Automatic Forecasting of Bike-Sharing Demand

Amir Eid

Dissertation written under the supervision of professor Pedro Afonso Fernandes

Dissertation submitted in partial fulfilment of requirements for the MSc in Business Analytics, at the Universidade Católica Portuguesa, January 2026.

Automatic Forecasting of Bike-Sharing Demand

Amir Eid

January 22, 2026

Abstract

This thesis aims to compare the forecasting accuracy and computational efficiency of the time series forecasting models SARIMA, Theta and Prophet based on the data from 2017-2019 of a bike-sharing company from San Francisco. The approach of the experiment was to do a rolling-origin forecast for univariate data to guarantee a suitable comparison despite the models differences. The results have shown that each model had each own strengths. SARIMA excelled at portraying seasonality and overall having the highest average forecasting accuracy, its amount of preprocessing steps and computational costs are significantly high, which makes it unsuitable for automatic forecasting. Theta has no preprocessing steps and produces forecasts almost instantly due to its simplicity, but has considerably worse forecasts in comparison to the other two models. Moreover, both models struggle with outliers and the impact of external covariates, which makes it difficult to apply to bike-sharing demand forecasting. Prophet proves to have a decent balance between computational efficiency and precision. Even though the model is highly inaccurate in the beginning of the rolling forecast approach, it manages to improve significantly to a comparable accuracy of SARIMA over time and furthermore succeeds at dealing with outliers and additional external factors. The findings show that there is a trade-off between forecasting accuracy and computational efficiency and the ideal balance and model depends on the needs of the business. Future work could include the automation of SARIMA's needed preprocessing steps, further testing on additional datasets or the inclusion of additional variables.

Keywords: Univariate Time Series, Automatic Forecasting, Bike-Sharing Demand.

Automatic Forecasting of Bike-Sharing Demand

Amir Eid

January 22, 2026

Resumo

Esta tese visa comparar a precisão de previsão e a eficiência computacional dos modelos de séries temporais SARIMA, Theta e Prophet, com base em dados de 2017-2019 de uma empresa de compartilhamento de bicicletas em São Francisco. A abordagem do experimento foi realizar uma previsão de origem móvel para dados univariados para garantir uma comparação adequada, apesar das diferenças entre os modelos. Os resultados mostraram que cada modelo possui seus próprios pontos fortes. O SARIMA destacou-se ao retratar a sazonalidade e obteve a maior precisão média de previsão, mas suas etapas de pré-processamento e custos computacionais são significativamente altos, tornando-o inadequado para previsões automáticas. O Theta não exige pré-processamento e produz previsões quase instantaneamente devido à sua simplicidade, mas apresenta previsões consideravelmente piores em comparação aos outros dois modelos. Além disso, ambos enfrentam dificuldades com outliers e o impacto de covariáveis externas, o que dificulta sua aplicação na previsão de demanda de bicicletas. O Prophet demonstra um equilíbrio razoável entre eficiência computacional e precisão. Embora seja altamente impreciso no início da abordagem de origem móvel, ele melhora significativamente até atingir uma precisão comparável à do SARIMA, lidando com sucesso com outliers e fatores externos. Os achados mostram que existe um trade-off entre precisão e eficiência, e o equilíbrio ideal depende das necessidades do negócio. Trabalhos futuros poderiam incluir a automação do pré-processamento do SARIMA, testes em dados adicionais ou a inclusão de novas variáveis.

Palavras-chave: Séries Cronológicas Univariadas, Previsão Automática, Procura de Bicicletas Partilhadas.

Contents

List of Figures

1 Introduction

Accurate forecasts are an important task in data science for business success. Therefore, it is crucial to develop high-quality algorithms and have trained employees who understand the logic and technical setup of the forecast. Although this can be possible for long-term forecasts, for short-term forecasts, such as hourly, daily, or weekly forecasts, these requirements can make the task expensive and difficult to scale. As a result, although high-quality forecasts can be produced, they usually are not produced at the pace that might be needed. For the use of multiple forecasts with a huge amount of data, this may not be sufficient, and improved methods might be necessary. Therefore, it can be said that to scale a huge variety of forecasts in a short time period, the forecasting methods should be suitable and configurable for people with skills in their respective domains, but without deep prior knowledge of time series methods. In order to achieve this, it is necessary for forecasts to become mostly automated, thus creating a significant amount of forecasts mostly produced by computing power rather than continuous human input. Therefore, forecasting modules will need to be able to efficiently evaluate and compare the efficiency of the model, whereas the only human input would be to analyze and adjust the model periodically or in case of significantly poor performance. (Taylor and Letham, 2018)

Moreover, it has to be considered that by increasing the amount of forecasts or forecasting frequency in addition to complex forecasting or Machine Learning models, the whole process would be computationally expensive and time-consuming. Therefore, the feasibility and flexibility of accurate short-term forecasts would be difficult to achieve. (Petropoulos et al., 2025)

A forecasting method, which has not gained broader attention yet, but could be suitable for the above-mentioned task, is the Theta method. The method gained traction in the M3 forecasting competition when it delivered better results than other models. Since it can be compared to Simple Exponential Smoothing (SES) with drift algorithm, the combination of simplicity while still being able to achieve accurate forecasts seems promising for achieving the goal of having feasible automatic forecasts. (Spiliotis et al., 2020)

On a personal matter, the motivation of this research would be to improve supply and demand constraints of mobility options in urban places in order to make human transportation more sustainable, more variable and less car-dependent. Moreover, since I was not able to find any empirical studies using the Theta method for bike-sharing demand forecasting, a further motivation on an academic level would be to bridge this gap in bike-sharing mobility literature. Furthermore, by achieving useful accuracy with the usage of smaller and more computationally efficient forecasts, such as Theta, forecasting could become more environmentally sustainable in the future. Therefore, the aim of this thesis would be to evaluate and compare the performance of the Theta method compared to well-known methods, such as ARIMA and Prophet, in terms of forecasting accuracy and computational efficiency for daily forecasts. In order to achieve that all three models will be used to forecast demand of a bike-sharing company based in San Francisco aggregated on a daily time level. Both forecasting accuracy and computational time needed will be evaluated. Moreover, in order to achieve a fair comparison across models the experiment will focus on univariate forecasts, which means that covariates, such as weather or holidays will not be taken into consideration. Concluding, the research question of this research paper would be: How does the Theta method perform in univariate forecasting for San Francisco bike-sharing demand compared to ARIMA and Prophet, in terms of both forecasting accuracy and computational efficiency on daily forecasts?

As a result, this research would lead to an improved understanding of automatic forecasting, which could lead to finding the most efficient and sustainable prediction methods in short-term urban mobility.

This thesis will have the following hypotheses:

- Even though the Theta model has a by far less complex structure than ARIMA or Prophet, the forecasting accuracy will be comparable to the other two models.
- ARIMA will achieve the highest forecasting accuracy but will be computationally inefficient and therefore not suitable for automatic forecasting in comparison to Theta or Prophet.
- Prophet will thrive at accurately forecasting trend changes and irregularities, whereas Theta may show its strengths by forecasting in the case of stable seasonal patterns.

2 Literature Review

2.1 Time-Series Forecasting

A time-series is a number of data points collected sequentially over time. Forecasts are computed from present and historical values and accurate forecasts are important in several areas, such as scientific, commercial, industrial or economic areas. Traditionally, forecasting methods could be classified into judgmental forecasts, univariate methods and multivariate methods. Judgmental forecasts are basically just built on base of intuition or subjective judgment, instead of statistical analysis. In univariate methods forecasts simply are only based on present and past data of a single series and can also be augmented by a time function like a linear trend. In multivariate forecasts the forecasted variable also depends to some degree on one or more additional time series variables, known as predictor or explanatory variables. (Chatfield, 2000)

A time series can be decomposed into different components, such as seasonality, trend, cycle and irregular fluctuations. Seasonality can be described as reoccurring patterns of behavior in the data observed at specific times of the year. A trend is the experience of steady upward growth or downward decline over several time periods. A cycle is explained as a regular variation in the data other than a year. A typical example of a cycle would be a business cycle. Last there are irregular fluctuations, which are variations in the data after the removal of trend, seasonality and other systematic components. These fluctuations could be explained as being random and therefore not possible to forecast. (Chatfield, 2000)

Another important concept for time series forecasting and thus models like ARIMA and Theta is stationarity. The properties of a stationary time series are not dependent on the time of a series, which means that time series with trends or seasonality are not considered stationary, since they would impact the observations of the series at different time points. Therefore, there would not be long term predictable patterns visible in a stationary time series and time plots would show that they are mostly horizontal with constant variance. (Hyndman and Athanasopoulos, 2018)

In order to stabilize non-stationary time series and making them useful for models like ARIMA or Theta, the concept of differencing is proven to be useful, which means to compute the difference between consecutive observations. By eliminating changes in the level of a time series the mean of a time series can get stabilized and as a result reduce seasonality and trend in the series. (Hyndman and Athanasopoulos, 2018)

Forecasting model types nowadays can be categorized in traditional statistical models or new machine learning models. Even though machine learning appears to be promising tool for a wide range of use cases it still seems to be limited in terms of times series forecasting due to preprocessing issues, time dependence and proneness to overfitting. Furthermore, the intense need of huge computational power for machine learning models make it more difficult to implement in high-frequent forecasting real life cases, such as automatic forecasting of weekly demand in bike-sharing mobility. (Makridakis et al., 2018a)

Moreover, the lack of interpretability makes the use of machine learning in time series forecasting more risky to implement in real life, since neither the validity nor the origin of the made forecasts could be explained. (Makridakis et al., 2018b)

2.2 Automatic Forecasting

Automatic forecasting uses algorithms, which decide an appropriate time series model, estimate the parameters and produce forecasts without or as little human intervention as necessary. They must be able to provide accurate forecasts regardless of unusual time series patterns and should be usable for large numbers of series, since it is common for a lot of businesses to have a lot of different products, which need forecasts in short time periods, such as daily or weekly. Generally, automatic forecasting models are either derived from exponential smoothing or ARIMA. This can also be seen in the M forecasting competitions, where they regularly perform really well (Makridakis et al., 2018a). The main characteristics of an automatic algorithm would be to automatically optimize the parameters for every possible model, select the best models based on the lowest Akaike's Information Criterion (AIC) and to finally produce point forecasts including prediction intervals for whatever time horizon needed. (Hyndman and Khandakar, 2008)

Even though the process of selecting forecasting models with the ARIMA model is often considered as a manual and subjective process, which would moreover also be difficult to apply, but there actually are several possibilities of automating ARIMA forecasting. (Hyndman and Khandakar, 2008) Furthermore, with exponential smoothing models it is possible to automatically decompose and optimize the model parameters, such as trend, seasonality, cycle and the error term without human intervention (Hyndman et al., 2008).

2.3 The Theta model

The theta method recently gained attention due to its outstanding performance in the M3 forecasting competition. It has been proposed by Assimakopoulos and Nikolopoulos in 2000 as a univariate forecasting method, splitting the data into two or more lines. (Spiliotis et al., 2020) The M3 competition was a forecasting competition, where automatic forecasts had to be generated from one model. The model had to forecast 3003 different time series of different types, mostly yearly, quarterly and monthly. The final conclusion was that the Theta model performed overall the best. (Makridakis and Hibon, 2000)

The theory behind the theta model is to adjust the local curvatures of a time series, which is achieved through the Theta-coefficient, which will be directly applied to the second differences of the time series.

$$X''_{\text{new}}(\theta) = \theta \cdot X''_{\text{data}}, \quad \text{where} \quad X''_{\text{data}} = X_t - 2X_{t-1} + X_{t-2} \text{ at time } t.$$

The lower θ is, the more flatter is the adjusted time series. If θ was 0, then the time series would get modified to a linear regression line. Moreover, the larger theta would be, the more focus would be laid on the short-term behavior.

There are six main steps behind the theta algorithm, which have been used during the M3 competition:

- Step 0: The time series gets tested for statistically significant seasonal behavior.
- Step 1: The time series gets deseasonalized by using the multiplicative classical decomposition method.
- Step 2: The time series gets decomposed into two or more theta-lines.
- Step 3: The theta lines get extrapolated. The classic way is to use linear regression for the first line, while simple exponential smoothing gets chosen for the second line.
- Step 4: Now that the forecasts have been produced from the extrapolation process, they get combined together with equal weights.
- Step 5: In the final step the final forecast gets reseasonalized.

The strength of the theta model lays in the decomposition of the original data. The first theta-line holds information about the long-term trend, whereas the second theta-line focuses on short-term fluctuations. (Assimakopoulos and Nikolopoulos, 2000)

The main limitations of the Theta method are not being accurate enough for time series of non-linear trends (e.g.: exponential trends) and connecting the components of trend and level additively, wheres seasonality is multiplied with its connected, even though time series components neither have to be connected additively nor multiplicatively. (Spiliotis et al., 2020)

Through the modification of the original Theta method, it can be transformed into an automatic forecasting algorithm, which can make generalized predictions for various business cases. It has been shown that through expanding the classic Theta model with considerations of linear and non-linear trends and the slope adjustment of these trends and also adding a multiplicative expression of the model to the additive one, AutoTheta can be a promising automatic forecasting tool, achieving higher or comparable forecasting accuracy to other forecasting algorithms, while keeping computational costs drastically lower. Results have shown that generalizing the classic Theta method this way, helps effectively its usefulness for automatic forecasts and improves forecasting accuracy by an average of 4%. On the other hand this improvement of AutoTheta also resulted with 40 times higher computational expenses, in comparison to just using classic Theta.(Spiliotis et al., 2020)

Another interesting use case of Theta in automatic forecasting is the theta intelligent forecasting information system (TIFIS). TIFIS uses the potential of knowledge-based techniques and the theta model for accurate forecasts for business. The most important processes of the theta algorithm for TIFIS are the deseasonalization and extrapolation processes. (Nikolopoulos and Assimakopoulos, 2003)

Another improvement idea of the classic theta method was to add a third theta line (theta = 1), which gets extrapolated using the damped exponential smoothing (DES) method. Moreover, since it is common that the trend of a time series changes over time, it is possible that the long-term and short-term trend move in different directions. Therefore, it could be that the long-term trend is an over- or underestimation of the short-term situation. As a result, it has been suggested that another linear trend line gets added to the theta model, which focuses on the short-term trend. This means that a linear trend would be fitted on the last k data points of the deseasonalized data.

There are two possible ways of specifying the value k: (Nikolopoulos and Thomakos, 2019)

$k = H$, where $H < n$ and is the required forecasting horizon,

$k = m$, where $m < n$ and is the periodicity of the data (e.g., $m = 12$ for monthly data).

It was considered that the Theta model could be a dynamic model, through choosing different theta lines and adapting the weights of the forecasts, which has not been done in the M3-Competition (Fiorucci et al., 2016). Therefore, the optimized Theta method has been suggested, which automatically finds the ideal theta lines and weights by minimizing the forecasting error through a loss function (Fioruci et al., 2015). To go even further, it has been taken into consideration with the Dynamic Optimized Theta Method (DOTM) to implement an improvement of the optimized Theta method by allowing the minimization of the loss-function to be dynamic by time. This means that the optimized Theta parameters can vary over time, which means that the forecasts do not have to be necessarily linear. Experiments have shown that the implementation of the DOTM lead to significant forecasting accuracy improvements, even though it has to be considered that the computational costs increased as well. Nonetheless, it would still be computationally far more efficient than the ARIMA model.(Fiorucci et al., 2016)

2.4 ARIMA

AutoRegressive Integrated Moving Average (ARIMA) is one of the most common models for time series forecasting, which has been introduced by Box et al. (2015) in the 1970s. The main goal of ARIMA models is to explain autocorrelations in the data. The ARIMA(p,d,q) model goes as follows:

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

where y't is the differenced series, which might have been differenced more than once, p is the order of the autoregressive part, d is the degree of first differencing and q is the order of the moving average part. (Hyndman and Athanasopoulos, 2018)

In order to include seasonal considerations to an ARIMA model, additional seasonal terms have to be implemented into the model. As a result Seasonal ARIMA (SARIMA) adds (P,D,Q)m to ARIMAs (p,d,q) parameters, where m is representing the number of observations per year. Therefore the uppercase letters are used for the seasonal parts of the model whereas the lowercase show the non-seasonal components of the model. Generally, the seasonal and non-seasonal parts of the model are similar apart from the seasonal components having lags of the seasonal period. Therefore, an ARIMA(1,1,1)(1,1,1) model with m = 4 for quarterly data can be formulated as followed: (Hyndman and Athanasopoulos, 2018)

$$(1 - \phi_1 B)(1 - \Phi_1 B^4)(1 - B)(1 - B^4)y_t = (1 + \theta_1 B)(1 + \Theta_1 B^4)\varepsilon_t$$

Implementing ARIMA for automatic forecasting has in general been a difficult task due to the subjective order selection process of the model. Moreover, as a result of exponential smoothing performing better in the M3 automatic forecasting competition it has been discussed that the larger model space of ARIMA might hurt overall forecasting performance, since it could include additional uncertainty. (Hyndman and Khandakar, 2008) Previous research in bike-sharing forecasting has shown that first of all the the parameters p and q of the ARIMA model had to be balanced to keep computational costs at a manageable level without losing too much forecasting accuracy. Furthermore, ARIMA's forecasting results appear as stable throughout the time series, since its strengths lay in finding linear parts of a time series. (Lin et al., 2024)

Wang (2024) shows that it is possible to achieve respectable forecasting result with ARIMA for univariate bike-sharing forecasting by just using rental volume and time as variables, and purposefully ignoring covariates, such as temperature, wind speed, humidity or other variables.

Moreover, it is visible that multivariate models, which make it possible to include additional variables, outperform ARIMA in terms of predictive performance in bike-sharing forecasting. (Wirtgen et al., 2022) Automatic ARIMA forecasting for COVID-19 cases have been proven to be useful in comparison to machine learning models, such as extreme gradient boosting (XGBoost), generalized linear model elastic net (GLMNet) and random forest methods. (Sardar et al., 2023)

2.5 Prophet

Prophet is an open-source forecasting model created by Meta and has been shown to be efficient in accurately fitting data patterns and seasons (Sharma et al., 2022).

The model uses a separable time series model including the three main model components trend, seasonality and holidays, which are added together in the following equation: (Taylor and Letham, 2018)

$$y(t) = g(t) + s(t) + h(t) + \epsilon(t) \quad (1)$$

.

where: g(t) represents the trend functions by modeling non-periodic changes in the data of the time series, while s(t) portrays periodic changes, such as weekly and yearly seasonality. An unique addition in the Prophet forecasting model is the inclusion of holiday effects with h(t), which can show the effects of possibly irregular time frames of one or more days. Lastly, the error term $\epsilon(t)$ displays changes which are not explained by the model.

Prophet has several strengths, such as the flexibility of easily accommodating seasonality with multiple periods, automatic removal of outliers, quick fitting and fairly interpretable parameters. These factors make it a suitable model for the purpose of automatic forecasting.(Taylor and Letham, 2018)

The trend component can either be implemented by a saturated growth model or a piecewise linear model. Since Prophet was built for Facebook, the most common use case of portraying growth forecasting would be to the expected population growth on the website. Therefore, this usually gets implemented through a logistic growth model: (Taylor and Letham, 2018)

$$g(t) = \frac{C}{1 + e^{-k(t-m)}}$$

where C is the carrying capacity, k is the growth rate and m is an offset parameter. If forecasting data is not showing saturating growth, a piece-wise constant growth rate model is used.

The Prophet model uses the Fourier series for providing flexibility of periodic effects, such as daily, weekly or yearly seasonality, in the forecasts. The flexibility of the seasonality can be controlled by the term N. The higher N is the faster it allows the model to fit seasonal patterns, but it also increases the risk of overfitting the model. The optimal choice of N can be automated by using a model selection procedure, such as AIC. (Taylor and Letham, 2018)

The Prophet model allows the inclusion of holiday and event effects in the forecast. These days are portrayed as additional regressors in the model and each date is represented by a binary outcome variable, which shows whether it is part of a holiday or event or not. (Taylor and Letham, 2018)

Research has shown that Prophet can handle seasonality exceptionally well, even though it struggles to deal with volatility (Yadav, 2022). Moreover, weak points of Prophet are comparatively possible low speed and computational complexity (Grekov et al., 2024)

3 Data and Methodology

In this chapter the data of this thesis and the used methodology will be explained. Following the variables and characteristics of the dataset, the necessary preprocessing tasks and the way of implementing and evaluating the forecasting models Theta, SARIMA and Prophet will be described. As the models have already been thoroughly explained in Chapter 2, the focus in this chapter will lay on the practical use, the experimental design and the used evaluation metrics for a justified fair comparison.

3.1 Data Description

The dataset used in this project is a public dataset from kaggle, which shows detailed trip-level data of Bay Wheels, a regional public bicycle sharing system in San Francisco, California. Trip data from 28 June 2017 to 1 April 2020 are covered in the dataset. Each row is an individual bike trip and has variables such as start and end time of the ride, start and end station name or whether the user is a subscriber. Overall, the dataset contains more than 5 million data points and a file size of 660 MB. In order guarantee a consistent and representative view of regular bike trip demand patterns, trips after 31 December 2019 have been excluded, since their patterns might be distorted because of the COVID-19 pandemic. As a result the final dataset used for this thesis' experiment contains 2.5 years of continuous pre-COVID bike sharing trips.

Choosing daily aggregation therefore is useful, since it can process the weekly cycle and is more useful for strategic planning reasons for bike-sharing companies, such as capacity management and scheduling bike maintenance, in comparison to hourly or weekly forecasts.

3.2 Data Preprocessing

In order to prepare the data for the forecasting experiment, the variable start_time got converted into a datetime object in order to guarantee chronological accuracy. Furthermore, it has been checked, whether the contains missing or duplicate data. Since, there were not any, no further steps had to be taken. Since the COVID pandemic could lead to misshaped forecasts, any data after 31 December 2019 has been removed. Lastly, in order to transform the data to a daily level aggregation, the pandas resample('D') function has been used.

The result of this data preprocessing is a time series with 917 daily ride observations with a range of 401 to 12287 rides per day.

3.3 Data Exploration

In order to obtain more detailed insight on the data an exploratory data analysis will be needed. Therefore, this subchapter is going to interpret the trend, seasonality, stationarity, autocorrelation and distribution of the dataset.

3.3.1 Overview

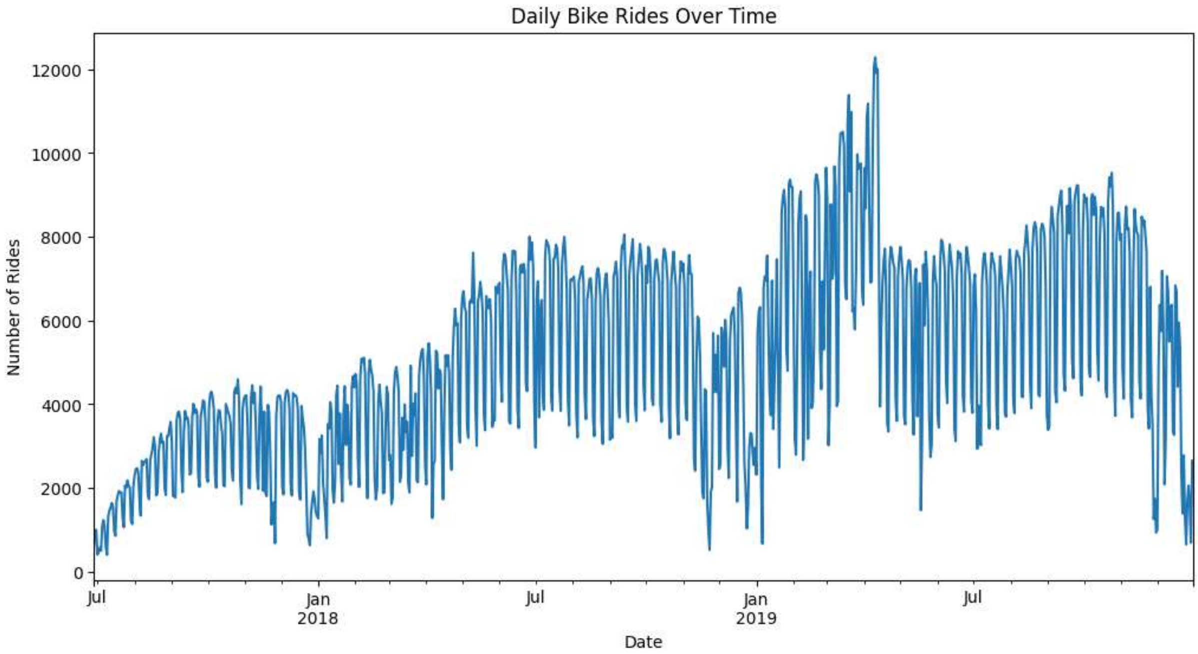

Figure 1: Daily Bike Rides Over Time

Figure 1 shows the number of daily bike rides over the whole time series. It is visible that the time series is in general showing an upward trend over time, with peaking in demand around May 2019. On one hand it can be seen that demand is the lowest in the colder winter months, such as December and January. On the other hand the highest demand for bike rides is usually in the spring to summer months from around April to August. In order to obtain better insight on short-term patterns of the data, Figure 2 shows the daily bike demand from June 2019 to August 2019, which shows a clear pattern of having a weekly cycle. The demand goes up from the beginning of the weak and peaks at the middle of the week and reaches a low on the weekend and the pattern repeats the week after.

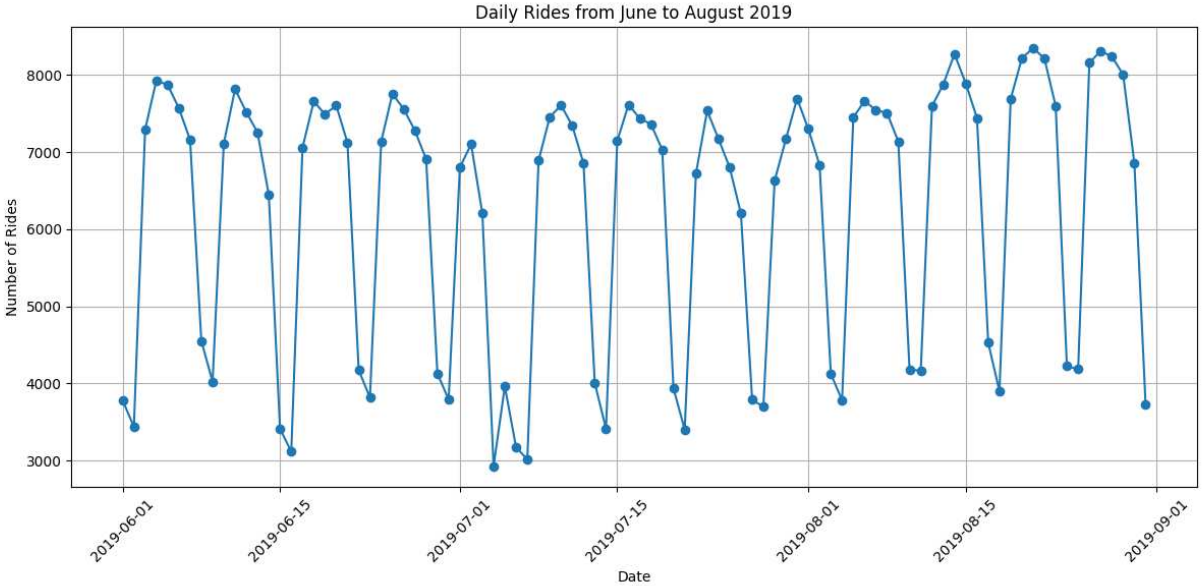

Figure 2: Daily Bike Rides from June 2019 to August 2019

In order to accurately display the above mentioned assumptions, Figure 3 shows a decomposition plot, which gives concise insight on the trend, seasonality and residuals of the time series. The trend in general has the tendency of going upwards. The variance of the residuals seems to have a slight but steady increase from 2017 to 2019 with a few outliers, suggesting the prevalence of heteroskedasticity. In between January 2019 and April 2019 the residuals have the highest variance, which might be explained by unobserved occurrences of external variables, such as more extreme winters than normally. The seasonal decomposition of figure 3 shows that the weekly cycle which has been observed in Figure 2 stays steady visible throughout the time series of the dataset.

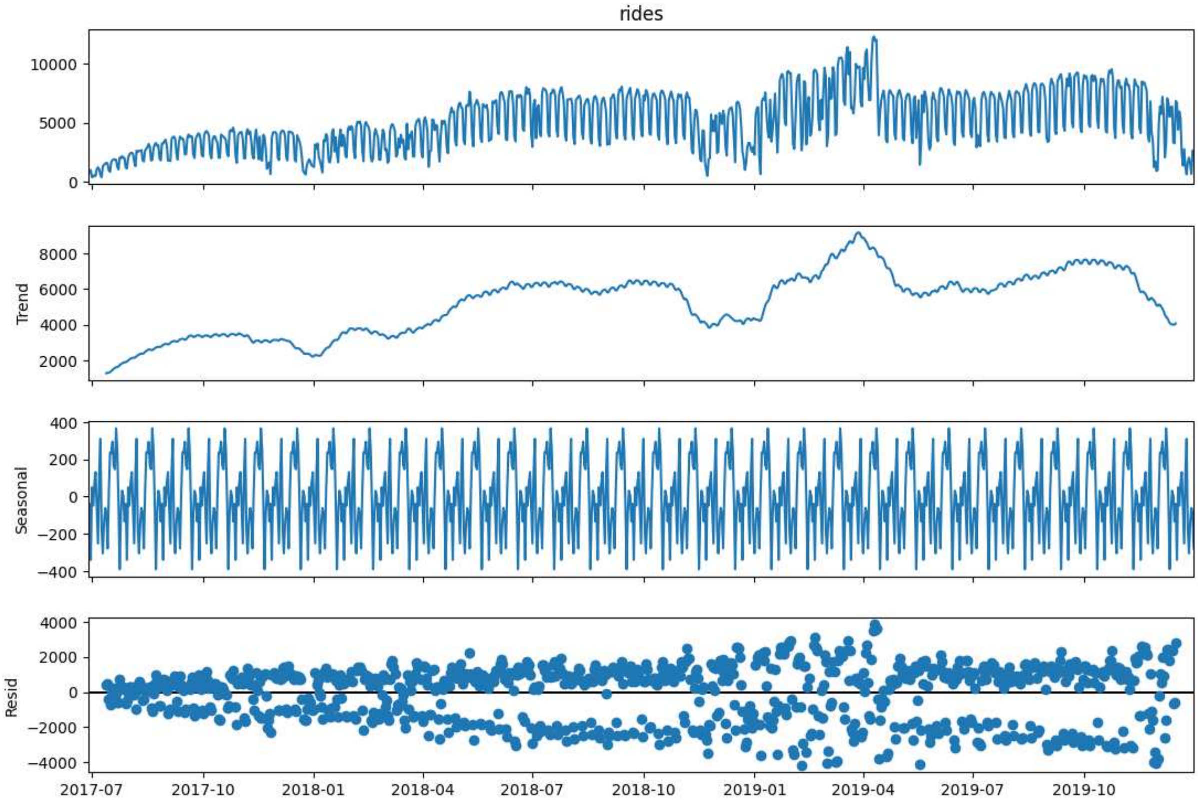

Figure 3: Time Series Decomposition

Figure 4 shows the number of rides by day of the week, where a clear difference in demand can be examined between weekdays and weekends. The median demand on Saturday and Sunday are clearly lower in comparison to the other days, which can probably be explained with the lack of work commutes on these two days. This further demonstrates the weekly cycle, which has been seen in Figure 2. Furthermore, the bike demand throughout the weekdays is roughly the same. It is worth noting that the number of rides on Fridays and Mondays have lower ranges in comparison to the other weekdays.

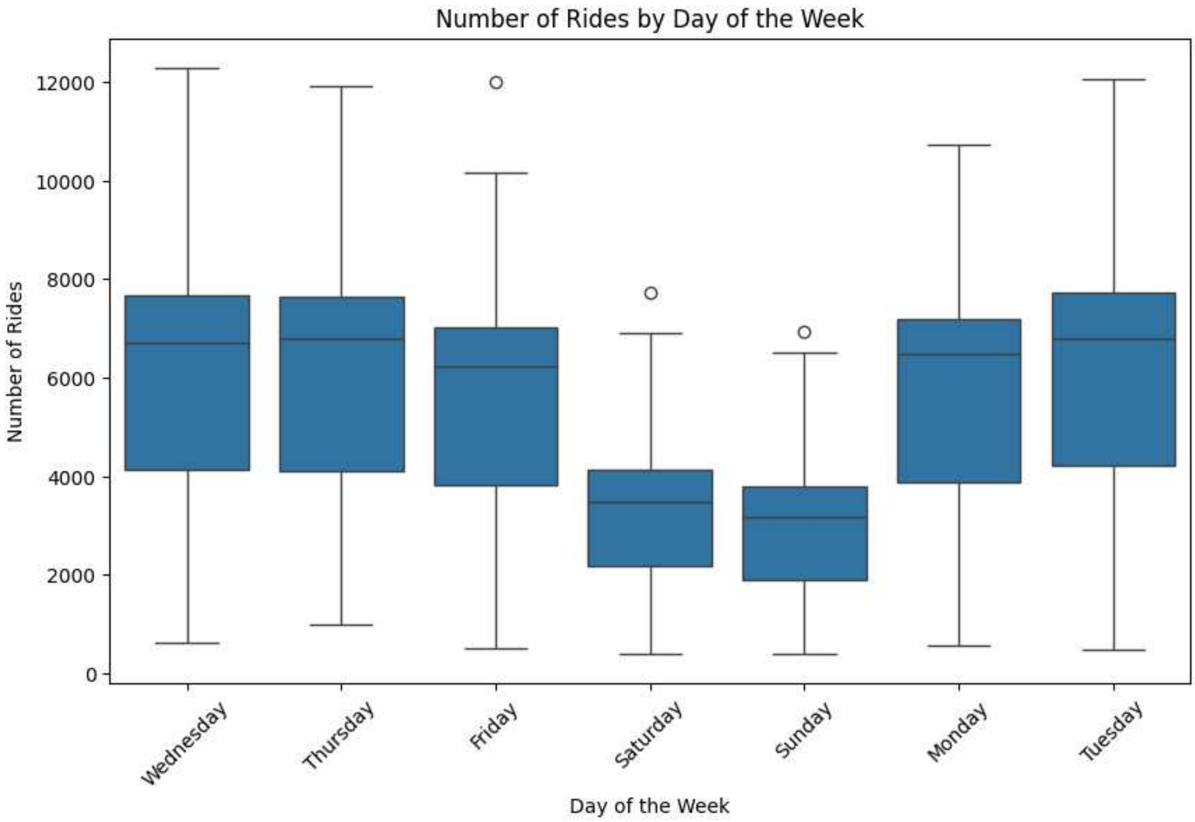

Figure 4: Number of Rides by Day of the Week

Moreover, in order to compare the demand variability throughout the year, Figure 5 demonstrates the demand of bike rides aggregated for each month. This visualization further confirms the theory of demand being the lowest in the winter months from December to February. As mentioned in Figure 1, the highest number of rides can be expected in the summer months, especially in May and June. The median number of rides decreases in June and July in comparison to May and June possibly due to holiday season. Lastly, the months March, April and November act as transitional months, which lead to higher differences in demand making it more difficult to forecast accurately.

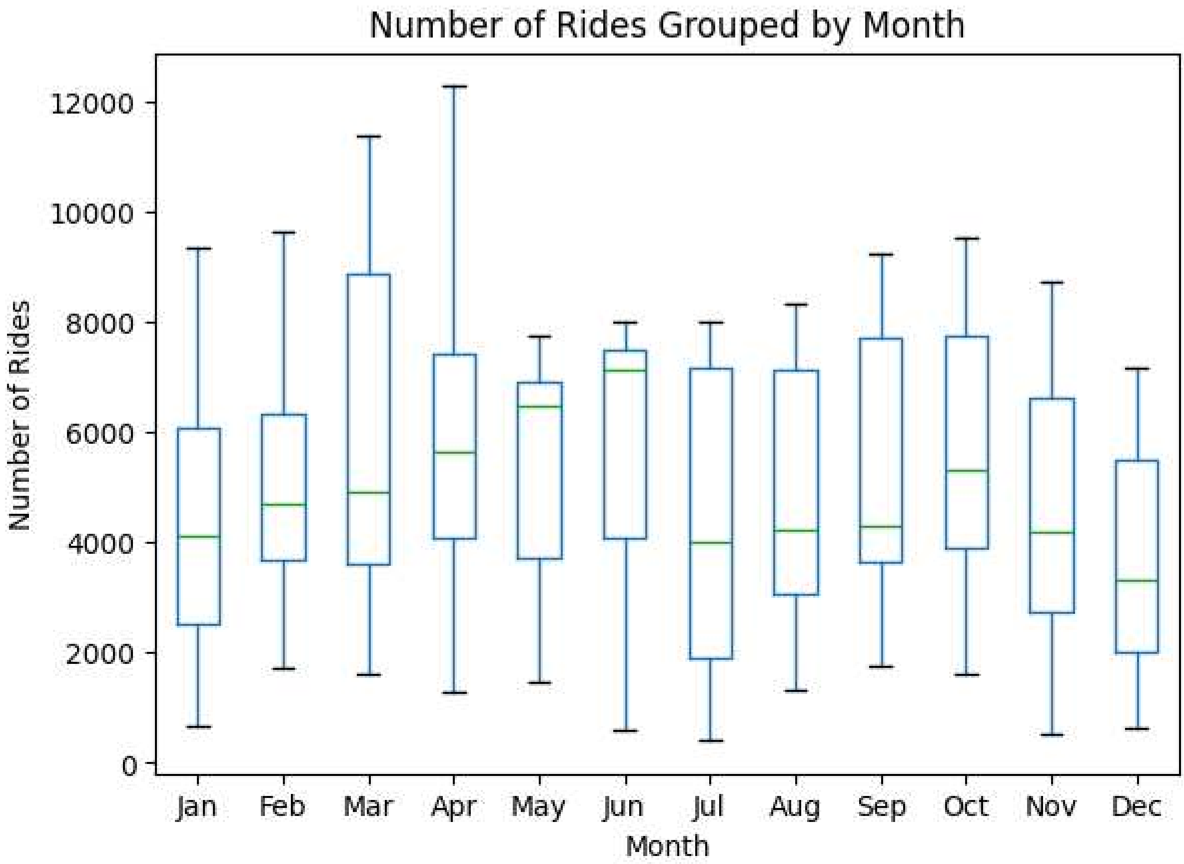

Figure 5: Number of Rides by Month

3.3.2 Stationarity Testing

It is necessary to test the stationarity of the time series, which means that it will be conducted whether the mean, variance and autocorrelation stay stable. Therefore, an an Augment Dickey-Fuller (ADF) test has been applied on the dataset, which checks for an unit root in a time series (Dickey and Fuller, 1979). The result of the test is that with a p-value of 0.195 the null hypothesis that the data is not stationary cannot be rejected. This aligns with the previously perceived trend and seasonality patterns of the time series. The non-stationarity is not an issue for the Theta and Prophet models. By finding the most appropriate parameters for the SARIMA model, this can also be handled well even though it requires further preprocessing steps, which are unsuitable for automatic forecasting.

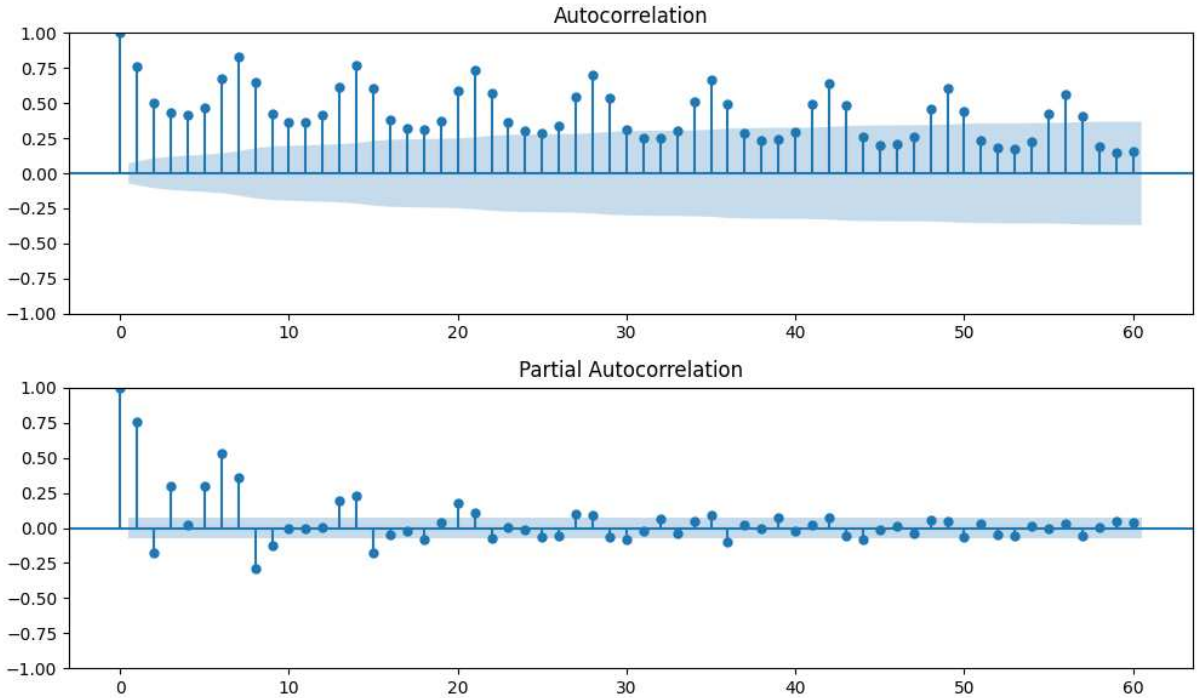

Figure 6: Autocorrelation and Partial Autocorrelation Plot

Figure 6 shows the autocorrelation (ACF) and Partial Autocorrelation (PACF) for the SARIMA model. The purpose of these plots is to see the structure and order of the autoregressive and moving average processes. First of all it can be seen that the ACF has steady significant correlation all up to Lag 60 and a slow decline, which further confirms the idea of non-stationarity in the time series from the ADF test. Moreover, it can be perceived that the ACF plot shows spikes every seven lags, which again shows that there is clear weekly seasonality in the dataset. The PACF plot shows strong significant spikes at Lag 1 and 2 leading to an instant negative decline at Lag 3.

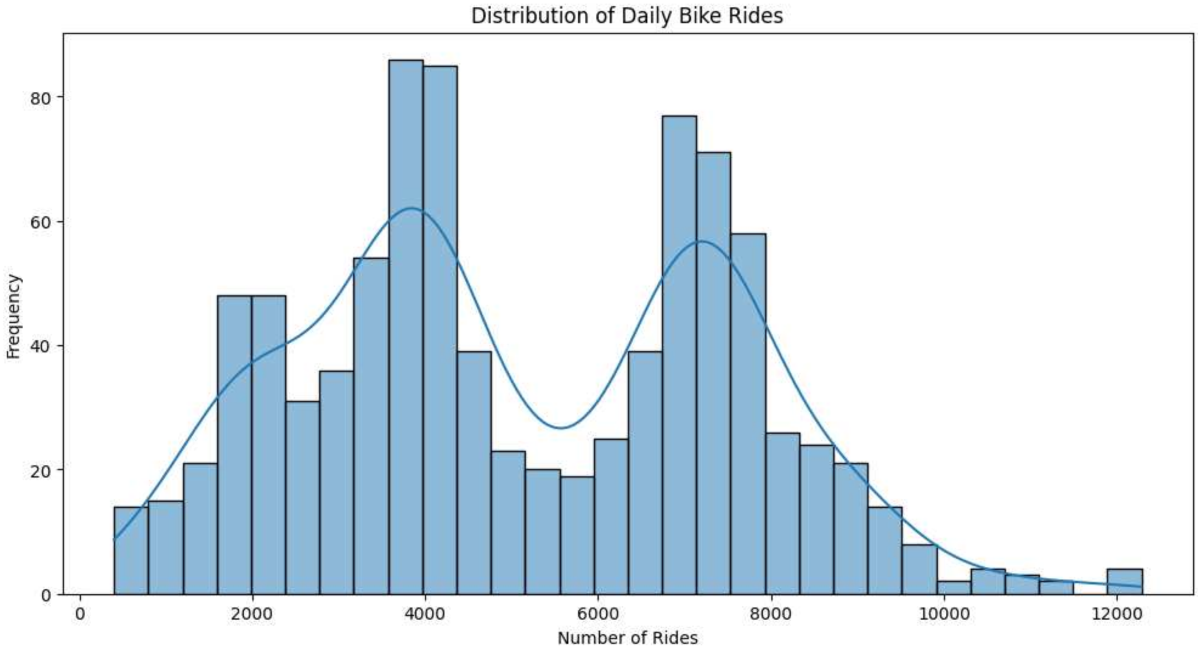

Figure 7: Distribution of daily bike rides

Figure 7 displays the distribution of counted frequencies of daily bike rides. The most common frequencies are around 4000 daily rides and around 7000 daily rides. This bimodality can be caused by the presence of weekdays with high demand and weekends with low demand on the contrary. Furthermore, it is visible that the data is right-skewed, since there are more days with a lower amount of bike rides than days with very a high amount of bike rides.

3.4 Model Implementation

As previously mentioned the goal of this study is to compare the models Theta, SARIMA and Prophet on forecasting accuracy and computational efficiency for an univariate daily demand series of a bike-sharing dataset. The tests were done on a MacBook Pro 2020 (Intel i5, 8 GB RAM) with Python 3.12.3.

The main libraries used for the experiments were:

- pandas and numpy for data handling.
- statsmodels for SARIMA and plotting ACF and PACF.
- autoarima for finding the ideal SARIMA model.
- prophet for Prophet.
- statsforecast for Theta.
- sklearn for the error metrics mean absolute error and mean squared error.

In order to accurately track the runtime of the model forecasts the variable 'start_time' has been formatted to a datetime data type.

3.4.1 Theta model

For this project, the Theta model has been implemented in Python by importing the default Theta module of the statsforecast library. Since the classic Theta model does not deal with extra parameters, no further parameter tuning was needed to prepare the Theta model for the automatic forecasting procedure. Since research showed that the use of AutoTheta results in 40 times higher computational expenses, it did not seem as a fitting model for this thesis (Spiliotis et al., 2020).

3.4.2 SARIMA model

Originally, it was planned to use ARIMA with the auto_arima module of the pmdarima library for the automatic forecast. Since the auto_arima module would in theory find the ideal paramters of the ARIMA model, no additional preprocessing steps would have been needed. In practice nonetheless, first runs of the ARIMA model would have very high computing times, as the code would still not have led to results, even after 16 hours. Therefore, the conclusion was that using ARIMA by letting auto_arima automatically select the parameters would neither be sustainable nor suitable for daily automatic forecasts. Therefore, it was necessary to implement a SARIMA model including manually done preprocessing in advance. By using a stepwise search to minimize AIC, it has been determined that the best SARIMA model for this bike-sharing data is SARIMA(0,1,2)(1,0,1)[7]. This means that for non-seasonal terms, there are no non-seasonal autoregressive terms, first differencing is applied for non-seasonal trend removals and two non-seasonal moving average terms are included in the model. Furthermore, there are one seasonal autoregressive term, no seasonal differencing and one seasonal moving-average term of seasonal components for a seasonality of period 7 in this SARIMA configuration. In the code, the statsmodels library has been used to apply the SARIMA model.

3.4.3 Prophet Model

The Prophet model has been applied with default parameters including the options of detecting yearly and weekly seasonality. Since the goal of this thesis is to provide an univariate comparison between the models, additional regressors, such as weather or holidays have not been taken into consideration for this model.

3.5 Rolling Forecast Implementation

Since it would be necessary to analyze the performance of the forecasting models over different time periods, a rolling forecast will be implemented in this thesis. Rolling forecast evaluations lead to more efficient series-splitting rules by successively updating the forecasting origin and producing more forecasts from these new origins. The models are trained on data up to time t and consequently produce a forecast for the predetermined time horizon. After that the forecasting origin will be moved forward by one period. (Tashman, 2000)

For this thesis time t starts at the first 90 days and the prediction horizon is always the next 14 days. The accuracy of each forecast will be measured with the accuracy metrics Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), since they have been theoretically relevant for a long time in history and have theoretical relevance (Hyndman and Koehler, 2006).

Using this method of rolling evaluation iterations helps to analyze and compare the models strengths and weaknesses over time revealing structural forecasting differences, seasonal effects and performance differences across time periods in general (Bergmeir and Benítez, 2012).

3.6 Evaluation Metrics

For this research evaluation metrics for forecasting accuracy and computational efficiency have been used for an accurate model comparison.

3.6.1 Accuracy Metrics

As already mentioned, MAE and RMSE have been used in this thesis to compare the forecasting accuracy of the models.

MAE is computed by: (Hyndman and Athanasopoulos, 2018)

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|$$

where y_t is the actual value at time t, $\hat{y}_t$ is the predicted value at time t and n are the total number of observations. Forecasts which manage to minimize the MAE are leading to forecasts of the Median (Hyndman and Athanasopoulos, 2018).

RMSE is computed with: (Hyndman and Athanasopoulos, 2018)

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

with the same notation as with MAE. Forecasts that minimize RMSE bring forecasts of the means of the observations (Hyndman and Athanasopoulos, 2018).

3.6.2 Efficiency Metric

In this project the computational efficiency is calculated by using the average run time of all forecasts in the rolling forecasting function for a model. When the rolling forecast is applied on a model, it is measured how long each forecast takes and gets appended to a list of total time.

Once all forecasts of a model are computed, the mean of the run time is computed. Therefore the metric can be formulated as:

$$\text{average run time} = \frac{1}{N}\sum_{i=1}^{N} t_i,$$

where t_i is the runtime in seconds of the model call i and N shows the total amount of forecasts in the rolling forecast.

3.7 Experimental Design

To sum up the workflow of the project can be broken into the following steps. First, in order to prepare the data for the forecasts, the trip observations are aggregated into daily trip counts and COVID related trip data is omitted. Through exploratory analysis the trend, seasonality and stationary of the data is analyzed. Since SARIMA needs preprocessing, its parameters were selected by doing stepwise search to minimize AIC. After that the models Theta, SARIMA and Prophet are fitted to the dataset. By using Rolling-origin, daily forecasts for each model with a horizon of 7 days are produced. Finally MAE, RMSE and computational run time are computed in order to be able to compare the models.

Nonetheless, it needs to mentioned that this thesis holds several limitations, which either go beyond the scope of this research or were not feasible. Even though the original dataset is provided with additional variables, such as if the user is a subscriber or the day is a weekend, these variables could not get included, since the goal of the thesis is to do a univariate comparison. Moreover, other time granularities might show different results of the models, since they might have different patterns. The same may be possible for applying the models to other unrelated datasets. Lastly, the computational time was only measured on one single laptop and as a result the hardware might influence the performance. Regardless, it should be sufficient to show the computational efficiency and comparison of the models and their suitability for automatic forecasting.

4 Results & Findings

In this study, Theta, SARIMA and Prophet have been tested as univariate forecasting models to forecast daily bike-sharing demand between 2017 and 2019 in San Francisco using a rolling-origin forecasting approach with a 14 day horizon and a 90 days training period.

4.1 Overall Results

Table 1: Forecasting Performance Metrics

Model	MAE	RMSE	Avg Time (s)
SARIMA	892.515561	1360.065844	0.567082
Theta	1712.510467	2131.391242	0.030482
Prophet	1296.979503	1772.742564	0.238052

Table 1 compares the average MAE, RMSE and computation time for one forecast for each model. It can be seen that the SARIMA model overall achieved the best accuracy metrics with an MAE of 892.52 and an RMSE of 1360.07, which shows that it has the best performance for short term forecasts in comparison to the Theta and Prophet model. With an average MAE of 1296.98 and an average RMSE of 1772.74 it is evident that the accuracy of the Prophet model was worse than the accuracy of the SARIMA model, even though it is not too far off. The Theta model performed significantly worse in terms of forecasting accuracy in comparison to the other two models with an MAE of 1712.51 and an RMSE of 2131.39. Nonetheless, in terms of needed computational time the results are different. Even though Theta achieved the worst forecasting accuracy, it is computationally the most efficient with only requiring 0.03 seconds for the production of one forecast. This is only a fraction of computational time needed in comparison to the SARIMA model with an average time needed of 0.57 seconds. Prophet again proves itself as a good middle ground, since it achieves a good forecasting accuracy and is still significantly faster than SARIMA with on average 0.24 seconds needed per forecast.

4.2 Model specific Results

In order to gain further insight on each model's performance, it is necessary to dive deeper into each models performance across different seasons and time periods.

4.2.1 SARIMA Results

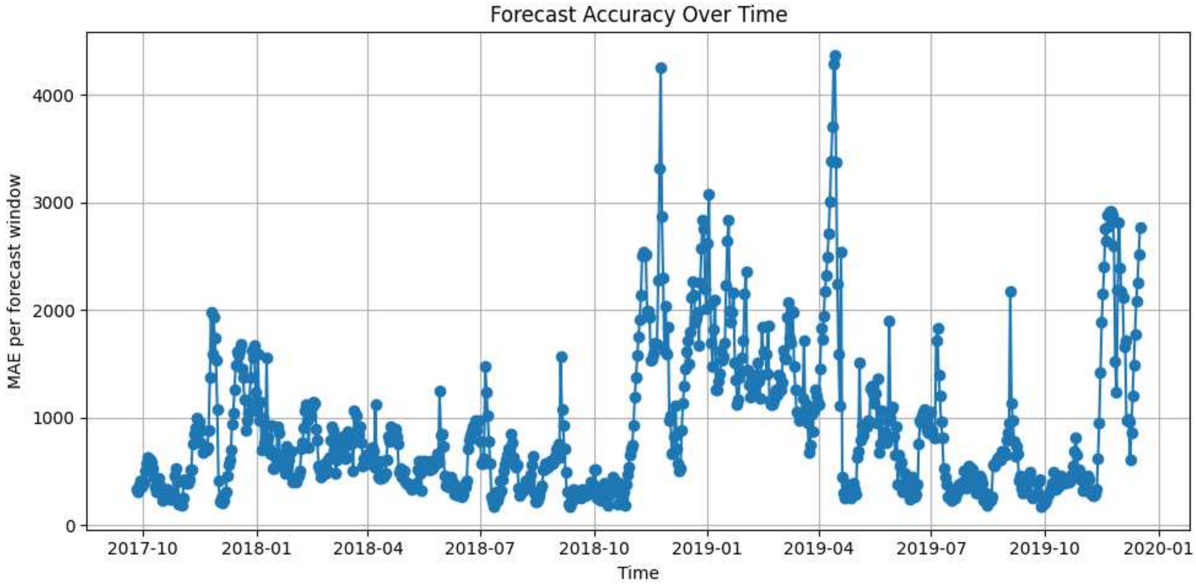

Figure 8: SARIMA Forecasting Accuracy

Figure 8 shows how the SARIMA model performed throughout the time series. The first thing standing out is that the baseline of the MAE is generally in between 250 and 750, implying that the model manages to forecast representative days pretty well. Regardless, it can be seen that the model's accuracy decreased significantly especially in the time periods in between November 2018 and April 2019 and from November 2019 till the end of the time series in December 2019. This can stem from the non-linear trend and increasing variance of the residuals in these time periods. Furthermore, it can be assumed that the increasing variance of the residuals and occurring outliers are not random, but are rather happening due to the effect of external variables in specific seasons, such as more extreme winter conditions. Therefore, it can be concluded that even though the SARIMA model excelled in terms of overall forecasting accuracy throughout the time series, it is prone to outliers and non-normality in the residuals.

In terms of computational efficiency it has to be considered that in addition to taking on average 0.57 to produce a forecast, further time consuming preprocessing steps are necessary in order to find the optimal SARIMA parameters. The original plan to use the auto_arima package to automatically find the best parameters and produce forecasts at the same time failed, since the code did not produce any output, even after running for 16 hours.

4.2.2 Theta Results

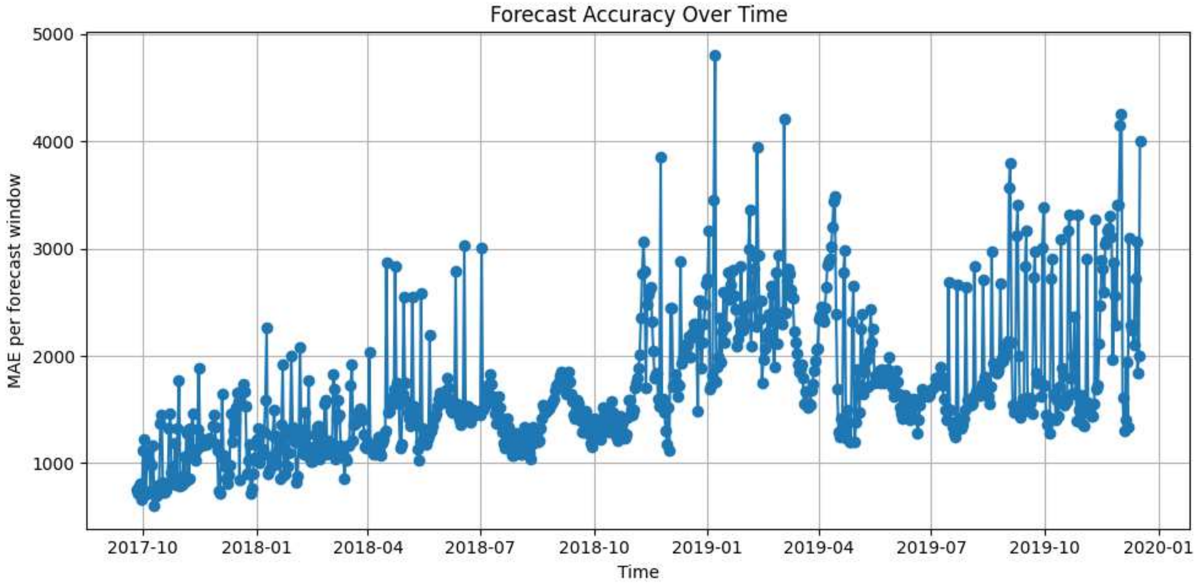

Figure 9: Theta Forecasting Accuracy

In figure 9 the difference in accuracy of the Theta model throughout the time series can be seen. Overall the MAE of the forecasts are hovering in between around 500 to 1500 bike rides. Since it manages to maintain to stay above this baseline, it can be concluded that the model manages to interpret most of the trend and seasonality successfully. This shows that the model's practice of decomposing the original theta-lines into two or more theta-lines proves to be useful for extracting the necessary information for the regular data points. Nevertheless, the Theta model had the worst forecasting accuracy out of the three models and a crucial reason for that is especially Theta's failure of being able to handle the heteroskedasticity of the errors and outliers. Since the variance of the errors increased over time in the dataset, the forecasting accuracy diminishes immensely leading to high fluctuations in the MAE and furthermore even almost achieving an MAE of nearly 5000 daily rides at its peak in January 2019. Considering that the highest amount of rides in one day in January 2019 was at around 8000 daily rides and the day with the highest amount of rides over the whole time series, an MAE of 5000 can be viewed as a significant forecasting error. The reason for this is simply since the forecasting computation with the Theta model by relying on simple exponential smoothing is mathematically easy, which leads to quickly produced time series forecasts, the model has no procedures of handling outliers or external covariates. Therefore, it can be said that even though Theta truly excels in terms of computational efficiency, its forecasting accuracy can crumble in unusual situations.

4.2.3 Prophet Results

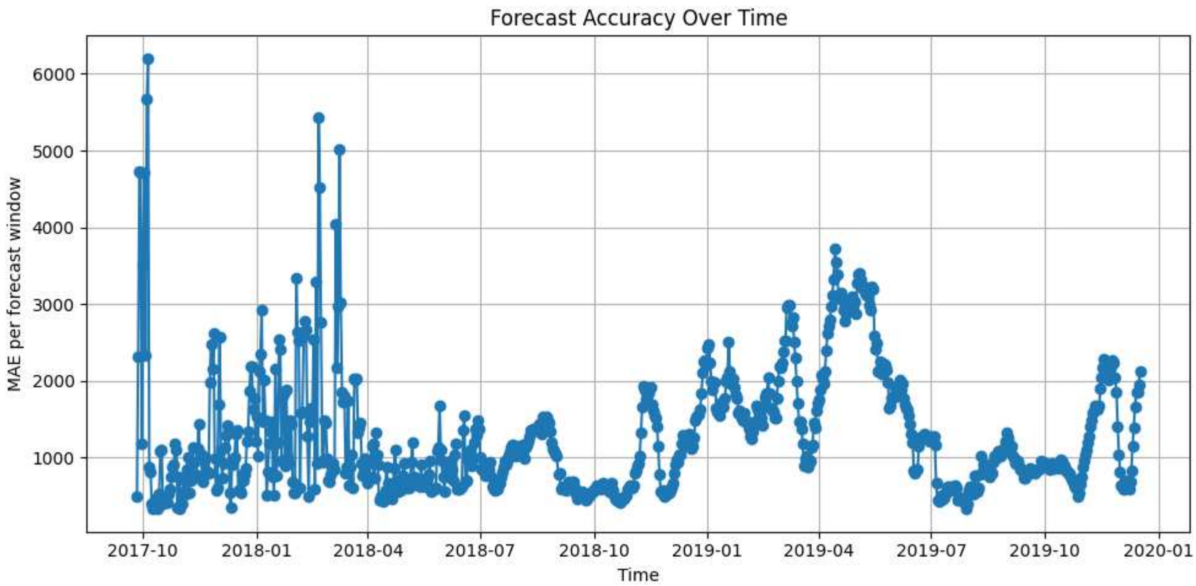

Figure 10: Prophet Forecasting Accuracy

Figure 10 displays the results of the forecasting accuracy of the Prophet model over the time series. On a first view it is interesting to see how different the forecasting pattern is in comparison to the other two models. Whereas both SARIMA and Theta started with a relatively low MAE and had significant spikes later on, Prophet had its biggest forecasting outliers in the beginning of the time series, especially in October 2017, where even MAEs over 6000 daily rides occurred, and March 2018. In that sense it can be assumed that the accuracy of the Prophet model improves over time, since it needs more data of the time series in order to properly adjust its trend and seasonality parameters. Nevertheless, it can be seen that overall Prophet manages to achieve a baseline of around 300 to 600 in stable seasons, which is pretty similar to the baseline of the SARIMA model, but without preprocessing issues and only a fraction of the needed computing time. Moreover, once the model manages to stabilize, it proves to forecast steady forecasts over time and is less fragile in the transitional months and the uncertainty in the winter months. Considering again the period starting from around January 2019, where both the SARIMA model and especially the Theta models had to deal with sudden upcoming spikes, even though the MAE also increased for Prophet, the shock was not as dramatic in comparison to the other two models. Furthermore, it manages to recover remarkably quickly to a low baseline of around 500 MAE. In conclusion, it can be said that even though the first forecasting results of the Prophet model are considerably inaccurate over time, its seasonality features prove to be useful.

5 Discussion

The obtained findings and insights from Chapter 4 can now build the base for a discussion regarding several important topics. First of all the trade-off between forecasting accuracy and computational efficiency will be discussed. Secondly it will be taken into consideration how robust and applicable the models are for automatic forecasting procedures. Moreover, the suitability for bike-sharing forecasting will be considered. It is also necessary to approach the limitations of this research, which could lead to further research possibilities in the future.

5.1 Trade-off between forecasting accuracy and computational efficiency

In regard to this experiment all three models showed its strengths and weaknesses. The results of the experiment show that even though Theta's forecasts computed almost instantly its forecasting accuracy was nonetheless considerably lower, which therefore proves the first hypothesis of this thesis as false. Therefore, it could be said that in Theta's case, its rapid forecasting generation comes at the cost of worse forecasting accuracy at the example of bike-sharing rental data. Regardless it has to be considered that the Theta model can be useful in use cases with less influence from external variables or where a higher error can be tolerated for the gain of faster forecasts. Using Theta to obtain a first approximate forecast might be a positive possibility as well.

On the contrary, SARIMA proved itself as reliable for accurate forecasts for a bike-sharing time-series. Nevertheless, it is necessary to address that the automatic optimization of its parameters were not feasible and as a result were necessary to determine manually. In addition to the troublesome preprocessing steps, the forecast generation in general takes more time into account in comparison. Therefore, it should be considered which use cases suit the use of a SARIMA model the best, since it might excel in scenarios where high accuracy is crucial, but speed is of secondary importance.

The Prophet model proves as balanced trade-off between the speed of Theta and the accuracy of SARIMA. As a matter of fact, after Prophet managed to learn from the initial winter shocks, its forecasting accuracy was comparable to the one of the SARIMA model and managed to deal with outliers and the heteroskedasticity of the errors remarkably well. This confirms the third hypothesis of this thesis, since Prophet managed to forecast trend changes and irregularities better than the other two models over time and even though Theta lacked in forecasting accuracy in comparison to Prophet and SARIMA, it can be argued that it succeeded in forecasting stable seasonal patterns. Nonetheless it also needs to be considered that the Prophet model is not fully automized either, since the user needs to specify seasonality aspects in the code manually.

5.2 Applications for automatic forecasting

The trade-off between forecasting accuracy and computational efficiency is a crucial aspect for automatic forecasting. First of all, it needs to be taken into consideration that the main goal of automatic forecasting is to use algorithms, which estimates the parameters and produces forecasts with as little human intervention as needed. Therefore, it can be concluded that the results show that ARIMA was not applicable for automatic forecasting, since the auto_arima package, which automatically chooses the parameters for the ARIMA model and consequently produces forecasts, could not run in a for automatic forecasting applicable time. This further confirms the second hypothesis of this thesis, which states that ARIMA will achieve the best forecasting accuracy, but will not be the most useful model for automatic forecasting due to its computational inefficiency. Regardless it has to be taken into account that this experiment has been conducted on a MacBook Pro 2020 (Intel i5, 8 GB RAM), which is in terms of computational power not the strongest. Therefore, it might be possible that even though the problem of ARIMA's efficiency and sustainability problem might still not be solved, with a stronger computer or the use of cloud-computing-services that the auto_arima package runs at a much faster time, which then might be useful for automatic forecasting.

Moving on to the Theta model, in theory it fulfills the requirements needed for automatic forecasting, since the forecasts are produced fully automatically without any necessary preprocessing or manual adjustments due to its mathematical simplicity. Regardless, it needs to be discussed on a case-to-case basis, whether the model is applicable. In most cases this will depend on the number of possible external variables and the stability of the observations of the time series. Furthermore, even though Theta manages to forecast stable seasonal patterns quite well, it still needs to be contemplated if possible higher accuracy deviations can be acceptable.

Lastly, the Prophet model's only manual adjustments are the settings of holidays and the daily, weekly and yearly seasonality, which makes it certainly applicable for automatic forecasting scenarios. Moreover, since Prophet automatically deals with outliers, it becomes an applicable tool for various real-life-cases, where stability and robustness in the time series might not always be guaranteed (Taylor and Letham, 2018). The only disadvantage of the model might be that it needs more observations to learn the patterns of the time series, else it might lead to drastic forecasting deviations in the beginning.

5.3 Suitability for daily demand bike-sharing forecasting

Lastly, the suitability of using Theta, SARIMA and Prophet for bike-sharing daily demand forecasting scenarios needs to be discussed. In general, it can be taken from the results of the experiment that considering the trade-off between forecasting accuracy and computational efficiency that the Prophet model seems as the best option. The main reasons for that is that the other two models simply were not able to find an appropriate balance of having a good forecasting accuracy and being computationally fast enough for being used in such a high-frequency forecasting environment. Even though SARIMA managed to forecast with high accuracy, which initially would make it suitable for bike-sharing forecasting, its inefficient computations are not sustainable for daily forecasting scenarios. Moreover, since bike-sharing forecasting can depend and fluctuate harshly dependent on various external variables, Theta struggles to deal with these outliers and uncertainties. This makes it as a result rather not suitable for the nature of demand forecasting of bike rides. On the contrary, Prophet manages to deal with this issues quite well and is able to stay computationally efficient at the same time.

5.4 Limitations

Even though further insights on daily demand forecasting in city bike-sharing environments have been gained from this thesis, there clearly have been limitations in the experiment, which simply were not feasible or went beyond the scope of this thesis. First of all, in order to provide a fair comparison environment it was necessary to implement an univariate forecasting experimental setup, which comes at the cost of only being able to producing forecasts based on one single aggregated daily demand time series. Consequently, additional variables had to be excluded, which as a result diminishes the possible forecasting accuracy. Furthermore, it has already been mentioned that because of computational reasons, instead of being able to use a fully automized ARIMA model, a simplified, manually configured SARIMA model had to be used. This may have lead to possibly not optimal parameter selection and as a result might not have fulfilled the whole potential of ARIMA.

It also needs to be said that in terms of testing the suitability of these models for automatic forecasting, only one dataset has been used in this thesis. Usually, the suitability of automatic forecasting models are validated on the base of a large number of time series, which would have gone beyond the scope of this experiment. Lastly, it is important to mention that even though this research was focused on bike-sharing demand in the city of San Francisco, it cannot be proved from this experiment, if these forecasts are making it possible to provide an actual impact on the rebalancing problem of bike-sharing companies. Also, station-based demand has not been taken into consideration, since demand has only been aggregated on the basis of overall city demand.

6 Conclusion

The aim of the thesis was to develop a rolling-origin forecasting setup, which provides the possibility to fairly compare the accuracy and computational efficiency of the forecasting models SARIMA, Prophet and Theta for automatic forecasting business cases. Since the models are inherently different, an univariate project approach has been used in order to assure the attempt of having a fair comparison. This project has been applied to a bike-sharing demand dataset from San Francisco. As the data was supposed to univariate the only two variables were time and demand, which have been aggregated to a daily level. Any other variables and external factors, such as temperature, weather or holidays have been purposefully excluded. The purpose of performing a rolling-origin forecasting setup, was to being able to see forecasting accuracy across different time periods, seasons and cycles. The metrics used in this project were MAE and RMSE for forecasting accuracy and average run time for computational efficiency.

The findings of the thesis showed that each model had its own strengths and weaknesses, which demonstrate the fundamental differences between the three models. Originally, it was planned to use the auto_arima package in Python to implement an automatic forecasting ARIMA model. This approach was not successful, since the code could not deliver results in an appropriate time frame. Nevertheless, the manual implementation of SARIMA led to having the best average forecasting accuracy, but it also took the most time to compute the forecasts. On the contrary, Theta had the fastest average run time by far, which shows its suitability for automatic and high frequency forecasting. Regardless, it was also visible that this computing speed comes at a cost, since it had worse overall forecasting accuracy than the other two models. Furthermore, by taking a deeper look into the rolling-origin forecasts, it could be concluded that both SARIMA and Theta struggled with dealing with outliers and external covariates. In terms of accuracy and computational efficiency Prophet was able to establish itself as the middle ground of the other two models, since it had clearly more accurate forecasts on average than Theta and computed at a more appropriate speed than SARIMA. The main difference between Prophet in comparison to the other models was that Prophet was able to handle outliers and external noise significantly better. Regardless, a further finding of Prophet was that Prophet's forecasts are highly inaccurate in the beginning and improve significantly over time.

The above mentioned findings lead to addressing the trade-off between forecasting accuracy and computational efficiency. Even though SARIMA might deliver qualitative forecasts, its number of necessary human intervention and additional run time makes it unsuitable for high frequency and automatic forecasting. Theta's speed might be useful in terms of speed but if its forecasts can be applied in real life would be dependent on the business case requirements and furthermore the kind of data, since it does not take seasonality nor external variables into consideration. Prophet, as a balanced option, proved itself to be suitable not only for automatic and high frequency forecasting, but also for bike-sharing demand forecasting, because of its adaptability for covariates and outliers.

Nevertheless, it has to be considered that this Thesis faced several limitations. Even though the univariate experiment structure was necessary for a fair comparison, a multivariate analysis would have been better for more applicable results in the bike-sharing sector and in addition might have led to different insights in general. Furthermore, since SARIMA had to be implemented manually it could not be guaranteed that the model has been fully optimized. Moreover, since automatic forecasting is usually tested on several hundred or thousand datasets whereas in this thesis only one dataset has been used for the forecasts.

To conclude, this thesis goal was to comprehend the usability of different forecasting models for bike-sharing companies, automatic forecasts and furthermore high frequency forecasts. This conducted research could lead to further development of an automated SARIMA preprocessing Python or R package. Furthermore, these models could be expanded to other data for comparison. Additionally, they could be expanded to station-based forecasts to conduct its usability for the bike-sharing system's rebalancing problem.

7 Appendices

```
def theta_forecast(train, horizon):
    df = pd.DataFrame({"unique_id": ["series_1"]* len(train),"ds":
       train.index, "y": train.values})
    sf = StatsForecast(models=[Theta()], freq="D")
    fcst  = sf.forecast(df=df, h=horizon)
    return fcst['Theta'].values
```

Listing 1: Theta model

```
def sarima_forecast(train, horizon):
    model = SARIMAX(train, order=(0,1,2),seasonal_order=(1,0,1,7),
       enforce_stationarity=False,enforce_invertibility=False)
    res = model.fit(disp=False)
    preds = res.forecast(steps=horizon)
    return np.array(preds)
```

Listing 2: SARIMA model

```
def prophet_forecast(train, horizon):
    df = pd.DataFrame({"ds": train.index, "y":train.values})
    model = Prophet(yearly_seasonality=True,
                    weekly_seasonality=True,
                    daily_seasonality=False)
    model.fit(df)
    future = model.make_future_dataframe(periods=horizon)
    forecast = model.predict(future)
    preds = forecast['yhat'].iloc[-horizon:].values
    return preds
```

Listing 3: Prophet model

```
def rolling_forecast(series, model_func, horizon=14, start=90):
    forecasts, actuals, times = [], [], []
    series = series.sort_index()
    for i in range(start, len(series) - horizon):
        train = series.iloc[:i]
        test = series.iloc[i:i + horizon]
        if len(train) < 10:
            continue

        start_time = pd.Timestamp.now()
        preds = model_func(train, horizon)
```

```
    runtime = (pd.Timestamp.now() - start_time).total_seconds()

    forecasts.extend(preds)
    actuals.extend(test.values)
    times.append(runtime)

mae = mean_absolute_error(actuals, forecasts)
rmse = np.sqrt(mean_squared_error(actuals, forecasts))
avg_time = np.mean(times)

return mae, rmse, avg_time
```

Listing 4: Rolling Forecast

References

V. Assimakopoulos and K. Nikolopoulos. The theta model: a decomposition approach to forecasting. *International journal of forecasting*, 16(4):521–530, 2000.

C. Bergmeir and J. M. Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, 2012.

G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

C. Chatfield. *Time-series forecasting*. Chapman and Hall/CRC, 2000.

D. A. Dickey and W. A. Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, 74(366a):427–431, 1979.

J. A. Fiorucci, T. R. Pellegrini, F. Louzada, F. Petropoulos, and A. B. Koehler. Models for optimising the theta method and their relationship to state space models. *International journal of forecasting*, 32(4):1151–1161, 2016.

J. A. Fioruci, T. R. Pellegrini, F. Louzada, and F. Petropoulos. The optimised theta method. *arXiv preprint arXiv:1503.03529*, 2015.

A. N. Grekov, E. V. Vyshkvarkova, and A. S. Mavrin. Forecasting and anomaly detection in bews: Comparative study of Θ, croston, and prophet algorithms. *Forecasting*, 6(2):343–356, 2024. doi: 10.3390/forecast6020019.

R. Hyndman, A. Koehler, K. Ord, and R. Snyder. *Forecasting with exponential smoothing: the state space approach*. Springer, 2008.

R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

R. J. Hyndman and Y. Khandakar. Automatic time series forecasting: the forecast package for r. *Journal of statistical software*, 27:1–22, 2008.

R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

H. Lin, Y. He, S. Li, and Y. Liu. Insights into travel pattern analysis and demand prediction: A data-driven approach in bike-sharing systems. *Journal of Transportation Engineering, Part A: Systems*, 150(2):04023132, 2024.

S. Makridakis and M. Hibon. The m3-competition: results, conclusions and implications. *International journal of forecasting*, 16(4):451–476, 2000.

S. Makridakis, E. Spiliotis, and V. Assimakopoulos. The m4 competition: Results, findings, conclusion and way forward. *International Journal of forecasting*, 34(4):802–808, 2018a.

S. Makridakis, E. Spiliotis, and V. Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3):e0194889, 2018b.

K. Nikolopoulos and V. Assimakopoulos. Theta intelligent forecasting information system. *Industrial Management & Data Systems*, 103(9):711–726, 2003.

K. I. Nikolopoulos and D. D. Thomakos. *Forecasting with the theta method: theory and applications*. John Wiley & Sons, 2019.

F. Petropoulos, Y. Grushka-Cockayne, E. Siemsen, and E. Spiliotis. Wielding occam's razor: Fast and frugal retail forecasting. *Journal of the Operational Research Society*, 76(8):1564–1583, 2025.

I. Sardar, M. A. Akbar, V. Leiva, A. Alsanad, and P. Mishra. Machine learning and automatic arima/prophet models-based forecasting of covid-19: Methodology, evaluation, and case study in saarc countries. *Stochastic Environmental Research and Risk Assessment*, 37(1): 345–359, 2023.

K. Sharma, R. Bhalla, and G. Ganesan. Time series forecasting using fb-prophet. In *ACM*, pages 59–65, 2022.

E. Spiliotis, V. Assimakopoulos, and S. Makridakis. Generalizing the theta method for automatic forecasting. *European Journal of Operational Research*, 284(2):550–558, 2020. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2020.01.007. URL `https://www.sciencedirect.com/science/article/pii/S0377221720300242`.

L. J. Tashman. Out-of-sample tests of forecasting accuracy: an analysis and review. *International journal of forecasting*, 16(4):437–450, 2000.

S. J. Taylor and B. Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

Y. Wang. Forecasting demand of shared bikes based on arima model. In *Proceedings of the 1st International Conference on Engineering Management, Information Technology and Intelligence – Volume 1: EMITI*, pages 659–664. SciTePress, 2024. ISBN 978-989-758-713-9. doi: 10.5220/0012961400004508.

C. Wirtgen, M. Kowald, J. Luderschmidt, and H. Hünemohr. Multivariate demand forecasting for rental bike systems based on an unobserved component model. electronics 2022, 11, 4146, 2022.

S. Yadav. A comparative study of arima, prophet and lstm for time series prediction. *Journal of Artificial Intelligence, Machine Learning and Data Science*, 1(1):1813–1816, 2022.