A Comparative Study of Univariate Time-series Methods for Sales Forecasting

by

Vishvesh Shah

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2019

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.
I understand that my thesis may be made electronically available to the public.

Abstract

Sales time-series forecasters, data scientists and managers often use time-series forecasting methods to predict sales. Nonetheless, it is still a question which time-series method a forecaster is best off using, if they only have time to generate one forecast. This study investigates and evaluates different sales time-series forecasting methods: multiplicative Holt-Winters (HW), additive HW, Seasonal Auto Regressive Integrated Moving Average (SARIMA) (A variant of Auto Regressive Integrated Moving Average (ARIMA)), Long Short-Term Memory Recurrent Neural Networks (LSTM) and the Prophet method by Facebook on thirty-two univariate sales time-series. The data used to forecast sales is taken from time-series Data Library (TSDL). With respect to the Root Mean Square Error (RMSE) evaluation metric, we find that forecasting sales with the SARIMA method offers the best performance, on average, relative to the other compared methods. To support the findings, both mathematical and economic reasoning on the drivers of the observed performance for each method are provided. However, a decision maker or an organization need to evaluate the trade-off between forecasting accuracy and the shortcomings associated with each method.

Acknowledgements

I would like to express my deep gratitude to my supervisor, Dr. Stanko Dimitrov for his continuous guidance, brilliant inspiration and valuable advice.

I am also grateful to Dr. Bon Koo and Dr. Brian Cozzarin for their time to review my thesis and for their insightful comments.

I want to thank the University of Waterloo, the Department of Management Sciences in particular, for giving me this wonderful opportunity.

I would like to thank my mentor and all my friends who have provided me with joy and warmth through these years.

Finally, my parents and my brother. Thank you for everything!!

Dedication

This thesis is dedicated to my parents and my brother who have always been there for me and encouraged me to pursue my dreams.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Forecasting sales is one of a multitude of actions a business must take to grow (Rao, 1985). Echoing this result, the Aberdeen Group (June 2011) shows that accurate sales forecasts help companies grow the total company revenue by 10% year-over-year, and teams are 7.3% more likely to hit overall attainment of quota (Ostrow, 2013). In addition, no sales forecasts lead to companies shrinking or ceasing operations (mindykim, 2019).

Time-series forecasting is of great importance not only in industry, but also academia. In fact, according to Adhikari and Agrawal (2013) "a lot of efforts have been done by researchers over many years for the development of efficient models to improve the forecasting accuracy" of time-series. However, sometimes sales time-series forecasting methods are used without fully understanding the limitations and applicability of each method (Winters, 1960). Using forecasts from a method that does not apply to the given time-series may lead to inaccurate predictions. Hence, a forecaster has to be careful in choosing a forecasting method suitable to the situation at hand (Purthan et al., 2014). In this dissertation we help sales time-series forecasters, data scientists and managers, determine which forecasting method to use. We help forecasters by testing the accuracy of the four most-common methods and one new method, all from the literature, for sales forecasting on thirty-two different sales time-series.

We use univariate time-series methods to forecast sales. The methods we consider, defined in greater detail in section 3, are Seasonal Auto-Regressive Integrated Moving Average method (SARIMA) (Box and Jenkins, 1990), Holt-Winters Exponential Smoothing (HW) method with additive and multiplicative seasonality (Holt, 2004, Winters, 1960), Long Short-Term Memory networks (LSTM) (Hochreiter and Schmidhuber, 1997); a type of Recursive Neural Network (RNN) and the Prophet method by Facebook (Taylor and Letham, 2018). We select these methods based on their ability to model trend and seasonality present in our data.

Our results suggest a sales time-series forecaster is best off using SARIMA to forecast future sales. We find a similar outcome on the sales data of an industry partner that motivated this study.

The contributions of this dissertation are:

1. We compare the four most-common sales time-series forecasting methods and one new sales time-series forecasting method on thirty-two univariate sales time-series.

2. We provide mathematical and economic reasoning on the drivers of the observed performance for each method.

In the remainder of this dissertation, we first discuss related work in section 2. Next, we describe the methods in section 3 and data in section 4. We present our analysis and results for each method in section 5. We combine and discuss the result for all methods in section 6. We conclude the dissertation in section 7.

Chapter 2

Related Work

Over the last few decades, time-series forecasting is performed using the established and commonly used HW and ARIMA methods. Conversely, there are two new methods that are growing in popularity also used to forecast time-series data, namely LSTM and Prophet. Hence we use two forms of HW, SARIMA (a variant of ARIMA), LSTM, and Prophet method in our study. All of the listed methods can identify complex relationships among time-series data along with trend and seasonality.

In our dissertation, we compare all five methods to one another. To our knowledge, no such comparison exists in the literature. However, two streams of literature compare a subset of the five methods. One stream compares the HW, SARIMA, and Artificial Neural Network (ANN) methods (a form of regression) for time-series prediction and another stream compares LSTM, ANN, SARIMA, and Prophet methods for time-series prediction. Not all papers in each of the streams use sales time-series and not all papers compare all of the methods to one another. As the literature on time-series forecasting spans multiple disciplines and decades, the list below is not exhaustive and there may be papers omitted due to fit with our work. Finally, to our knowledge there is no other that compares all five methods simultaneously. We now present and discuss the two stream of literature in turn and at the end of presenting each stream we compare our work to the entire stream.

2.1 Comparison of HW, SARIMA, and ANN time-series methods

Cranage and Andrew (1992) uses 79 months of restaurant sales data to compare an econometric method with time-series methods (SARIMA and HW). Their results show that the SARIMA method performs better in both the initial period and in the seven month forecasting period. The authors highlight that restaurant operators with limited time and skills are best to consider HW but may improve accuracy using the SARIMA method. However, the study was only specific to one restaurant.

Purthan et al. (2014) forecast Indian motorcycle sales using HW method and SARIMA method. Using the Least Mean Square Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) as evaluation measures, authors conclude that the HW method is more precise and accurate than SARIMA method.

Makatjane and Moroke (2016) show that HW method has greater predictive power in predicting sales relative to SARIMA method using 19 years of monthly car sales in South Africa. The authors conclude this on basis of a power test.

Udom (2014) forecast the sales for five different products of a single distributor in the plastic industry using the Moving Average method (averaging observations with normalized weights summing to one put on each observation), HW method, and SARIMA method. Using MAPE as the evaluation measure, the authors find that the SARIMA method forecast sales best.

Frank et al. (2003) employ single seasonal exponential smoothing (SSES) (a form of HW without seasonality or trend), the HW, and ANNs methods for women's apparel sales forecasting. They find that ANNs show satisfactory goodness-of-fit statistics (R^2), but the HW method performs better in comparing actual sales vs the forecasted sales.

Chu and Zhang (2003) forecast out-of-sample retail sales using both linear and nonlinear neural networks. According to their result, nonlinear neural networks are preferred over linear neural networks. The authors state: "deseasonalized data (nn-deseason) perform the best overall, while ARIMA and neural networks using the original data (nn-direct) perform about the same." Note that ARIMA is SARIMA without the seasonality components.

Alon et al. (2001) using ANN, HW, SARIMA method, and multivariate regression compare out-of-sample forecasts of monthly aggregated retail sales. The researchers conclude that for data without seasonality, HW and SARIMA methods perform well. However, for data with trend and seasonality, ANN outperforms all the other methods. On the other hand, Zhang and Qi (2005) find that ANNs do not handle seasonal patterns in data very well.

Even though all the above literature uses sales data for forecasting, the results are not consistent with one another. Using one time-series, one method outperforms other methods, while using another time-series, the once best performing method is no longer best, all using the same evaluation metric. Hence, the work of the first stream of literature is insufficient to address the topics addressed in our dissertation, i.e., what method should a manager or a data scientist use to forecast sales time-series. We do not use ANNs in our study as ANNs have no memory of previous time-steps for sequential data. We also do not use RNNs in our study due to RNNs time-dimension susceptibility to the vanishing or exploding gradient problem, i.e., RNNs are not able to capture the features of time-series within a longer time-frame. Instead we use a specialized RNN layer called Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). Similar to some papers in the first stream, we use HW and SARIMA methods in our dissertation.

2.2 Comparison of LSTM, ANN, SARIMA and Prophet time-series methods

The results of comparing ANN and LSTM to SARIMA and the Prophet are thus far, somewhat mixed.

Yu et al. (2018) use the LSTM method to forecast sales. They analyze 66 products consisting of 45 weeks of data. The data has little to no seasonality. Their results show that LSTM provides accurate predictions for 17 out of 66 products. This may be due to the author not considering seasonality in their LSTM network, considering only one LSTM network for all 66 products and insufficient time-series length.

Hu et al. (2018) propose a data driven method that analyzes relations between precipitation and runoff time-series for flood forecasting. The experimental data is from 98 rainfall-runoff events where almost 88% of the data is used for training and the remainder for testing. The authors conclude that LSTM and ANN are better than conceptual and physical based methods, found in the environmental sciences literature. In addition, the results of R^2 and Nash-Sutcliffe Efficiency (NSE) (Nash and Sutcliffe, 1970) show that LSTM method is more stable than ANN method.

Samal et al. (2019) proposes two approaches for pollution forecasting based on the historical data ranging from 2005 to 2015 for the city of Bhubaneswar, India. Their result based on RMSE and MSE shows that both the SARIMA and the Prophet methods provides a good quality of accuracy but the Prophet method on log transformation of the original data is the most accurate method.

Weytjens et al. (2019) compare SARIMA and Prophet to ANN and LSTM methods to predict the cash flow. In their work, they introduce a new performance measure i.e. Interest Opportunity Cost (IOC). Using IOC and MSE as their cost functions, they conclude that LSTM is the best method for forecasting cash flow.

The work of the second stream of literature is insufficient to address the questions of interest our dissertation as no paper compares all five methods, and no paper uses the breadth of time-series we use in our dissertation. Similar to some papers in the second stream, we use LSTM and Prophet in our dissertation as LSTM allows us to learn long-term temporal dependencies as well as read and write information from previous time-steps unlike ANN, RNN, or Fuzzy time-series methods, which we do not consider in our dissertation. For the most studies, the important evaluation measure, RMSE, is never considered. We use the RMSE as the evaluation metric and is discussed in more detail in 5.

Chapter 3

Method Selection

Time-Series methods employed in this study are the Multiplicative and Additive Holt-Winters method, SARIMA method, LSTM (RNN) method, and the Prophet method by Facebook. Each method may have an instance, we refer to as a *model* in this study. This section presents a brief overview of these methods.

3.1 Holt-Winters Exponential Smoothing method

Holt-Winters Exponential Smoothing (HW) method is one of the simplest and most widely used method in industry for sales data with seasonal patterns and trends. It comprises of the forecast equation and three smoothing equations (defined in greater detail below): one for the level, one of a trend component, and one of a seasonal index, each with a corresponding smoothing parameters: α, β, γ, respectively.

The two main variations of the HW method differ in the nature of the seasonal equation. One method uses an additive seasonal equation (called additive HW) and the other a multiplicative seasonal equation (called multiplicative HW). The additive method is preferred, when the seasonal variations are of the same magnitude throughout the data set, while the multiplicative method is preferred when the magnitude of seasonal variations changes with time. The multiplicative HW method is not applicable if the time-series has null or negative values. Due to this restriction, the multiplicative HW method cannot be applied to sales data with zero sales, something we encounter in one time-series of this study. We apply both variants of the HW method in this dissertation. We next present the parameterized mathematical equations of the HW method that are fit on a given time-series (Hyndman, 2010).

Suppose the time-series, X, where $X = x_1, x_2, ...x_t$ for each time period t, S is the seasonal period (e.g., $S = 12$ for monthly data). Let $\hat{y}_{t+m|t}$ represent the forecast of the time-series value m-steps ahead of the observed value at time t.

Then basic equations for multiplicative HW method are as follows:

$$Level : l_t = \alpha(x_t/s_{t-S}) + (1-\alpha)(l_{t-1} + b_{t-1}), \tag{3.1}$$

$$Trend : b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}, \tag{3.2}$$

$$Seasonal : s_t = \gamma(x_t/(l_{t-1} + b_{t-1})) + (1-\gamma)s_{t-S}, \tag{3.3}$$

$$Forecast : \hat{y}_{t+m|t} = (l_t + b_t m)s_{t+m-S}. \tag{3.4}$$

Above, l_t is the level of sales at time t, b_t is the trend of sales at time t, s_t is the seasonal component at time point t, s_{t+m-S} is the seasonal component given seasonality S and step m. The parameters estimated in the HW method, α, β, and γ are restricted to lie between 0 and 1.

Similarly, the basic equations for the additive HW method are as follows:

$$Level : l_t = \alpha(x_t - s_{t-S}) + (1-\alpha)(l_{t-1} + b_{t-1}), \tag{3.5}$$

$$Trend : b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}, \tag{3.6}$$

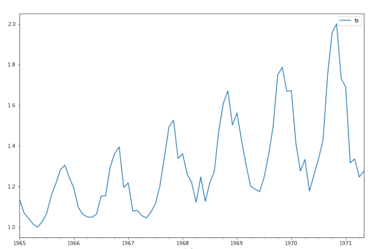
$$Seasonal : s_t = \gamma(x_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-S}, \tag{3.7}$$

$$Forecast : \hat{y}_{t+m|t} = l_t + b_t m + s_{t+m-S}. \tag{3.8}$$

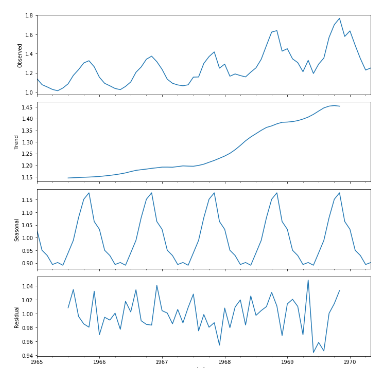
The trend equation remains the same as the multiplicative method, but the other equations replace seasonal products and ratios with additions and subtractions, respectively. The level equation is the weighted average of the seasonally adjusted observation, first term on the right-hand side of (3.5), and non-seasonal forecast; the seasonal equation, (3.7), is the weighted average of the current seasonal index and the seasonal index of the same season the previous season.

We next present how the multiplicative HW method is used to forecast sales of company X (Chatfield and Prothero, 1973). We also present how the additive HW method is used to forecast monthly sales of carpet (Montgomery and Johnson, 1976). We use statsmodel (Seabold and Perktold, 2010) in Python for both HW methods. For a time-series, *level* is the average value of the time-series, *trend* is the tendency of the time-series to increase or decrease over time (used to determine y_{t+1} from x_t), *seasonality* is the cyclical patterns of fixed frequency in the series (used to determine y_{t+S} from x_t).

Below figure 3.1a shows the time-series of sales of company X, Jan. 1965 to May 1971 (Chatfield and Prothero, 1973). Its decomposition plot is shown in figure 3.1b below. We can see from both the plots that data has consistent upward trend, exhibit a seasonal pattern and the amplitude of the seasonal cycle increases over time. Hence the time-series may be predicted using the multiplicative HW method.



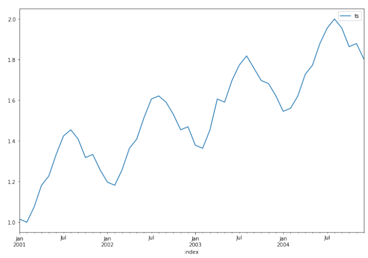
(a) time-series of monthly sales of company X



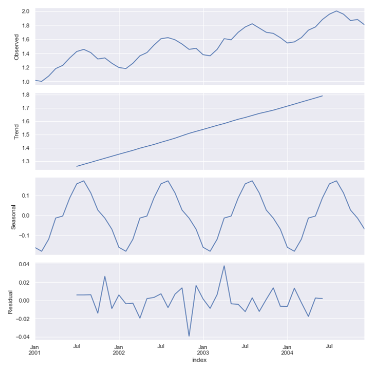
(b) Decomposition plot of monthly sales of company X

Figure 3.1: Time-series plot and decomposition plot of company X

Below figure 3.2a shows the time-series of monthly demand of carpet (Montgomery and Johnson, 1976). Its decomposition plot is shown in figure 3.2b below. We can see from both the plots that data has consistent upward trend, exhibit a seasonal pattern and the amplitude of the seasonal cycle remains constant over time. Hence the time-series may be predicted using the additive HW method.



(a) Time-series of monthly sales of carpet



(b) Decomposition plot of monthly sales of carpet

Figure 3.2: Time-series plot and decomposition plot of sales of carpet

3.2 Seasonal Auto Regressive Integrated Moving Average method

Seasonal Auto Regressive Integrated Moving Average (SARIMA) is a modification of the Box-Jenkins method that proposed Auto Regressive Integrated Moving Average (ARIMA) (Box and Jenkins, 1990). Before we discuss SARIMA, we will briefly discuss ARIMA. ARIMA uses three principles: the Auto Regression (AR), Moving Average (MA), and an integrated term (I). In notation, $ARIMA(p, d, q)$ comes from three parts where p is the autoregressive order which allows the method to incorporate past values in forecasting future values, d is number of nonseasonal differences needed for stationarity, and q is the moving average order which relies on number of lagged forecast errors for obtaining the forecast values.

When determining whether to use ARIMA or SARIMA, we must look at the time-series seasonality, as defined in section 3.1. The ARIMA method is used when the time-series does not show any seasonal patterns while the SARIMA method is used otherwise. To apply ARIMA method one must first determine the differencing order, d so that the method may be applied to a stationary time-series. With SARIMA method, in addition to the differencing order, d to make the resulting time-series stationary, we must also conduct seasonal differencing, D, to account for time-series seasonality. This additional seasonal differencing is the main difference between ARIMA and

SARIMA.

The SARIMA method includes both nonseasonal and seasonal factors in a multiplicative method and is expressed as $ARIMA(p,d,q)(P,D,Q)$, where p is nonseasonal AR order, d is nonseasonal differencing, q is nonseasonal MA order, P is seasonal AR order, D is seasonal differencing, Q is seasonal MA order, and S is number of periods per season (e.g. S is 12 for monthly data).

Thus a multiplicative SARIMA method is obtained which has the form (Shumway and Stoffer, 2010):

$$\Phi_p(B^s)\phi_p(B)\nabla_s^D\nabla^d x_t = \Theta_Q(B^s)\theta_q(B)\epsilon_t \tag{3.9}$$

Where the time-series, X, where $X = x_1, x_2, ...x_t$ for each time period t, ϵ_t is Gaussian white noise process, B is a backward shift operator which means "shift by one time unit" and is defined by $B^k x_t = x_{t-k}$.
The operator polynomials are

$$\text{Ordinary autoregressive} : \phi_p(B) = (1 - \phi_1 B - ... - \phi_p B^p) \tag{3.10}$$

$$\text{Ordinary moving average} : \theta_p(B) = (1 - \theta_1 B - ... - \theta_p B^p) \tag{3.11}$$

$$\text{Ordinary difference} : \nabla^d = (1 - B)^d \tag{3.12}$$

$$\text{Seasonal autoregressive} : \Phi_p(B) = (1 - \Phi_1 B - ... - \Phi_p B^p) \tag{3.13}$$

$$\text{Seasonal moving average} : \Theta_p(B) = (1 - \Theta_1 B - ... - \Theta_p B^p) \tag{3.14}$$

$$\text{Seasonal difference} : \nabla_s^D = (1 - B^s)^D \tag{3.15}$$

Box-Jenkins method is an iterative approach that consists of the following 3 steps:

1. Address the time-series stationarity and seasonality, if necessary, then determine autoregressive and moving average components of the time-series (Model identification)

2. Fit selected model to data (Estimation and Evaluation)

3. Diagnostic test to check for autocorrelation in the model, i.e., the error term is independent of the estimates

Any model that passes through the three steps above may then be used for forecasting the fitted time-series. In the remainder of the section we discuss each one of the three steps above in greater detail.

We must first determine if the time-series is stationary or not. To test the stationarity of the time-series, we use the Augmented Dickey–Fuller (ADF) test (Dickey and Fuller, 1979). The null hypothesis of the ADF test is that there is a unit root in the time-series. As indicated in our analysis, most of the time-series result in an ADF test that fails to reject the null hypothesis.

We now move to step 2 where we fit a SARIMA or ARIMA method, depending on the existence of seasonality. As a sub-step we must first make the time-series stationary. The time-series we consider may be non-stationary due to either trend or seasonality or both. We address both by differencing observations, but trend may also be addressed by taking the log transform of the time-series. After transforming the time-series we run the ADF test on the transformed time-series to confirm it is stationary. We now move to determining the lags to use in forecasting the time-series, p, q, P and Q. We use the Auto-Correlation Function (ACF) and Partial Auto-Correlation Function (PACF) plots to help determine p, q, P and Q. On basis of ACF and PACF plots and the suggested rules (Nau, 2014), we identify and determine the parameters of the SARIMA or ARIMA method. Next, we use Akaike's information criterion (AIC) (Akaike, 1998) to select most parsimonious method for forecasting. AIC is an estimator of the relative quality of statistical methods for a given time-series and is given by (Akaike, 1998):

$$AIC = 2k - 2ln(\hat{L}), \tag{3.16}$$

where k is the number of estimated parameters in the method and L be the maximum value of the likelihood function for the method. Moreover, we use automatic method selection method, $auto\_arima()$ in python that performs grid search to give best combination of SARIMA method on the basis of AIC.

Next, we fit the method and conduct a diagnosis test through LjungeBox Q-statistics (Ljung and Box, 1978). As a part of a diagnostic check, we also examine the autocorrelation function of the residual to check for overfitting by the method. Once the SARIMA method is statistically appropriate, i.e., p-value of LjungeBox Q-statistics is larger than the 0.05 and residuals are both random and approximately normal, we perform out-of-sample forecasting and measure the RMSE forecasting accuracy.

3.3 LSTM Neural Network method

LSTM stands for Long Short-Term Memory Recurrent Neural Network and are used for time-series forecasting. LSTMs are capable of learning long-term dependencies by creating special type of structures called memory cells and gate units. LSTMs address the problem of insufficient, decaying error back-flow in Vanilla Recurrent Neural Networks (RNNs), i.e., RNNs are not able to capture the features of time-series within a longer time horizon (Hochreiter and Schmidhuber, 1997). LSTMs are applied to forecast time-series.

Figure 3.3 shows an LSTM memory cell. We can see from figure 3.3 that the previous output, h_{t-1} and C_{t-1}, is processed together along with the current input, x_t through all three steps of a memory cell: forget, input, and output. The forget gate defines what information is removed from cell state while input gate and output gate is used specify what information is added and used respectively from the cell state (Hu et al., 2018).

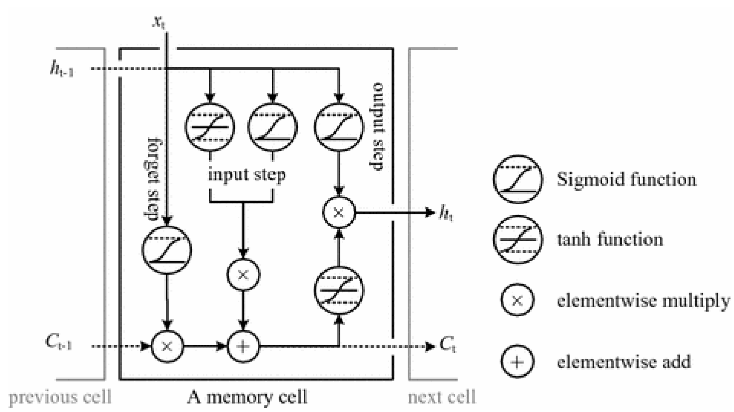


Figure 3.3: LSTM memory cell reproduced from Yu et al. (2018)

In figure 3.3 we use the notation by Yu et al. (2018) and denote the time series as $X = x_1, x_2, ...x_t$ and only x_t is the input at each period t, h_t is the output with respect to x_t, h_{t-1} is the output of the previous memory cell (h_{t-1} may also be referred to as a hidden state), C_t is the cell state while C_{t-1} is the cell state of the previous cell.

Figure 3.4 shows an LSTM network with three memory cells for time series forecasting. Here three consecutive time-steps are used to produce the fourth time-step as an output.

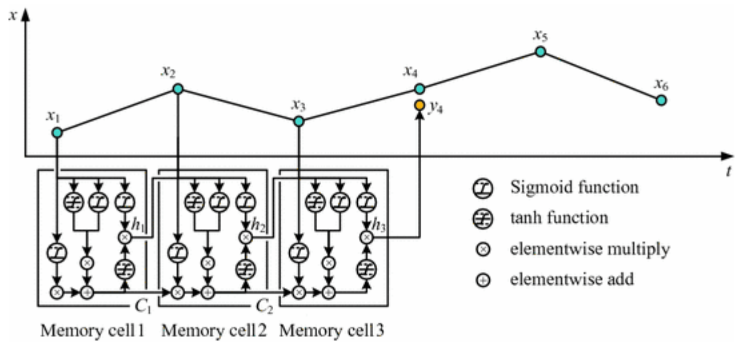


Figure 3.4: LSTM network with three memory cells for time series forecasting reproduced from Yu et al. (2018)

We next present how LSTM method is used in this study. We implement LSTM methods using the python library Keras Chollet et al. (2015). We define a sequential model with 50 to 200 LSTM memory cells (depending on the size of data and efficiency of model) in the hidden layer and an output layer for prediction. The model is fit using the *Adam* version of stochastic gradient descent, defined by Kingma and Ba (2014), and model parameters are determined relative to the mean squared error (mse) and root mean squared error (rmse) loss functions. We then perform out-of-sample forecasting and measure the RMSE forecasting accuracy.

3.4 Prophet method

nProphet is an open-source forecasting tool published by Facebook's core data science team and is available in Python (Van Rossum and Drake Jr, 1995), and R (R Core Team, 2013). Prophet is developed for typical Facebook issues such as predicting user activities. This makes the Prophet method convenient for predicting seasonalities, special events, data with holidays, data showing outliers and data with varying trend.

The Prophet method uses a framework called "Analyst-in-the-Loop" as shown in figure 3.5 below.

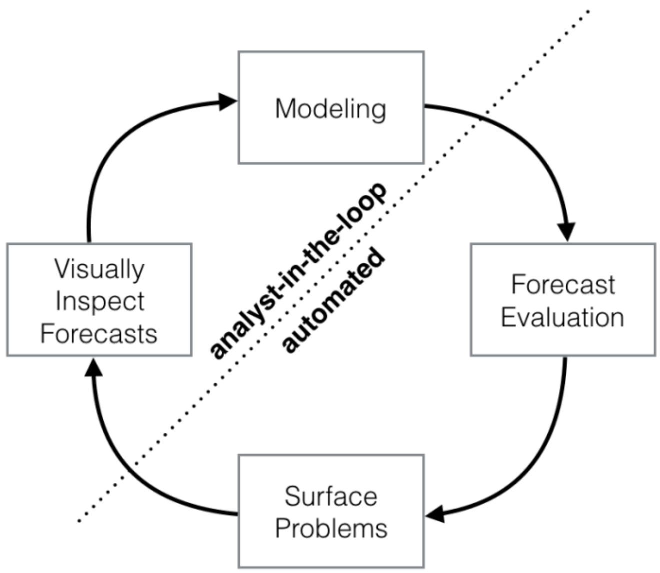


Figure 3.5: Analyst-in-the-Loop Modeling reproduced from Taylor and Letham (2018)

The framework is double-sided where on one side model fitting is automated assuming that the user has no statistical knowledge, while on other side the framework allows the same user to input information based on their domain/industry knowledge.

The Prophet procedure is an additive regression method which belongs to the Generalized Additive Model (GAM) family with the following components and functional form:

$$y(t) = b(t) + s(t) + f(t) + \epsilon_t. \tag{3.17}$$

In equation (3.17), $b(t)$ captures trend in the time-series, $s(t)$ captures time-series seasonality, $f(t)$ captures holidays or special events in the time-series and ϵ_t is an irreducible error term. In any instance of the Prophet method, only ϵ_t is always present, the remaining three terms may not always be present, as they have to be provided by the user. Now, we will explain each component in greater detail.

$b(t)$ models non-periodic changes (trend) in the time-series. The Prophet library implements piece-wise linear trend model where the growth rate remains constant and nonlinear trend model where growth rate decreases with time t. In the current implementation of Prophet, growth rate cannot increase with time.

$s(t)$ Seasonality represents periodic changes (daily/weekly/monthly/yearly seasonality) in the time-series.

$f(t)$ Holidays component contributes information about holidays and events as is provided by the user.

The Prophet method uses a curve fitting technique for the time-series fit. The method is fitted automatically using Stan code (Carpenter et al., 2017) that takes seasonality, trends, and holidays into account. Prophet's robustness, ease of configuration and being fast to fit attracts non-experts and users with limited statistical knowledge to deploy Prophet within their organization.

Chapter 4

Data

The study compares the five prediction method, additive HW, multiplicative HW, SARIMA, LSTM, and Facebook's Prophet on 32 different sales time-series. The time-series comes from the time-series Data Library (TSDL) (Hyndman and Yangzhuoran, 2019). TSDL is continually evolving and new time-series are added, for this document we use most recent version of the library as of September 14, 2019. In total, TSDL has about eight hundred time-series and the library is created and managed by Rob Hyndman, Professor of Statistics at Monash University, Australia. These time-series comes from different areas and are widely used in the applied statistics literature. There are a total of forty-six sales time-series in TSDL that contain both univariate and multivariate time-series. The dataset contains daily, weekly, monthly, quarterly and yearly time-series. We use thirty two univariate time-series for our analysis, as this project is motivated by a project conducted by an industry collaborator interested in forecasting a univariate sales time-series. Due to privacy concerns from our industry collaborator, we do not use any of their data in our study. We consider data of each granularity for our study, i.e., daily, weekly, monthly, quarterly and annually. We use 1 daily, 7 weekly, 20 monthly, 3 quarterly and 1 yearly time-series for forecasting. We normalize each time-series to fall within the range of 1-2, further details are provided below in the preprocessing steps. The two reasons we choose TSDL are as follows:

1. It provides freely accessible, high-quality data.

2. Many time-series data exhibits variable behavior including but not limited to trends and seasonal patterns.

Readers can refer to the TSDL package (Hyndman and Yangzhuoran, 2019) for more information about the time-series. A complete list of the transformed time-series, their sources, their descriptions and some necessary information about each is provided in Table 4.1.

Table 4.1: Summary of thirty-two transformed sales time-series data from Time Series Data Library (TSDL)

Time-series	Mean	Granularity	No. of Observations	Stationary/ Non-stationary	Source	Description
1	1.444741	Monthly	192	Non-Stationary	Abraham and Ledolter (1983)	Monthly gasoline demand Ontario gallon millions 1960-1975
2	1.456246	Monthly	132	Non-Stationary	Abraham and Ledolter (1983)	Monthly sales of U.S. houses (thousands) 1965-1975
3	1.439682	Monthly	108	Non-Stationary	Abraham and Ledolter (1983)	Monthly car sales in Quebec 1960-1968
6	1.312274	Monthly	36	Non-Stationary	Bowerman et al. (1993)	Monthly sales of Tasty Cola
10	1.09596	Monthly	36	Non-Stationary	Makridakis et al. (1998)	Sales of product C
11	1.223384	Daily	1067	Stationary	John C Nash	Daily net retail sales. 5 May 2000-6 April 2003
12	1.364915	Weekly	104	Non-Stationary	Makridakis and Wheelwright (1989)	Der Stern: Weekly sales of wholesalers 71-72
13	1.415841	Weekly	104	Non-Stationary	Makridakis and Wheelwright (1989)	Der Stern: Weekly sales of wholesalers A, 71-72
14	1.386321	Weekly	104	Non-Stationary	Makridakis and Wheelwright (1989)	Der Stern: Weekly sales of wholesalers A, 71-72
15	1.383766	Weekly	104	Non-Stationary	Makridakis and Wheelwright (1989)	Der Stern: Weekly sales of wholesalers B, 71-72
16	1.292798	Weekly	104	Non-Stationary	Makridakis and Wheelwright (1989)	Der Stern: Weekly sales of wholesalers B, 71-72
17	1.267788	Monthly	105	Non-Stationary	Makridakis and Wheelwright (1989)	Perrin Freres monthly champagne sales millions 71-72
18	1.511335	Monthly	147	Non-Stationary	Makridakis and Wheelwright (1989)	CFE specialty writing papers monthly sales
22	1.122828	Monthly	84	Non-Stationary	Makridakis et al. (1998)	Monthly sales for a souvenir shop on the wharf at a beach resort town in Queensland, Australia. Jan 1987-Dec 1993
23	1.40489	Quarterly	24	Non-Stationary	Makridakis et al. (1998)	Quarterly reports of a French company
24	1.435189	Monthly	275	Stationary	Makridakis et al. (1998)	Monthly sales of new one-family houses sold in the USA since 1973
25	1.516926	Monthly	107	Non-Stationary	Makridakis et al. (1998)	Sales of new one-family houses, USA, from Jan 1987 through Nov 1995
28	1.343522	Monthly	36	Non-Stationary	Makridakis et al. (1998)	Sales of shampoo over a three year period
29	1.495071	Monthly	60	Stationary	Makridakis et al. (1998)	Monthly sales of product A for a plastics manufacturer
30	1.629764	Monthly	120	Non-Stationary	Makridakis et al. (1998)	Industry sales for printing and writing paper (in Thousands of french francs) January 1963-December 1972
31	1.525568	Monthly	48	Non-Stationary	Montgomery and Johnson (1976)	Monthly demand for carpet (p.272: Montgomery: Fore. & T.S.)
32	1.261598	Monthly	96	Non-Stationary	Montgomery and Johnson (1976)	Monthly champagne sales (in 1000's) (p.273: Montgomery: Fore. & T.S.)
34	1.394231	Monthly	65	Stationary	Montgomery and Johnson (1976)	Weekly sales for a novelty item (p.37-38: Montgomery)
35	1.522088	Weekly	100	Non-Stationary	Montgomery and Johnson (1976)	Weekly demand for a plastic container (Montgomery & Johnson)
37	1.588659	Weekly	100	Non-Stationary	Montgomery and Johnson (1976)	Weekly sales of a cutting tool (p.270, Montgomery: Fore. & T.S.)
39	1.472862	Monthly	48	Non-Stationary	Montgomery and Johnson (1976)	Monthly sales of soft drink (hundreds of cases) (p.272: Montgomery)
40	1.307122	Monthly	77	Non-Stationary	Chatfield and Prothero (1973)	Monthly sales of company X Jan 65 - May 71
41	1.438964	Monthly	178	Non-Stationary	O'Donovan (1983)	Monthly empolyees wholes/retail Wisconsin 61-75 R.B.Miller
43	1.245656	Monthly	64	Non-Stationary	Roberts (1992)	Monthly unit sales, Winnebago Industries, Nov. 1966 - Feb. 1972
44	1.450525	Quarterly	28	Non-Stationary	Roberts (1992)	Quarterly sales of SPSS manual, second edition, Jan. 1976 through April 1982
45	1.33656	Annualy	108	Non-Stationary	Roberts (1992)	Annual domestic sales and advertising of Lydia E, Pinkham Medicine, 1907 to 1960
46	1.377261	Quarterly	39	Non-Stationary	NA	Quarterly retail turnover: $m current. Jun 1982 - Dec 1991

Most of the time-series in Table 4.1 are non-stationary. In addition, of the non-stationary time-series some show only seasonality, some exhibit only a trend, while some possess both seasonality and trend. Time-series that we find, therefore, require pre-processing. The following steps are implemented as a part of preprocessing:

1. Impute missing sales observations in the time-series using linear interpolation method whenever needed.

2. We then use the Min-Max scaling to normalize sales data. This normalization can be formally defined as follows:

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}} + 1 \tag{4.1}$$

   Here, X_{sc} represent the normalized sales data. As a consequence of normalization, data ranges from 1-2. Please note that the addition of 1 in (4.1) is our variant of the algorithm and we add 1 in order to apply multiplicative version of the HW method to the time-series.

We split the time-series data into two sets for cross-validation purposes: training and testing. The training data was the first $80 - 85\%$ of the time-series data for each time-series. The testing data was the remaining $15 - 20\%$ of the time-series data. For example, if a time-series consists of 100 days, numbered 1 through 100, then the training data of the first 80% is days 1 through 80, inclusive, and the testing data is days 81 through 100, inclusive. We consider only one one-step ahead forecasting for our study, i.e., we fit our model only once. We use pandas (Mckinney, 2010), numpy (Oliphant, 2006), matplotlib (Hunter, 2007), statsmodels (Seabold and Perktold, 2010), scikit-learn (Pedregosa et al., 2011), keras (Chollet et al., 2015), seaborn (Virtanen et al., 2019), and fbprophet (Taylor and Letham, 2019) to implement the models in the Python environment (Van Rossum and Drake Jr, 1995). Analysis is done in Jupyter Notebook (Kluyver et al., 2016).

Chapter 5

Results

The following section presents how sales are predicted by each model.

5.1 Performance Criteria

We use Root Mean Square Error (RMSE) as the evaluation measure. RMSE can be defined as follows:

$$RMSE = \sqrt{\frac{\Sigma_{i=1}^{n}(\hat{y}_t - x_t)^2}{m}} \tag{5.1}$$

where x_t is the actual value at time t, $\hat{y}_t$ is the prediction value at time t and m is the number of time steps. Lower RMSE value implies better accuracy. We use RMSE as the evaluation metric in our study for the following reasons:

1. RMSE punishes large errors more than MAE and MAPE, two commonly used evaluation metrics in the literature. For this reason our industry partner requested we use RMSE, as large errors are especially unacceptable in their setting.

2. RMSE avoids using absolute value unlike MAE. It is highly undesirable, from a computation perspective, to use absolute value in measuring model error sensitivities or for data assimilation applications (Chai and Draxler, 2014).

3. RMSE is derived from the sample average, while MAE is derived from the sample median. If data is skewed, then the median falls below the

average which is not be acceptable, at least to our industry partner. In addition, MAE results in poor forecasts for intermittent data (data with many periods of no demand) (Vandeput, 2018).

4. MAE is less sensitive to long tailed distributions, due to approximating the median. This is an undesirable property for our study, as our industry partner is concerned with extreme values of the distribution.

5.2 Time-Series Fit Results

In order to compare the five-time-series forecasting methods, we fit each of method to each of the thirty-two sales time-series. After preprocessing the time-series (as described in section 4), we fit each of the time-series forecasting methods using the training data and measure the fitted model on the testing data to determine each methods forecasting accuracy, measured using RMSE. The RMSE values for each time-series and time-series forecasting method are presented in table 5.1. For example, the performance of SARIMA on time-series 10 is 0.115019086. For each row, time-series, the RMSE value of the best forecasting method for is the one with a white background in table 5.1. We use a grey color gradient to capture the RMSE values in ascending order, with the highest RMSE value having the darkest grey background.

Table 5.1: The obtained out-of-sample forecasting results

Time-Series	Holt-Winters Method		SARIMA	LSTM	Prophet
	Additive	Multiplicative			
1	0.052718476	0.047197678	0.04379352	0.0565074	0.063169587
2	0.362147886	0.339159135	0.216381818	0.114771638	0.233960167
3	0.081430551	0.084518051	0.088315154	0.114717893	0.09347953
6	0.105357268	0.079594544	0.011710945	0.071115282	0.15177958
10	0.362472624	0.380228945	0.090905397	0.052263324	0.532592047
11	0.169044221	0.15123353	0.063372383	0.216759328	0.064594451
12	0.457803373	0.716419407	0.118663012	0.426921697	0.33688178
13	0.273247357	0.423289868	0.124115385	0.37293235	0.417652742
14	0.423605793	0.551439853	0.153340255	0.164004374	0.344218624
15	0.322388095	0.451495754	0.068349105	0.236286005	0.308324887
16	0.477080293	0.509248565	0.062846976	0.237123006	0.347998358
17	0.036343298	0.046946161	0.027581641	0.048944797	0.054571149
18	0.303222317	0.310348304	0.327293065	0.475275638	0.315665002
22	0.453506917	0.498518553	0.173066273	0.36836919	0.13699539
23	0.086800418	0.071112304	0.077070271	0.056690928	0.077232278
24	0.043057577	0.060095778	0.076900427	0.076009868	0.079034887
25	0.104473268	0.12102555	0.109445989	0.168706033	0.1767096
28	0.227664811	0.267840435	0.337413552	0.203460253	0.386050892
29	0.635004862	0.371527685	0.168286419	0.294129172	0.581238058
30	0.070358196	0.062185129	0.070714207	0.095460833	0.089778998
31	0.060220246	0.120796422	0.023162358	0.045112556	0.026566921
32	0.082502227	0.069788167	0.058709611	0.051181117	0.056373376
34	0.140073523	0.142644695	0.14244647	0.132333604	0.216321316
35	0.952119524	1.15274414	0.44595928	0.123842699	0.598523979
37	0.60429911	0.799753769	0.115187179	0.652799928	0.330400601
39	0.172658603	0.190830059	0.120100865	0.095619582	0.067967304
40	0.100362698	0.097844025	0.130138999	0.105133063	0.14320691
41	0.029135486	0.009732091	0.005462345	0.036034472	0.028491471
43	0.290354341	0.278373674	0.300301405	0.280337001	0.298611941
44	0.194622891	0.211984287	0.166228902	0.168538519	0.14476153
45	0.098849004	0.122250752	0.065715554	0.149223962	0.238204715
46	0.029990256	0.028664263	0.037490268	0.096501547	0.05206334

The following can be observed from the results obtained in table 5.1:

1. The SARIMA method performs significantly better in terms of RMSE (15 out of 32) than the other methods.

2. The next best performing model is the LSTM method, which may also be considered for forecasting sales.

3. HW Multiplicative, HW Additive and the Prophet methods all perform nearly equal in forecasting sales.
4. Overall the SARIMA and LSTM methods are favoured for sales forecasting over either the Prophet or both HW methods.

We summarize table 5.1 by showing the percentage of time-series each method is the best performing method in table 5.2.

Table 5.2: The overall performance in percentage of the compared methods on the 32 sales time-series

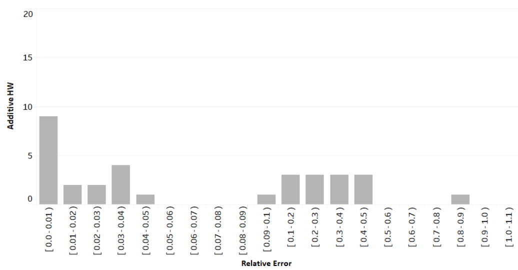
Method	**% Time Best**
SARIMA	47%
LSTM	22%
Multiplicative HW	13%
Additive HW	9%
Prophet	9%

One can observe in table 5.2 that the SARIMA method performs the best (47%) followed by LSTM (22%),Multiplicative HW (13%), and Additive HW (9%) and lastly Facebook's Prophet method (9%).

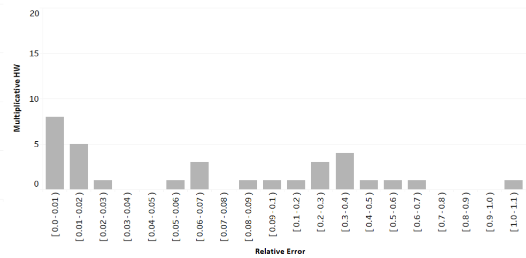
5.3 Insight into the comparison results

The following section will provide the comparison and insights of the thirty-two univariate sales time-series as mentioned in the in section 1. To seek more insight into the comparison results, we take the minimum RMSE value of each time-series, across all forecasting methods, and deduct the minimum RMSE value from the RMSE values of all the time-series forecasting methods. This means that for each time-series, there will be at least one "modified" RMSE with a value of zero, which is the minimum RMSE value. We do not show the modified version of table 5.1, for space considerations. However, we superimpose the modified RMSE values in figure 5.1 to provide insight into the relative distributions of the RMSEs across all five time-series forecasting methods.

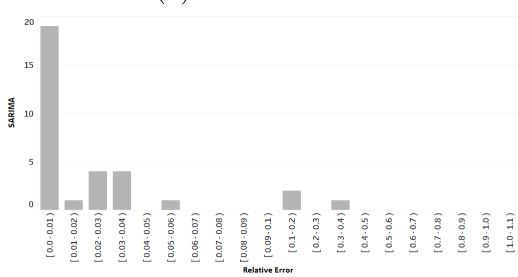
Few things of interest to keep in mind when reading figure 5.1:



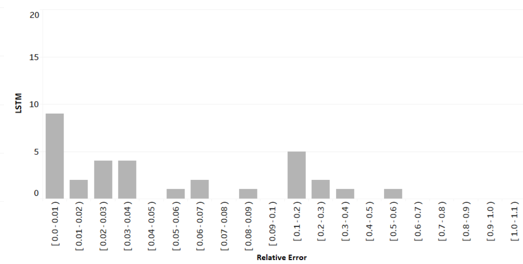
(a) Additive HW



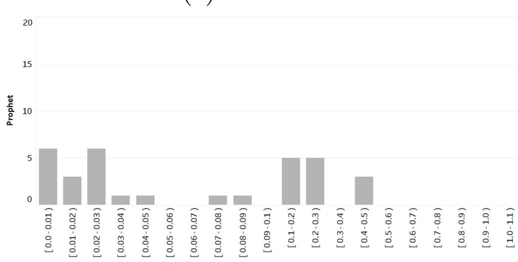
(b) Multiplicative HW



(c) SARIMA



(d) LSTM



(e) Prophet

Figure 5.1: Insights into comparing results

1. Bins begin at the same interval of 0.01 from 0 to 0.1 and then rise to 0.1 from 0.1 to 0.9.

2. The closer a method appears on the left of the figure, the better that method at predicting the true value.

3. The figure indicates the relative difference in performance between the methods and not the individual performance.

One can see in figure 5.1c distribution of relative errors for SARIMA is left skewed (echoing the results in table 5.2) indicating the SARIMA method is likely the best for forecasting sales time-series. One can also see from the figure 5.1d that the LSTM method tends to be very close to zero suggesting that the LSTM method for sales forecasting may also be appropriate. However, the LSTM relative error may also be quite large as it appears multiple times in bins capturing values greater than 0.1. This indicates the LSTM method may not be best method to use. The remaining three methods, Prophet, multiplicative and additive HW are not skewed to the left. This means that all methods, other than the SARIMA method, may have large predictive errors, relative to the best performing method, in expectation. This attribute, of not being left skewed, suggests that Prophet, multiplicative and additive HW are not a good method to use for an unknown time-series.

Chapter 6

Discussion

In Section 5 we compared the five sales time-series forecasting methods. In this section we provide mathematical and economic reasoning on the drivers of the observed performance.

6.1 Mathematical reasoning

The mathematical reasoning on the drivers of the observed performance for each method are as follows:

1. The effects of forecasting using the HW method often depend on the pattern of the last trend period, i.e., the most recently observed sample. As seen in table table 5.1 and table 5.2, neither type of HW method work well in predicting future sales. Therefore, we are left to postulate that the time-series used in our study depend on more than just the last trend period (Kalekar, 2004). According to Gelper et al. (2007), any sort of transformation may improve prediction accuracy, something not considered in this dissertation.

2. We note that SARIMA models may have many combinations of parameters and terms. One may use the auto_arima function in python to automatically build a SARIMA model. However, we find that by fine-tuning parameters ourselves, we can improve the model's accuracy. The fine tuning helped us find the correct AR and MA terms, as suggested by Nau (2014), using the ACF and PACF plots. This suggests

that our fine tuning may result in SARIMA performing better than one may expect find when only using the auto_arima function.

3. For the LSTM model, we use a sequential model for each time-series with 50 to 200 LSTM memory cells in the hidden layer and an output layer. The model fits using the stochastic gradient descent version of *Adam* and is optimized using the loss functions of MSE and RMSE. However, it remains unclear if accuracy may be improved if we use a different LSTM structure, something currently not explored in this study. Fine tuning the LSTM structure may improve its performance.

4. We built the Prophet model on the original data automatically for our research. However, no holiday details are used for any of the time-series. As suggested in the paper (Taylor and Letham, 2018), domain/industry knowledge can boost the model's accuracy, something not available in this research study and may limit the performance of the Prophet model.

6.2 Economical reasoning

The economical reasoning on the drivers of the observed performance for the methods are as follows:

1. Including managerial expertise is one of the important and difficult steps in the sales forecasting process. A decision-maker or forecaster is often familiar with issues such as a possible strike and expected regulatory change. Since these events are rare they may not appear in the time-series. The use of the SARIMA time-series forecasting method and including these independent variables will aid businesses in decision making, something we have not considered in this dissertation due to only considering univarite time-series.

2. Business users need the fastest and simplest possible short-term forecasting. The time to find the best fit model varies for each time-series forecasting method. The Prophet model is an automated system that takes a few seconds to run, and managers and users may prefer it to other time-series forecasting methods. In addition, due to their robustness and unique features, HW methods do not take long to fit, and

there is an advantage of not having to set parameters. Unlike the HW methods, both SARIMA and LSTM require parameters to be set. In addition, both methods take longer than than HW to fit on a given dataset.

3. Though SARIMA methods perform best on sales time-series, the need to set parameters may make them less appealing to decision makers not comfortable with statistics. As such, it is important to keep in mind the tradeoff between no forecasts, qualitative (gut/experiential) forecasts, quantitative (data-driven) without parameters, and quantitative (data-driven) with parameters. It is up to a decision maker and an organization to determine where on the spectrum the like to lie. This research study sheds light on the last two ways to generate forecasts: quantitative (data-driven) without parameters and quantitative (data-driven) with parameters. Our findings suggest that for quantitative (data-driven) with parameters SARIMA is the the best sales time-series forecasting method, and for quantitative (data-driven) without parameters multiplicative HW is best. With that said, as previously discussed this study is a needed first step, and additional explorations are needed to definitely make the claims regarding which method to use.

Chapter 7

Conclusion and Future Work

The main goal of this study is to help sales time-series forecasters, data scientists and managers by a comparative study using Multiplicative HW, Additive HW, SARIMA, LSTM and Prophet methods of time-series forecasting. These methods are chosen according to their abilities to identify complex relationships among time-series data with trend and seasonality. In this dissertation, we presents the empirical comparative evaluation of the performance of the each mentioned methods for short-term sales forecasting on thirty-two different sales time-series taken from time-series Data Library (TSDL) (Hyndman and Yangzhuoran, 2019).

Our findings suggest that for quantitative (data-driven) methods with external parameters SARIMA is the the best method to use for sales time-series forecasting. For quantitative (data-driven) methods without external parameters multiplicative HW is the best method to use for sales time-series forecasting. This conclusion is based on the comparison of the smallest RMSE values on each of the thirty-two different sales time-series. We support our findings by providing mathematical and economic reasoning on the drivers of the observed performance for each method.

We show that even though SARIMA methods perform well overall, it has computational and practical shortcomings. Computationally, a SARIMA method may take much longer to fit to a dataset than a HW method, in the order of hours for small time-series, but much longer for larger time-series. Practically, due to the need for external parameters, the SARIMA method requires training to use and may not be easily accessible to a novice. These two shortcomings may make SARIMA methods less appealing to decision makers not comfortable with statistics. A decision maker and an organization need to evaluate the trade-off between forecasting accuracy and the above-mentioned SARIMA shortcomings. We also show that both the LSTM neural network method and Prophet method are unable to fully capture the behavior of sales time-series.

In the future, one needs to further examine the impact of other external factors in order to increase the reliability of the generated time-series forecasts. In order to generalise the results, the same five methods need to be further tested on time-series with different attributes. For example, different length of time-series, as well as multivariate and not only univariate time-series need to be considered. In addition to the adding additional time-series lengths and observation types, one will ideally incorporate domain/industrial knowledge (LSTM and Prophet), holiday events (Prophet), and/or various model structures (LSTM and SARIMA) in future studies. Though we have not been able to incorporate the listed next steps in this dissertation, we think this dissertation a necessary first step in determining the appropriate method to use when predicting sales time-series. We also hope future studies conduct analogous studies to this one for other types of time-series. Inherently sales time-series are a model of demand, but another aspect of profit is costs, which may be inherently contingent on a firms policies, such as maintenance, growth, or hiring practices. We suspect that the methods listed above may not perform the same for cost time-series as they do for sales time-series.

Bibliography

B. Abraham and J. Ledolter. *Statistical methods for forecasting*. Wiley, 1983. ISBN 9780470317297.

R. Adhikari and R. K. Agrawal. *An Introductory Study on Time Series Modeling and Forecasting*. LAP LAMBERT Academic Publishing, 2013. ISBN 3659335088.

H. Akaike. A New Look at the Statistical Model Identification BT - Selected Papers of Hirotugu Akaike. In E. Parzen, K. Tanabe, and G. Kitagawa, editors, *Selected Papers of Hirotugu Akaike*, pages 215–222. Springer New York, 1998. ISBN 978-1-4612-1694-0. doi: 10.1007/978-1-4612-1694-0{_}16. URL https://doi.org/10.1007/978-1-4612-1694-0_16.

I. Alon, M. Qi, and R. J. Sadowski. Forecasting aggregate retail sales:a comparison of artificial neural networks and traditional methods. *Journal of Retailing and Consumer Services*, 2001. doi: 10.1016/s0969-6989(00)00011-4.

B. L. Bowerman, R. T. O'Connell, and B. L. Bowerman. *Forecasting and time series : an applied approach*. Duxbury Press, 1993. ISBN 9780534932510.

G. E. P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Inc., San Francisco, CA, USA, 1990. ISBN 0816211043.

B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1), 2017. ISSN 15487660. doi: 10.18637/jss.v076.i01.

T. Chai and R. R. Draxler. Root mean square error (RMSE) or mean absolute error (MAE)?-Arguments against avoiding RMSE in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014.

C. Chatfield and D. L. Prothero. Box-Jenkins Seasonal Forecasting: Problems in a Case. *Journal of the Royal Statistical Society: Series A (General)*, 136(3):295–315, 1973.

F. Chollet et al. Keras, 2015. URL https://keras.io.

C.-W. Chu and G. P. Zhang. A comparative study of linear and nonlinear models for aggregate retail sales forecasting. *Int. J. Production Economics*, 86:217–231, 2003. doi: 10.1016/S0925-5273(03)00068-9.

D. A. Cranage and W. P. Andrew. A comparison of time series and econometric models for forecasting restaurant sales. *International Journal of Hospitality Management*, 1992. doi: 10.1016/0278-4319(92)90006-H. URL https://doi.org/10.1016/0278-4319(92)90006-H.

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(366):427, 6 1979. ISSN 01621459. doi: 10.2307/2286348.

C. Frank, A. Garg, A. Raheja, and L. Sztandera. Forecasting women's apparel sales using mathematical modeling. *International Journal of Clothing Science and Technology*, 15(2):107–125, 2003. doi: 10.1108/09556220310470097. URL https://www.emerald.com/insight/content/doi/10.1108/09556220310470097/full/pdf?title=forecasting-womens-apparel-sales-using-mathematical-modeling.

S. Gelper, R. Fried, and C. Croux. Robust Forecasting with Exponential and Holt-Winters Smoothing. *SSRN Electronic Journal*, 2007. ISSN 1556-5068. doi: 10.2139/ssrn.1089403. URL http://www.ssrn.com/abstract=1089403.

S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://www.mitpressjournals.org/doi/pdfplus/10.1162/neco.1997.9.8.1735.

C. C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1):5–10, 1 2004. ISSN 01692070. doi: 10.1016/j.ijforecast.2003.09.015. URL https://linkinghub.elsevier.com/retrieve/pii/S0169207003001134.

C. Hu, Q. Wu, H. Li, S. Jian, N. Li, and Z. Lou. Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. *Water (Switzerland)*, 10(11), 10 2018. ISSN 20734441. doi: 10.3390/w10111543.

J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science and Engineering*, 9(3):99–104, 5 2007. ISSN 15219615. doi: 10.1109/MCSE.2007.55.

R. Hyndman. Initializing the Holt-Winters method, 2010. URL https://robjhyndman.com/hyndsight/hw-initialization/.

R. Hyndman and Y. Yangzhuoran. tsdl: Time Series Data Library, 2019. URL https://finyang.github.io/tsdl/,https://github.com/FinYang/tsdl.

P. S. Kalekar. Time series Forecasting using Holt-Winters Exponential Smoothing. *Kanwal Rekhi School of Information Technology*, 4329008, 2004.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 12 2014. URL http://arxiv.org/abs/1412.6980.

T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing, and J. Development Team. Jupyter Notebooks ? a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90. IOS Press, 2016. doi: 10.3233/978-1-61499-649-1-87. URL https://eprints.soton.ac.uk/403913/.

G. M. Ljung and G. E. Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 8 1978. ISSN 00063444. doi: 10.1093/biomet/65.2.297.

K. D. Makatjane and N. D. Moroke. Comparative study of holt-winters triple exponential smoothing and seasonal arima: Forecasting short term seasonal car sales in South Africa. *Risk Governance and Control: Financial Markets and Institutions*, 6(1):71–82, 12 2016. ISSN 20774303. doi: 10.22495/rgcv6i1art8.

S. Makridakis and S. C. Wheelwright. Forecasting Methods for Management, 1989.

S. G. Makridakis, S. C. Wheelwright, and R. J. Hyndman. *Forecasting : methods and applications*. John Wiley \& Sons, 1998. ISBN 9780471532330.

W. Mckinney. Data structures for statistical computing in python. *PROC. OF THE 9th PYTHON IN SCIENCE CONF*, 445:51–56, 2010.

mindykim. The 5 Financial Impacts of Missing Your Next Sales Forecast - Datahug, 2019. URL https://datahug.com/blog/5-financial-impacts-missing-next-sales-forecast/.

D. C. Montgomery and L. A. Johnson. *Forecasting and time series analysis*. McGraw-Hill, 1976. ISBN 0070428573.

J. E. Nash and J. V. Sutcliffe. River flow forecasting through conceptual models part I - A discussion of principles. *Journal of Hydrology*, 10(3):282–290, 1970. ISSN 00221694. doi: 10.1016/0022-1694(70)90255-6.

R. Nau. Rules for identifying ARIMA models, 2014. URL https://people.duke.edu/~rnau/arimrule.htm.

T. M. O'Donovan. *Short term forecasting : an introduction to the Box-Jenkins approach*. Wiley, 1983. ISBN 0471900133.

T. E. Oliphant. *A guide to NumPy (Vol. 1)*. Trelgol Publishing USA, 2006.

P. Ostrow. Sales Forecasting How Top Performers Leverage the Past, Visualize the Present, and Improve Their Future Revenue. *Recuperado el*, 2013. URL www.aberdeen.com.

F. Pedregosa, Varoquaux Ga{\"e}l, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12:2825–2830, 2011. URL http://scikit-learn.sourceforge.net.

D. Purthan, H. Shivaprasad, K. Kumar, and M. Manjunath. Comparing SARIMA and Holt-Winters' forecasting accuracy with respect to Indian motorcycle industry. *Transactions on Engineering and Sciences*, 2:25–28, 2014. URL https://www.researchgate.net/publication/262804803.

R Core Team. R: A Language and Environment for Statistical Computing, 2013. URL http://www.R-project.org/.

S.-k. Rao. An Empirical Comparison of Sales Forecasting Models. *Journal of Product Innovation Management*, 2(4):232–242, 12 1985. ISSN 0737-6782. doi: 10.1111/1540-5885.240232. URL http://doi.wiley.com/10.1111/1540-5885.240232.

K. K. R. Samal, K. S. Babu, S. K. Das, and A. Acharaya. Time Series based Air Pollution Forecasting using SARIMA and Prophet Model. In *Proceedings of the 2019 International Conference on Information Technology and Computer Communications - ITCC 2019*, pages 80–85, New York, New York, USA, 2019. ACM Press. ISBN 9781450372282. doi: 10.1145/3355402.3355417. URL http://dl.acm.org/citation.cfm?doid=3355402.3355417.

S. Seabold and J. Perktold. Statsmodels: Econometric and Statistical Modeling with Python. In *PROC. OF THE 9th PYTHON IN SCIENCE CONF*, page 57. Scipy, 2010. URL http://statsmodels.sourceforge.net/.

R. H. Shumway and D. S. Stoffer. *Time Series Analysis and Its Applications: With R Examples (Springer Texts in Statistics)*. Springer, 2010. ISBN 144197864X. URL http://www.amazon.com/Time-Series-Analysis-Its-Applications/dp/144197864X.

S. Taylor and B. Letham. prophet: Automatic Forecasting Procedure, 2019. URL https://facebook.github.io/prophet/.

S. J. Taylor and B. Letham. Forecasting at Scale. *American Statistician*, 72(1):37–45, 1 2018. doi: 10.1080/00031305.2017.1380080. URL https://doi.org/10.1080/00031305.2017.1380080.

P. Udom. A comparison study between time series model and ARIMA model for sales forecasting of distributor in plastic industry. *IOSR Journal of Engineering*, 4(2):32–38, 2 2014. ISSN 22788719. doi: 10.9790/3021-04213238.

G. Van Rossum and F. L. Drake Jr. *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995. URL https://dev.rbcafe.com/python/python-3.5.1-pdf/tutorial.pdf.

N. Vandeput. *Data science for supply chain forecast*. Independently published, 2018. ISBN 1730969437.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors. SciPy 1.0– Fundamental Algorithms for Scientific Computing in Python. *arXiv preprint arXiv:1907.10121*, 57:61, 7 2019. URL http://arxiv.org/abs/1907.10121.

H. Weytjens, E. Lohmann, and M. Kleinsteuber. Cash flow prediction: MLP and LSTM compared to ARIMA and Prophet. *Electronic Commerce Research*, 2019. doi: 10.1007/s10660-019-09362-7. URL https://doi.org/10.1007/s10660-019-09362-7.

P. R. Winters. Forecasting Sales by Exponentially Weighted Moving Averages. *Management Science*, 6(3):324–342, 4 1960. doi: 10.1287/mnsc.6.3.324. URL https://www.jstor.org/stable/2627346.

Q. Yu, K. Wang, J. O. Strandhagen, and Y. Wang. Application of Long Short-Term Memory Neural Network to Sales Forecasting in Retail—A Case Study. In *Lecture Notes in Electrical Engineering*, volume 451, pages 11–17. Springer Verlag, 2018. ISBN 9789811057670. doi: 10.1007/978-981-10-5768-7{_}2.

G. P. Zhang and M. Qi. Neural network forecasting for seasonal and trend time series. In *European Journal of Operational Research*, pages 501–514, 1 2005. doi: 10.1016/j.ejor.2003.08.037.