SMU Data Science Review

Volume 3
Number 3 *Fall 2020*

Article 7

2020

Automated Machine Learning Framework for Demand Forecasting in Wholesale Beverage Alcohol Distribution

Jenna Ford
Southern Methodist University, jenna.a.ford@gmail.com

Christian Nava
Southern Methodist University, cjnava@mail.smu.edu

Jonathan Tan
Southern Methodist University, jhtan@mail.smu.edu

Bivin Sadler
Southern Methodist University, bsadler@mail.smu.edu

Automated Machine Learning Framework for Demand Forecasting in Wholesale Beverage Alcohol Distribution

Jenna Ford, Christian Nava, Jonathan Tan, Bivin Sadler
Master of Science in Data Science, Southern Methodist University,
Dallas, TX 75275 USA
{jennaf, cjnava, jhtan, bsadler}@smu.edu

Abstract. This paper covers the development, testing, and implementation of an automatic framework for analyzing and forecasting demand for an alcoholic beverage distributor's products at varying levels of granularity. Rather than look at macroscale geographic demand for a product from a distribution center, this framework will look at the localized customer level demand for that product before aggregating total demand. The approach will better capture individual behavior variations for each customer and allow for a more accurate estimation of the total monthly demand for that product. To best account for each product's influencing factors, each product is analyzed separately per customer with both traditional time series and contemporary machine learning models to identify the best performing forecasts. This research sets up an AutoML framework to individually identify the best forecasting model for different product and customer combinations.

1 Introduction

Demand forecasting is a vital part of retail business establishments. For a wholesale beverage alcohol distribution company, having too much supply on-hand can lead to excess storage costs, while not having enough supply on-hand leaves revenue on the table. In the United States, getting alcoholic beverages from producer to consumer is a three-tiered distribution process where a producer sells directly to a wholesale distributor who, in turn, sells to a direct retailer who then sells to a consumer. Predicting retail demand, or forecasting demand, for a wholesale distributor can be a valuable tool, which would allow the distributor to more accurately stock its inventory throughout the year. Demand forecasting is particularly valuable when dealing with a perishable product like beer, which may be discarded by, or returned for disposal to, the distributor, increasing the distributor's costs.

The demand profile for each product varies according to region, customer, and time of year. This may require a different model for each customer and product combination, which can result in a costly investment of resources. Additionally, not every product has enough historical data to be modeled and going through each store and product combination can be a laborious task for which a retail organization may not have adequate staff. Employing an automated machine learning (AutoML) solution can allow retail organizations with smaller teams to extract meaningful insights while lowering technical barriers. This study focuses on creating an AutoML approach to

demand forecasting where all product and customer combinations are forecasted and a winning model is selected with no supervision. Traditional time series models as well as deep learning models are incorporated into the framework. The overarching goal for this wholesale beverage alcohol distribution company is to forecast demand at a product level. Thus, product and customer combinations are aggregated together, after being forecasted, to create a forecast for demand at the product level.

Previous work on beverage alcohol distribution demand forecasting for this company focused on aggregated sales by product, and single product and customer combinations [1,2]. Traditional time series techniques and deep learning methods were employed in both studies. The results varied for the different product and customer combinations as to which model performed best.

There are two primary concerns to address for the AutoML framework for time series analysis: stationarity and whether the original dataset is white noise. Whether a time series is stationary will determine which models are appropriate and if any transformations need to take place. Stationarity is often determined based on visual inspection of the time series. However, with an AutoML framework, statistical tests will need to be used to evaluate stationarity. Additionally, if a time series is deemed to be white noise, there is no need to model the series and a simple equal means forecast is sufficient.

The benefit of an AutoML approach is the speed at which forecasts are delivered. However, prediction accuracy is expected to suffer slightly without human intervention. This kind of AutoML framework could be useful to make quick and impactful changes to the supply chain while a more in-depth analysis of each individual time series is undertaken.

The remainder of this paper is organized as follows: In Section 2, related literature is reviewed. Section 3 provides information about the dataset and time series that will be evaluated. Section 4 goes through a traditional exploratory data analysis (EDA) for one of the time series in the dataset. In Section 5, the AutoML framework is reviewed along with in-depth descriptions of how determinations are made for whether a time series is deemed to be white noise and whether a time series is deemed to be stationary. Section 6 provides an overview of the models used in the AutoML framework. In Section 7, model evaluation techniques are reviewed for identifying the winning model. Section 8 provides the results from the AutoML framework. Section 9 presents a Graphical User Interface developed for practical application of the AutoML framework. Section 10 highlights conclusions of this research and, in Section 11, topics for further research are explored.

2 Related Work

Previous work on demand forecasting has been done for the same company and location as this study. Aurora et al. (2020) performed demand forecasting on aggregated case sales for two products: Taaka Vodka 80 1L and Jack Daniel's Whiskey [1]. Aurora et al. used S-ARIMA, Vector Auto-Regression (VAR), Long Short-Term Memory Networks (LSTM), and ensemble models to forecast monthly case sales. A weighted ensemble model combining forecasts for S-ARIMA, VAR and LSTM was used for

Taaka Vodka and an average ensemble model was used for Jack Daniel's Whiskey. A rolling-window ASE was used to determine the best model for each product. For both products, the LSTM model achieved the lowest rolling-window ASE. Due to concerns of overfitting with the LSTM models, Aurora et al. identified the ensemble models as the models with the best fit. The root mean squared error (RMSE) for forecasting monthly case sales with the ensemble model was reduced 50% for Taaka Vodka and 33.5% for Jack Daniel's Whiskey, compared to the naïve forecasts of the same period from the previous year.

Jiang et al. (2020) also worked on demand forecasting for the same company, but a different location [2]. Jiang et al. focused on vodka products and noted that there are different seasonal patterns found for different products within the vodka category. Two vodka products for three different customers were ultimately selected for forecasting. One of the products displayed a strong seasonal trend and the other did not. The following models were run for each of the six time series: naïve using the monthly value from the previous year as the forecast, naïve using an average of the monthly value from the previous two years as the forecast, ARMA, ARIMA with d=1, ARUMA with s=5, signal-plus-noise, Multiple Linear Regression (MLR), biLSTM, CNN LSTM and a multivariate LSTM. The results indicated that in five of the six time series being forecasted, Jiang et al. were able to improve forecast accuracy compared to the naïve models. The conclusion was that there is no single model that performed best in all instances. This conclusion, in addition to the findings in Aurora et al., lead to the prospect of an AutoML approach to identify different models for different time series to achieve higher forecasting accuracy.

AutoML is a quickly growing field in Data Science with a goal of reducing human interaction in the process of model development [3]. AutoML algorithms typically create a static AutoML template by performing data preprocessing and feature selection followed by the primary task such as classification or regression [4]. This static template presents issues with parameter optimization and scalability [4]. A variety of AutoML tools are increasingly available in both for-purchase and open source environments. In reviewing open-source options, Budjač et al. notes that these tools are limited by the tasks they can be applied to and are not one-size-fits-all solutions [3].

Literature on AutoML for traditional time series applications is sparse. This is not surprising given the need to visually inspect the data to determine if the conditions for stationary are met before proceeding with modeling. Newer, deep-learning models do not typically suffer from the same constraints. A generalized regression neural network (GRNN) is a fast-learning model with a single design parameter that does not rely on the assumption of stationarity [5]. The model was awarded best prediction in the NN3 time-series competition among 60 models submitted. Additional work that involves time-series models and AutoML includes the use of multiple kernel learning (MKL) to automatically select the optimal size of sliding windows and find the pattern of the time series [6]. Allen and Balaji's work benchmarked the auto-sklearn and TPOT frameworks against H20's AutoML using datasets from OpenML and found auto-sklearn outperformed for classification datasets and TPOT outperformed for regression datasets [7]. However, Allen did not include mention of time series data. In their review of AutoML frameworks from a computer science and biomedical perspective, Waring, Lindvall, and Umeton (2020) present an effort to better utilize "off-the-shelf" machine

learning models [8]. They focus on open-source AutoML tools and find efficiency limitations of AutoML on large-scale datasets.

The authors are unaware of research on AutoML applied specifically to supply chain logistics or for retail demand forecasting. With respect to demand forecasting, Ahmed et al. (2010) compare several machine learning methods on business-type time series, more specifically, a subset of the monthly M3 time series competition data [9]. They found that multilayer perceptron, Gaussian processes, and Bayesian neural networks performed best among the models compared, and they noted that preprocessing can have a large impact on performance. Other studies have shown that holidays or special days can pose a challenge when forecasting retail demand. Huber and Stukenschmidt (2020) address the problem of forecasting daily demand in the presence of special days for a bakery chain by using artificial neural networks and gradient boosted decision trees [10]. They found that classification-based approaches outperformed regression-based approaches.

The objective of the AutoML application of this research is to use the resulting forecasts to make decisions on purchasing inventory. This implies that the accuracy of the forecasts is more important than their interpretability. As such, "black-box" models that are not easily interpretable are can be explored. Elsayed, Maida, and Bayoumi (2019) compare a long short-term memory (LSTM) model to a gated recurrent unit (GRU) model to create a hybrid, fully convolutional GRU (FCU-GRU) model [11]. They found that the FCU-GRU model outperformed the LSTM model for a univariate time series.

Improving the performance of AutoML models can be achieved by combining models or using pretrained models. Combining algorithms has also shown to improve performance [12, 13]. Noh et al. (2020) used a hybrid model using a genetic algorithm and a gated recurrent unit (GA-GRU) where the GA model was used to find the optimal hyperparameters of the GRU model [12]. They found the GA-GRU model outperformed ARIMA, LSTM, and RNN models. Helmini, Jayasinghe, and Perera (2019) use an LSTM with "peephole connections" on the Rossmann data set for sales forecasting and found that the peephole connection LSTM outperformed extreme gradient boosting (XGB) and random forest models [14]. Additionally, LSTM models tend to outperform traditional ARIMA models in certain use cases. Weytjens, Lohmann, and Kleinsteuber (2019) use an LSTM model to forecast cash flows and compared the LSTM model's performance to ARIMA, multiple-layer perceptron (MLP), Facebook's Prophet forecasting tool [15]. They found that the LSTM model outperformed ARIMA, MLP, and Prophet for periods between 1 and 30 days.

Pretrained models are those that have been trained on other datasets that are similar to the data of interest. Using metadata or pretrained models can lead to increased speed in AutoML, which can benefit use cases where data with a similar distribution are generated on a frequent basis [16]. This is particularly interesting for an alcohol beverage distributor that generates weekly sales data.

When developing an AutoML framework for time series, it is critical to evaluate stationarity. Many formal tests have been developed over the years to test for stationarity. One type of formal test tests the null hypothesis that a unit root is present, such as the augmented Dickey-Fuller (ADF) test and the Phillips-Perron test [17, 18]. A unit root is a factor of $(1 - B)$ from a characteristic equation of a time series; the presence of which indicates the series is not stationary. Unit root tests have difficulty

distinguishing between a unit root and factors close to a unit [17]. Another category of stationarity tests test the null hypothesis that the time series is stationary around a mean or linear trend against the alternate hypothesis that there is a unit root. Examples of this type of unit root test are the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) and Leybourne-McCabe (LMC) tests [18]. KPSS tests are frequently used in combination with ADF tests since they test for different kinds of stationarity. These tests for stationarity test for only one unit root at a time. To test for multiple unit roots, these tests are performed multiple times after transforming the data. Taylor (2003) performed Monte Carlo simulations to support an idea that tests like KPSS are negatively impacted when there are additional unit roots present in the time series and recommends pre-filtering the time series with a first difference before performing the test for stationarity to minimize this [19]. A variety of non-parametric tests for stationarity are becoming popular in recent literature to test for stationarity. van Delft, Characiejus, and Dette (2018) propose an L^2 distance test which measures the difference between the spectral density of a non-stationary time series and the best approximation of its stationary counterpart [20]. Jin, Wang, and Wang (2015) propose an automated test to determine if the autocorrelation structure of a time series changes when taking systematic samples of the data [21]. Woodward, Gray and Elliott (2017) recommending using tests for stationarity in combination with other knowledge about the time series to make a determination on stationarity [17].

3 Dataset

The dataset was provided by a large, U.S.-based, beverage alcohol distribution company. The data is monthly case sales from 2013-2019 for a single U.S. metropolitan region. Each row of data represents one month of standard case sales for one product and one customer. Each customer represents a unique store location. There are 4,017 different products, 34 different customers, and a combined total of 37,391 different combinations of products and customers to forecast.

Thirty-one columns of data are provided. The most important are the customer IDs, product names, number of standard cases sold, and total purchase price per transaction. Other variables are included about each product such as alcohol content, product categorization, and container volume, however, these are not as relevant.

Many of the product and customer combinations found in the dataset have a sparse number of records, which suggests that the product is not purchased on a consistent monthly basis. Missing observations were filled in with case sales of zero and a total purchase price of zero dollars. Product and customer combinations missing more than fifty percent of monthly data for the 2013-2019 period were removed.

4 Exploratory Data Analysis

Exploratory data analysis (EDA) was conducted for a sample of ten product and customer combinations. What follows in this section is a detailed EDA for one of the sampled products and customer combinations.

Determination of stationarity is generally the first step in time series analysis. A stationary time series will have a constant mean, constant variance, and autocorrelations that depend only on the time between observations. As shown in the top plot of Fig. 1, there appears to be wandering behavior and possibly a difference in the mean in the latter half of the series. There is insufficient evidence to suggest that the variance is not constant. Autocorrelation (ACF) plots are used to help determine if the autocorrelation structure differs for different segments of time. Autocorrelations that change over time imply a non-stationary time series. This can be seen in the bottom graphs of Fig.1 where the ACF plots for the first half of the time series differs from the second half of the time series and suggests that the autocorrelation structure is not constant over time. This time series does not appear stationary since it does not meet the conditions of constant mean and constant autocorrelation.

Time series modeling requires stationary data. For the time series in Fig. 1, a transformation is necessary since the time series is not stationary. Taking a first difference is a typical transformation to apply. Reexamining the same plots used in Fig. 1 after taking a first difference of the data indicates that the time series now appears to be stationary. Another possible transformation is differencing for a seasonal component. The Parzen window in Fig. 1 shows a slight peak at 0.1667 and at 0.25, which suggests possible seasonal pattern of s=6 or s=4, respectively. Reexamining the same plots used in Fig. 1 for each of the seasonal differences indicates that these transformations do not appear to make the time series stationary. A first difference is the appropriate transformation for this time series.

A determination of whether the time series is white noise needs to be made as well. The middle left graph in Fig. 1 is a plot of the autocorrelations (ACF). At a 95% confidence level, approximately one lag out of 20 would be expected to cross outside of the blue stripped bands if this series was white noise. With six lags extending beyond the 95% confidence level, this series does not appear to be white noise. Ljung-Box tests with K=10 and K=24 were run as another test for white noise. At as significance level of 0.05, the chi-square value for K=10 was 74.99 with a p-value less than .0001 and the chi-square value for K=24 was 124.31 with a p-value less than .0001; we reject the null hypothesis that this dataset is white noise. The Ljung-Box test indicates this dataset may not be white noise.

Human-performed data analysis relies on visual inspection of a variety of plots and performing statistical tests. In an Auto-ML framework, however, visual inspection is omitted as a process, and the issues of stationarity and white noise need to be addressed by statistical tests alone.

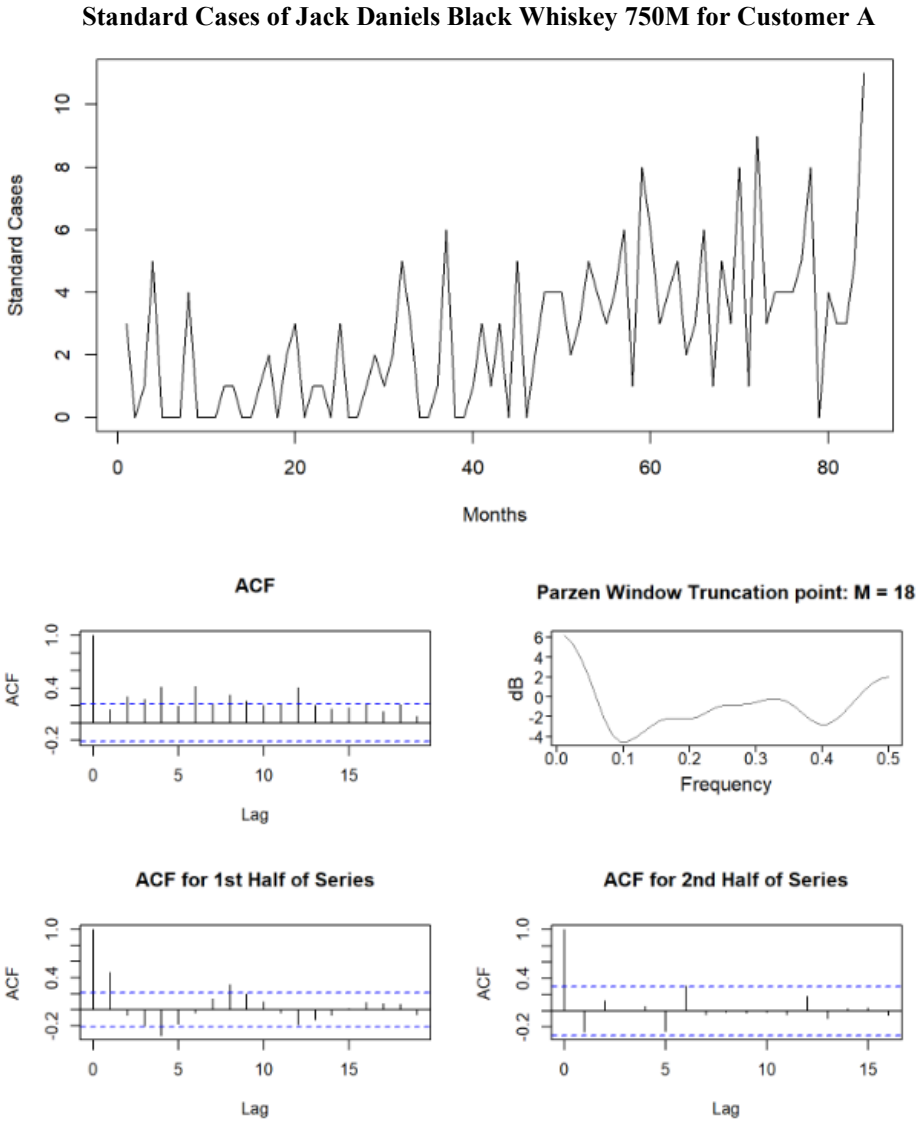


Fig. 1. EDA performed for one of the product and customer groups.

5 AutoML Framework

The primary goal for the AutoML framework is to determine which model most accurately forecasts the number of standard cases sold, without the need for human intervention. This framework is indifferent to whether a times series is sent through aggregating all customers for a product or at a product and customer level.

The AutoML framework begins by making determinations on white noise and stationarity. Ljung-Box test and an evaluation of ARMA parameters are used to indicate if the time series is deemed to be white noise. If a time series is truly white noise, the equal means model, explained in section 6.1, is expected to outperform other models. A similar approach is taken for stationarity. Augmented Dickey-Fuller (ADF) and Kwiatkowski–Phillips–Schmidt–Shin (KPSS) tests are run to test for stationarity. Section 5.2 explains why both tests need to agree for a determination to be made on stationarity. Time series that meet the minimum number of observations, as discussed in Section 3, are modeled independent of the determination of white noise and stationarity. Sections 5.1 and 5.2 explain how the determinations of white noise and stationarity are made and why they are important in the AutoML framework. Section 6 will provide details about the forecasting models used inside the AutoML framework. For each product and customer combination forecasted, the winning model will be displayed, along with a note indicating whether the time series was deemed to be white noise and stationary.

5.1 White-Noise

As reviewed in the exploratory data analysis, Section 4, visualization of the time series is typically the first thing a data scientist does when attempting to determine if the time series is white noise. Forecast residuals are also typically reviewed after modeling to determine if the residuals are white noise. If the residuals are not white noise, this suggests that further modeling may better explain the behavior in the data.

Whether reviewing a time series or forecast residuals after modeling, a visual inspection and a Ljung-Box test can be employed to assist with making a determination on white noise. In the AutoML framework developed here, where a visual inspection of the time series is not applicable, a Ljung-Box test and the presence of an ARMA(0,0) model as one of the top five ARMA models are used to evaluate stationarity. The Ljung-Box test approaches the autocorrelations as a group to determine if the residuals are white noise. It tests the null hypothesis, H_0, that all autocorrelations, ρ, are zero (i.e., the residuals are white noise). If at least one autocorrelation is not zero, then white noise is not present.

$$H_0: \rho_1 = \rho_2 = \cdots = \rho_K = 0$$
$$H_a: at\ least\ one\ p_k \neq 0\ for\ 1 \leq k \leq K. \quad \textbf{(1)}$$

The number of autocorrelations to test needs to be determined to run the Ljung-Box test. Hyndman and Athanasopoulos (2018) suggest using $K = \min(10, n \div 5)$ when the time series does not have a seasonal pattern and the $K = \min(2 * seasonal\ period, n \div 5)$ when the time series does have a seasonal pattern [22]. Respective to the Hyndman and Athanasopoulos suggestions, Ljung-Box tests with K=10 and K=24 are performed in the framework here ($N = 84$ for all time series forecasted in this paper). If the results from these two tests differ, then the Ljung-Box test is inconclusive.

Estimating parameters for an ARMA model may offer insight into whether the time series is white noise. It is used in this framework as an additional piece of evidence. The top five ARMA models are generated using the aic5.wge function of the R package tswge. The Bayesian Information Criterion (BIC) is used to evaluate different models and the five models with the lowest BIC are determined. The presence of an ARMA(0,0) selected as one of the top five models by the BIC process suggests the dataset may be white noise. This evaluation method is not as conclusive as the Ljung-Box test and serves as an additional piece of information gathered about the determination of white noise.

An AutoML approach reduces the time and effort put into identifying an optimal model for the data. For this AutoML framework all potential models are fit on the dataset regardless of the determination of white noise. Indications will be given to the user of the framework as to the determination of white noise. If the winning model happens to be something other than the equal means model explained in Section 6.1, then the user can determine if the equal means model is more appropriate.

5.2 Stationarity

As discussed in Section 4, a data scientist visually inspects a time series to see if the three conditions for stationarity are met. This section discusses conditions of stationarity in more detail and how stationarity will be accounted for in the AutoML framework.

The first condition of stationarity is constant mean. If a time series has a constant mean, the mean does not depend on time. A linear trend where the mean increases over time or a seasonal or cyclic pattern are possible reasons for a non-constant mean. For example, monthly temperature data would show higher temperatures in summer months and lower temperatures in winter months. This pattern is predictable and expected.

The second condition of stationarity is constant variance. If a time series has constant variance, the variance does not depend on time. This condition is more difficult to evaluate. If the first condition of stationarity is not met, it is increasingly difficult to make a determination about constant variance. If multiple realizations of a time series can be imagined, the variances for each time point should not change throughout the series, if the variance is constant. For example, consider the time series in Fig. 2 that represents the daily difference between the high and low temperature in Austin, Texas from 2014-2016. The variation seen in winter months is much larger than the variation seen in summer months. This dataset does not have constant variance.

The third condition of stationarity is that autocorrelations depend only on how far apart the observations are, not where in time the observations are. This can be viewed by creating separate ACF plots for different ranges of the times series. As discussed in Section 4, the bottom row of ACF plots in Fig. 1 shows an ACF plot using the first half of a time series and another ACF plot using the second half of the same time series. The patterns in these two plots should match if the third condition of stationarity is met. For the time series in Fig. 1, the patterns in the ACF plots do not match and this time series does not meet the third condition of stationarity.

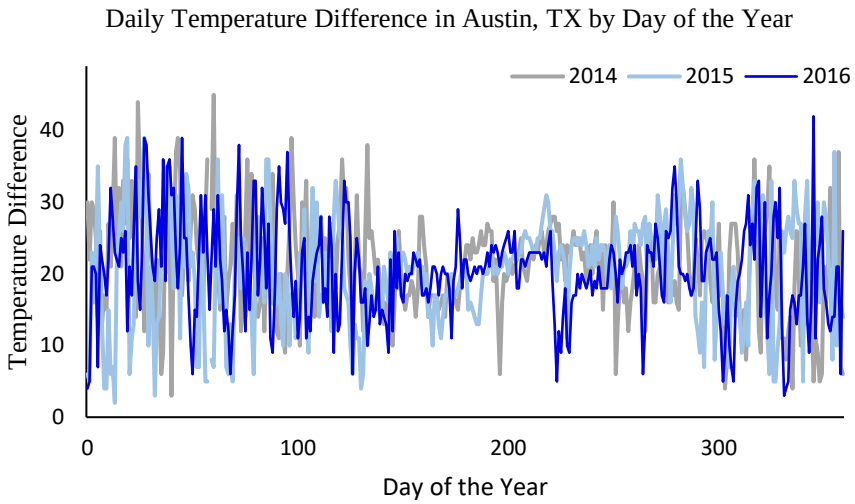


Fig. 2. Daily difference between high and low temperatures in Austin, TX.[1]

Many forecasting models assume that a given time series is stationary. If a non-stationary time series was modeled using a model that assumes stationarity, the forecast residuals would be larger than if the time series was first transformed and then modeled. Typical transformation options are differencing or averaging. Transforming the time series with a first difference would remove a linear trend in the data. Other differencing techniques are used to remove seasonal or cyclic behaviors in the dataset. Stationarity is then reassessed on the transformed data. This process repeats until a stationary dataset is identified.

Where no human interaction exists, evaluating how and when to transform data poses a problem for an AutoML framework. For the AutoML framework developed in this paper, a first difference and a twelve-month seasonal difference are applied as transformations. If non-stationary data is fit on a model that assumes stationarity, then larger forecast residuals would be expected. Therefore, there is little concern regarding the appropriateness of the winning model since the model will minimize the residuals.

ADF and KPSS tests for stationarity will be used to provide the user with guidance about stationarity. The ADF tests the null hypothesis that there is a unit root. The KPSS tests the null hypothesis that the time series is stationary around a mean or linear trend. Agreement between these two tests would lend evidence to support a determination of stationarity. However, it is recommended to use these tests in conjunction with other known information about the time series. For the AutoML framework for this paper, an indicator will be displayed to show the determination of stationarity by these two tests.

[1] Gruben, M. (2017, August). *Austin Weather* (Version 3) [Data set]. Retrieved July 10, 2020 from https://www.kaggle.com/grubenm/austin-weather/data#.

6 Forecasting Models

Traditional equations and algorithms, such as ARMA/ARIMA models, that represent past, present, and future values as stochastic variables are the most common and basic form of time series analysis. These algorithms forecast future values of a time series by calculating the statistical likelihood of future values.

More complex and recent approaches such as decision trees, multilayer perceptron, and long short-term memory networks have only recently made practical advances in computing capacity and speed. These algorithms forecast the future variables of a time series in different ways and do not require stationarity as do traditional time series models. What follows in this section are descriptions of the models used in the AutoML framework used for this research.

6.1 Equal Means

The equal means model takes the mean of the time series and uses that value as the forecast. Residuals are calculated as the difference between the forecasted value and the mean. This model is most appropriate for a time series that is white noise. In a white-noise time series, previous observations do not help forecast future observations, and a mean is the best forecast available.

6.2 ARMA

Autoregressive moving average, ARMA(p, q) is a traditional model for univariate time series analysis. The autoregressive (AR) portion of the model uses regression to represent each value of the time series relative to previous values by expressing the current value, X_t, as a function of past values and a white noise term, ϵ_t.

$$X_t = \mu + \sum_{i=1}^{p} \varphi_i X_{t-i} + \epsilon_t \quad . \tag{2}$$

The moving average (MA) portion of the model uses a moving average with q number of coefficients. It quantifies the moving average of error terms for each point in the series, where error is the difference between expected and observed values, q, is the number of error terms that are averaged by the model. If, for example, $q = 3$, then the previous three terms are averaged for each point.

$$X_t = \mu + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \quad . \tag{3}$$

An additional variation of the ARMA model used in this AutoML framework is an AR(p) model where $q = 0$.

6.3 ARIMA

An ARIMA(p, d, q) model is an autoregressive integrated moving average model that contains the same AR and MA components as an ARMA model with an additional integrated component. This integrated component represents the order of difference applied to the time series.

$$X_t' = X_t - X_{t-1} \quad . \tag{4}$$

The first order difference (d) of a time series is the difference between a single point in the series and its neighbor, as expressed in (4) above. This can be useful for making non-stationary time series appear more stationary by stabilizing the variance and allowing the application of stationary-dependent techniques. An additional variation of this model where $q = 0$ is also used in the AutoML framework.

$$X_t' = X_t - X_{t-s} \quad . \tag{5}$$

Seasonality is another component of time series analysis that must be taken into account when modeling. Seasonality is the presence of an identifiable pattern within the time series such as cyclical, consistent increases or decreases in values. Removing the seasonal component is accomplished by taking the difference between a single point in the series and a single previous point S observations before, as expressed in (5) above. Seasonal trend can be identified by several methods, such as autocorrelation plots, spectral density estimation, or visual inspection of a realization. A seasonal ARIMA model with the term $S = n$ can account for cyclical changes that repeat every n terms in the time series. Examples of pattern identification would be $S = 7$ in a daily data for a weekly pattern, $S = 26$ in weekly data for a biannual pattern, and $S = 12$ in hourly data for a 12-hour pattern. An additional variation of this model where $q = 0$ is also used in the AutoML framework.

6.4 MLP

A multilayer perceptron (MLP) is a type of artificial neural network. MLP models for univariate time series forecasting usually contain a single hidden layer of nodes or neurons and an output layer used to make the prediction. Backpropagation is used to adjust the weight and bias of each neuron to approximate the relationship between points of the time series. Variations in parameters such as the number of layers, layer size, and training repetitions are used to find a balance between getting close to the expected results and overfitting.

6.5 Random Forest

By treating the next possible value of a time series as a selection from a finite number of choices, a decision tree can be used to forecast a single future-value of a time series. A random forest is an ensemble of outputs from multiple decision trees fit to the data. The advantages of such a model are the relative transparency and ease of use. A random forest model is nonlinear and does not make assumptions about the data of the time series like an ARIMA model or other statistically based models. The disadvantages of this approach are that a random forest can technically only predict $t + 1$ time periods. Forecasting more than one unit into the future requires using a previous forecast as the basis for the next forecast. While the decrease in accuracy as the forecast horizon is extended is common in all forecasting methods, the results of a longer-term, decision-tree-based forecast is entirely based on how well the model represents the initial data before it starts propagating outwards.

7 Model Evaluation Method

There are various ways to identify the winning model. The AutoML framework for this paper uses a rolling-window average squared error (ASE) to identify the model that has the most accurate forecasts over time. Section 7.1 details the process and calculations for computing a rolling-window ASE.

Even if the time series appears to not be white noise and the winning model is not the equal means model, it is useful to check if there is a statistically significant difference between the equal means model and the winning model. An analysis of variance (ANOVA) test can be performed to determine if there is a statistically significant difference between the model with the lowest rolling-window ASE and the equal means model. Section 7.2 provides the methodology to determine if there is a statistically significant difference between models.

7.1 Rolling-Window ASE

A rolling-window ASE will be used to measure the goodness of fit for model performance. The ASE measure takes the sum of the square of the difference between the predicted value (forecast), $\hat{y}_i$, and the actual value, y_i. It then averages the error over the number of observations. A lower ASE value indicates a more accurate model.

$$ASE = \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n} \quad . \tag{5}$$

It should be noted that the ASE is a snapshot in time and can vary for the same dataset depending on the size of the training data. It uses $n - k$ values, where n is the length of the time series, to train the model and then uses the last k values to validate forecasted values.

A more useful approach is to shorten the training period and fit the model on a smaller training set (a shorter "window" of time) and then validate the data on the subsequent k values. In a process similar to cross-validation, the training set, or window, then “rolls” or “slides” to the subsequent period and is evaluated repeatedly. Fig. 3 shows which observations are included in the training set and which observations are forecasted for different windows. Once the windowing process has completed to the end of the dataset, the ASE values for each window are averaged together to get a rolling-window ASE.

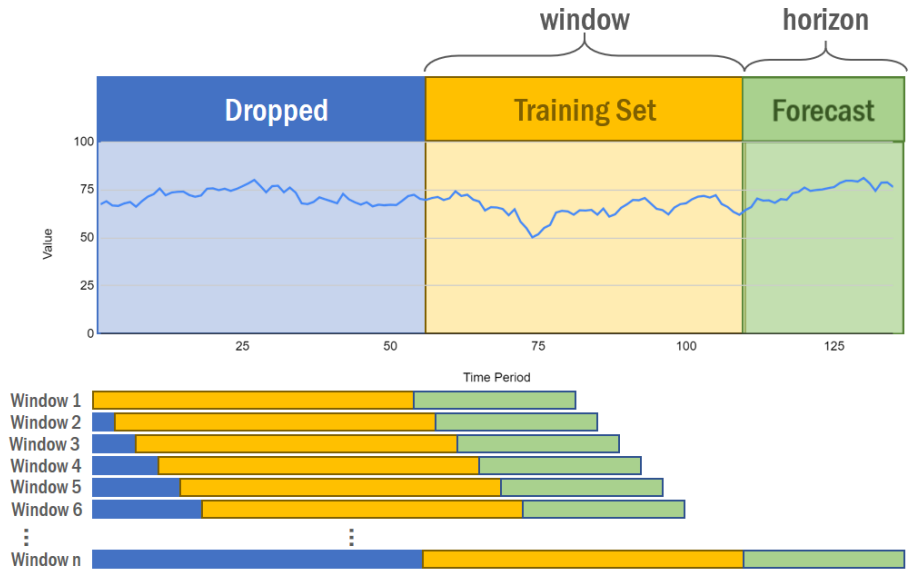


Fig. 3. Rolling-window training and test splits.

The rolling-window ASE method can prove to be a more stable representation of the overall model ASE. For example, if there was some particularly odd behavior in previous observations of a time series, a single ASE could be misleading. The winning model is determined as the model with the lowest rolling-window ASE.

7.2 ANOVA

An ANOVA is performed to determine if the AR models are statistically different than the equal means model. This calculation is performed for three models, regardless of whether one is the winning model: AR, ARIMA with $d = 1$ and $q = 0$, and ARIMA with $s = 12$ and $q = 0$.

Fig. 4 shows the ANOVA table used to calculate an F-statistic. In order to calculate the F-statistic, the degrees of freedom and sum of squared residuals must be calculated for each model. Fig. 4 compares the equal means model to the AR model. The equal means model has $n - 1$ degrees of freedom. The AR model has $n - (p + 1)$ degrees

of freedom, where n is the number of observations in the dataset, and p is the number of autoregressive components in the model. The number of parameters, p, is increased by 1 if the mean of the model is not specified. If the p-value associated with the F-statistic calculated from the ANOVA table with (df_{Extra}, df_{AR}) degrees of freedom is less than 0.05, then the null hypothesis, that the AR and equal means models do not differ in forecasting precision, can be rejected.

	Degrees of Freedom	Sum of Squared Residuals	MS	F-statistic
Extra Sum of Squares	$df_{Extra} = df_{EM} - df_{AR}$	$SS_{Extra} = SS_{EM} - SS_{AR}$	$MS_{Extra} = \frac{SS_{Extra}}{df_{Extra}}$	$\frac{MS_{Extra}}{MS_{AR}}$
AR Model	$df_{AR} = n - (p + q + 1)$	$SS_{AR} = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$	$MS_{AR} = \frac{SS_{AR}}{df_{AR}}$	
Equal Means Model	$df_{EM} = n - 1$	$SS_{EM} = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$		

Fig. 4. ANOVA table to determine if there is a statistically significant difference between the equal means model and the AR model for a time series.

A white noise time series should be best modeled by an equal means model because there is no identifiable, repeatable pattern. However, it is possible for another model to occasionally be a better fit for a white noise time series. In this instance, it is useful to determine whether there is a statistically significant difference between the models. To accomplish this, a simulation was performed using the ANOVA table in Fig. 4 to determine the distribution of the F-statistic. Ten-thousand white noise time series were generated with one-hundred observations each. An equal means model and an AR(2) model were generated for each of the ten-thousand time series and an F-statistic was calculated. The resulting density plot in Fig. 5 shows the distribution of the F-statistic for this simulation and compares this to an F-distribution with (3, 97) degrees of freedom.

This same simulation was attempted with ARMA models. However, adding in the moving average of the error terms did not produce an F-statistic distribution similar to the true F-distribution. Therefore, the analysis to determine if a model is statistically different than the equal means model cannot be done on models with an MA component. This same simulation was also not run for the MLP or RF models. Inferences, however, can be made regarding the models with an MA component and the MLP and RF models. For example, assume the RF model was the winning model for a particular time series because it had the lowest ASE. Also, assume that the AR model performed better than the equal means model. If the ANOVA test comparing the AR model to the equal means model indicates that these models are statistically different, then it can be inferred that the RF model is also statistically different than the equal means model because the RF model has a lower rolling-window ASE than the AR model.

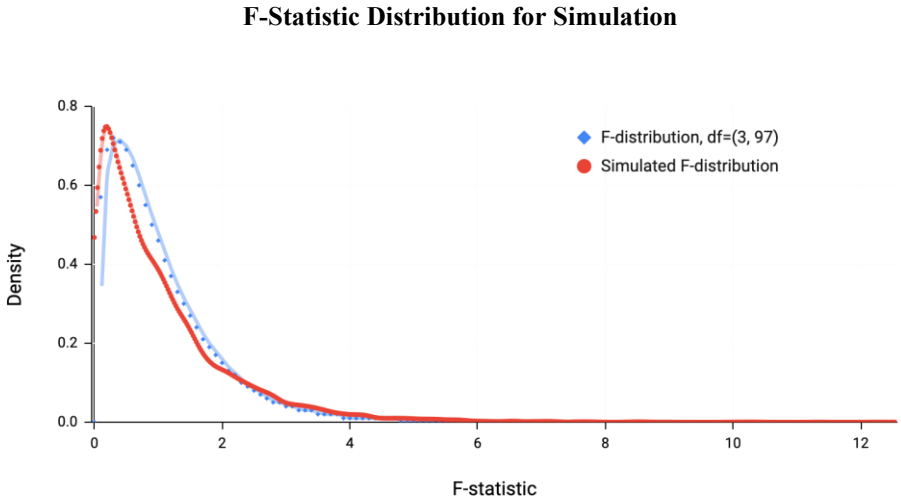


Fig. 5. F-statistic distribution for a simulation of 10,000 white noise time series with associated equal means and AR(2) models.

8 Results and Analysis

A simple, random sample of ten product and customer combinations was run through the AutoML framework. Detailed results for three product and customer combinations are reviewed in the remainder of this section. Results for forecast horizons of one, three, and twelve months are provided in Tables 1-3.

8.1 Jack Daniels Black Whiskey 750M for Customer A

Table 1 shows model results for Jack Daniels Black Whiskey 750M for customer A. All methods for detecting white noise indicate that this time series is not white noise. The ADF test yielded a *p*-value of 0.01, which rejects the null hypothesis that a unit root is present. This would indicate that the time series is stationary. The winning model for the one-month and thee-month forecast horizons is the ARMA model. The winning model for the twelve-month forecast horizon is the ARIMA model, where a first difference is applied (d=1) and no MA term is used ($q = 0$).

The ANOVA test results are displayed in the last column of Table 1. If at least 70% of the rolling-window *p*-values indicate that there is a difference in the model compared to the equal means model, then the models are said to be different. If only 30% of the rolling-window *p*-values indicate that there is a difference in the models, the models are said to be the same. Percentages between 30% and 70% are inconclusive. For this time series, the ARIMA model with $q = 0$ and d=1 is statistically different than the equal means model. It can also be inferred that the ARMA model is statistically different than the equal means model. This is because the AR model is statistically

different than the equal means model, and for the one-month and three-month forecasts, the ARMA model has lower ASE values. For the twelve-month forecast this inference can be made because the ARIMA model with $q = 0$ and d=1 is statistically different than the equal means model and the ARMA model has a higher ASE.

Table 1. Model results for Jack Daniels Black Whiskey 750M for Customer A.

White Noise Results	Stationarity Results	Model	Rolling-Window ASE 1-Month Forecast	3-Month Forecast	12-Month Forecast	F-statistic Conclusion
Not white noise	Stationary	Equal Means	9.06	9.19	10.61	
		AR	5.61	6.41	7.41	Different
		ARMA	5.54	6.36	6.87	
		ARIMA, q=0, d=1	6.01	6.60	6.79	Different
		ARIMA, d=1	8.54	8.67	9.31	
		ARIMA, q=0, s=12	8.77	9.92	9.99	Inconclusive
		ARIMA, s=12	9.15	10.05	9.98	
		RF	6.01	6.99	7.01	
		MLP	7.98	12.28	14.52	

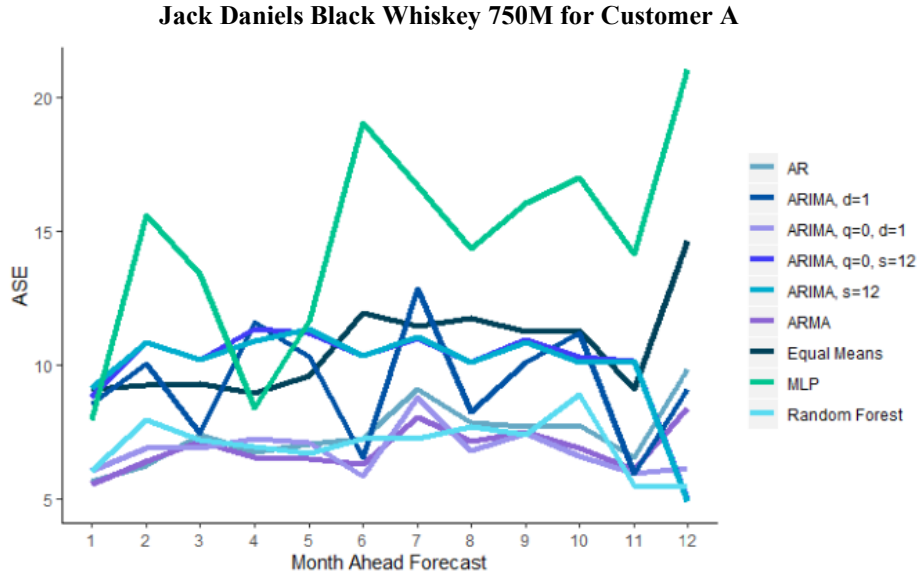


Fig. 6. ASE results by month-ahead and model for Jack Daniels Black Whiskey 750M for Customer A.

It may also be of interest to see how the different models perform at different forecast horizons. Fig. 6 shows the average ASE by forecast month (not calendar month) for the models in the AutoML framework. Month 1 in the figure averages the ASE for the first month of the forecast from each rolling-window. Month 2 averages the ASE for the

second month of the forecast from each rolling-window, etc. Fig. 6 shows that the forecast accuracies for the ARIMA model with *d*=1 and the MLP models have wide swings in prediction accuracy from month to month.

8.2 Tortilla Silver Tequila DDS 1.75L for Customer B

Table 2 shows model results for Tortilla Silver Tequila DDS 1.75L for Customer B. All methods for making a determination on white noise indicate that this time series is white noise. Stationarity tests indicate the time series is not stationary. The winning model for the one-month and three-month forecast horizon is the equal means model, as is expected for a white noise time series. For the twelve-month forecast horizon, the equal means model and the ARIMA model with *d*=1 have the same ASE. However, the determination that this time series may be white noise indicates that the winning model should be the equal means model. The indication that this time series may not be stationary is irrelevant if the time series is truly white noise. Furthermore, stationarity is not an assumption for the equal means model. Since the winning model was an equal means model, the conclusions from the ANOVA tests are also irrelevant.

Table 2. Model results for Tortilla Silver Tequila DDS 1.75L for Customer B.

White Noise Results	Stationarity Results	Model	Rolling-Window ASE			F-statistic Conclusion
			1-Month Forecast	3-Month Forecast	12-Month Forecast	
White noise	Not stationary	Equal Means	0.57	0.56	0.72	
		AR	0.69	0.63	0.74	Same
		ARMA	0.72	0.68	0.75	
		ARIMA, q=0, d=1	0.71	0.67	0.74	Same
		ARIMA, d=1	0.64	0.62	0.72	
		ARIMA, q=0, s=12	1.18	1.05	1.07	Same
		ARIMA, s=12	2.03	1.65	2.18	
		RF	0.70	0.64	0.89	
		MLP	0.60	0.66	0.79	

The average ASE by forecast month (not calendar month) for the models in the AutoML framework is plotted in Fig. 7. Fig. 7 shows that the forecast accuracies for the ARIMA with $s = 12$ are worse than the other models. Many of the models have very similar ASE values and are difficult to distinguish in the plot, as is expected from the results in Table 2. As would be expected, these models show that as the forecast horizon gets further out, the ASE increases.

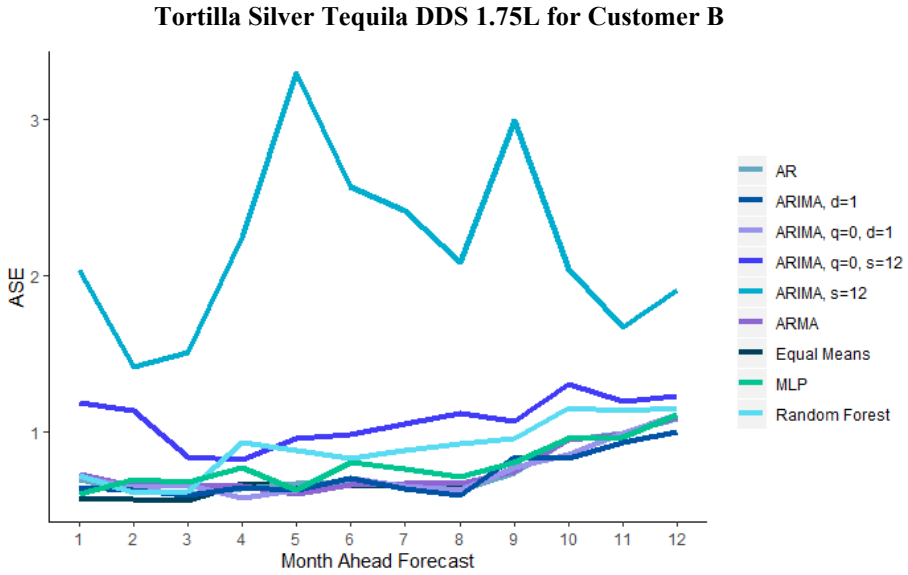


Fig. 7. ASE results by month-ahead and model for Tortilla Silver Tequila DDS 1.75L for Customer B.

8.3 Casa Noble Crystal Tequila 6PK 750M for Customer C

Table 3 shows model results for Casa Noble Crystal Tequila 6PK 750M for Customer C. The different methods for determining if the time series is white noise both indicate the times series may not be white noise. The ADF test failed to reject the null hypothesis that a unit root is present, indicating the time series may not be stationary. The winning model for the one-month, three-month and twelve-month forecast horizons is the equal means model. If the model is truly not white noise, then it is reasonable to assume that a model could be found that would provide a better forecast than the equal means model. More transformations and models could be added to the ML framework for time series such as this one.

The average ASE at each month-ahead forecast by the different models in the AutoML framework is plotted in Fig. 8. Fig. 8 shows that the forecast accuracies for the MLP model fluctuate significantly. The equal means, ARMA, and ARIMA with $d = 1$ models are grouped together at the bottom of the plot and have consistent ASE values throughout the twelve-month forecast.

Table 3. Model results for Casa Noble Crystal Tequila 6PK 750M for Customer C.

White Noise Results	Stationarity Results	Model	Rolling-Window ASE 1-Month Forecast	3-Month Forecast	12-Month Forecast	F-statistic Conclusion
Not white noise	Not stationary	Equal Means	3.17	2.76	1.89	
		AR	4.60	3.44	2.07	Inconclusive
		ARMA	4.62	3.46	2.10	
		ARIMA, q=0, d=1	5.73	4.12	2.36	Inconclusive
		ARIMA, d=1	5.77	4.04	2.33	
		ARIMA, q=0, s=12	5.71	7.42	6.65	Same
		ARIMA, s=12	6.31	8.20	6.84	
		RF	5.37	6.87	5.32	
		MLP	25.17	18.46	13.76	

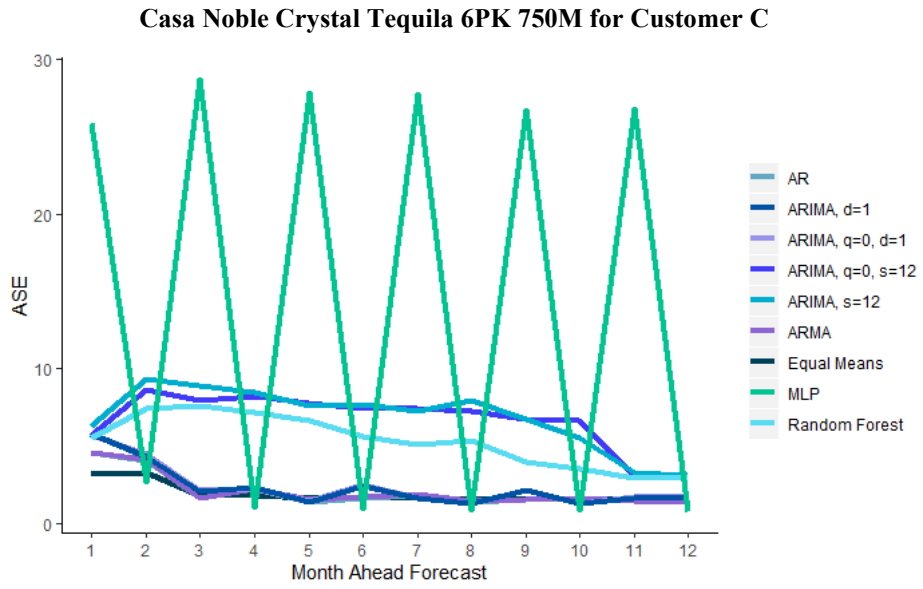


Fig. 8. ASE results by month-ahead and model for Casa Noble Crystal Tequila 6PK 750M for Customer C.

8.3 Aggregated Forecasts vs. Forecasted Aggregate

Aurora et al. (2020) forecasts Taaka Vodka 80 1L by first aggregating all standard case sales and then forecasting [1]. Using the AutoML framework discussed in this paper, a test was run to compare the results from aggregating detailed customer level demand forecasts to the results from forecasting based on the aggregated data.

Forecasting demand for Taaka Vodka 80 1L standard case sales at the customer level may identify and account for unique sale patterns at individual stores. Time series for all Taaka Vodka 80 1L standard case sales by customer were run through the AutoML framework. As noted in Section 3, not all time series meet the criteria of having enough data points to be able to forecast standard case sales. Of the thirty-three customers who purchased Taaka Vodka 80 1L standard cases between 2013 and 2019, four customers did not have enough data points to forecast. For these customers, the mean standard-case sales were used as the forecast. For the remaining twenty-nine customers, the AutoML framework used the smallest rolling-window ASE value for a twelve-month forecast to determine the winning model. The monthly forecasts for 2019 for each customer's winning model were summed with the mean from the four customers without enough data to create a total demand forecast for Taaka Vodka 80 1L. Fig 9 shows the forecast results of the AutoML framework for this method in the red line. The ASE for the months in 2019 was 4,726.

The data for all customers who purchased Taaka Vodka 80 1L between 2013 and 2019 was aggregated by month in order to compare the results from aggregating forecasts to forecasting from aggregated data. This single time-series was sent through the AutoML framework. The winning twelve-month model based on the lowest rolling-window ASE was the equal means model. The mean value is used as the forecast and can be seen by the black line in Fig. 9. The ASE for the months in 2019 was 9585, which is significantly higher than the ASE from aggregating the customer level forecasts.

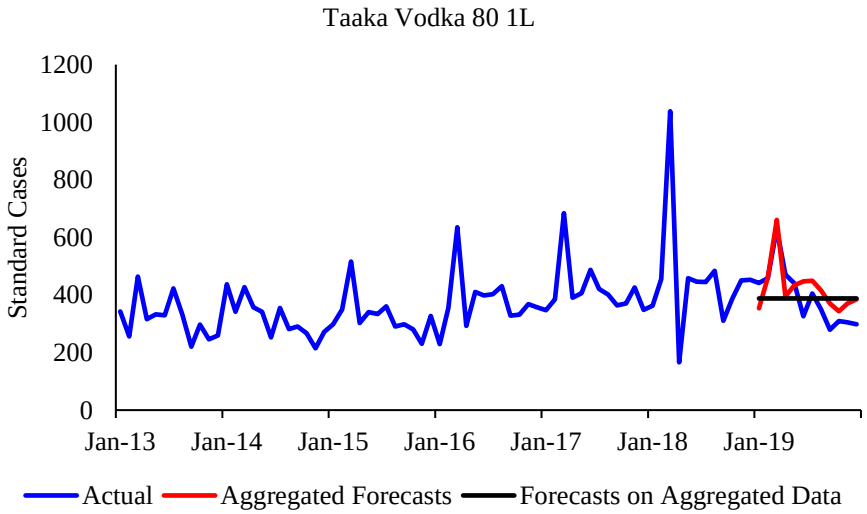


Fig. 9. Aggregated Taaka Vodka 80 1L standard case sales with 12-month forecasts. Aggregated forecasts are the sum of forecasts at the customer level. Forecasts on aggregated data sum all sales and then forecast.

9 Practical Application

The AutoML framework developed in this paper was implemented into an R Shiny graphical user interface (GUI). The purpose of the GUI is to forecast demand for a product by aggregating individual customer level forecasts. The user selects the product to forecast, as seen in Fig. 10. The output in Fig. 11 graphically shows historical case sales by month and a twelve-month forecast. These results can assist in making inventory decisions over the course of the next twelve months.

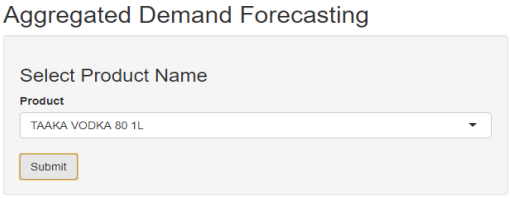
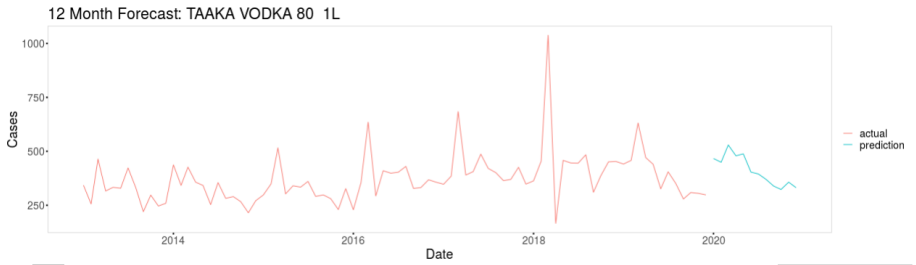


Fig. 10. GUI for the AutoML framework. User selects the product and then clicks 'Submit' to obtain forecasts for the aggregated demand of the selected product.



Show entries Search:

	date	cases
1	2020-01-01	466.285889740109
2	2020-02-01	449.326733431107
3	2020-03-01	529.232899965201
4	2020-04-01	479.165292958682
5	2020-05-01	487.929699500315
6	2020-06-01	403.463998228551
7	2020-07-01	394.883040064846
8	2020-08-01	370.002869143944
9	2020-09-01	338.901067907515
10	2020-10-01	323.064348742371
11	2020-11-01	356.924288493628
12	2020-12-01	331.450281959055

Fig. 11. Displayed results from the R Shiny GUI.

10 Conclusion

In this paper, a time series AutoML framework was developed to identify the model with the most accurate forecasts of standard case sales for a large, beverage alcohol distribution company in the United States. The AutoML framework includes tests for white noise and stationarity and then evaluate the performance of a variety of different models, both traditional and deep learning. A rolling-window ASE was used to identify the model with the most accurate forecasts over time. This framework allows for different models to be identified as the best model for different product and customer combinations. It should not be assumed that one model will work best for every combination.

Tests for white noise were performed to see if modeling in fact needs to occur for the time series. The framework did not stop models from being run on time series that were identified as possibly being white noise. The results of thirty-nine different product and customer combinations for Taaka Vodka 80 1L were reviewed and ten of those were deemed to possibly be white noise. Of these ten, an equal means model was selected as the winning model for the twelve-month forecast horizon for only six. It is recommended to review the results for the remaining four product and customer combinations to determine whether an equal means model may be more appropriate.

Tests for stationarity were also performed to assist with checking assumptions for the traditional ARMA-type models. All models were run, regardless of the proposed determination of stationarity. A reason for running all models independent of a determination of stationarity is that the ADF unit root test used has low power and has trouble distinguishing between a unit root and a root close to a unit root.

As shown in Section 8, the AutoML framework can be used at different levels of data found in the dataset. Forecasting at the product and customer level and then aggregating the forecasts produced a more accurate forecast than forecasting on aggregated data. This may not be the case for all datasets, but the AutoML framework developed here makes testing this quick and simple.

11 Future Work

For an AutoML framework to be useful for a different dataset, some modifications to the programming are required. These modifications should include the ability to add any number of explanatory variables and generalizing the dataset and variable names throughout the code. Data preparation would typically be required regardless of any generalizability obtained in the code, but these changes would make it faster for a user to drop a time series into the framework to identify the best model.

This paper introduced tests for stationarity but did not integrate these tests with possible transformations. For the purposes of this paper, two transformations were used, regardless of the outcome of the tests for stationarity. Stationarity test integration with data transformation would be especially useful when using an AutoML framework on a different dataset. This could enable better results from the various models in the process.

Finally, there are more models that could be added to this framework. For example, LSTM and multivariate models could be added to improve model performance. Ensemble models could also be evaluated as Aurora et al. (2020) were able to achieve lower rolling-window ASE values with this method [1].

References

1. Arora, T., Chandna, R., Conant, S., Sadler, B., & Slater, R. (2020). Demand forecasting in wholesale alcohol distribution: An ensemble approach. *SMU Scholar*.
2. Jiang, L., Rollins, K. M., Ludlow, M., & Sadler, B. (2020). Demand forecasting for alcoholic beverage distribution. *SMU Scholar*.
3. Budjač Roman, Marcel, N., Peter, S., Zahradníková Barbora, & Janáčová Dagmar. (2019). Automated machine learning overview. Research Papers.Faculty of Materials Science and Technology.Slovak University of Technology in Trnava, 27(45), 107-112. doi:10.2478/rput-2019-0033.
4. Mohr, F., Wever, M., & Hüllermeier, E. (2018). ML-plan: Automated machine learning via hierarchical planning. Machine Learning, 107(8-10), 1495-1515. doi:10.1007/s10994-018-5735-z.
5. Yan, W. (2012). Toward automatic time-series forecasting using neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 23(7), 1028-1039. doi:10.1109/TNNLS.2012.2198074.
6. Widodo, A., Budi, I., & Widjaja, B. (2016). Automatic lag selection in time series forecasting using multiple kernel learning. *International Journal of Machine Learning and Cybernetics*, 7(1), 95–110. https://doi.org/10.1007/s13042-015-0409-7.
7. Allen, A., Balaji, A. (2018). Benchmarking Automatic Machine Learning Frameworks. arXiv.org. http://search.proquest.com/docview/2092781703/.
8. Waring, J., Lindvall, C., & Umeton, R. (2020). Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artificial Intelligence in Medicine, 104*. doi:10.1016/j.artmed.2020.101822.
9. Ahmed, N. K., Atiya, A. F., Gayar, N. E., & El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. Econometric Reviews: The Link between Statistical Learning Theory and Econometrics: Applications in Econometrics, Finance, and Marketing, 29(5-6), 594-621. doi:10.1080/07474938.2010.481556.
10. Huber, J., & Stuckenschmidt, H. (2020). Daily retail demand forecasting using machine learning with emphasis on calendric special days. *International Journal of Forecasting*, 36(2). doi:10.1016/j.ijforecast.2020.02.005.
11. Elsayed, N., Maida, A. S., & Bayoumi, M. (2019). Gated recurrent neural networks empirical utilization for time series classification. *IEEE*. doi:10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00202.
12. Jiseong Noh, Hyun-Ji Park, Jong Soo Kim, & Seung-June Hwang. (2020). Gated Recurrent Unit with Genetic Algorithm for Product Demand Forecasting in Supply Chain Management. *Mathematics* (Basel), 8(565). https://doi.org/10.3390/math8040565.
13. Weng, T., Liu, W., & Xiao, J. (2019). Supply chain sales forecasting based on lightGBM and LSTM combination model. *Industrial Management & Data Systems*, 120(2), 265–279. https://doi.org/10.1108/IMDS-03-2019-0170.
14. Helmini, S., Jayasinghe, M., & Perera, S. (2019). Sales forecasting using multivariate long short term memory network models. PeerJ PrePrints. https://doi.org/10.7287/peerj.preprints.27712v1.

15. Weytjens, H., Lohmann, E., & Kleinsteuber, M. (2019). Cash flow prediction: MLP and LSTM compared to ARIMA and Prophet. *Electronic Commerce Research*, 1–21. https://doi.org/10.1007/s10660-019-09362-7.
16. Tuggener, L., Amirian, M., Rombach, K., Lorwald, S., Varlet, A., Westermann, C., & Stadelmann, T. (2019). Automated Machine Learning in Practice: State of the Art and Recent Results. *2019 6th Swiss Conference on Data Science (SDS)*, 31–36. https://doi.org/10.1109/SDS.2019.00-11.
17. Woodward, W., Gray, H., & Elliott, A. (2017). *Applied Time Series Analysis with R* (2nd edition). Boca Raton: Taylor & Francis.
18. Gimeno, R., Manchado, B., & Mínguez, R. (1999). Stationarity tests for financial time series. *Physica A: Statistical Mechanics and its Applications*, 269(1), 72-78. doi:https://doi-org.proxy.libraries.smu.edu/10.1016/S0378-4371(99)00081-3.
19. Robert Taylor, A.,M. (2003). Robust stationarity tests in seasonal time series processes. *Journal of Business & Economic Statistics*, 21(1), 156-163. doi:10.1198/073500102288618856.
20. van Delft, A., Characiejus, V., & Dette, H. (2018). A nonparametric test for stationarity in functional time series. Ithaca, United States Ithaca, Ithaca: Cornell University Library, arXiv.org. doi:http://dx.doi.org.proxy.libraries.smu.edu/10.5705/ss.202018.0320.
21. Jin, L., Wang, S., & Wang, H. (2015). A new non-parametric stationarity test of time series in the time domain. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(5), 893-922. doi:10.1111/rssb.12091.
22. Hyndman, R.J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. Accessed on September 18, 2020.