STOCKHOLM SCHOOL OF ECONOMICS
Department of Economics
5350 Master's Thesis in Economics
Academic Year 2022-2023

Demand Forecasting using Machine Learning Methods

An Empirical Study on Walmart Retail Sales Forecast

Author: Yongjai Lee (42135)

Abstract

Economists put considerable effort and research into demand estimation, modeling consumer behavior uncovering causal relationships between product attributes, consumer preferences, prices, and demand. However, demand forecasting is less explored in the economics literature. This paper provides insights into machine learning methods and econometric methods that applied economists can use to forecast demand. In particular, Long Short-Term Memory (LSTM), and eXtreme Gradient Boosting (XGBoost) are compared with the more traditional econometric model, ARIMA in forecasting product demands from a multinational retail corporation, Walmart. Machine learning models performed better in terms of prediction accuracy compared to ARIMA. LSTM demonstrated the highest performance in efficiently capturing non-linear components in sales data.

Keywords: Demand forecasting, machine learning, time series, LSTM, ARIMA

JEL: C22, C44, C55, D12

Supervisor:	Rickard Sandberg
Date submitted:	May 15, 2023
Date examined:	May 26, 2023
Discussants:	Ivelina Ilkova, Mambuna Njie
Examiner:	Karl Wärneryd

Acknowledgement

I would like to thank my supervisor, Rickard Sandberg, for his patience and advice. Furthermore, I am grateful to my family to have supported me in all times.

Contents

List of Abbreviations **ii**

List of Figures **iii**

List of Tables **iv**

1 Introduction **1**

2 Background **2**
2.1 Supply Chain Managment 2
2.2 Demand estimation and forecasting 3
2.3 Time series 4

3 Theoretical Background **6**
3.1 ARIMA 6
3.2 LSTM 7
3.3 Extreme Gradient Boosting 9

4 Data **12**
4.1 Data sources 12
4.2 Data Selection 12

5 Methodology **14**
5.1 Model Development 14
5.2 Model Comparison 14

6 Results **16**
6.1 ARIMA 16
6.2 LSTM 17
6.3 XGBoost 19

7 Discussion **20**

8 Conclusion **22**

References **25**

A Appendix **26**

List of Abbreviations

RNN - Recurrent Neural Network
LSTM - Long Short Term Memory
XGBoost - eXtreme Gradient Boost
ARIMA - Autoregressive integrated moving average
ML - Machine Learning
SCM - Supply Chain Management

List of Figures

2.1 Supply Chain Planning Matrix (Fleischmann et al., 2002) 2

3.1 Recurrent Neural Networks 8
3.2 Structure of modules in an LSTM 8
3.3 Decision Tree Model 9
3.4 Example of regression tree with four terminal nodes 10

6.1 ACF from ARIMA 16
6.2 PACF from ARIMA 16
6.3 Predicted forecasts on test data for ARIMA 17
6.4 Predictions from LSTM and XGBoost 18

A.1 Example autocorrelation for product series 26
A.2 Example First differenced autocorrelation for product series 26
A.3 Feature importance for ML models 27
A.4 Feature importance for ML models 27
A.5 Example ARIMA output 28

List of Tables

6.1 Prediction Accuracy of Top Demand Products - Baseline 18
6.2 LSTM model with 5 days ahead predictions 19
6.3 LSTM model with 25 days ahead predictions 19

1 Introduction

Over the past few decades, demand forecasting has become an essential tool for businesses to optimize their supply chain and production processes. Accurate forecasting enables businesses to make informed decisions regarding inventory management, marketing strategies, and resource allocation. In recent years, the application of machine learning algorithms in demand forecasting has gained significant attention due to their ability to handle complex and non-linear relationships between variables. Demand forecasting differs from traditional demand estimation research in economics which puts emphasis on identifying a demand equation that quantifies the links between the level of demand for a product and the variables that determine it. Demand forecasting provides less insight into the causes of observed demand, and focuses more on predictions based on past records. Nevertheless, Business and Economics literature continues to adopt machine learning in diverse research areas.

In general, businesses model consumer demand as a sequential data of consumer demand over time. Unlike demand estimation where economists use panel data, businesses have to predict sales on daily bases. Popular statistical time series methods such as ARIMA assume that the time series assumes linearity in components. However, many real world data consists of non linear patterns. Computational intelligence techniques such as artificial neural networks (ANN), support vector machine (SVM), K-nearest neighbors (KNN) have been used for modeling non-linearity in time series prediction recently. Other machine learning methods such as gradient-boosted trees offer an efficient non-parametric solution to regression problems.

In this thesis, I aim to compare two prominent machine learning methods, namely LSTM and XGBoost, with the standard time series econometric model, Autoregressive moving average (ARIMA), in forecasting demand for Walmart's hierarchical time series data on sales.

We start by providing a brief overview of the literature on demand forecasting and the applications of machine learning algorithms in this field. We then describe the data used in our analysis, including the hierarchical structure of Walmart's sales data and the preprocessing steps taken to prepare the data for modeling. Next, we present our methodology for training and evaluating the three models. We explain the hyperparameters chosen for each model and the performance metrics used to evaluate the forecasting accuracy. We also discuss the limitations and assumptions of each model.

Finally, we present our findings and conclusions. Our results show that LSTM and XGBoost perform similarly in terms of forecasting accuracy, with LSTM performing slightly better in terms of efficiency. We discuss the implications of our findings and provide recommendations for future research in this area.

Overall, this thesis contributes to the growing body of literature on demand forecasting using machine learning algorithms and provides practical insights for economists seeking to improve their forecasting accuracy and efficiency.

2 Background

In this section, we present a comprehensive literature review from several literature areas. First, we discuss a selection of business theory, namely Supply chain management. Then, we review literature from economics that focuses on demand estimation and industrial organization. Lastly, we go over literature from time series forecasting.

2.1 Supply Chain Managment

Supply chain management is a critical function in any business that deals with the production and distribution of goods and services. It involves the coordination of various activities, including procurement, production, logistics, and distribution. One of the emerging trends in supply chain management is the use of machine learning, which has the potential to improve efficiency, reduce costs, and enhance the overall performance of the supply chain. This literature review aims to explore the use of machine learning in supply chain management from a business theory perspective.

Supply chain management involves the coordination of various activities, including sourcing, production, inventory management, transportation, and warehousing, to ensure the smooth flow of goods and services from suppliers to customers. (Christopher, 2016) Effective supply chain management requires a deep understanding of the supply chain network, including suppliers, distributors, and customers.

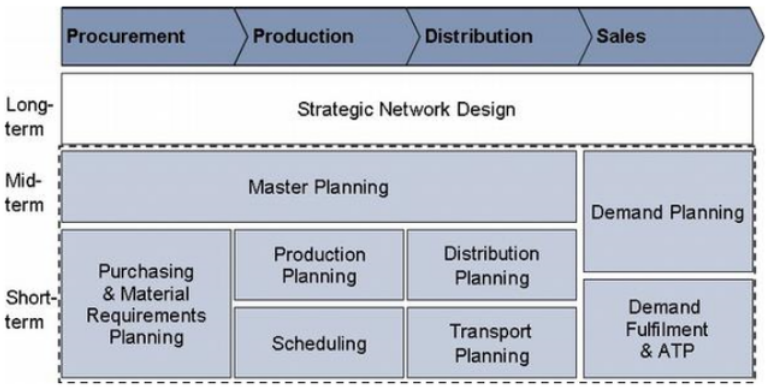

Figure 2.1: Supply Chain Planning Matrix (Fleischmann et al., 2002)

From a business theory perspective, supply chain management can be viewed as a value chain, which involves the identification and creation of value for customers via upstream or downstream links in different processes and activities. The value chain consists of primary activities such as inbound logistics, operations, outbound logistics, marketing, and service, as well as support activities such as procurement, technology development, and human resource management. (Porter, 1985) The goal of the value

chain is to create a competitive advantage by optimizing each activity and maximizing the overall value created for customers.

Machine learning is used to improve procurement by analyzing supplier data and predicting supplier performance. (Feigin et al., 2021) It can also be used to optimize production by predicting machine failures and scheduling maintenance proactively. In logistics and transportation, machine learning can be used to optimize route planning and delivery schedules by analyzing real-time data on traffic and weather conditions. Machine learning has the potential to transform supply chain management by improving demand forecasting, and optimizing inventory management which is the area of interest in this study. (Wenzel et al., 2019)

Demand planning in SCM is essential and in many cases uses econometric methods for quantitative analysis. Univariate time series models such as AR or ARIMA are commonly used to model retail product demands. These traditional time series forecasting methods are applied on the assumption that past demand serves as a statistical indicator of future demand. Typically, these methods perform well in markets with relatively stable demand. Recently, machine learning, a classification technique, has been newly applied in supply chain management to capture exogenous elements, unstable trends, and nonlinearity.

2.2 Demand estimation and forecasting

The following section provides a synopsis of demand estimation and forecasting described in academic and economic literature. Demand estimation and forecasting have roots in the economics literature. Much of demand estimation follows traditional models from IO research. Demand estimation in IO strives to find demand systems using more causal methods such as natural experiments. Bresnahan (1987) uses characteristics of other products as IV to find a relationship between price and quantity in the 1950s Auto Market. The empirical IO research focuses on solving the endogeneity problem from supply and demand using market-level panel data and supply-side instruments. (Berry et al., 1995) A popular empirical IO model is the BLP method of random-coefficients logit model in the differentiated products market. Some economists have started using techniques from machine learning literature in demand estimation problems. Bajari et al. (2015) tests machine learning models such as Random Forest, Support Vector Machines (SVM), and Bagging into a canonical demand estimation problem and finds that it produces superior predictive accuracy compared to the logit model.

SCM literature focuses much on time series demand forecasting and uses various empirical methods. In a retail context, demand is closely related to actual sales, and many studies employ models based on historical sales data to forecast future demand. Auto-regressive integrated moving average (ARIMA) time series model developed by Box et al. (1967), is frequently used in both SCM and econometric applications. Recently, nonparametric models from machine learning have gained attention. Islek and Ögüdücü (2017) investigated the challenges of demand forecasting as warehouses and product quantities increase using MLP, Bayesian Network, Linear Regression, and SVM, to improve forecasting accuracy. Weng et al. (2019) compared the performance of the AutoRegressive Integrated Moving Average (ARIMA) model, backpropagation (BP) network method, and recurrent neural network (RNN) method in forecasting agricultural product prices. Mittal et al. (2019) proposed a method that combined Support Vector Regression (SVR) with Particle Swarm Optimization (PSO) to forecast retail sales using United States Census Bureau data.

Energy economics has seen a rising interest in time series modeling and forecasting. Forecasting in energy economics generally focuses on predicting energy prices, modeling energy consumption or demand, and policy analysis. Econometric methods such as Vector Auto Regression (VAR) or ARIMA have been used to model and forecast oil, and electricity prices. Oil market VAR models have become the standard tool for understanding the real price of oil and its impact on the macro economy. (Kilian and Zhou, 2020) Energy commodity price series often exhibit complex and challenging features including non-linearity, lag-dependence, non-stationarity, and volatility clustering. (Cheng et al., 2019) ML methods provide new opportunities for innovative research in the field of energy. Recent publications suggest that Artificial Neural Networks (ANN), and Support Vector Machines (SVM) are frequently utilized in energy price modeling with Deep Learning (DL) being less common in this area. There are opportunities in applying DL to energy economics. Moshiri and Foroutan (2006) forecast the daily series of futures oil prices using a nonlinear ANN model. Mirakyan et al. (2017) used ensemble methods for modeling the electricity market.

2.3 Time series

Demand forecasting is the process of estimating the future demand for a product or service. It involves analyzing historical data, market trends, customer behavior, and other relevant factors to make predictions about the quantity of goods or services that customers will likely purchase in the future. The primary goal of demand forecasting is to support effective planning and decision-making within businesses, enabling them to optimize inventory levels, production schedules, pricing strategies, and resource allocation. Consumer demand is a sequential data of customer demands over time and therefore demand forecasting can be developed as a time series forecasting problem. (Villegas and Pedregal, 2018)

Time series forecasting methods are categorized into two main types statistical and computational intelligence methods. (Khashei and Bijari, 2011) ARIMA, a well-accepted method in various fields, including economics, and finance, provides a flexible framework to capture both short-term and long-term patterns in predicting time series data. Some critical assumptions of ARIMA include stationarity, linearity, independence, normality, as well as the absence of outliers. These assumptions have some limitations in some data. Differencing time series can cause a loss of information. Nonlinear statistical models such as general autoregressive conditional heteroscedastic (GARCH) have been developed to bypass the linearity assumption.

Machine learning models including artificial neural networks (ANN), support vector machine (SVM), and K-nearest neighbors (KNN) have been recently used for time series prediction problems. Artificial neural networks, (ANN) computational models inspired by the structure and functioning of the human brain, are designed to learn and recognize patterns in data. It has several advantageous characteristics such as the ability to capture nonlinear patterns, and not assume a specific probability distribution for the input data. A specific type of ANNs is recurrent neural networks (RNNs). RNNs are specifically designed to process sequential data. RNNs have a recurrent connection that allows them to retain information from previous time steps and use it to influence the processing of subsequent inputs. This makes it a suitable technique for processing sequence data (Parmezan et al., 2019) Variations of RNN, such as LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) networks were developed to overcome an issue called "vanishing gradient" that traditional RNNs suffer from. (Bengio et al., 1994; Parmezan et al., 2019) Vanishing gradients restrict RNNs' ability to capture long-

term dependencies. To overcome this issue, variations of RNNs have been developed, such as the LSTM (Long Short-Term Memory) (Hochreiter and Schmidhuber, 1997) and GRU (Gated Recurrent Unit) networks, which integrate gating mechanisms to better manage and retain information over longer sequences. (Wu et al., 2018; Xin et al., 2018) Some related empirical work includes the following. Babu and Reddy (2014) proposed a novel hybrid model of ARIMA and ANN that yields more accurate forecasting from sunspot data, electricity price data, and stock market data. Sagheer and Kotb (2019) utilized deep LSTM recurrent networks in petroleum production data.

Some models in Machine learning offer another relaxation of statistical assumption, and does not rely on strong assumptions about the underlying data. Non-parametric supervised learning algorithm such as decision tree are often used for classification and regression problems. Ensemble method combines these multiple individual models and make more accurate predictions than any single model, boosting overall performance. A popular iterative ensemble method is Extreme Gradient Boosting (XGBoost) where it creates a predictive model by combining multiple weak or base learners, typically decision trees, where models are trained sequentially. Many practitioners and scholars use ensemble methods for predictive modeling in diverse types of fields. Pesantez-Narvaez et al. (2019) used logistic regression and XGBoost to predict the occurrence of accident claims in motor insurance. Wang and Guo (2020) proposed a hybrid model with greater predictive performance than of a single ARIMA model or a single XGBoost model in predicting stock price in financial markets.

3 Theoretical Background

This section is about methodology and discusses different methods in econometrics and machine learning in forecasting retail demand.

3.1 ARIMA

ARIMA (Autoregressive Integrated Moving Average) is a popular time series econometric method used for forecasting future values based on historical data. It is a combination of three components: autoregression (AR), differencing (I), and moving average (MA).

Autoregression (AR): The autoregressive component of ARIMA focuses on the relationship between an observation and a certain number of lagged observations. The variable of interest is forecasted using a linear combination of past values of the variable. (Hyndman and Athanasopoulos, 2018)

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

Where ε_t is the error term, white noise. The order of autoregression, denoted as p, represents the number of lagged observations used in the model.

Differencing (I): The differencing component of ARIMA is used to make the time series stationary. (Kwiatkowski et al., 1992) Many time series models such as ARIMA assume stationarity, and constant statistical properties over time, in data. Differencing involves taking the difference between consecutive observations to remove trends and seasonality which allows for constant mean, variance, and autocovariance.

$$y'_t = y_t - y_{t-1}$$

The order of differencing, denoted as d, represents the number of times differenced to obtain stationarity. Similar to the above equation, second-order differencing would yield y''_t.

Moving Average (MA): The moving average component of ARIMA focuses on the relationship between an observation and past forecast errors. It considers the effect of previous error terms on the current value of the time series.

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$

Where ε_t is the error term, white noise. The order of the moving average, denoted as q, represents the number of lagged errors used in the model.

ARIMA model combines these three components (AR, I, MA) to capture patterns and relationships in time series data.

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

The above equation outlines the ARIMA(p,d,q) model where p represents the order of the autoregressive part, d the degree of differencing, and q the order of the moving average part. The parameters (p, d, q) of the ARIMA model are estimated using statistical techniques, and the model is then used to forecast future values based on the observed past data.

Popular statistical techniques for ARIMA parameter determination are Maximum likelihood estimation (MLE) and Information Criteria such as AIC and BIC. MLE finds the values of the parameters that maximize the probability of obtaining the data that we have observed minimizing the mean squared error term.

$$AIC = -2\log(L) + 2(p + q + k + 1)$$

Where L is the likelihood of the data, and $k = 1$ if $c \neq 0$ and $k \neq 0$ if $c = 0$. c represents the constant term in the ARIMA model. Akaike's Information Criterion (AIC) is helpful in determining the orders of ARIMA models. (Hurvich and Tsai, 1989)

$$AICc = AIC + \frac{2(p + q + k + 1)(p + q + k + 2)}{T - p - q - k - 2}$$

Corrected AIC accounts for the number of parameters and sample size in the ARIMA model.

$$BIC = AIC + [\log(T) - 2](p + q + k + 1)$$

Bayesian Information Criterion (BIC) is also a criterion for model selection amongst a set of competing models. BIC evaluates the trade-off between model fit and complexity, placing a stronger penalty on complex models which helps to prevent overfitting and encourages the selection of a simpler model with less number of parameters. (Aho et al., 2014)

3.2 LSTM

To understand LSTM, one needs to understand Recurrent Neural Network. (RNN) RNN is a type of Artificial neural network with loops making information persist. It imitates the structure of neural networks in the human brain and how it functions in the following way. Human brains have persistent thoughts which affect understanding of the next reasoning.

A, a neural network, takes in input x_t and outputs a value h_t with a loop allowing information to persist and flow from one step to the next. An RNN can be viewed as multiple copies of the same network, each passing a message to a network of the next time step. This allows RNN to be well-suited for tasks that involve processing and understanding sequential data allowing it to be used for a variety of tasks, including sequence classification, language modeling, and speech recognition.

RNN however suffers from the problem of long-term dependencies. It works when we only have to look at recent information to perform the present task. Sometimes, however, information from further back as well as recent information are required to

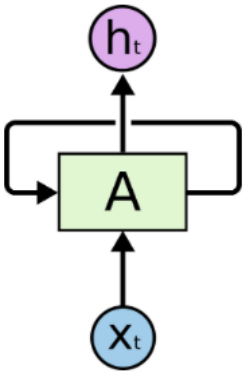

Figure 3.1: Recurrent Neural Networks

perform or predict the next task. When the relevant information is too many time steps back from the point where it is needed to become very large, RNNs become unable to learn to connect the information. (Bengio et al., 1994)

Long Short Term Memory networks were introduced to solve this problem of long-term dependencies. (Hochreiter and Schmidhuber, 1997) They are designed to store and remember information for long periods of time.

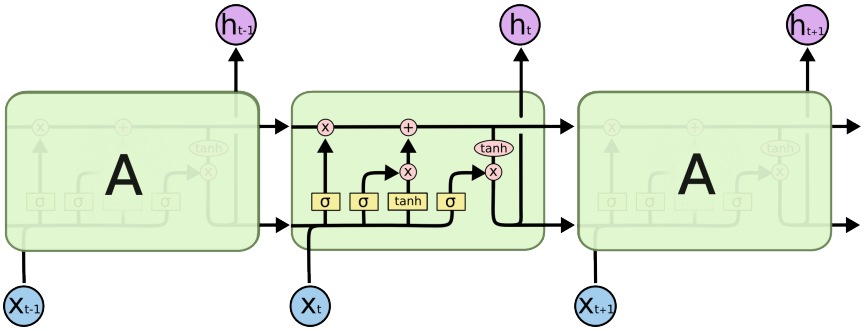

Figure 3.2: Structure of modules in an LSTM

LSTM has the structure of repeating modules of a neural network with four neural network layers consisting of σ and tanh. (Olah, 2015) The horizontal line that goes through the upper side of modules is called the cell state. Cell state allows information to flow from each neural network module with information being added or removed from structures called gates. Gates are composed of a sigmoid neural net layer that lets additional information into cell state.

The first process in LSTM is a sigmoid layer called the "forget gate" which decides what information to disregard from the cell state.

$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$

The forget gate f_t calculates the information to be preserved in C_{t-1} using inputs x_t and h_{t-1} where x_t and h_t are input and output values

Next, LSTM updates new information to store it in the cell state. A sigmoid layer called the "input gate layer" decides which value to update. A tanh layer then creates a list of candidate values, $\tilde{C}_t$ possibly for the addition to the cell state.

$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$

$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right)$

These forget and input gates work together to update the old cell state into the new cell state.

$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$

The final step of the LSTM module is the output stage through the "output gate" consisting of both a sigmoid and a tanh layer. A sigmoid layer decides the parts of the cell state to output. Then, the cell state goes through tanh outputting only the parts that are designed to.

$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$

$h_t = o_t * \tanh\left(C_t\right)$

LSTM learns from the forward learning, steps that happen in LSTM modules and produces outputs. It computes the error between the resulting data and the input data of each layer. The computed error is then transmitted back to the input gate, cell, and forget gate. Finally, the Optimization algorithm updates the weight of each gate based on the error term.

3.3 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is a supervised learning algorithm frequently used for regressions and classification problems. XGBoost is a gradient-boosted trees algorithm which combines hundreds of decision tree models. It iteratively trains decision trees, where each subsequent tree improves on the mistakes made by the previous trees. Time Series Forecasting can be modeled as a Supervised Learning Problem using machine learning.

Before explaining ensemble methods and boosted tree algorithms, one should describe decision trees. A decision tree is a non-parametric supervised machine learning algorithm for classification and regression problems. It is a tree-like model where each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label or a numerical value. The decision tree algorithm builds the tree by dividing the data based on the attribute that provides the best split. This process continues until a stopping criterion is met.

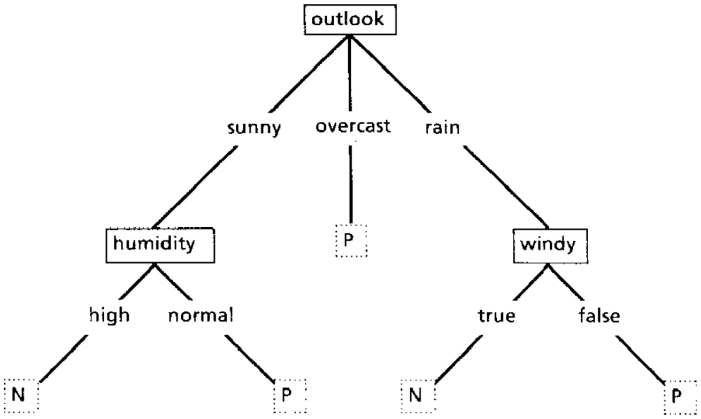

Figure 3.3: Decision Tree Model

Recursive partitioning, dividing data into smaller subsets based on conditions, is the core concept behind the decision trees theory. The decision tree algorithm aims to maximize the information gain or minimize the resulting subsets' impurity. (Quinlan, 1986) There are three commonly used measures of impurity including entropy, Gini index, and classification error.

At each internal node, the algorithm evaluates based on features different splitting criteria calculating the impurity of the resulting subsets. The following measures of impurity quantify the quality of the split.

Entropy measures uncertainty or randomness in a set of data. It quantifies the average amount of information required to describe the sample.

$H(S) = -\sum_{i=1}^{n} p_i * \log(p_i)$

where $p(i)$ represents the ratio of class i in set S

The Gini index is a measure of inequality in a sample with 1 being the highest inequality and 0 being a perfectly equal sample.

$Gini(S) = 1 - \sum_{i=1}^{n} p_i^2$

The classification error is also a simple measure of impurity. It calculates the error rate in the subset.

$Error(S) = 1 - \max p_i$

The decision tree algorithm uses measures of impurity to find the best split at each node that leads to the greatest information gain or smallest impurity. This process is repeated recursively for each resulting subset until a stopping criterion is met.

Popular decision tree algorithms include Classification and Regression Tree (CART), ID3, CHAID, and C4.5. Each algorithm has its way of building decision trees based on different strategies and criteria. CART minimizes Gini impurity to measure of impurity a node in the tree based on the distribution of classes of the node. ID3 uses entropy, the average amount of information or randomness in a set of class labels, for feature classification. (Choi, 2017) CHAID utilizes chi-square tests to see if there is a statistical significance between categorical features and the target variable, selecting the most significant feature for splitting. (Rokach and Maimon, 2005) Lastly, C4.5 works similarly to ID3 but then uses instead the Information Gain Ratio normalizing the information gain by taking into account the intrinsic information of each feature. This reduces bias. C4.5 also incorporates pruning techniques, removing certain branches or nodes, to prevent the overfitting of trees and increase the accuracy of predictions on test data.

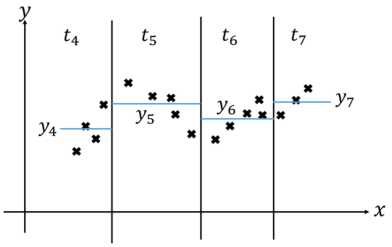

Figure 3.4: Example of regression tree with four terminal nodes

Combining multiple trees and creating a stronger and more accurate predictive model is called ensemble methods. A popular ensemble method is XGBoost (eXtreme Gradient

Boosting), an implementation of gradient boosting. Gradient boosting is a method that combines weak learners sequentially, where each new learner is trained to reduce the errors from previous trees. XGBoost introduced enhancements to better performance and scalability.

The objective function, loss function and regularization, of XGBoost is the following. (Chen and Guestrin, 2016)

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t\left(\mathbf{x}_i\right)\right) + \Omega\left(f_t\right)$$

where x_i and y_i are training sets, l the function of CART learners and consecutive additive trees.

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, \gamma\right)$$

The objective function originates from the initial model. A tree is fitted to pseudo residuals computed in the following way.

$$r_{im} = -\left[\frac{\partial L\left(y_i, F\left(x_i\right)\right)}{\partial F\left(x_i\right)}\right]_{F(x)=F_{m-1}(x)} \qquad \text{for } i = 1, \ldots, n$$

Pseudo-residuals represent errors or discrepancies between the actual target values and the predictions made by the current ensemble. This residual is fitted to a base learner, $h_m(x)$.

$$f_t(x_i) = \gamma h_m(\mathbf{x}_i)$$

The function, f_t, from the loss function can be expressed as the above equation where γ is a constant multiplier. Individual model in the ensemble is trained to correct the mistakes or errors made by the previous models. The new multiplier γ_m is calculated by minimizing the loss function.

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

The model follows an iterative method where each new tree is fitted to the negative gradient of the loss function. This then is multiplied by a constant and added to the value from the previous iteration. XGBoost aims to minimize a specific loss function through this iterative method. In a regression setting, mean squared error (MSE) typically serves as a loss function.

4 Data

This section provides an overview of the dataset utilized in the study. Then it delves into the discussion of data handling as well as splitting of the datasets into train, validation, and test sets for forecasting purposes.

4.1 Data sources

Walmart released a large set of actual sales data publicly for the M5 forecasting competition. The Makridakis Open Forecasting Center (MOFC) at the University of Nicosia conducts cutting-edge forecasting research and is known for its Makridakis Competitions. Its 5th occurrence provides hierarchical sales data from Walmart which covers stores in three US States, California, Texas, and Wisconsin. The data includes item level, department, product categories, and store details from one of the world's largest retail corporations. The dataset consists of 30490 household and food products from 7 departments in 10 different stores with daily sales ranging from 2011 to 2016. The length of the data is 1913 daily observations. The large number of observations allows for a fine time series model as the model has more observation than parameters in the time series, as well as is long enough to capture the phenomena of interest accounting for annual seasonality.

The dataset also consists of explanatory variables such as price, promotions, day of the week, and special events. These additional master data are useful for retail sales forecasting and can be used for features: a synonym for the independent variable in machine learning. The dataset is organized in a Hierarchical structure where the data is organized into nested levels: item, department, product category, as well as store levels. This hierarchical structure enables the representation of complicated relationships and dependencies within the data.

4.2 Data Selection

The output variable, also known as the dependent variable in statistics, is the total sales for an item. Many series were examined to see if they serve as a good fit for demand forecasting. The top 10 selling products were chosen for the series to be forecasted as products in high demand are typically the ones that need effective demand planning. Estimating future demand ensures that the right quantity of products is available at the right time to meet customer demand. With the forecasts, businesses optimize their inventory levels, production schedules, and supply chain operations to meet customer needs and minimize any potential supply disruptions. Other series such as bottom 10 selling products, and random products, were analyzed as well, however, seemed to be inadequate for time series analysis as many data points included zero sales.

The dataset is subsetted into training and test sets. The training set is used to train or build predictive models. It contains a labeled dataset where the input variables

(features) are paired with the corresponding output variable (target). The model learns the patterns and relationships within the training data to make predictions. The data was divided into one month for testing and the rest of the data to train our model. The test set, the latest one-month period in the dataset, is used to evaluate the performance of the trained model. The model makes predictions on the test set based on the patterns it learned during training. The predicted outputs are compared with the actual values from the test set to assess the model's accuracy and performance. A popular method for splitting the dataset into test and train sets is randomly splitting the dataset. However, as the problem at hand is future retail sales forecasting, the last 100 days were selected as the test set.

In addition to training and test sets, a validation set is typically used for model development purposes to fine-tune parameters, select the best model architecture, or make decisions about feature selection. 100 days prior to the test set were allocated for the validation set. The validation set is independent of the test set to provide an unbiased estimate of the model performance on new data. This ensures a more reliable evaluation of the model's performance on the test set and helps in selecting the optimal model for deployment.

5 Methodology

This section provides an overview of the overall methodology employed for time series demand forecasting. Firstly, the methodology for the baseline model, ARIMA, is explained. Then, the methodology pertaining to machine learning models is described. Ultimately, the overall methodology employed to compare the predictive performance of the selected models is discussed.

5.1 Model Development

ARIMA model first uses ACF and PACF in identifying the parameters for modeling. Then, Augmented-Dickey-Fuller (ADF) is used to determine if the time series is stationary or not. Series is differenced in the case that the series is not stationary. Akaike information criteria (AIC) provides a more robust check on model criteria. It provides accurate model identification in selecting the autoregressive (AR) order (p), the differencing (I) order (d), and the moving average (MA) order (q).

For machine learning methods, data is divided into training, validation, and test sets with features obtained from the feature importance method. LSTM design requires defining the number of LSTM layers, the number of memory cells or units in each layer, and the activation functions. LSTM model training involves forward and backward propagation. The propagations adjust the model's parameters using optimization algorithms, Adam. XGBoost requires that the number of trees and other hyperparameters is defined for its model design. The hyperparameters include tree depth and learning rate.

5.2 Model Comparison

Forecasting predictions are compared to the actual sales data in the test set using various accuracy measures: Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and symmetric Mean Absolute Percentage Error (sMAPE).

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

RMSE calculates the average magnitude of the differences between predicted and actual values. RMSE provides an overall measure of forecasting error but is sensitive to outliers and large errors. These appropriate metrics provide performance measures. (Hita-Contreras et al., 2018)

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

MAPE measures the average percentage difference between predicted and actual values. MAPE is commonly used in business forecasting and is useful for understanding the relative magnitude of the forecasting error. MAPE has a limitation of becoming infinite when the actual value is zero or close to zero.

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|\hat{y}_t - y_t|}{\frac{|\hat{y}_t|+|y_t|}{2}}$$

sMAPE is an alternative to MAPE which addresses the asymmetry in percentage errors issue. sMAPE offers a symmetric measure of percentage error, making it suitable for comparing forecasts of different scales.

6 Results

This section presents the results of the study. It provides overview of the models developed. The prediction accuracy of each model is compared, and analysis of results is presented.

6.1 ARIMA

The first stage of ARIMA model involes investigating if the time series is stationary, or difference stationary. Of the ten sales series to be investigated, five were stationary. Other time series were considered unsuitable for ARIMA modelling as either they exhibited complex patterns including non-linear relationships between lagged dependent variable, or non stationary time series. Some of these non stationary product sales experience demand shocks or periods of time with no sales. Model identification was executed through analyzing the autocorrelation function (ACF) plots, and partial autocorrelation function (PACF) plots to decide the order of autoregressive (AR) and moving average (MA) components of the model.

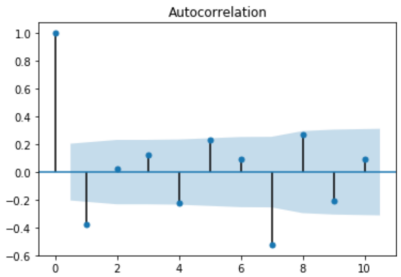

Figure 6.1: ACF from ARIMA

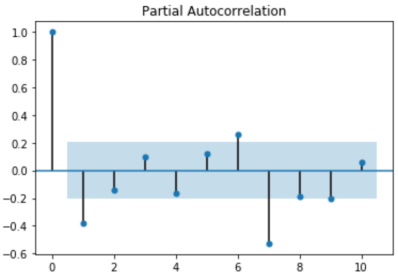

Figure 6.2: PACF from ARIMA

For example, for a product series FOODS_3_252, ARIMA model of (0, 1, 1) was selected based on ACF and PACF. The selection of 1 moving average parameters indicate

that the time series depend on previous error term but 0 auto-regressive lag implies that the demand for the product is not dependent on the previous daily demands. Following this procedure, Akaike Information Criterion (AIC) is used to assess the model's performance. Finally, the model is used to generate forecasts for the last 100 days time period.

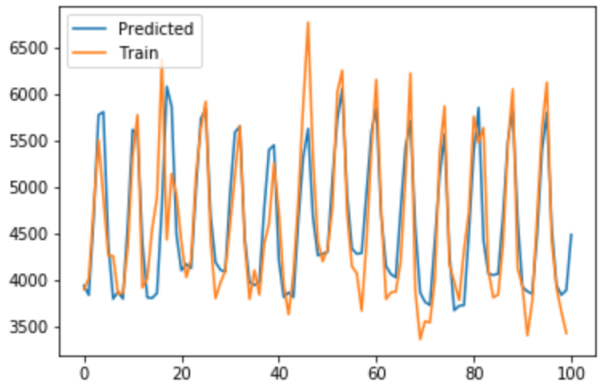

Figure 6.3: Predicted forecasts on test data for ARIMA

6.2 LSTM

The first stage of LSTM analysis was the selection of features. Firstly, the model incorporates time components as features allowing it to capture the cyclic and seasonal patterns that may exist in the data. Year, quarter, month, day of week were selected as time component features. Following this, special events: Thanksgiving, Christmas, independence day, NHL finals, as well as the three state categories were selected as features. Finally, price of product in analysis were selected as a feature. Beginning of a year generally saw more sales and the features allow the model to account for the temporal aspect and improve the accuracy of predictions.

Last three months, 100 days, were split to be assigned to the test set. THe rest of the data points were assigned to trasing set. LSTM looked at 28 time steps to look back in the past for prediction. As the goal of the prediction is retail sales prediction in a month period of time, the number of time steps is appropriate. LSTM uses this length of the input sequence and the model processes the data in sequences of 28 consecutive data points to determine the current time step. The LSTM network is defined with 1 LSTM layer of 4 units (4 LSTM cells in the layer), with 1 dense output layer with one unit. 8 gives overview of the theory behind the LSTM model setting. The LSTM neural network is then trained over multiple epochs, 10 with a batch size of 1. These hyperparameter in neural network training affect the model performance and the training process. Epoch determines the number of times the network iterate over the whole training dataset. The batch size, on the other hand, refers to the number of smaller subsets that the model can divide into for training of samples. Larger batch size improves training efficiency while smaller batch sizes provides better generalization and prevents the model from memorizing the training examples.

The results suggest that machine learning methods suggest more accurate prediction capabilities than the more traditional econometric ARIMA approach. The machine learning models provided similar accuracy results in comparison to each other with

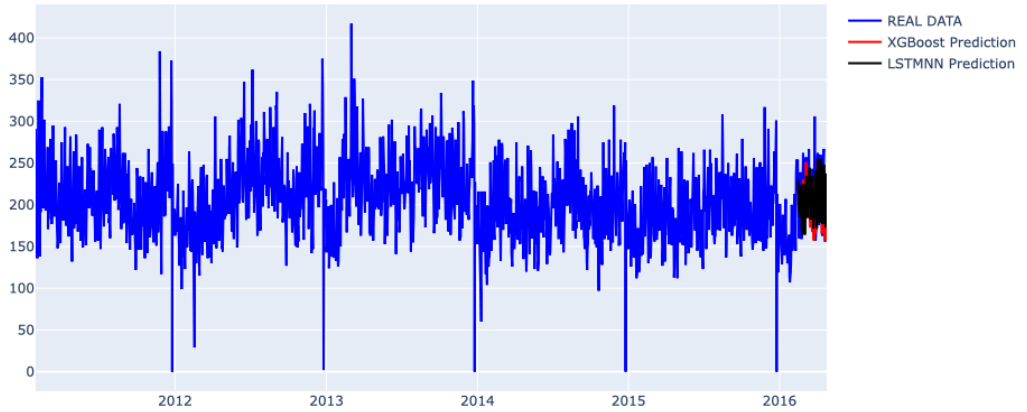

Figure 6.4: Predictions from LSTM and XGBoost

		ARIMA			**LSTM**			**XGBoost**	
item id	RMSE	MAPE	SMAPE	RMSE	MAPE	SMAPE	RMSE	MAPE	SMAPE
FOODS_3_694	30.89	2448.52	11.64	21.47	1731.47	8.31	24.86	1861.96	8.84
HOBBIES_1_354	18.19	1403.73	21.58	12.26	990.71	17.70	10.23	833.40	15.63
FOODS_3_444				0.71	70.51	200.00	0.16	833.40	200
FOODS_3_555	35.74	2741.97	14.81	26.81	2153.17	9.32	34.97	2666.27	11.46
FOODS_3_252	60.98	3970.46	18.23	42.35	3321.58	11.63	45.65	3540.68	12.86
FOODS_3_587				38.32	3034.70	11.81	37.59	3069.07	12.73
FOODS_3_202				30.92	2171.78	11.54	84.03	6274.30	44.51
FOODS_3_090				99.53	7059.44	12.77	163.17	13551.83	27.44
FOODS_3_120				85.90	6181.77	63.81	123.04	10756.21	79.17
FOODS_3_586	73.59	5034.80	14.84	49.62	3775.59	9.13	44.51	3493.33	8.45

Table 6.1: Prediction Accuracy of Top Demand Products - Baseline

LSTM performing better in more product series than XGBoost. LSTM perform better than XGBoost in six out of ten product series with RMSE as a metric. The comparison improves, LSTM being more accurate in seven series, with SMAPE as the alternative indicator of accuracy. Overall, the results are consistent with the anticipated outcomes derived from theory and previous findings.

Our results also suggest that for retail forecasting where sales observe fluctuations, including zero sales, machine learning methods are more useful for use of forecasting. Even of the top demand products, only five series were stationary. ARIMA could not be utilized in other series where they did not meet the assumption of stationary even after differencing. LSTM can be used on unstationary data and is specifically designed to handle sequences and time-dependent patterns, providing leniency in modeling and predicting diverse time series data.

However, interpretation is a downside of machine learning methods compared to ARIMA. While machine learning models are more accurate and effective in making predictions, econometric model ARIMA provides more interpretable results with explicit relationships between variables based on statistical principles and assumptions. Interpretability is important in academic research and business contexts where understanding the causal relationships is important.

	LSTM		
item id	RMSE	MAPE	SMAPE
FOODS_3_694	21.78	1783.71	8.49
HOBBIES_1_354	11.07	909.55	16.53
FOODS_3_444	1.60	160.05	200.00
FOODS_3_555	27.08	2191.46	9.58
FOODS_3_252	42.67	3290.98	11.63
FOODS_3_587	40.73	3064.56	11.97
FOODS_3_202	30.73	2178.15	11.53
FOODS_3_090	126.09	10116.78	18.12
FOODS_3_120	86.96	6068.87	64.58
FOODS_3_586	49.06	3726.80	9.02

Table 6.2: LSTM model with 5 days ahead predictions

	LSTM		
item id	RMSE	MAPE	SMAPE
FOODS_3_694	21.99	1861.99	8.93
HOBBIES_1_354	10.72	894.56	15.66
FOODS_3_444	2.25	225.11	200.00
FOODS_3_555	33.32	2742.20	12.08
FOODS_3_252	47.79	3464.27	12.24
FOODS_3_587	42.44	3536.49	13.39
FOODS_3_202	31.17	2205.67	11.78
FOODS_3_090	117.53	8315.68	15.22
FOODS_3_120	99.23	6848.86	87.82
FOODS_3_586	53.54	4203.36	10.06

Table 6.3: LSTM model with 25 days ahead predictions

Further exploration is done with LSTM model to compare prediction accuracies. Different number of days ahead for predictions are attempt. LSTM uses a window size to predict the next day retail sales. However, it is quite likely that businesses seek to predict in advance. Different number of days ahead, 5 days, and 25 days are used for analysis. This resulted in loss of accuracy in general and is intuitively clear. As the forecasting horizon extends further into the future, more potential factors influence the outcome, making it harder to accurately capture all the variables and their interactions. Long-term predictions depend on complexity of the underlying patterns in the historical data. The inherent uncertainty associated with long-term predictions makes them more challenging and less precise.

6.3 XGBoost

Feature selection was identical to that of the LSTM model for the XGBoost model. The eXtreme Gradient Boosting uses 200 decision trees during the training process with the maximum depth of each decision tree being 6 nodes. Optimal hyperparameter values vary on the dataset and problem at hand. The parameter tuning process helps improve the performance of the models. The forecasting results are given in Table 6.1.

7 Discussion

A notable constraint in many time series studies is that the models are only tested on one series. This is a common scenario in time series studies, particularly when analyzing macroeconomic variables that lack the same level of detail as individual products. In this particular study, multiple products were examined within a specific time period, allowing further exploration and validation of the findings through replication and extension.

Furthermore, the study's modeling process prioritized a "fair assessment" of the models by adhering to a general methodology, potentially leading to the selection of models with suboptimal specifications. The general methodology employed grid-search algorithms to choose hyperparameters from pre-set "rule of thumb" values. These choices made by the researchers encompassed both the selection of possible hyperparameter values and the hyperparameters themselves. However, it is worth noting that these "rule of thumb" choices may not have included the optimal values, which could have influenced the final results rather than solely relying on the models' inherent capabilities.

In practice, a researcher utilizing a single model could iterate and refine the model by revisiting the parameter choices after validation rounds to further optimize its performance. However, since the study aimed to compare multiple models, this agile approach was deliberately avoided to maintain a "fair" assessment for each model. This highlights a drawback of machine learning methods, as their flexibility allows for the creation of well-constructed models but also poses challenges in determining the optimal specifications for a given study. This is where the widely recommended technique of cross-validation can offer significant value.

Another drawback of machine learning methods is their inability to provide statistically stable confidence intervals for predictions. (Shrestha and Solomatine, 2006) Confidence intervals hold significant value in business applications, particularly in supply chain forecasting for physical goods. (Dalrymple, 1987) Businesses dealing with physical goods typically prefer to make inventory decisions based on the confidence intervals of demand forecasts to ensure sufficient inventory levels. While this study focuses on a product without concerns to inventory, it is crucial to consider this limitation when applying these methods to analyze the demand for physical goods.

The findings of the study suggest that machine learning methods have the potential to add value to demand forecasting efforts. Additionally, the findings imply that the demand for these products follows a discernible pattern, potentially resembling a product life cycle. This insight could serve as a starting point for validating product life cycles across products within the retail industry.

The lack of consensus extends beyond modeling approaches to the broader use of machine learning for prediction purposes. As highlighted in the study, there are no strict recommendations on which models consistently outperform others in general. As machine learning gains prominence in economics literature, it becomes crucial for research to continue exploring the strengths and limitations of these models. The results of this study contribute additional evidence to the ongoing research into this question.

Future studies should continue applying different models to diverse time series to shed light on the limitations of different methods in various use cases. The study examined a demand series related to a physical good. Further research can be done on series related to non-physical good such as services.

8 Conclusion

In this study, we provide additional evidence to the demand forecasting literature by comparing econometric methods to machine learning approaches. The neural network model LSTM and tree-based model XGBoost are compared with the baseline model ARIMA. These models were employed in multiple product-level time series to forecast its daily sales. The study evaluates the performance of these methods and compares their effectiveness in capturing and predicting demand patterns.

Overall, compared to the benchmark ARMIMA model, the machine learning models showed higher forecasting accuracy. With retail product level data being highly fluctuating and inconsistent, nonlinearity assumptions in machine learning approaches were beneficial. Both of the machine learning models achieved a similar magnitude of accuracy. LSTM achieved slightly higher performance in more product series than XGBoost.

In this study multiple time series were analyzed allowing for higher scope. Many time series studies involve examination of single time series puts a several constraint the research. By exploring many product level time series, the study lift those typical constraints: limited variability, lack of generalizability, incomplete understanding of data characteristics, lack of bench marking, and limited insights into broader trends.

The results of the study demonstrated that each method has its strengths and weaknesses in retail demand forecasting. ARIMA, a traditional time series model, struggled with data with non-linear and non-stationary patterns. LSTM, a deep learning model designed for sequential data, showed superior performance in understanding complex temporal dependencies and non-linear patterns. XGBoost, a gradient boosting algorithm, demonstrated robustness in handling diverse features and capturing both linear and non-linear relationships.

Overall, this research contributes to the understanding of the strengths and limitations of ARIMA, LSTM, and XGBoost in economic forecasting. It provides insights into forecasting tools and methods that econometricians and practitioners can apply in the field of demand planning and other. Future research should continue to utilize modern tools in applied economic problems and attempt to shed light on improving interpretability of machine learning methods.

Bibliography

Ken Aho, DeWayne Derryberry, and Teri Peterson. Model selection for ecologists: the worldviews of aic and bic. *Ecology*, 95(3):631–636, 2014.

C Narendra Babu and B Eswara Reddy. A moving-average filter based hybrid arima–ann model for forecasting time series data. *Applied Soft Computing*, 23:27–38, 2014.

Patrick Bajari, Denis Nekipelov, Stephen P Ryan, and Miaoyu Yang. Machine learning methods for demand estimation. *American Economic Review*, 105(5):481–485, 2015.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Steven Berry, James Levinsohn, and Ariel Pakes. Automobile prices in market equilibrium. *Econometrica: Journal of the Econometric Society*, pages 841–890, 1995.

George EP Box, Gwilym M Jenkins, and David W Bacon. Models for forecasting seasonal and non-seasonal time series. Technical report, WISCONSIN UNIV MADISON DEPT OF STATISTICS, 1967.

Timothy F Bresnahan. Competition and collusion in the american automobile industry: The 1955 price war. *The Journal of Industrial Economics*, pages 457–482, 1987.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

Fangzheng Cheng, Tian Li, Yi-ming Wei, and Tijun Fan. The vec-nar model for short-term forecasting of oil prices. *Energy Economics*, 78:656–667, 2019.

Hyeong In Choi. Lecture 9: Classification and regression tree (cart). 2017.

Martin Christopher. *Logistics & supply chain management*. Pearson Uk, 2016.

Brian Dalrymple. Novel rearrangements of is 30 carrying plasmids leading to the reactivation of gene expression. *Molecular and General Genetics MGG*, 207:413–420, 1987.

Valery L Feigin, Benjamin A Stark, Catherine Owens Johnson, Gregory A Roth, Catherine Bisignano, Gdiom Gebreheat Abady, Mitra Abbasifard, Mohsen Abbasi-Kangevari, Foad Abd-Allah, Vida Abedi, et al. Global, regional, and national burden of stroke and its risk factors, 1990–2019: a systematic analysis for the global burden of disease study 2019. *The Lancet Neurology*, 20(10):795–820, 2021.

Bernhard Fleischmann, Herbert Meyr, and Michael Wagner. Advanced planning. *Supply chain management and advanced planning: concepts, models, software and case studies*, pages 71–96, 2002.

Fidel Hita-Contreras, Juan Bueno-Notivol, Antonio Martínez-Amat, David Cruz-Díaz, Adrian V Hernandez, and Faustino R Pérez-López. Effect of exercise alone or combined with dietary supplements on anthropometric and physical performance measures in community-dwelling elderly people with sarcopenic obesity: A meta-analysis of randomized controlled trials. *Maturitas*, 116:24–35, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Clifford M Hurvich and Chih-Ling Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, 1989.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

Irem Islek and Sule Gündüz Ögüdücü. A decision support system for demand forecasting based on classifier ensemble. In *FedCSIS (Communication Papers)*, pages 35–41, 2017.

Mehdi Khashei and Mehdi Bijari. A novel hybridization of artificial neural networks and arima models for time series forecasting. *Applied soft computing*, 11(2):2664–2675, 2011.

Lutz Kilian and Xiaoqing Zhou. Oil prices, gasoline prices and inflation expectations: A new model and new facts. 2020.

Denis Kwiatkowski, Peter CB Phillips, Peter Schmidt, and Yongcheol Shin. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics*, 54(1-3):159–178, 1992.

Atom Mirakyan, Martin Meyer-Renschhausen, and Andreas Koch. Composite forecasting approach, application for next-day electricity price forecasting. *Energy Economics*, 66:228–237, 2017.

Mandeep Mittal, Prabodh Ranjan Swain, and Hemant Rana. A nature inspired optimisation method for supply chain management problem. In *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pages 505–509. IEEE, 2019.

Saeed Moshiri and Faezeh Foroutan. Forecasting nonlinear crude oil futures prices. *The energy journal*, 27(4), 2006.

Christopher Olah. Understanding lstm networks. 2015.

Antonio Rafael Sabino Parmezan, Vinicius MA Souza, and Gustavo EAPA Batista. Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information sciences*, 484:302–337, 2019.

Jessica Pesantez-Narvaez, Montserrat Guillen, and Manuela Alcañiz. Predicting motor insurance claims using telematics data—xgboost versus logistic regression. *Risks*, 7 (2):70, 2019.

Michael E Porter. Technology and competitive advantage. *Journal of business strategy*, 5(3):60–78, 1985.

J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

Lior Rokach and Oded Maimon. Decision trees. *Data mining and knowledge discovery handbook*, pages 165–192, 2005.

Alaa Sagheer and Mostafa Kotb. Time series forecasting of petroleum production using deep lstm recurrent networks. *Neurocomputing*, 323:203–213, 2019.

Durga L Shrestha and Dimitri P Solomatine. Machine learning approaches for estimation of prediction interval for the model output. *Neural networks*, 19(2):225–235, 2006.

Marco A Villegas and Diego J Pedregal. Supply chain decision support systems based on a novel hierarchical forecasting approach. *Decision Support Systems*, 114:29–36, 2018.

Yan Wang and Yuankai Guo. Forecasting method of stock market volatility in time series data based on mixed model of arima and xgboost. *China Communications*, 17 (3):205–221, 2020.

Yuchen Weng, Xiujuan Wang, Jing Hua, Haoyu Wang, Mengzhen Kang, and Fei-Yue Wang. Forecasting horticultural products price using arima model and neural network based on a large-scale data set collected by web crawler. *IEEE Transactions on Computational Social Systems*, 6(3):547–553, 2019.

Hannah Wenzel, Daniel Smit, and Saskia Sardesai. A literature review on machine learning in supply chain management. In *Artificial Intelligence and Digital Transformation in Supply Chain Management: Innovative Approaches for Supply Chains. Proceedings of the Hamburg International Conference of Logistics (HICL), Vol. 27*, pages 413–441. Berlin: epubli GmbH, 2019.

Yuting Wu, Mei Yuan, Shaopeng Dong, Li Lin, and Yingqi Liu. Remaining useful life estimation of engineered systems using vanilla lstm neural networks. *Neurocomputing*, 275:167–179, 2018.

Yang Xin, Lingshuang Kong, Zhi Liu, Yuling Chen, Yanmiao Li, Hongliang Zhu, Mingcheng Gao, Haixia Hou, and Chunhua Wang. Machine learning and deep learning methods for cybersecurity. *Ieee access*, 6:35365–35381, 2018.

A Appendix

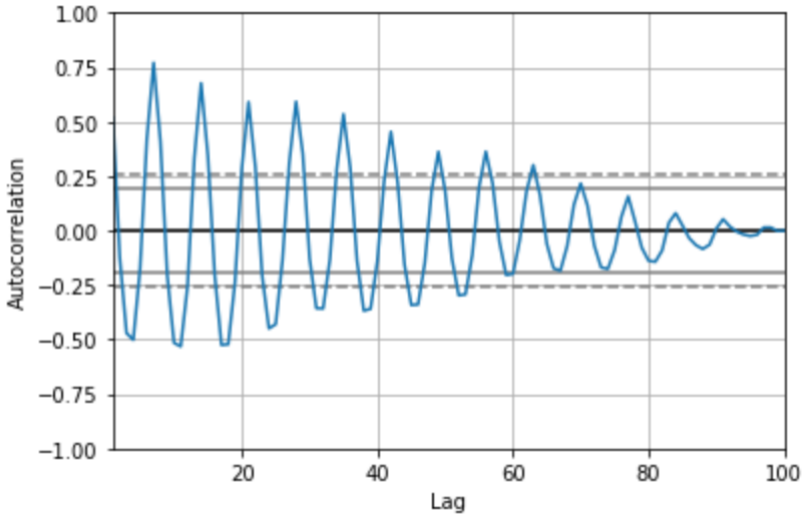

Figure A.1: Example autocorrelation for product series

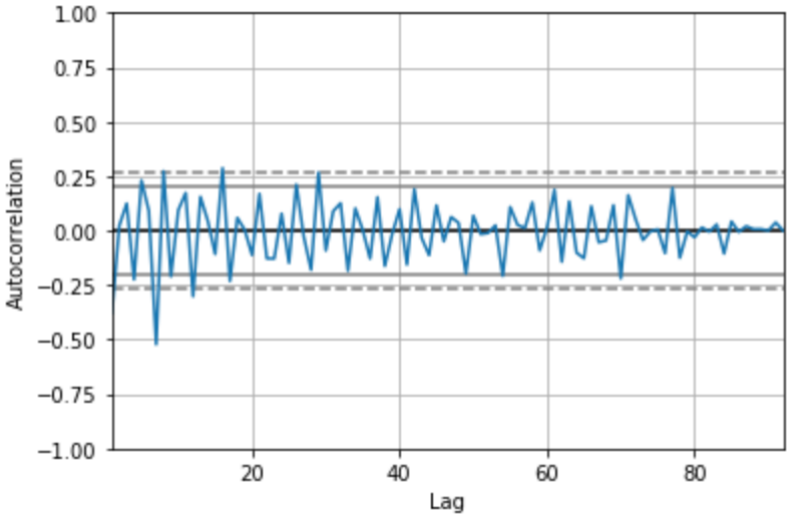

Figure A.2: Example First differenced autocorrelation for product series

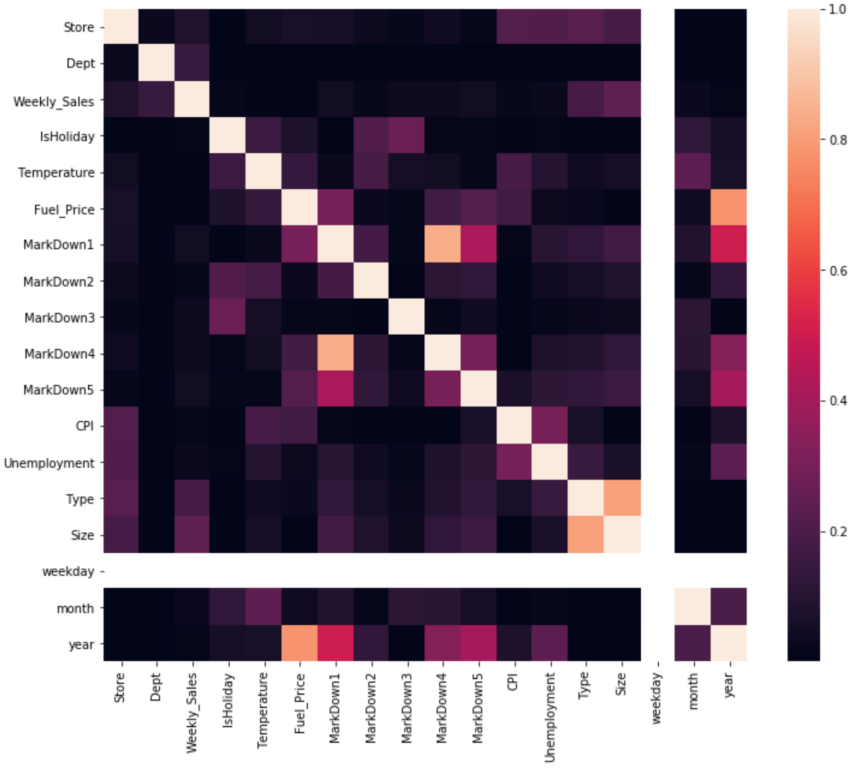

Figure A.3: Feature importance for ML models

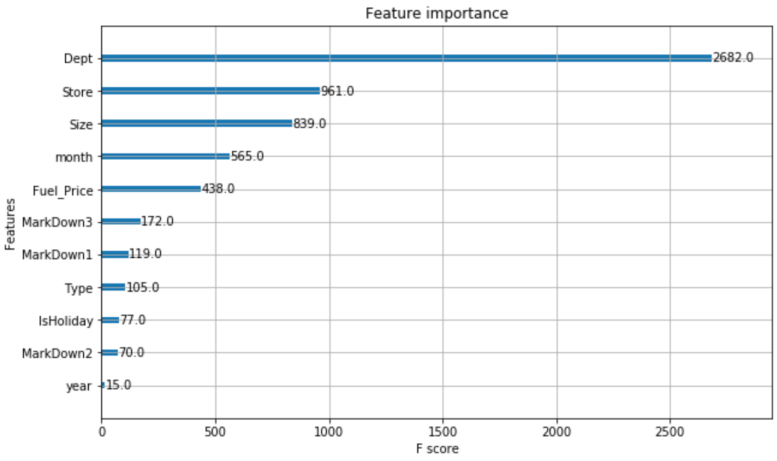

Figure A.4: Feature importance for ML models

```
                                SARIMAX Results
==========================================================================================
===
Dep. Variable:                                 y   No. Observations:                 1
880
Model:             SARIMAX(0, 1, 1)x(0, 1, 1, 7)   Log Likelihood              -14119.
716
Date:                           Sun, 29 Mar 2020   AIC                           28245.
432
Time:                                   14:27:59   BIC                           28262.
036
Sample:                                        0   HQIC                          28251.
549
                                          - 1880
Covariance Type:                             opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -0.7561      0.012    -63.660      0.000      -0.779      -0.733
ma.S.L7       -0.9588      0.008   -124.950      0.000      -0.974      -0.944
sigma2      2.063e+05   2390.600     86.289      0.000    2.02e+05    2.11e+05
===================================================================================
Ljung-Box (Q):                      396.80   Jarque-Bera (JB):             19499.40
Prob(Q):                              0.00   Prob(JB):                         0.00
Heteroskedasticity (H):               1.23   Skew:                            -1.61
Prob(H) (two-sided):                  0.01   Kurtosis:                        18.48
===================================================================================
```

Figure A.5: Example ARIMA output