SMU Data Science Review

Volume 3 | Number 1 Article 5

2020

Demand Forecasting for Alcoholic Beverage Distribution

Lei Jiang
Southern Methodist University, leij@smu.edu

Kristen M. Rollins
Southern Methodist University, kmrollins@smu.edu

Meredith Ludlow
Southern Methodist University, mludlow@smu.edu

Bivin Sadler
Southern Methodist University, bsadler@smu.edu

Demand Forecasting for Alcoholic Beverage Distribution

Lei Jiang, Meredith Ludlow, Kristen Rollins, and Dr. Bivin Sadler

Master of Science in Data Science
Southern Methodist University
Dallas, TX 75275 USA
{leij, mludlow, kmrollins, bsadler}@smu.edu

Abstract. Forecasting demand is one of the biggest challenges in any business, and the ability to make such predictions is an invaluable resource to a company. While difficult, predicting demand for products should be increasingly accessible due to the volume of data collected in businesses and the continuing advancements of machine learning models. This paper presents forecasting models for two vodka products for an alcoholic beverage distributing company located in the United States with the purpose of improving the company's ability to forecast demand for those products. The results contain exploratory data analysis to determine the most important variables impacting demand, which are time of year and customer. For each of the two products, models were built to predict demand for three major customers. For each product/customer combination, this paper compares time series and deep learning models to a naive model to see if the prediction accuracy can be improved. For five out of six products, the time series models reduced error by 2.5–66.7% compared to the naive models. Also, for one product, a hybrid CNN model developed for this paper outperformed the time series models by 3–10% and reduced error by 49% compared to the naive models.

1 Introduction

With the explosion of digital data in the modern world, it can be difficult for a business to draw insights from the data they have collected. This may be especially true for an established company that has many years worth of data collected, but does not know where to begin in order to obtain business insights from this data. This is the exact problem that an alcoholic beverage distribution company presented to the researchers of this paper. The company provided their customer data and requested any business insights that could be found.

This paper presents time series and deep learning models that forecast the demand of two vodka products for the alcoholic beverage distribution company. Demand forecasting was selected as the focus of the paper because, as a wholesale product distributor, the distribution company must manage relationships with both manufacturers and retailers, and anticipation of product demand is a vital component of the company. Demand forecasting is one of the most important aspects of supply chain management since it relates to optimizing stocks, reducing

costs, and increasing profits and customer loyalty [11]. The data provided by the company contained many products. This paper focuses on predicting demand for two products only, because initial data analysis showed that individual products have very different buying patterns over time. Also, for a company with many different products, demand will need to be known on a product-by-product basis.

To provide the company with insights, the first step was performing Exploratory Data Analysis (EDA) on the data provided. The EDA determined which variables are most important in predicting demand. It also led to the discovery that for a given product, each of the three major customers had different buying patterns. This led to the decision to build models for each product/customer combination; six in total. For each combination, statistical time series models and deep learning models were built, which were then compared to a naive model to see if the developed models improve prediction ability. The naive model was developed as an estimation of how the distribution company could be predicting demand without using advanced modeling techniques. Each model was run to predict 6, 12, and 23 months in the future. The models were compared by calculating an average mean absolute error (MAE). The average MAE was computed using a rolling window format.

There is a lot of research in the area of demand forecasting, but no research has been done in the alcoholic beverage distribution industry. This paper aims to provide specific insight for the distribution company and also provide general insight for the industry. The contributions of this paper are time series and deep learning models developed for the distribution company that forecast demand better than naive models, including a hybrid convolutional neural network (CNN) model containing an architecture that has not been employed for alcohol demand forecasting.

The rest of the paper is presented as follows. Section 2 discusses other research and work that has been done in the area of demand forecasting. Section 3 discusses the dataset provided by the alcoholic beverage distribution company, and Section 4 presents the exploratory data analysis performed on the dataset to find the most important features. Section 5 gives an overview of the three types of models used in this paper: naive models, time series models, and deep learning models. Section 6 lays out the methods used to validate the accuracy of each model. Section 7 displays the model results and analyzes those results. Section 8 presents the conclusions drawn from this project. Finally, Section 9 discusses future work that could be done to expand on what was accomplished in this paper.

2 Related Work

Because forecasting product demand is such a valuable business endeavor, there is much literature exploring predictive models and techniques in various industries. Levy and Sheflin (1985) estimated the total demand for alcoholic beverages for the whole U.S. using related time series data from 1940–1980 [13]. In their article they use demand theory to estimate alcohol consumption using variables

such as personal income and beverage pricing. Although the focus of this paper is seeking insights for the same industry, contemporary data (2013–2019) was used to predict future alcohol sales. Levy and Sheflin (1985) predicted current demand for the whole country. While they used demand theory, this paper achieves these forecasts with both autoregressive models and neural networks, and compares the results to determine the most effective forecasting techniques for the alcoholic beverage industry.

While more modern models for forecasting have not been applied to the alcohol industry, they are being applied in other industries. Bandara et al. (2019) proposed using LSTM models on time series data to forecast demand for an e-commerce business. In their research, they noted that previous state-of-the-art methods, such as exponential smoothing methods and ARIMA methods, for forecasting demand were univariate. This means that for predicting the demand of specific product, the model only looked at historical data for that product. In a business, specifically an e-commerce business, products in the same sub-category/category/department are related to each other and thus have sales patterns and demand that are correlated. As such, the authors proposed that using historical data from related products to forecast the demand of a product would result in better predictions. They used an LSTM model that incorporated historical time series data of related products to forecast demand of products on a category level and a super-department level using data from Walmart.com. They found that their models performed better than the univariate state-of-the-art methods currently being used [2]. Lancon et al. (2019) also used LSTM to forecast spot pricing from Amazon Web Services (AWS) [12]. They also compared their LSTM models to baseline ARIMA models and found that the LSTM models performed better. This paper takes the idea from Lancon et al. to compare deep learning models to time series models.

Machine Learning algorithms have also been applied successfully in sales time series forecasting. However, applying machine learning methods to non-stationary sales often yields biased results. The prediction is heavily dependent on the assumption that historical data will be repeated in the future. Thus, sales prediction is more of a regression problem than a time series problem, and regression approaches often outperform time series methods [20]. In 2019, Pavlyshenko proposed a stacking approach to build ensemble predictive models to improve the performance of predictive models. In this approach, the prediction results on the validation set is used as input for the next level models. The first layer is based on the XGBoost algorithm. The algorithms for the second layer could be a variety of choices such as linear regression, Random Forest, or Neural Network. The results from the second layer are summed up with weights on the third layer. Because traditional cross validation is not applicable to time series data, the historical data is split into training and validation sets by period splitting. An out-of-sample set which is not used in model training is used to calculate stacking errors. Validation error and out-of-sample error are both used to evaluate the performance of models. Based on these evaluation metrics, their stacking model has the lowest error on both types. They concluded that the effect

of machine learning generalization can capture the patterns in the whole set of data and the stacked model takes into account the differences in the results of multiple models to improve the prediction accuracy. However, while this model averaging approach could help improve prediction accuracy, the computational complexity is something to be considered.

Another study of note that involves deep learning forecasting was performed by Salinas, Flunkert, and Gasthaus (2019), who used auto-regressive recurrent networks on large amounts of time series data to generate probabilistic forecasts. This process was called DeepAR and can be applied to demand forecasting [22]. Using deep learning and time series models to forecast demand has been researched in many areas, from demand forecasting to predicting the weather [23]. This paper's goal is to bring some of that insight to the alcoholic beverage distribution industry.

3 Dataset

The alcoholic beverage distribution company provided a large dataset describing customer transactions with the distributing company. This dataset was pulled from the company's MicroStrategy platform, where they maintain detailed transaction records. It was helpful to filter the records to contain purchases by 3 major customers, located in a single city, and filtered to only liquor sales.

While upwards of 3,000 attributes and metrics are documented and stored by the distributor, this was an unwieldy amount to pull and filter through without intimate personal knowledge of the business. Therefore, the company contact provided 47 attributes, consisting of those most routinely used in the business, as well as some the company suspected might be useful in forecasting product sales. These were fields such as customer, chain, vendor, product, product type, and dollar sales. The target variable that was forecasted is standard cases. Standard cases is the unit the company uses to measure sales of their liquor. A negative value of standard cases represents a return.

The data was in the form of monthly transactions for a span of 7 years (January 2013–November 2019), resulting in a dataset with nearly half a million rows. Exploration of the dataset was required next to determine which features were redundant or correlated, or useful in forecasting product demand.

4 Exploratory Data Analysis (EDA)

Predicting demand for a company with many products and customers is a large task, particularly because different customers will have different inputs and needs. Coming up with one model for all customers and all products may not be accurate enough to be useful. To narrow the scope of the project, the focus was set on predicting demand for two products for the three different customers whose data the distribution company provided. The intent was to be able to dig in deep for these specific products instead of having a broader, shallower approach for the business as a whole. Exploratory data analysis was conducted

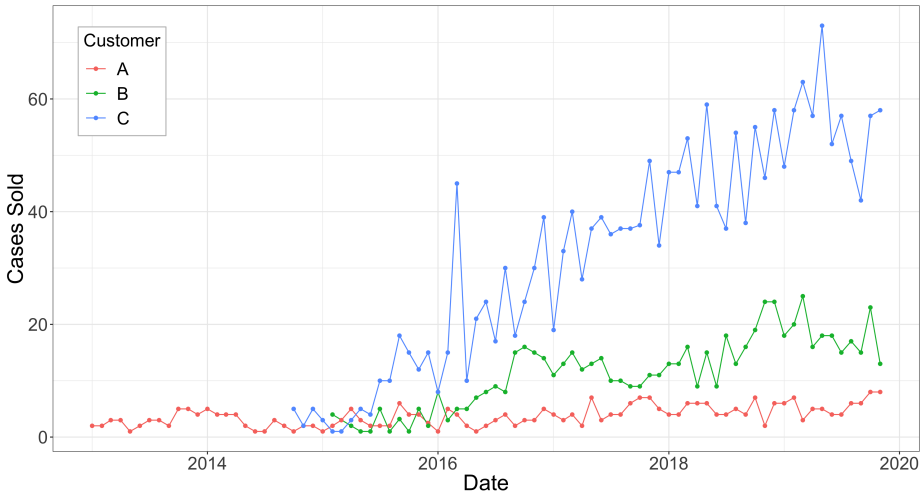


Fig. 1. Product 1 Sales by Retail Customer

on two popular vodka products to see which attributes influence demand. These products are referred to as Product 1 and Product 2, while the three customers are referred to as Customer A, Customer B, and Customer C. The distribution company requested that all of their customer information be kept anonymous so pseudonyms were used.

4.1 Product 1

Graphing the number of cases sold from 2013–2019 by month for the top 5 most popular vodkas revealed that some vodkas have a pretty even trend and some have a seasonal pattern. This seasonal pattern shows large spikes in purchases every three months: in February, May, August, and November. This seasonal pattern for some of the vodkas is so strong that it influences the pattern of purchases when all vodka products are aggregated together. For Product 1, looking at the of the amount of cases purchased for each month aggregated together shows that Product 1 might follow this pattern, if only slightly.

The data came from three different major chains of liquor stores, Customers A, B, and C. These retailers buy their product from the distribution company. Figure 1 shows the demand over time for Product 1 for each of the three customers. The graph shows that each customer has very different buying patterns. Customer A has been buying Product 1 over the full span of the 2013–2019 data and their buying patterns appear to be pretty stable and consistent. Both Customers B and C did not start buying Product 1 until around 2015. Customer B's sales increased but appeared to have leveled off recently. Customer C, however, has increased rapidly over time and seems to be following an upward trend. It also seems to have more variability than the other two customers.

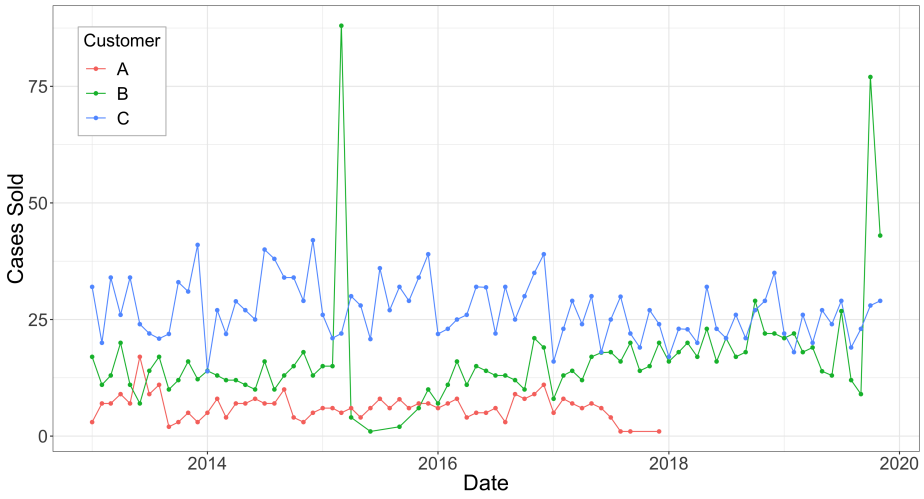


Fig. 2. Product 2 Sales by Retail Customer

4.2 Product 2

Aggregating the total purchases by month for Product 2 does not indicate any strong seasonality in the data. All months have similar averages and spreads. Figure 2 shows the purchases over time by customer for Product 2. The first major observation is that Customer B has several very large spikes in sales at the beginning of 2015 and the end of 2019. Aside from those two outliers, the other two customers have fairly consistent buying patterns, although Customer A stopped buying this product at the end of 2018.

4.3 Summary

Based on the exploratory analysis, a couple of variables could potentially play an important role in predicting the sales of Products 1 and 2. The time of year appeared to have a slight effect on the buying patterns for Product 1. A seasonal component should be added to the models for Product 1. Separating the sales by customer showed that each customer has different buying patterns and trends. This justifies the need for creating separate models for each product for each customer.

5 Forecasting Models

There are three types of models used in this study to forecast alcohol demand: Naive models, Time Series, and Deep Learning. The naive models are described in Section 5.1, as a baseline for how the company might currently forecast demand. The statistical time series models are presented in Section 5.2, while Section 5.3 describes the Deep Learning neural architectures that were explored.

5.1 Naive Models

Before building time series or deep learning models, first a naive model was built as a baseline to compare to the performance of the more advanced models. This baseline model is a conceivable method that the alcohol distribution company could currently use to make forecasts and manage their inventory. In this naive model, data for each month is forecasted to be the value from the data 12 months prior; i.e. sales for June 2019 is predicted to be the sales from June 2018 and so on. One additional naive model was built that goes slightly further and takes the average of the two years prior, so that sales from June 2019 would be forecasted as the average of the sales from June 2017 and June 2018. These are very simple but plausible models for a company that has not incorporated data science techniques into their business strategy. These naive models operate as a baseline to compare with the time series and deep learning models developed.

5.2 Time Series

Traditional time series methods were use to compare to the performance of the naive baseline models when forecasting wholesale alcohol demand. The most well-known time series model is ARIMA, which stands for autoregressive integrated moving average. This model is an non-stationary extension of the stationary ARMA (autoregressive moving average) process. An ARIMA model is a process with orders p, d, and q, where if the data is differenced d times it satisfies and ARMA(p,q) model [26]. The notation for an ARIMA model is found in Equation 1.

$$\phi(B)(1-B)^d X_t = \theta(B)a_t \tag{1}$$

To identify an appropriate ARIMA model, the Box-Jenkins approach of differencing the data until the realization, autocorrelations, and spectral densities appear stationary was used [19]. From there, the Akaike Information Criterion (AIC) was employed to identify an ARMA(p,q) model, and then maximum likelihood estimates (MLE) were used to estimate the parameters of that model.

In addition to the popular ARIMA time series model, the more generalized ARUMA model was utilized where appropriate. This model incorporates a seasonal factor $(1-B^s)$ to the equation above if seasonality s is observed, resulting in the model in Equation 2.

$$\phi(B)(1-B)^d(1-B^s)X_t = \theta(B)a_t \tag{2}$$

To identify seasonality, the autocorrelations and spectral densities of the series were plotted, and overfit factor tables to match factors of the suspected seasonal factor were created.

Depending on the behavior of the data, deterministic Signal-plus-noise models could be appropriate. This model is of the form found in Equation 3.

$$X_t = s_t + Z_t \tag{3}$$

In this equation, s_t is a deterministic signal and Z_t is a stationary noise component that may or may not be autocorrelated. For this study's data, only linear trend signals were observed, but harmonic signals can also be represented with this model.

The final time series model built for each product was a Multiple Linear Regression (MLR) model with correlated errors. This model incorporated additional variables such as time, customer, and month, to create a multivariate setting in addition to the univariate models described above. This is an extension of a multiple regression model, but as time series data points are not expected to be independent of one another, the model fits an ARMA model to the residuals to account for this correlation. The Cochrane-Orcutt procedure was used to determine whether a linear trend was present in the data before applying this model [1].

All of the time series methods described are traditional models that can be found throughout literature. These have been found to be effective in forecasting, but this paper wanted to pursue more recent developments in deep learning as well and compare their effectiveness in the alcohol distribution industry.

5.3 Deep Learning

There are several reasons why Deep Learning methods were used in addition to traditional time series models for this project. First, there are around 3,000 features in the dataset with half a million rows. Feature selection done manually is not practical. Deep Learning can automatically perform feature extraction without any preprocessing [5]. Second, traditional methods like ARIMA are built with the assumption that the series can be stationarized by differencing. If the series does not meet this assumption, then the predictions will be of lesser quality. Sample statistics—mean, variance, co-variance—are useful as descriptors of future behavior only if the series is stationary. In addition, accuracy is where Deep Learning may have the greatest potential to distinguish itself from other algorithms in many domains. Very high accuracy in the models is desired for this real-world business application, in which even 1 percent difference could mean millions of dollars of cost to the distribution company.

Recurrent Neural Networks (RNNs) are one of the most commonly used methods on time series data. However, RNN has a few drawbacks. First, the input must always be the same length. Also, it suffers from the vanishing gradient problem, which means the gradient of some of the weights starts to become too small or too large if the network is unfolded for too many time steps [8]. A type of architecture to solve this problem is called Long Short-Term Memory (LSTM) as shown in Figure 3. LSTM is resistant to the vanishing gradient problem and works better with dropout. One particularly popular variant of LSTM is the Gated Recurrent Unit (GRU). This is a simplified version of LSTM and seems to have similar performance [9]. It has no memory cell, and is composed of only forget and update gates.

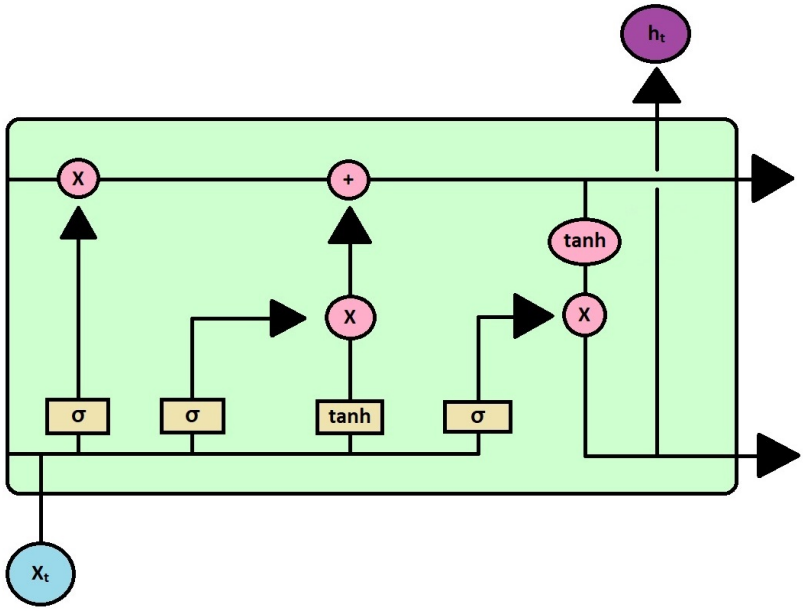


Fig. 3. Long Short-Term Memory (LSTM) Cell [18]

Promise of Deep Learning: Neural Networks learn arbitrary mapping functions from inputs to outputs. In Deep Learning, Neural Networks are used to build deep, hierarchical models that can be exponentially more efficient at representing some functions when compared to more "shallow" models. There are several advantages in Deep Learning in general. Firstly, it is non-linear and requires no strong assumptions, which makes it a good fit for problems with complex non-linear dependencies. Secondly, during data preparation, it does not require a scaled or stationary time series as input. In addition, it is robust to noise in the input data. Most importantly, it handles multivariate inputs and is capable of multi-step forecasts. Makridakis et al. published their study in 2018 on a collection of 1,045 univariate monthly time series, in which they found that classical methods like ARIMA outperform machine learning and deep learning methods for one-step and multi-step forecasting, based on the evaluation metric of symmetric Mean Absolute Percentage Error (sMAPE) [14]. However, deep learning is promising due to its capability to handle high dimensional multivariate inputs, which is its unique advantage over classical methods. Therefore, this paper performed some experiments with multivariate models. Due to time constraints, exhaustive experimentation with multivariate models was not done, but it is worth exploring further in future work.

The specific types of deep learning algorithms all have their own unique advantages. LSTM as a type of RNN handles the order of observations when learning by using sequence as a new dimension. It is able to learn long-term

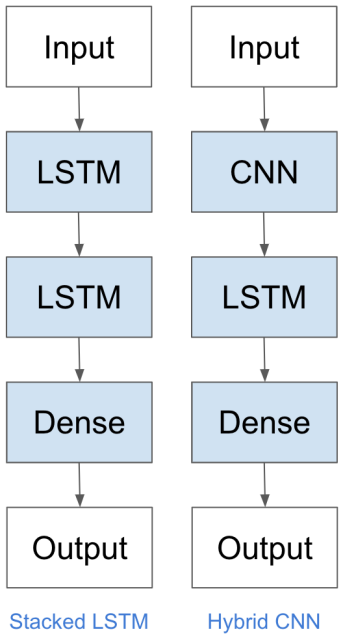
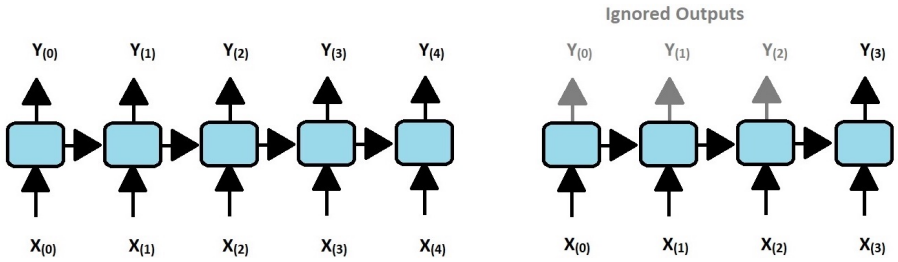


Fig. 4. Neural Architectures [5]

correlations in a sequence, which is the main reason for its popularity in time series prediction studies. On the other hand, CNN models support efficient feature learning that can help transform the data input. In conclusion, automatic feature learning and the capability to handle high dimensional inputs distinguish deep learning methods from traditional statistical methods.

Neural Architectures: Two types of neural architectures in a `tf.keras` workflow were utilized for this project. Those were a stacked LSTM model (including a stacked biLSTM) and a CNN LSTM model as shown in Figure 4, which were implemented with Keras in the TensorFlow framework [4], [5].

There are a variety of architectures in Deep Learning, perhaps only limited by imagination. The choices of neural architectures were determined based on literature review and preliminary experiments on a boarder range of architectures. Stacked LSTM is a type of deep neural network that is commonly used in time series regression. It can be defined as an LSTM model comprised of multiple LSTM layers. Stacking multiple recurrent hidden states potentially allows the hidden state at each level to operate at different timescales. The depth of the network has been found to be more important than the number of memory cells in a given layer to model skills [5]. The depth of this architecture allows each layer to process some part of the task and pass it to the next, as shown



(a) Sequence-to-sequence (b) Sequence-to-vector

Fig. 5. RNN Architectures [9]

in Figure 5a, a sequence-to-sequence architecture. The last layer of the LSTM produces a vector, as shown in Figure 5b, the sequence-to-vector architecture.

Previous research found that a stacked LSTM was able to learn higher-level temporal patterns without prior knowledge of the pattern duration [16]. Stacking several independent neural network layers to work together has also been proven to produce better results than existing shallow structures [8].

Moreover, a Bidirectional LSTM (biLSTM) model allows learning in both directions. Attempting both stacked LSTM and stacked biLSTM (up to three layers of biLSTM) models resulted in the biLSTM performing better than the LSTM. This is why biLSTM structures were used in the stacked architecture.

Various studies have reported that hybrid models such as a CNN-LSTM architecture outperform pure CNN or LSTM models [3], [24]. Therefore, hybrid models with this architecture were built to compare to the stacked LSTM architecture, expecting that the CNN layer for automatic feature extraction could help improve the performance of the models.

For all the Deep Learning models, in the final layer a multilayer perceptron (MLP) was used as the dense layer for the output. Parts of the code used are cited from a book and online lectures [5], [17].

With time constraints, the GRU algorithm was not tested, but it is expected to produce similar results as LSTM with less computational complexity.

Data Preparation: For the Deep Learning approach, a lot of preparation is required before the time series data can be used to train in a model. The time series data was transformed into a three-dimensional structure in order to train convolutional and LSTM neural network models. In addition, a sliding window method was used to transform the time series into a supervised learning setting [6]. For a univariate time series, for example, supervised learning means using the first 12 months of data to predict the 13th data point. The 13th data point becomes the y label, and the 12 previous data points are represented as a feature vector with 12 dimensions. In this way the data is transformed to a structure with 12 inputs and 1 output. For multiple-step prediction, the out-

put can be greater than 1. More specifically, a customized function in Python called `windowed_dataset` was used to split data into window chunks. Data in this structure can be used directly to train a simple neural network like an MLP since it requires that the shape of the input portion of each sample be a vector. However, for the data to be usable for CNNs and LSTMs, the data needed to be further processed into a three-dimensional structure, because the input to every CNN and LSTM layer must be three-dimensional with samples, time steps, and features, which can be summarized using the array shape notation of [samples, timesteps, features]. One sequence is one sample. Several samples make up a batch. A NumPy function called `reshape()` was used to achieve this three-dimensional structure of data.

In the three-dimensional feature matrix X created, the first dimension is the number of samples. The second dimension is the number of time steps per sample. The last dimension specifies the number of parallel time series or the number of variables [5]. In this study, both univariate and multivariate models with month as an additional predictor were created.

In addition, the deep learning modeling was run with transformed data, i.e. trend or seasonality was removed before inputting the data to the models via differencing or similar transformations. The results showed that transforming the data did not make the deep learning models more skillful.

Hyper-parameter Tuning for Deep Learning: The window size parameter was determined to be 12 based on the yearly seasonality found during EDA and the relatively small sample size. When the window is aligned to a seasonal boundary, each window would be equally affected by any seasonality.

Learning rate is one of the most important hyper-parameters needed to be tuned for deep learning networks. To tune the learning rate, a `LearningRateScheduler` object using `tf.keras.callbacks.LearningRateScheduler` was set up and used in the callbacks namespace and assigned that to the callback. Each epoch allows the learning rate to change a little so that it steps all the way from 10^{-8} to 10^{-4}. A Loss versus Learning Rate Curve was generated for each neural architecture to help visualize the lowest loss point as the optimal learning rate as shown in Figure 6. In this plot, the learning rate around 10^{-5} appears to be optimal for this particular neural architecture and product combination. In this way, the optimal learning rate for each of the models was determined.

A loss function called Huber loss was used to find the optimal learning rate. The Huber loss is less sensitive to outliers in the data than the squared error loss. It can be used to balance between the Mean Absolute Error, or MAE, and the Mean Squared Error, MSE. It is therefore a good loss function for when there are only a few outliers in the datasets which were identified during EDA [17].

The choice of optimizer was a Stochastic gradient descent and momentum optimizer (`tf.keras.optimizers.SGD`) since gradient descent with momentum tends to works faster than the standard gradient descent algorithm. Finally, to configure the optimal number of epochs, a grid search method was used.

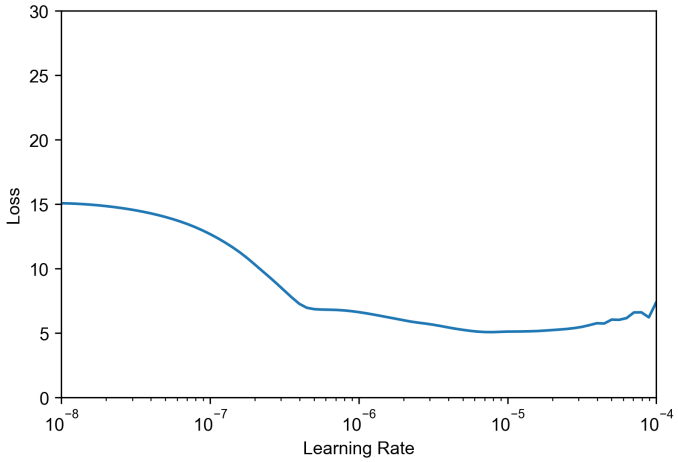


Fig. 6. Loss vs. Learning Rate Curve with Stacked LSTM for Product 1, Customer C

In addition to learning rate and number of epochs, there are many other hyper-parameters to tune in order to improve the models, such as type of activation function, type of optimizer and its momentum value, dropout rate, batch size, number of nodes and layers in the neural network. Due to time constraints and the computational complexity, not all of these parameters were tuned in the course of this study. However it would be worth trying given a sufficient time budget.

6 Model Validation

In order to compare the performance of the models, a method of model validation needed to be implemented. Section 6.1 details why MAE was used to as the metric for model accuracy, and Section 6.2 describes the rolling window cross validation scheme that was used for model validation.

6.1 Model Evaluation

Once all of the forecasting models were identified and created, the metric used to validate and compare these models was Mean Absolute Error (MAE). This loss function takes the average of the magnitude of all the errors on the test set, as expressed in Equation 4.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}| \tag{4}$$

This metric does not penalize large errors as much as other metrics such as Mean Squared Error (MSE) or Root Mean Squared Error (RMSE). It also can be

interpreted nicely, as it is in the same units as the value being predicted, and it simply indicates the average amount by which the predictions were off. Moreover, MAE has been found to be a more natural measure of average error. Therefore, inter-comparisons of average model performance error should be based on MAE [25]. It may be more appropriate to calculate model error sensitivities using MSE or RMSE [7], but these were not a priority in this study.

In addition, multiple methods of validation were used to simulate the different use cases for the models. An initial method was forecasting the last 23 data points (January 2018–November 2019) for each product, using the remainder of the points as training data. This gave an indication of which models performed best for different behaviors of the data. However, this scheme does not account for possible changes in the structure of the data over the time because there is only one estimation of the model.

6.2 Rolling Window

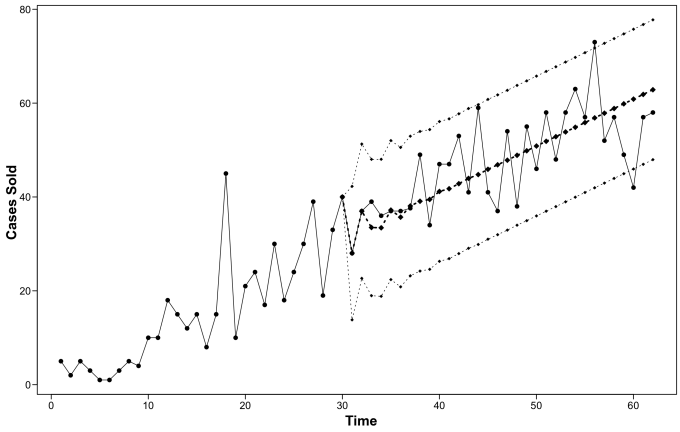
To solve this problem, a more reliable rolling-window cross validation scheme was developed. The rolling window was 24 data points long and started at the beginning of each dataset. Those 24 data points would be used to forecast either 6, 12, or 23 data points into the future. An MAE was calculated for those predictions. Then the window would be shifted forward one time step and the new set of 24 points would predict and another MAE was calculated. The window would keep shifting until it reached the end of the dataset. All the MAEs that were calculated were then averaged to get a total MAE for the model. This process calculated an average MAE for each product/customer combination, for each type of model, for a prediction horizon of 6, 12, and 23. This scheme is more robust on testing model performances with its capability to check the stability of the models over time [27]. However, there is a concern about this method: picking the right size of the window for rolling window method is crucial. A parameter search on the window size could be performed for optimization [21].

7 Results and Analysis

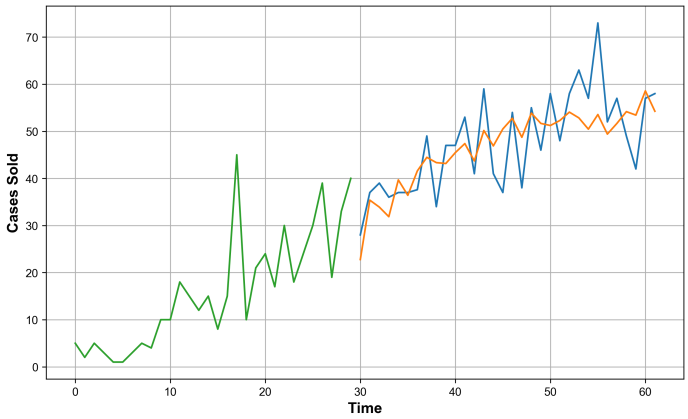
This section presents the results from the three types of forecasting models created, and examines the results found. Section 7.1 first describes the forecasting behavior of the models developed. Section 7.2 then provides detailed quantitative results for each forecasting model, and Section 7.3 draws comparisons between the models.

7.1 Model Behavior

Before applying the full rolling-window evaluation scheme, the behavior of the models was examined when forecasting the last few points of each product's sales. While these forecasts were not used to formally evaluate the models, they were used to investigate the characteristics of the forecasts from each model. The



(a) Signal-plus-noise

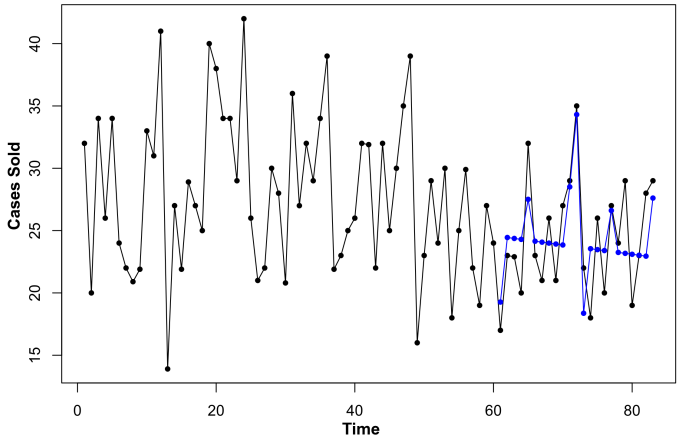


(b) Bidirectional LSTM

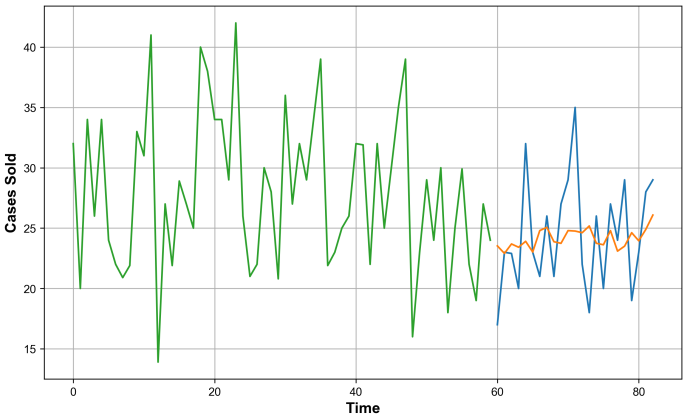
Fig. 7. Product 1, Customer C Forecasts

behavior of the models may be a critical part of how the distribution company will choose models for forecasting, as each model has its own explicit and implicit assumptions [10]. The distributor should evaluate whether a candidate model has reasonable assumptions.

For example, forecasts for Product 1 as purchased by Customer C are found in Figure 7, for two contrasting types of models. On the top are forecasts (the bold dotted line) from a time series Signal-plus-noise model, while the bottom graph shows forecasts (the orange line) from a deep learning Bidirectional LSTM model. While both of these models appear to perform well, they have very different characteristics. The Signal-plus-noise model propagates a deterministic linear trend, while the LSTM trends upwards while attempting to reproduce patterns found in the training set.



(a) MLR with correlated errors



(b) Bidirectional LSTM

Fig. 8. Product 2, Customer C Forecasts

As another example, forecasts for Product 2 bought by Customer C are found in Figure 8, for two different models. The top graph depicts forecasts (the blue line) for an MLR model with correlated errors, while the bottom graph provides forecasts (the orange line) made by another Bidirectional LSTM model. The MLR model appears to get closer to the actual values than the LSTM model, but the time series model may rely too heavily on the categorical month variables included in the model. Such behavior and assumptions are important factors to consider for the distributor, as a supplement to the quantitative metrics presented below.

Table 1. Product 1, Customer A: Average MAE for 24-Point Rolling Window Horizons

Model	6 mo. horizon	12 mo. horizon	23 mo. horizon
Naive 1	1.710692	1.751773	1.891908
Naive 2	1.555031	1.596631	1.751510
ARMA	1.471306	1.548560	1.670737
ARIMA ($d = 1$)	1.486079	1.547913	1.613743
ARUMA ($s = 5$)	1.755732	1.666493	1.811270
Signal-plus-noise	1.342166	1.469629	1.602031
MLR w/ correlated errors	1.758978	1.868723	1.921021
biLSTM	1.837374	2.085293	2.600941
CNN LSTM	2.159886	2.431345	3.223500
Multivariate LSTM	2.000546	3.961921	-

7.2 Model Performance

This section presents the results for all the forecasting models that were developed for each product/customer combination. In each of the tables presented, the color of the model names indicates whether it is a Naive (pink), Time Series (purple), or Deep Learning (blue) model. The darker green cells in the tables highlight the best MAE obtained for each of the 6, 12, and 23 month forecasting horizons, and the lighter green cells highlight the second best MAE found for each horizon.

For example, in Figure 1 it was initially observed that the behavior of Product 1 bought by Customer A was fairly stable over time. From the results in Table 1, a Signal-plus-noise model had the best results for all three forecasting horizons of 6, 12, and 23 months. This was one of the classical time series types of models. The next best models were also traditional time series models, ARMA and ARIMA.

In addition, Table 2 shows the full results for Product 2, Customer B sales. Figure 2 showed that the buying patterns for this product were very unstable, as there were two very large outliers at seemingly arbitrary times. For this high variance data, the hybrid CNN LSTM model performed best across all three horizons, and reduced the error 3–10% more than the next best ARMA time series model. This deep learning model may have performed best on this data because it was able to detect non-linear relationships, while traditional methods are limited to linear relationships.

Three additional tables are included in the Appendix, showing results for three more product/customer combinations. Table 6 shows that a Signal-plus-noise model greatly reduced the error for data that was consistently trending upward, for all three forecasting horizons. Tables 4 and 5 demonstrate more varied results, as no single model performed best for all three horizons. In these cases, a model should be chosen based on how many values into the future need to be forecasted.

Finally, Table 3 illustrates the results when forecasting demand for Product 2, Customer C. The buying pattern for this product was very seasonal, and each month this retailer bought the product in similar quantities from the previous

Table 2. Product 2, Customer B: Average MAE for 24-Point Rolling Window Horizons

Model	6 mo. horizon	12 mo. horizon	23 mo. horizon
Naive 1	8.700000	8.419149	9.616425
Naive 2	8.710220	8.372606	9.320441
ARMA	6.068292	5.646206	5.733381
ARIMA ($d = 1$)	8.060017	7.658063	7.931171
Signal-plus-noise	7.399033	7.603976	10.25761
MLR w/ correlated errors	8.159100	8.246929	11.07309
biLSTM	8.837178	9.322320	10.60560
CNN LSTM	5.810758	5.297931	4.773196
Multivariate LSTM	16.47646	10.70044	-

Table 3. Product 2, Customer C: Average MAE for 24-Point Rolling Window Horizons

Model	6 mo. horizon	12 mo. horizon	23 mo. horizon
Naive 1	4.576101	4.654255	5.139251
Naive 2	4.154245	4.204433	4.807548
ARMA	5.100733	5.063569	5.490468
ARIMA ($d = 1$)	6.441118	6.063819	6.697477
ARUMA ($s = 12$)	5.482179	5.291624	5.839697
Signal-plus-noise	4.982179	5.198895	5.564578
MLR w/ correlated errors	4.824322	5.321764	5.726380
biLSTM	4.795140	5.229527	5.711579
CNN LSTM	4.735278	4.886445	5.244143
Multivariate LSTM	7.101707	8.143090	-

year. Thus the second naive model, which averages the values from the two previous years, performed best across all three forecasting horizons. Future work could explore additional models or include more variables to try to outperform the naive model for this product.

7.3 Model Comparison

To understand these results at a higher level, the results from the tables above were aggregated into bar graphs, depicting the best MAE for each model type for a given horizon. Figure 9 illustrates these aggregated results for a 6 month forecasting horizon. Similar graphs were produced for 12 and 23 month forecasting horizons, found in Figures 10 and 11 in the Appendix.

Examining Figure 9, each bar represents the MAE of the best model for each class of models, which were Naive, Time Series, and Deep Learning. In this graph, a smaller bar represents lower error and thus a better model fit. From the plot it can be observed that the model performance varies across the different product/retailer combinations, but with five out of the six products at least one of the models showed improvement over the naive model. The greatest improvement over the naive model for this 6 month horizon came for Product 1, Customer C, where a time series model improved the forecasting error by 51.4%,

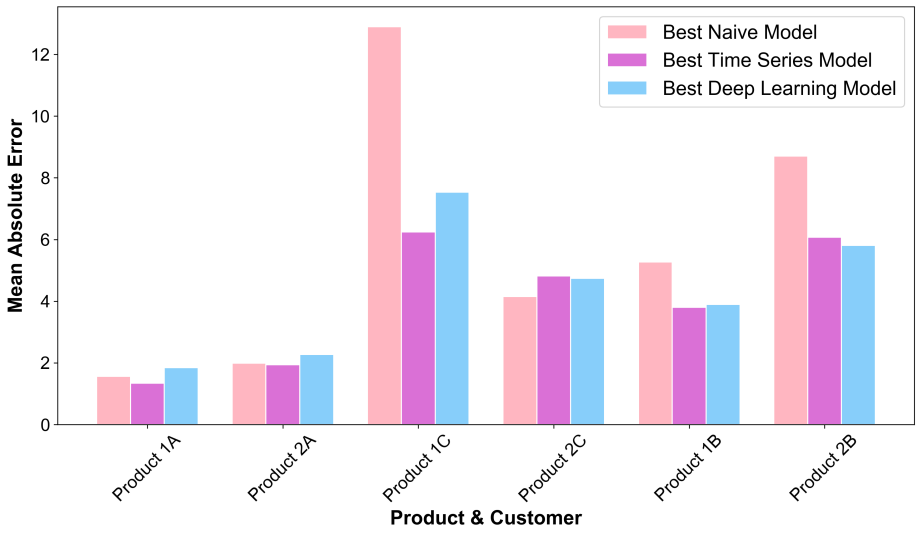


Fig. 9. Model Comparison for 6 Month Forecasting Horizon

meaning that it more than halved the MAE from the naive model. The smallest improvement occurred for Product 2, Customer A, where a time series model edged out the naive model by just 2.5%. Nevertheless, small percentages like these could translate to considerable money saved in a large-scale distribution setting.

For the short-term 6 month forecasts, it was observed that there were some products where a time series model outperformed the naive models, but none of the deep learning models did. Namely, this situation occurred for the sales of both Products 1 and 2 to Customer A. This was surprising because of the promise of deep learning methods. However, this may be due to most of the deep learning models being univariate, and traditional time series models have been found to often outperform machine learning methods in a univariate setting [14]. The variables in the dataset that were useful for prediction were limited, so mining for or generating additional variables that affect alcohol supply chains may have the potential to improve the performance of these deep learning models.

Similar patterns were found for the medium- and longer-term forecasts of 12 and 23 months, found in Figures 10 and 11 in the Appendix. For both horizons, improvement to the naive model was made by either a time series or deep learning model in five of the product/retailer combinations. For 12 months of forecasts, the smallest improvement observed was 5.2% (Product 2, Customer A), while the smallest improvement over 23 months was an 8.6% error reduction (Product 1, Customer A). On the other hand, the largest improvement was 52.4% for 12 months and 66.7% for 23 months (both for Product 1, Customer C). From these results, when the forecasting horizon was increased, the reduction in error improved as well. This is assuring because, even as uncertainty increases

when predicting farther into the future, the models showed progressively better performance compared to naive predictions.

It should be noted that for some products, even within the same class of model, different models actually performed best for each forecasting horizon. The individual-level product/customer tables should be consulted to find the best performing model for the desired horizon. However, all in all, considering all three forecasting horizons the models achieved between a 2.5% and 66.7% error reduction from the naive model, for the five products where improvement was seen. It was also observed that for most products, the best time series and best deep learning models had similar performance. The largest reductions in error occurred for Product 1 purchased by Customer C, which had an upward trending buying pattern.

8 Conclusions

The alcoholic beverage distribution company wanted data driven business insight into their company. This paper was able to present them with models that can forecast demand more accurately than a naive model for five out of the six products. The ability to accurately forecast demand will help the distribution company better manage their supply chain.

The biggest takeaway from the results presented above is that there is no one model that performs best overall. Each customer for each product has their own purchasing patterns, and one type of model does not generalize for all of them. The best way for the distribution company to forecast demand would be to model on the individual level and then add the predictions of multiple models to get an overall total estimate of demand for a product or a customer.

Despite the lack of an overall best model, there were some general conclusions that could be drawn. Classical time series models performed well overall and were able to outperform a naive model for five of the six product/customer combinations. Time series models performed best for stable or seasonal data and could reduce the error up to 67% compared to the naive model. The hybrid CNN model performed best for high variance data, reducing the error up to 49%. For one product/customer combination, no model was able to outperform the naive models. This data was extremely seasonal and no advanced model tested was able to capture that as effectively as the naive models.

9 Future Work

Data analysis for the alcoholic beverage distributing company will be continued by at least one additional data science capstone group. The models produced in this study were specific to products and customers, so a future group could investigate generalizing the models to a customer or product as a whole. In addition, the dataset was filtered to a particular city, so future work could research whether these results apply across regions. Another line of analysis could be seeing how the methods proposed here could be applied in different industries.

Other modeling approaches could be taken to provide more insight into the data. This project focused primarily on univariate approaches, which time series models excel at. To harness deep learning's ability to handle high-dimensional, multivariate inputs, a deeper dive into multivariate models could be taken. In addition, other hybrid approaches such as ensemble ARIMA and deep learning models have been found to be effective for short-term predictions [8]. Multilevel stacking models built on a combination of XGBoost models and Neural Network models for forecasting as described in previous research could be another avenue of investigation [20].

Beyond forecasting, an additional objective could be to implement anomaly detection for alcoholic beverage demand. Previous research used stacked LSTM networks for anomaly/fault detection in time series [16]. An Encoder-Decoder scheme was also used to detect anomalies from predictable, unpredictable, periodic, aperiodic, and quasi-periodic time series [15]. Previous research has explored methods for detecting anomalies that could be built upon and applied to the alcoholic beverage industry.

References

1. Altman, N.: An iterated Cochrane-Orcutt procedure for nonparametric regression. Journal of Statistical Computation and Simulation **40**(1-2), 93–108 (1992). https://doi.org/10.1080/00949659208811368
2. Bandara, K., Shi, P., Bergmeir, C., Hewamalage, H., Tran, Q., Seaman, B.: Sales demand forecast in e-commerce using a Long Short-Term Memory neural network methodology. In: Neural Information Processing. pp. 462–474. Springer International Publishing, Cham (2019)
3. Bao, T., Zaidi, S.A.R., Xie, S., Yang, P., Zhang, Z.: A CNN-LSTM hybrid framework for wrist kinematics estimation using surface electromyography (1 2020)
4. Brownlee, J.: Long Short-Term Memory Networks with Python. MachineLearningMastery (2017)
5. Brownlee, J.: Deep Learning for Time Series Forecasting Predict the Future with MLPs, CNNs and LSTMs in Python. MachineLearningMastery, v1.6 edn. (2019)
6. Brownlee, J.: Introduction to Time Series Forecasting with Python. MachineLearningMastery, v1.8 edn. (2019)
7. Chai, T., R.R.Draxler: Root mean square error (RMSE) or mean absolute error (MAE)? arguments against avoiding RMSE in the literature. Geosci. Model Dev. (2014). https://doi.org/10.5194/gmd-7-1247-2014
8. Gamboa, J.C.B.: Deep learning for time-series analysis (2017)
9. Geron, A.: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2nd edn. (9 2019)
10. Hyndman, R., Athanasopoulos, G.: Forecasting: principles and practice. OTexts (2018)
11. Kilimci, Z.H., Akyuz, A.O., Uysal, M., Akyokus, S., Uysal, M.O., Atak Bulbul, B., Ekmis, M.A.: An improved demand forecasting model using deep learning approach and proposed decision integration strategy for supply chain. Complexity **2019** (2019). https://doi.org/https://doi.org/10.1155/2019/9067367

12. Lancon, J., Kunwar, Y., Stroud, D., McGee, M., Slater, R.: AWS EC2 instance spot price forecasting using LSTM networks. SMU Data Science Review **2**(2) (2019), https://scholar.smu.edu/datasciencereview/vol2/iss2/8/
13. Levy, D., Sheflin, N.: The demand for alcoholic beverages: An aggregate time-series analysis. Journal of Public Policy & Marketing **4**, 47–54 (1985)
14. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Statistical and machine learning forecasting methods: Concerns and ways forward. PloS one **13**(3) (2018)
15. Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G.: LSTM-based Encoder-Decoder for multi-sensor anomaly detection (2016)
16. Malhotra, P., Vig, L., Shroff, G., Agarwal, P.: Long Short Term Memory networks for anomaly detection in time series. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. pp. 89–94. Bruges (Belgium) (4 2015)
17. Moroney, L.: Sequences, time series and prediction, deeplearning.ai on www.coursera.org (12 2019), [Lecture; viewed 1-January-2020]
18. Olah, C.: Understanding LSTM networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (8 2015), [Online; accessed 14-October-2019]
19. Pankratz, A.: Forecasting with Univariate Box-Jenkins Models: Concepts and Cases. Wiley Series in Probability and Statistics, Wiley (2009)
20. Pavlyshenko, B.: Machine-learning models for sales time series forecasting. Data **4**, 15 (1 2019). https://doi.org/10.3390/data4010015
21. Perera, S.: Rolling Window Regression: a Simple Approach for Time Series Next value Predictions (6 2016), [Online; accessed 10-April-2020]
22. Salinas, D.: DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting (2019), https://www.sciencedirect.com/science/article/pii/S0169207019301888
23. Salman, A.G.: Weather forecasting using merged Long Short-Term Memory model (LSTM) and autoregressive integrated moving average (ARIMA) model. Journal of Computer Science (2018). https://doi.org/10.3844/jcssp.2018.930.938
24. Sun, J., Di, L., Sun, Z., Shen, Y., Lai, Z.: County-level soybean yield prediction using deep CNN-LSTM model. Sensors (Basel) **19(20): 4363.** (2019). https://doi.org/10.3390/s19204363
25. Willmott, C., Matsuura, K.: Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate research **30**, 79–82 (2005)
26. Woodward, W.A., Gray, H.L., Elliott, A.C.: Applied Time Series Analysis with R. CRC Press, Boca Raton, FL, 2nd edn. (1 2017)
27. Zivot, E., Wang, J.: Modeling Financial Time Series with S-PLUS. Springer-Verlag, Berlin, Heidelberg (2006)

Appendix

The tables and graphs in this section display the rest of the results for each product/customer combination. Their contents are discussed in Section 7, but were moved here to save space in the main body of the paper. Tables 4, 5, and 6 present the respective MAEs for each model for Product 2 Customer A, Product 1 Customer B, and Product 1 Customer C. Figures 10 and 11 show the best model for each model type for 12 and 23 month horizons respectively.

Table 4. Product 2, Customer A: Average MAE for 24-Point Rolling Window Horizons

Model	6 mo. horizon	12 mo. horizon	23 mo. horizon
Naive 1	1.995556	1.941319	2.357860
Naive 2	2.336389	2.251910	2.606438
ARMA	2.171123	1.977445	2.026780
ARIMA ($d = 1$)	1.996195	1.979661	1.873152
Signal-plus-noise	1.941775	2.038029	2.498614
MLR w/ correlated errors	2.118729	2.161306	2.433706
biLSTM	2.494719	2.455633	2.033428
CNN LSTM	2.271335	2.259781	2.201930
Multivariate LSTM	3.649512	5.677353	-

Table 5. Product 1, Customer B: Average MAE for 24-Point Rolling Window Horizons

Model	6 mo. horizon	12 mo. horizon	23 mo. horizon
Naive 1	5.392857	5.405303	5.047431
Naive 2	5.274405	5.209470	6.042095
ARMA	3.798131	5.064172	6.763355
ARIMA ($d = 1$)	3.963743	4.035309	5.100716
Signal-plus-noise	4.793923	5.778166	6.385668
MLR w/ correlated errors	4.955327	5.532659	5.858596
biLSTM	3.894328	5.149653	6.595758
CNN LSTM	5.224087	4.851543	4.439607
Multivariate LSTM	6.516032	20.88898	-

Table 6. Product 1, Customer C: Average MAE for 24-Point Rolling Window Horizons

Model	6 mo. horizon	12 mo. horizon	23 mo. horizon
Naive 1	12.89062	12.82692	18.83768
Naive 2	18.79167	18.72212	23.58986
ARMA	10.26938	13.62094	19.70817
ARIMA ($d = 1$)	17.76884	19.47173	29.57861
Signal-plus-noise	6.245323	6.105684	6.279858
MLR w/ correlated errors	6.840409	7.020570	6.659496
biLSTM	8.726529	10.31896	13.87718
CNN LSTM	7.534186	8.523174	10.70080
Multivariate LSTM	9.490489	23.44610	-

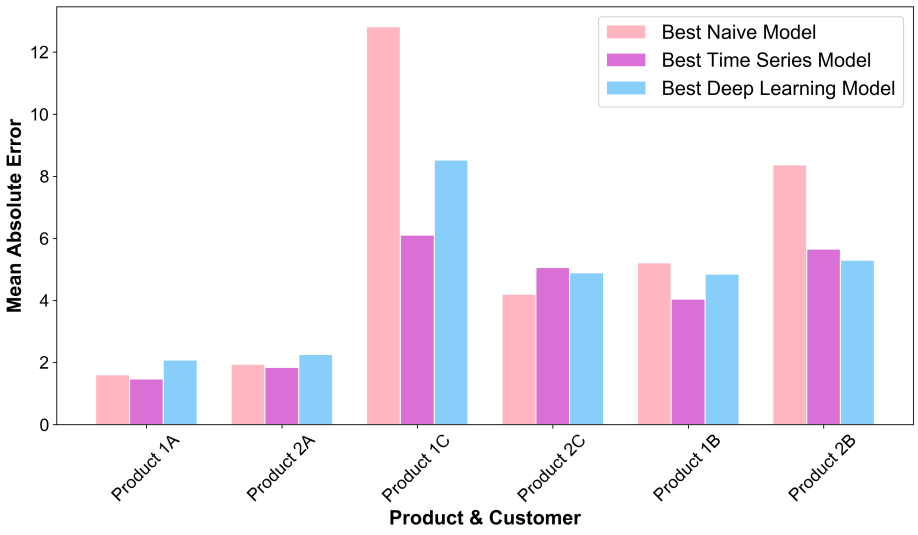


Fig. 10. Model Comparison for 12 Month Forecasting Horizon

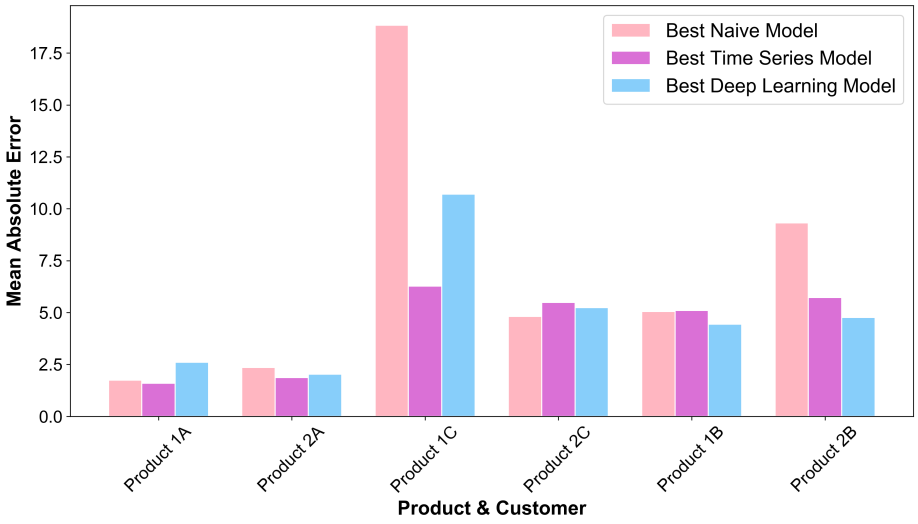


Fig. 11. Model Comparison for 23 Month Forecasting Horizon