growth or decline, followed by a gradual flattening of the curve.
2. Seasonality: Prophet accounts for recurring patterns in the data, such as weekly, monthly, or yearly seasonality. It uses Fourier series to model these seasonal components.
3. Holidays: Prophet incorporates the impact of holidays or important events that may affect the time series. It allows users to provide a custom list of holidays or automatically detects holidays based on country-specific datasets.
4. Error: Prophet assumes that the observed time series is a combination of trend, seasonality, and noise. It models the residual errors using a non-parametric approach based on historical data.

Equation of the Prophet Model is:
$y(t) = g(t) + s(t) + h(t) + e(t)$.

g(t): trend
s(t): seasonality
h(t): holiday effects
e(t): error term/noise

Prophet offers several advantages for time series forecasting:

- Flexibility: It can handle time series data with irregular gaps, missing values, and outliers.

- Automatic changepoint detection: Prophet automatically detects changepoints in the trend, allowing for capturing abrupt changes in the time series.
- Intuitive model interpretation: Prophet provides clear and interpretable visualizations, including trend, seasonality, and holiday effects, enabling users to understand the underlying patterns.
- Scalability: Prophet can handle large-scale time series datasets efficiently and can be parallelized for faster computation.

To use the Prophet model, users need to provide a historical time series dataset with a timestamp column and a corresponding value column. Prophet then fits the model to the data, estimates the model parameters, and provides forecasts for future time points.

Prophet has gained popularity due to its ease of use, flexibility, and ability to generate accurate forecasts for various types of time series data. It is particularly useful for business forecasting, demand planning, and other applications where interpretable and robust time series models are required.

TSMC

TSMC (Taiwan Semiconductor Manufacturing Company) is a global leader in semiconductor manufacturing and the world's largest dedicated independent semiconductor foundry. Established in 1987, TSMC has played a pivotal role in shaping the technology landscape and driving innovation in the semiconductor industry.

As a foundry, TSMC specializes in the fabrication of advanced semiconductor products for a wide range of applications, including consumer electronics, automotive, telecommunications, and industrial devices. The company's cutting-edge manufacturing processes enable the production of high-performance chips, delivering enhanced computational power, energy efficiency, and miniaturization.

Most of the leading fabless semiconductor companies such as AMD, Apple, ARM, Broadcom, Marvell, MediaTek, Qualcomm, and Nvidia, are customers of TSMC, as are emerging companies such as Allwinner Technology, HiSilicon, Spectra7, and UNISOC.

Data and Insights

All datasets utilized in this project were sourced exclusively from the official website of TSMC, ensuring data authenticity and reliability. The dataset employed for time series analysis comprises monthly revenue data spanning from January 1999 to April 2023. The revenue values are reported in units of 1 billion NTD (New Taiwanese Dollar).

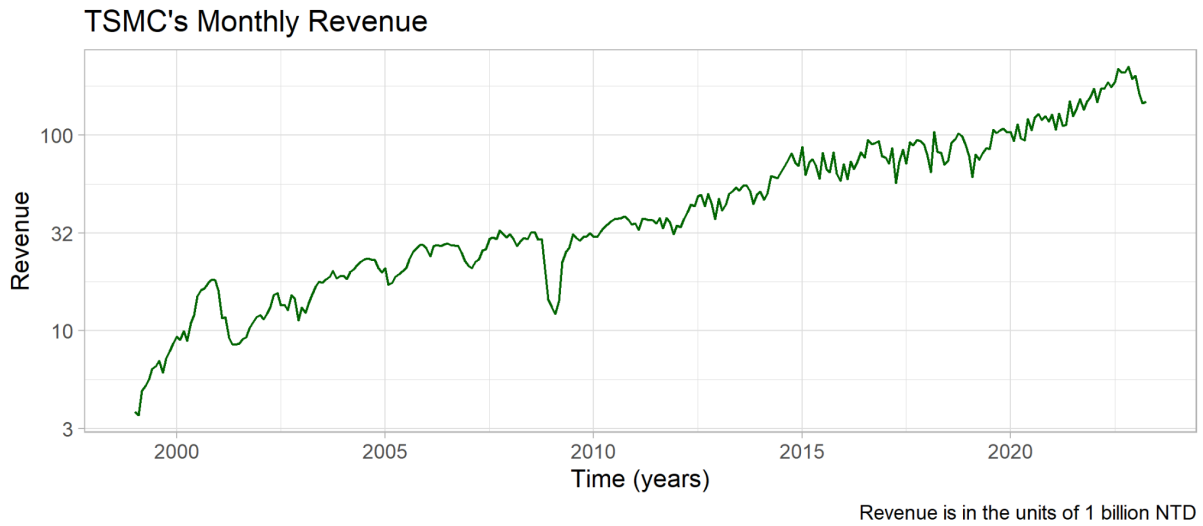
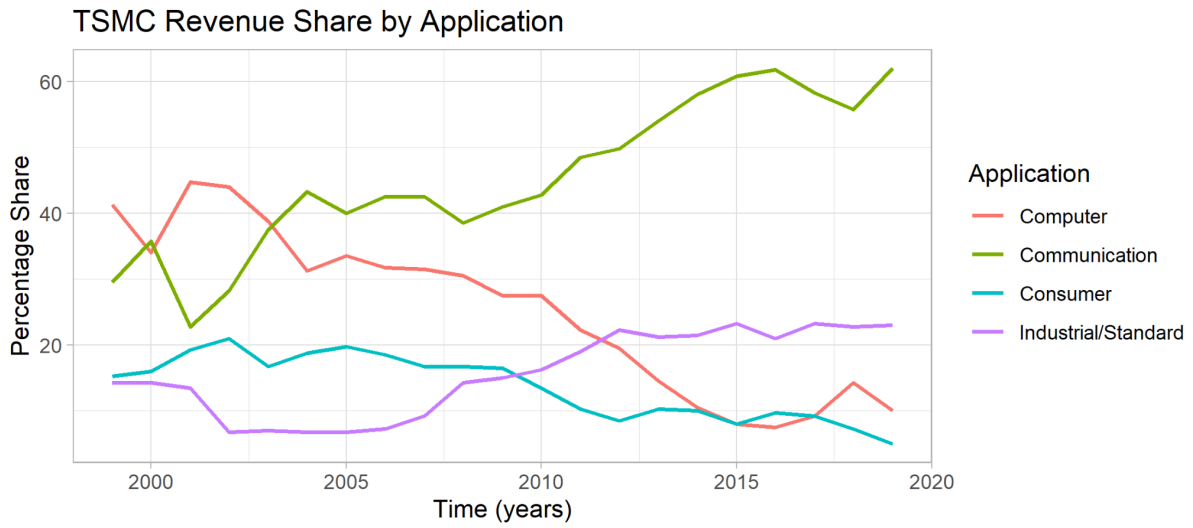
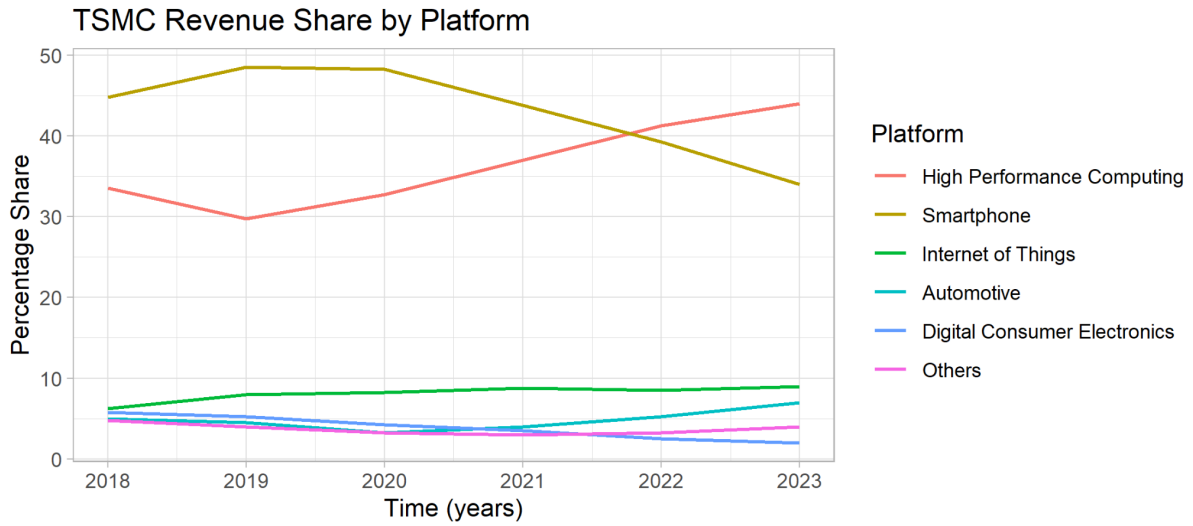
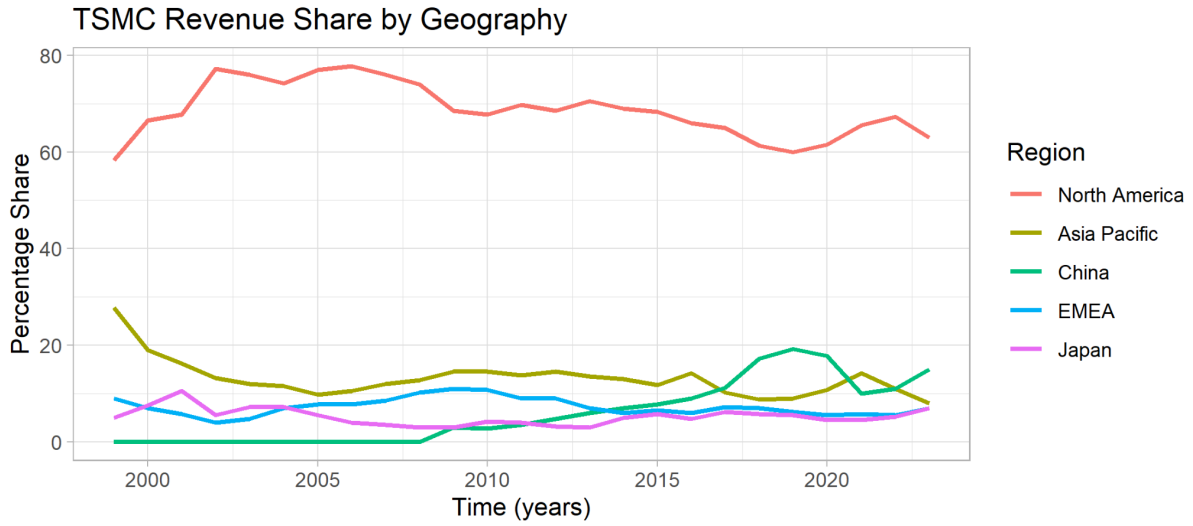


Figure (1) Monthly revenue of tsmc from 1999 to 2023





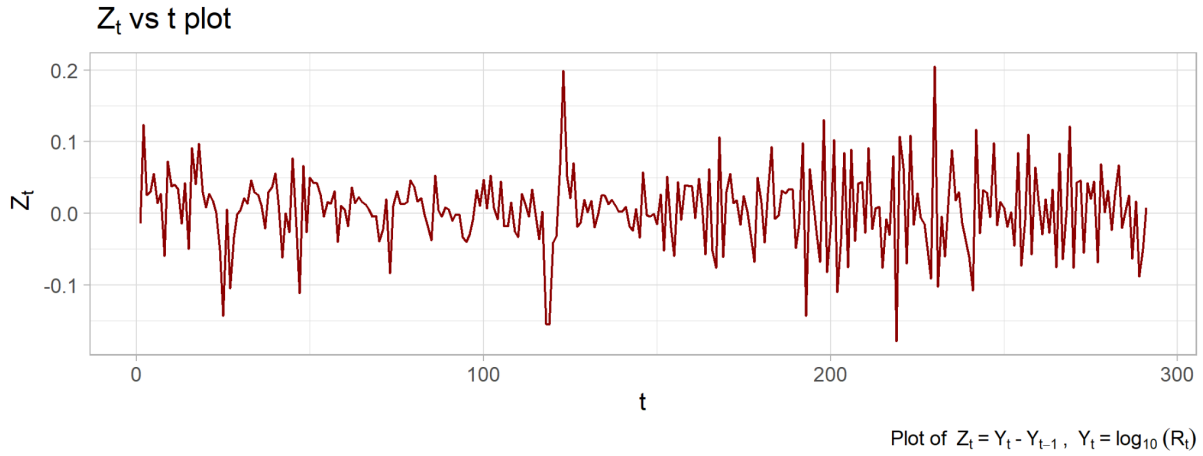


Analysis and Results

Initially, we extract valuable insights from the plotted data. As depicted in Figure (1), the monthly revenue exhibits an approximately linear trend when visualized on a logarithmic scale. This observation leads us to consider employing models that assume a linear trend in the data, such as ARIMA, Holt Winter's, and FB Prophet. For the purpose of analysis, we adopt a logarithmic transformation using a base of 10.

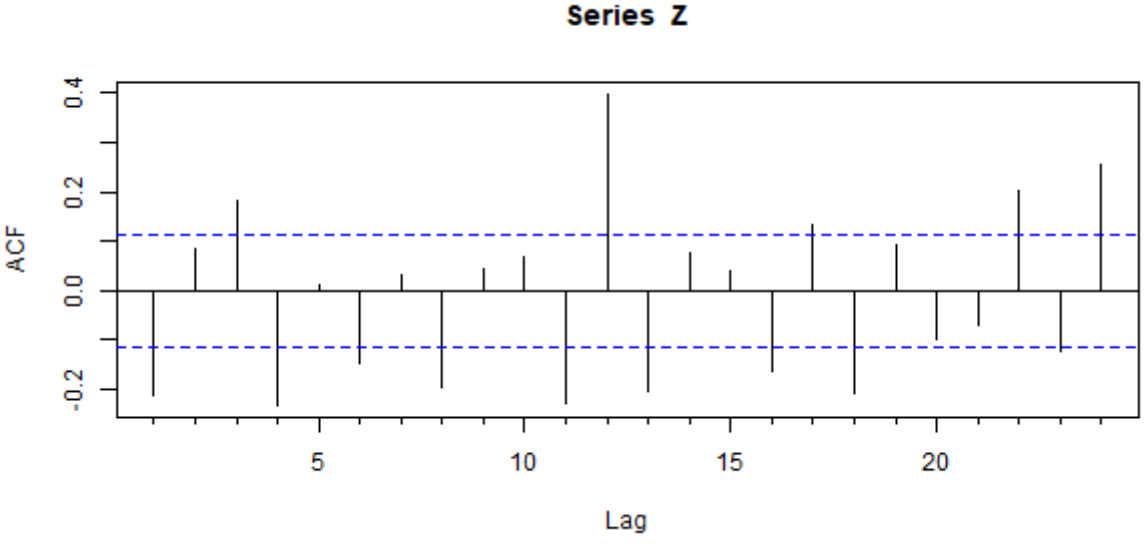
SARIMA

Let us denote our original revenue timeseries by R_t . We have 292 data points from January 1999 to April 2023 hence $t \in \{1, 2, \ldots 292\}$. In order to fit the seasonal arima model we first need to make the time series stationary. As mentioned earlier we take log of R_t to get $Y_t = log_{10}(R_t)$. We apply a first order differencing to Y_t to get $Z_t = Y_t - Y_{t-1}$.
Figure (2) shows the plot of Z_t.

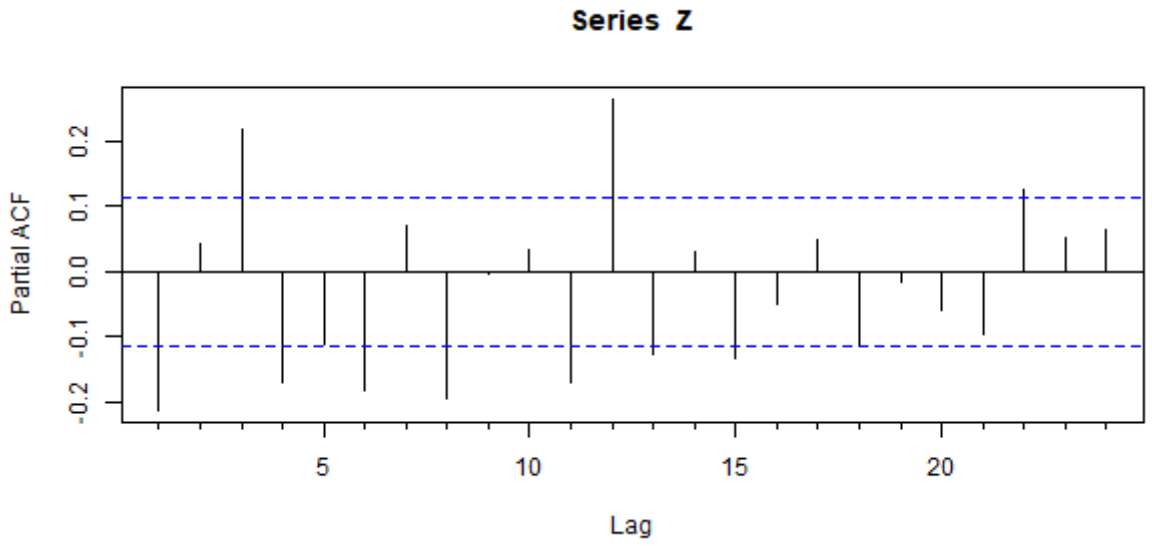


Figure(2) Plot of the differenced timeseries

Z_t does not have any trend hence we conclude that in the SARIMA model the order of differencing $d = 1$. Now to find the value of p, P, Q, and q we look at the ACF and PACF plots. The PACF plot shows the direct correlation between Z_t and Z_{t-k} where k is the number of lags. The largest k for which the PACF value is significantly different from zero gives a good measure of p. The ACF plot shows the direct correlation for Z_t and Z_{t-k}. The largest k for which the ACF value is significantly different from zero give a good measure of q. P, Q, and m are also found similarly. Since some correlation might arise from random chance and may not be reflective of the underlying data generation process we have to consider multiple values for p and q and then select the best. AICc value is used to select the best model.



Figure(3) ACF plot for Z



Figure(4) PACF plot for Z

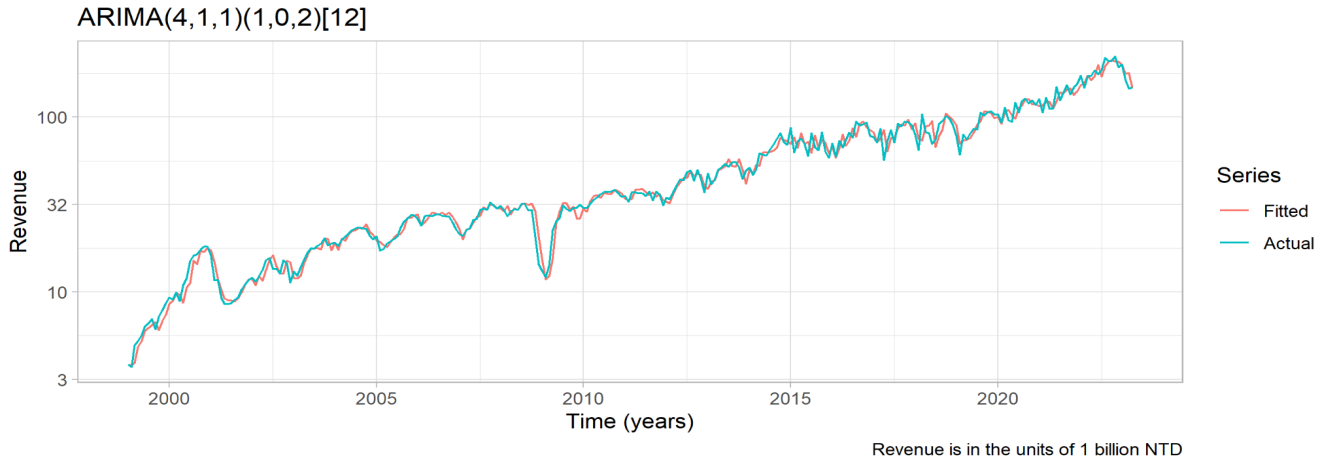
The spike in the PACF plot at lag = 12 suggests that the seasonal period (m) is 12. Using the plots we guess that $p \in \{1, 3, 4\}$, $q \in \{1, 3, 4\}$, $D \in \{0\}$, $P \in \{1\}$, $Q \in \{0, 1, 2\}$. Now we take all possible combinations of p, q, and Q and calculate the AICc value for each of them. The values of p, q, and Q which minimises AICc are 4, 1, and 2 respectively. Hence the arima model that we get is ARIMA(4,1,1)(1,0,2)[12].

	ar1	ar2	ar3	ar4	ma1	sar1	sma1	sma2
Value	0.8218	0.1028	0.2128	-0.3186	-0.9289	0.9892	-0.7017	-0.2032
S.D.	0.0672	0.0749	0.0746	0.0571	0.0387	0.0216	0.0814	0.0783

In order to get the final model equation we need to find out which coefficients are statistically different from 0. If the value of the coefficients is within two standard deviations of 0 then we say that that coefficient is statistically equivalent to 0 hence we will not include it in the model equation. We find that ar2 should not be included.

$$Z_t = 0.822\, Z_{t-1} + 0.213\, Z_{t-3} - 0.319\, Z_{t-4} - 0.929\, \varepsilon_{t-1} + 0.989\, Z_{t-12} - 0.702\, \varepsilon_{t-12} - 0.203\, \varepsilon_{t-24}$$

$$Y_t = Z_t + Y_{t-1}$$



Figure(5) Fitted Arima model

If our model is good then the residuals should be IID normal random variables i.e. white noise. Ljung-Box test with number of lags = 24 on our residuals gives us a p-value of 0.36 which shows that the residuals are in fact white noise.

Holt-Winters

Following the same procedure as in the SARIMA model we fit our model on Z_t. We use the same seasonal period = 12. The smoothing parameter α, β, and γ are estimated by minimising the mean squared error.

$\alpha = 0.0008, \beta = 0.0008, \gamma = 0.0485.$

The estimated initial values are

$l_{12} = mean(Z_1, Z_{13}, \ldots) = 0.0111$

$b_{12} = mean(\frac{Z_{13}-Z_1}{12}, \frac{Z_{14}-Z_2}{12}, \ldots) = -0.003$

$s_i = Z_i - l_{12}, \forall i \in \{1, 2, \ldots 12\}$

s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12
0.0071	-0.029	-0.0286	0.0181	-0.0064	0.0258	0.0061	0.0142	0.0188	-0.0099	0.0477	-0.0639

$\forall t > 12,$

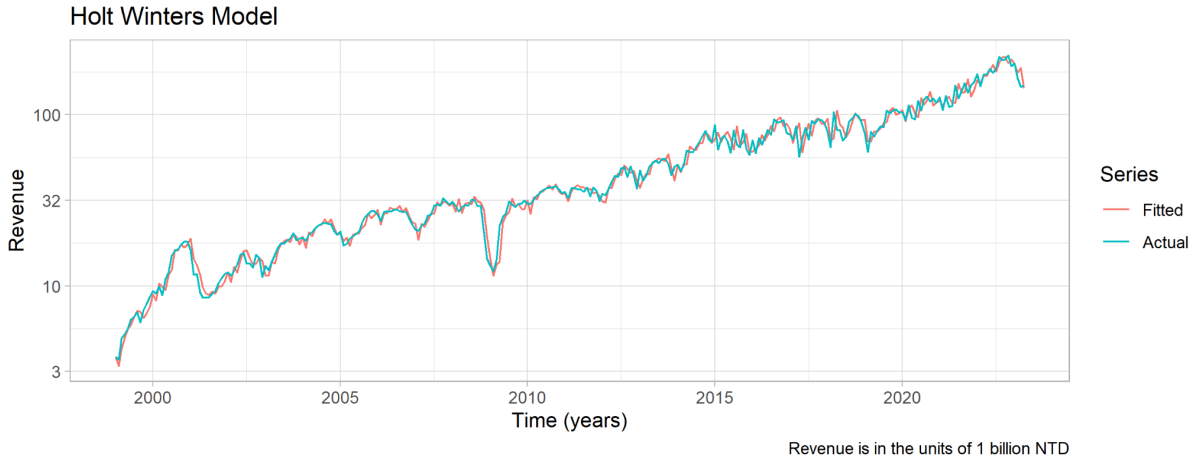
$l_t = \alpha(Z_t - s_{t-12}) + (1-\alpha)(l_{t-1} - b_{t-1})$

$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}$

$s_t = \gamma(Z_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-12}$

$Z_t = l_{t-1} + b_{t-1} + s_{t-12}$

$Y_t = Z_t + Y_{t-1}$



Figure(6) Holt Winters Model

Prophet

The model is fitted on Z_t with growth set to 'linear' and seasonality mode set to additive.

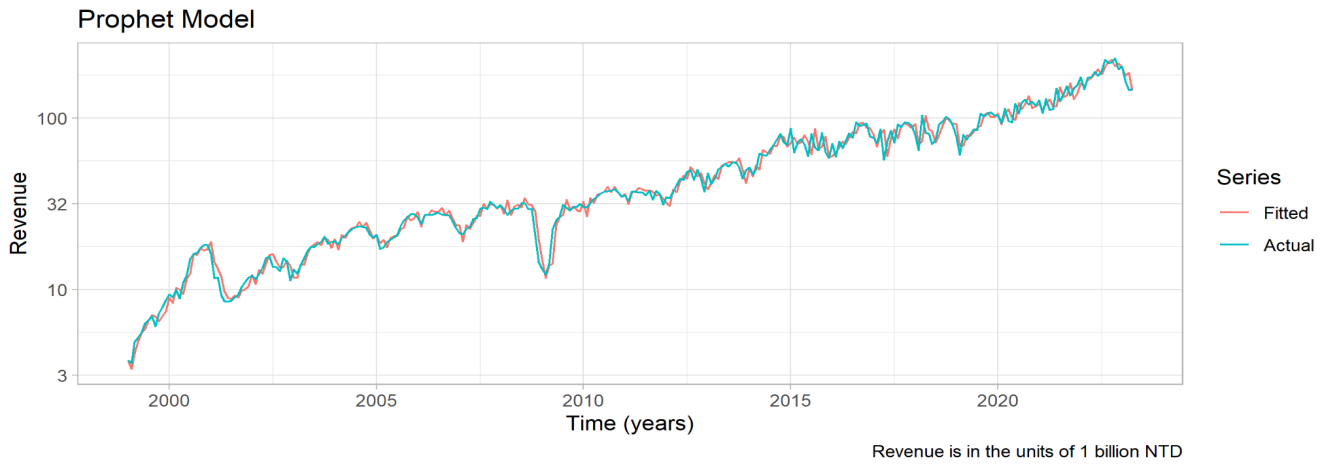


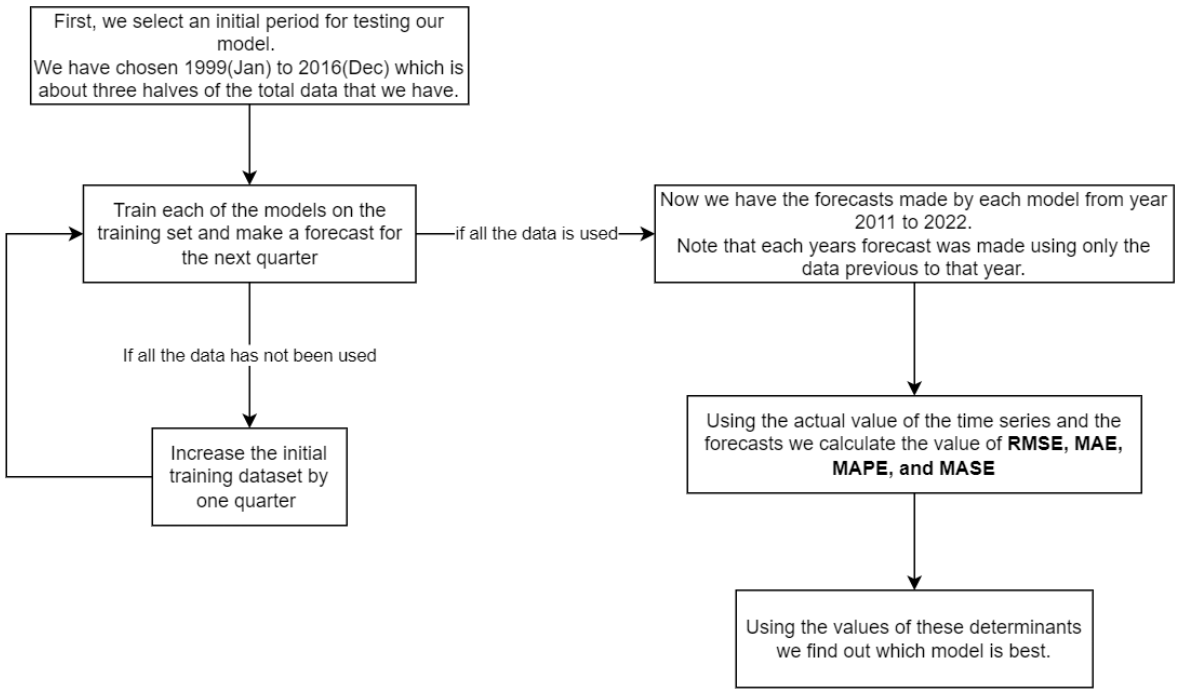
Figure showing the Prophet model

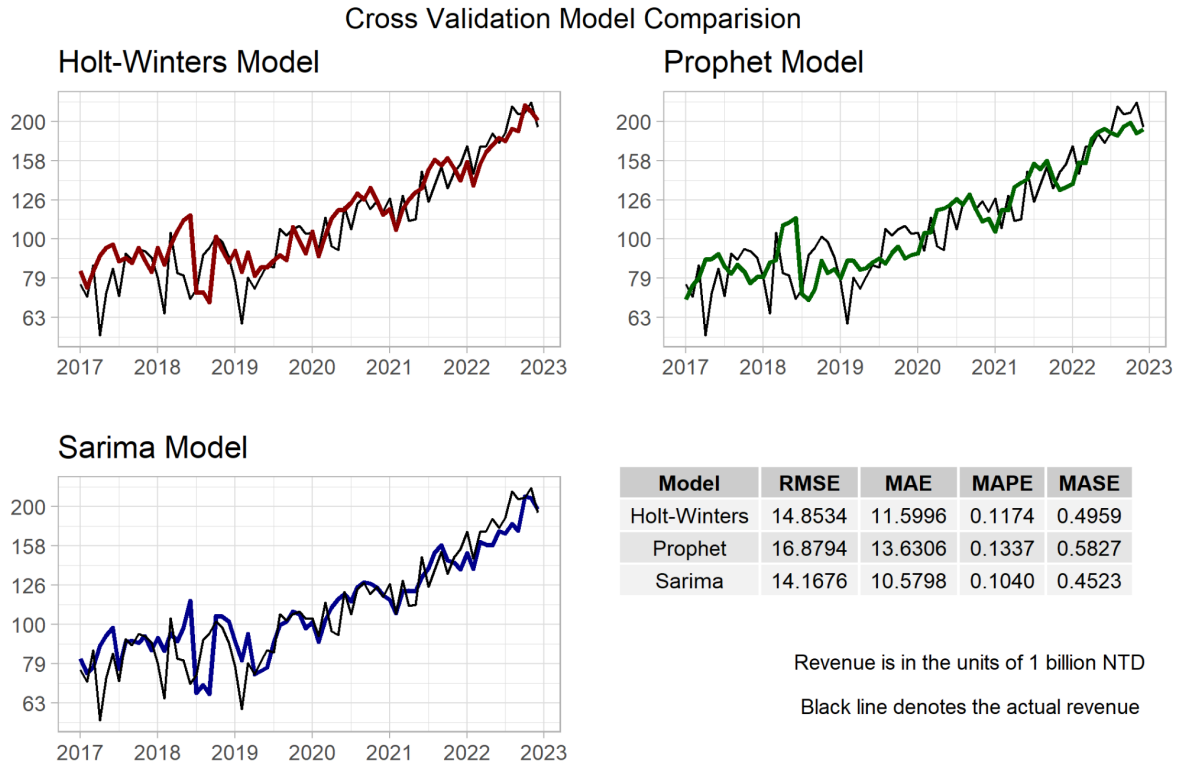
LSTM

Cross Validation

The objective of our study is to identify the most effective model for predicting revenue for the upcoming quarter, specifically targeting a 3-month forecasting horizon. To evaluate the performance of the models, we have adopted the rolling forecast origin cross-validation technique. This technique involves a systematic process of training and testing the models using sequential time periods.

The methodology can be summarized as follows:





Figure() Plot showing comparison of the three models

Conclusion

Based on the analysis and evaluation of all four determinants, it is evident that the Prophet model does not outperform the other two models, namely Holt-Winters and SARIMA. However, it is important to acknowledge that this study is limited to a specific context, focusing on predicting the future monthly revenue of TSMC for a 3-month period.

It should be recognized that the performance of the Prophet model may vary depending on the particular time series under analysis and the forecasting period considered. Different datasets and forecasting objectives might yield different results. Therefore, the findings of this study cannot be generalized to all scenarios and must be interpreted within the scope of this specific investigation. Further research and experimentation are necessary to explore the performance of different models across diverse datasets and forecast scenarios.

Considering the specific case of predicting the future monthly revenue of TSMC for a 3-month period, the SARIMA model emerges as the superior choice compared to both Holt-Winters and the Prophet model. The SARIMA model demonstrated better accuracy and reliability in forecasting revenue for the given timeframe.

References

Things left to do

1. References dalne h
2. Prophet ki theory likni h(Analysis and Results wale part me) simiar to SARIMA

3. Residual ke bare me sochna h
4. Ljung box test ko pura likhna h