Local Business Short-Term Demand Forecasting Exploration Leveraging Neural Network Architectures

Carmen Pelayo Fernández

May 2024

Abstract

In an era where precision in operational management can dictate the success of a business, accurate demand forecasting is more crucial than ever, especially for industries like the restaurant sector dealing with perishable goods. This paper presents a comprehensive analysis comparing the effectiveness of traditional and deep learning forecasting models on real-world data from a local café. Traditional models, including ARIMA and Holt-Winters, are juxtaposed against modern deep learning models, specifically NeuralProphet and N-BEATS, to assess their capabilities in a practical business context. The results indicate that deep learning models, particularly NeuralProphet, significantly outperform traditional methods in terms of accuracy, as quantified by the Mean Absolute Percentage Error (MAPE). Additionally, this research explores the enhancement of forecasting accuracy through the integration of multivariate data inputs such as weather conditions and annotations of special events, demonstrating that incorporating external information and qualitative data from the business activity can substantively improve model predictions. For those interested in implementing the discussed models or exploring the dataset and methodology further, the complete code and resources are available on GitHub at `https://github.com/carmenpelayo/DLDemandForecast`. Overall, this paper not only validates the superior performance of neural network-based models in a real-world setting but also provides a framework for local businesses to leverage cutting-edge AI technologies for enhanced decision-making.

1 Introduction

Demand forecasting plays a key role in every business, as it can dramatically help in optimizing costs. Restaurants particularly benefit from forecasting tools, as they deal with perishable goods that require rapid consumption. Other advantages include maximizing customer satisfaction, optimizing labor allocation or enhanced menu planning (5-OUT, n.d.). Acknowledging the critical value of

accurate forecasts, this paper aims to develop a framework that enables local businesses to anticipate market shifts and better tailor their services.

In recent years, deep learning methods have significantly advanced the state of forecasting, surpassing traditional statistical models in their ability to learn from time series, capture nonlinear relationships, and automatically detect seasonality and trends without explicit specification. This paper explores both classical forecasting methods, which have provided a reliable baseline for understanding traditional forecast dynamics, as well as modern deep learning approaches, potentially offering a new paradigm in predictive analytics that this research seeks to validate.

2 Related Work

Time series forecasting has long been a focal point of both academic research and practical application. Traditional methods such as *Autoregressive Integrated Moving Average (ARIMA)* and the *Seasonal Holt-Winters method* have established strong foundations by effectively handling seasonal patterns and trends in time series data. These models are well-understood and serve as benchmarks in many studies, including this one, due to their simplicity and proven track record in various industries.

In recent years, the introduction of deep learning techniques has reshaped the landscape of forecasting. The *NeuralProphet* model, for instance, integrates the robust time-series decomposition ability of Facebook's Prophet model with the flexibility of neural networks, providing enhanced accuracy through the automatic detection of seasonality and trends –which are often manually specified in traditional models. Similarly, the *Neural Basis Expansion Analysis Time Series (N-BEATS)* model offers a unique approach by decomposing the forecast into a series of basis functions, thus allowing for an interpretable yet powerful forecasting tool.

This research distinguishes itself from previous works by not only comparing traditional and modern approaches under the same experimental conditions but also by focusing on a specific challenge faced by many local businesses, forecasting their service demand. Unlike many studies that deploy these models on standardized datasets, this paper utilizes real-world data from a local café, integrating external factors such as weather conditions, which significantly impact operations. Furthermore, this study leverages qualitative data to achieve superior accuracy. Overall, this research aims provide a simple, yet comprehensive exploration and subsequent comparison of the existing forecasting options available today for local businesses.

3 Methodology

The focus of this study is to test different forecasting methods in a practical scenario, so as to observe the real-life usability of both classical and neural-network-based prediction models. For illustration purposes, this study will revolve around the case of a local café that seeks to predict their daily meal counts. This particular task was chosen as it is representative of the basic forecasting needs that local businesses may face. The variable to be predicted here was chosen to be the number of meals served daily, as it is the fundamental indicator of this café's operational flow and was simple to collect, properly accommodating for the limited data collection resources that local businesses may face. Other businesses may benefit from predicting different indicators (e.g., revenue, product sales, etc.).

This study employs time series analysis to forecast daily meal counts, a continuous dataset that inherently displays seasonality. This seasonality offers a clear pattern for forecasting models to leverage, although it introduces the challenge of non-stationarity—where the mean, variance, or covariance vary over time—requiring transformations such as differencing to stabilize the series for more accurate predictions. The forecasting horizon was strategically set to two weeks. This period aligns with the café's operational logistics, particularly its supply chain and staffing requirements. However, this prediction window could be adjusted to suit different businesses' needs, which may require shorter or longer prediction intervals based on their specific operational dynamics.

The approach to forecasting in this study is twofold: univariate and multivariate. The univariate approach relies solely on historical meal count data, providing a baseline by focusing purely on internal historical trends. In contrast, the multivariate approach incorporates external variables such as weather conditions, special events, or revenue data, potentially enriching the model's context and enhancing prediction accuracy. This study aims to explore both approaches to determine their effectiveness in real-world applications.

Overall, the study seeks to answer the following questions:

1. *Do deep learning models outperform traditional methods in forecasting the demand of a business?*
2. *Which deep learning models are most effective for short-term demand forecasting?*
3. *Can the accuracy of forecasting models be improved by incorporating additional variables (e.g., weather, revenue, etc.)?*
4. *How can we maximize information extracted from basic datasets collected with limited resources in local businesses?*

5. *How much can local businesses rely on artificial intelligence (AI) to guide their operational decisions?*

To answer these, a double study will be conducted, encompassing both (a) the training of short-term forecasting models, and (b) text analysis methods to leverage the qualitative data in regular business datasets.

3.1 Short-Term Forecasting

Identifying the optimal forecasting model is the primary goal of this project, as such, this will constitute the central part of the study.

First, classical forecasting models will be explored, setting an initial accuracy baseline. On the one hand, some *ARIMA (AutoRegressive Integrated Moving Average)* models will be fitted, following the ideas developed by George Box and Gwilym Jenkins in the 1970s. ARIMA is a traditional model that combines autoregressive terms, differencing, and moving averages to handle data with trends and seasonal effects. Its robustness and simplicity make it suitable for time series with strong seasonal patterns, often observed in daily business operations like those in this café. On the other hand, the *Seasonal Holt-Winters Method* is a smoothing-based model that was developed by Charles Holt in 1957 and later extended by Peter Winters. It is particularly well-suited for forecasting where historical patterns, such as weekly or annual cycles, play a significant role, as it is the case here.

Next, two deep learning architectures will be explored, looking to validate the superior capability of neural networks against classical methods in short-term demand forecasting. *NeuralProphet* is an adaptation of Facebook's Prophet algorithm, which integrates neural networks with decomposable time series models and was enhanced to capture seasonality and trends dynamically. This model is advantageous for datasets where these patterns are not strictly regular but vary over time. Additionally, *Neural Basis Expansion Analysis Time Series (N-BEATS)*, introduced by researchers at Element AI in 2019, uses a series of fully connected neural network blocks, focusing on different time series components. This model is excellent for handling complex and noisy data, providing robust forecasts even in the absence of explicit seasonal or trend components.

The key difference between traditional methods (*ARIMA, Holt-Winters*) and deep learning approaches (*Neural Prophet, N-BEATS*) lies in the latter's ability to automatically adapt to data complexities without extensive manual tuning, providing potentially more accurate and flexible performance in diverse scenarios.

3.2 Text Analysis

This project will later investigate the incorporation of qualitative data (in the form of notes from the business owner in this case) to enhance the forecasting

accuracy, making it a multivariate problem. By making use of natural language processing (NLP) techniques, the sentiment in daily notes will be scored using both traditional (*VADER*) and deep learning large language models (*transformers*). The aim is to determine whether the sentiments obtain correlate with fluctuations in meal demands, potentially providing a predictive edge.

First, a simple sentiment score will be assigned (on the 0-10 scale) to text notes using a *Valence Aware Dictionary and sEntiment Reasoner (VADER)*, which works well with informal language. VADER uses a combination of a sentiment lexicon, which is a list of words labeled according to their semantic orientation as either positive or negative, along with a set of rule-based modeling for text analysis.

Further deepening the analysis, deep-learning transformer models are employed. Transformers are particularly effective in sentiment scoring, as they consider the full context of a word by using both its left and right surroundings. This is different from simpler NLP models (like VADER), which consider each word's meaning separately. In this study, three of the most popular sentiment-scoring transformers for the English language in HuggingFace –a platform hosting large language models– were chosen. All three architectures are variations of the original *BERT (Bidirectional Encoder Representations from Transformers)*, an architecture developed by researchers at Google AI Language in 2018 that is bidirectionally trained, enabling a deeper language context than single-direction language models.

This dual approach using traditional and cutting-edge NLP methods allows the project to leverage both established and novel techniques to enrich the forecasting model with additional variables.

4 Data

This study uses data from the Vanilla Bean Café's (located in Pomfret, CT) spanning over two decades (01/01/2000 to 12/31/2023). The dataset includes information about the number of daily breakfast, lunch, and dinner servings, as well as the revenue earned and some notes about the weather, day of the week, and general events. Figure 1 displays the raw dataset for reference.

4.1 Cleansing

To prepare the data for the analysis, various initial preprocessing actions were performed, including the drop of unnecessary columns, the renaming of untitled columns, and the validation of data formats.

Subsequently, an exploratory analysis was conducted to verify the quality of the data and/or detect possible anomalies. A total of 164 missing values were

	Unnamed: 0	Unnamed: 1	Date	Day	Breakfast	Lunch	Dinner	Total	Weather	Notes	Unnamed: 10	Avg Spend
0	NaN	NaN	2023-12-31	Sunday	88	105	0	193	Overcast	NaN	3649.27	18.908135
1	NaN	NaN	2023-12-30	Saturday	56	149	28	233	PM Rain	NaN	4643.05	19.927253
2	NaN	NaN	2023-12-29	Friday	30	154	17	201	PM Rain	NaN	3715.30	18.484080
3	NaN	NaN	2023-12-28	Thursday	26	154	21	201	PM Rain	NaN	3677.01	18.293582
4	NaN	NaN	2023-12-27	Wednesday	43	138	13	194	PM Rain	NaN	3239.03	16.696031
...	...	...	...	...	...	...	...	...	...	...	...	...
9074	NaN	NaN	2000-01-05	Wednesday	8	110	20	138	NaN	NaN	1320.27	9.567174
9075	NaN	NaN	2000-01-04	Tuesday	5	92	0	97	NaN	NaN	1077.91	11.112474
9076	NaN	NaN	2000-01-03	Monday	3	79	0	82	Part Sunny	Warm	840.04	10.244390
9077	NaN	NaN	2000-01-02	Sunday	55	161	29	245	NaN	NaN	2259.14	9.220980
9078	NaN	NaN	2000-01-01	Saturday	2	91	0	93	Sunny	Open 11-5 Slow Start	1064.21	11.443118

Figure 1: Raw Dataset

then identified (i.e., the count of meals served was 0), likely representing days where the restaurant was closed (maybe due to holidays, construction, or other unusual events). This is important since leaving these observations just in blank can be interpreted by the algorithm as days with very bad business performance, instead of considering that there were no operations at all. In every data science project, missing values should either be dropped or filled in with interpolated data to prevent the impact of null values on predictions. Given the time series nature of the problem, dropping values would discontinue the series, potentially leading to inaccuracies. Therefore, missing values will rather be interpolated from nearby data. There are multiple methods to do so, including the *mean imputation*, *median imputation*, *Last Observation Carried Forward (LOCF)*, *Text Observation Carried Backward (NOCB)*, *linear interpolation* or the *spline imputation*. Given that there are only a small number of missing values, we can perform a *spline interpolation*, which is the most computationally expensive method, but also the most accurate for capturing complex trends and subtle changes in time series data. It estimates missing values by fitting a flexible, curved line through the data points.

Figure 2 depicts the evolution of meal counts after the dataset preprocessing (the data in the plot was resampled to quarterly observations just for visualization purposes). From this plot, several insights can be obtained:

- First, the data seems to follow a clear seasonality, with a fairly constant trend over the years and low variance (at least until 2021). Throughout the entire series, we can see that the first quarter of the year (i.e. the months of January, February, March) produces the lowest earnings in the year, with the second (April, May, June) and third (July, August, September) quarters (the warmer months) being the most profitable ones.

- Interestingly, the observed seasonality is broken with the presence of a clear outlier during 2023. This year, revenue almost reached 3 times the value of other years' revenue. This could be due to either the effect of external factors or issues in the data collection process. Duplicated values

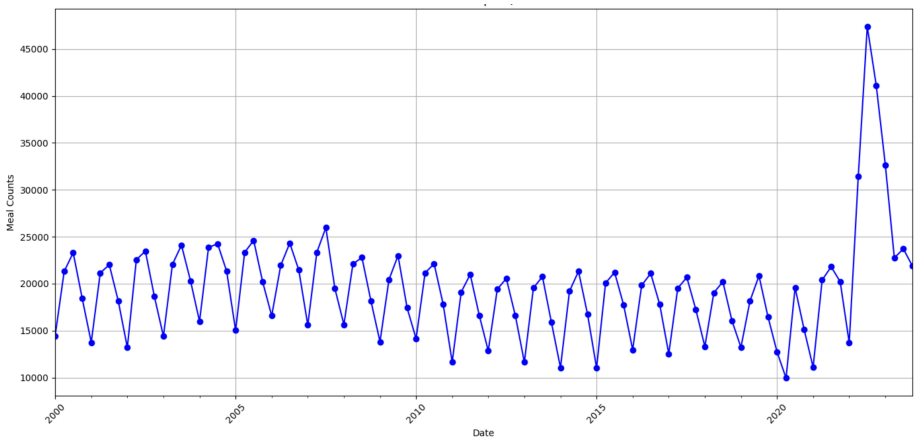

Figure 2: Preliminary Visualization of the Evolution of Meal Servings

were then investigated in the dataset, identifying 314 repeated values, which were eliminated (keeping the first observation only).

After addressing this issue, the evolution of meal counts was re-plotted, now exposing a much more reasonable pattern (see Figure 3).

Another thing to note is that variables in the dataset are linearly dependent on others. For instance, the meal count in a day corresponds to the sum of breakfast, lunch, and dinner servings; and the average customer spending is obtained by dividing the revenue by the meal counts. This reinforces the choice of the total count of meals as the target variable.

4.2 Additional Information

The next step in the data preparation process is to encode the categorical variables into numerical values, so that forecasting algorithms can incorporate them into mathematical computations that will result in a numerical output. Two key text columns will be encoded in this study: the weather and the special event annotations.

4.2.1 Weather

The weather annotations column in the original dataset included 40 different values, so they were classified into one of seven categories (*Warm, Hot, Cloudy, Rain, Snow, Cold* or *Neutral*) and then encoded using a *one-hot encoding* approach. This ensures a binary vector with all zeros represents each category except for one at the category's index.

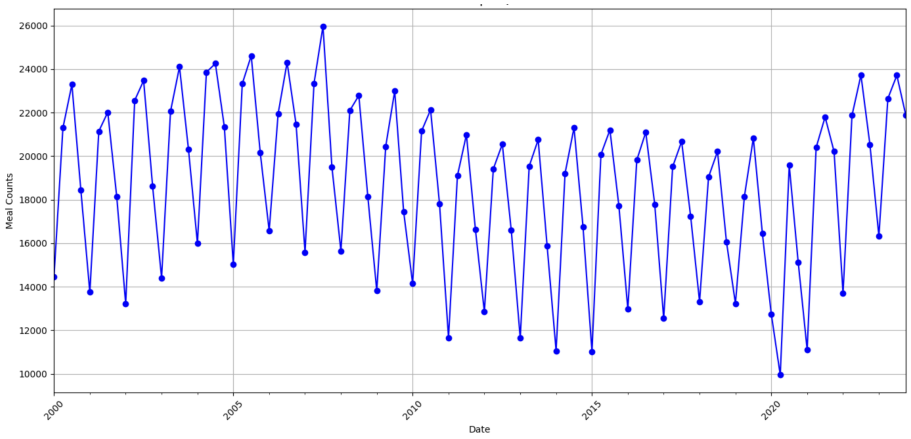

Figure 3: Final Visualization of the Evolution of Meal Servings

Additionally, external weather data was incorporated into the dataset for potentially increased prediction accuracy. Information on daily minimum, average, and maximum temperatures, along with precipitation and wind indicators, was scrapped from the *Meteostat* application programming interface (API). This API provides reliable and accurate weather indicators measured in Celsius degrees, adding an extra level of granularity to the weather variable that could potentially help in predictions. More specifically, data was obtained from the closest weather station available, that of *Willimantic/Mansfield Hollow*, located approximately 20 miles from Pomfret, CT (where the café studied is). Additionally, it is important to note that this weather data was only available from January 1st, 2006.

4.2.2 Special Events

Secondly, the notes column, which includes annotations about special events or observations of each day's business activity was addressed. Unfortunately, 78% of the notes in this dataset were blank. However, it was still interesting to conduct some Natural Language Processing on the column to observe whether useful information could be gained from it. With this objective, we used VADER and transformers to assign a sentiment score to each non-blank note in the dataset.

VADER: Valence Aware Dictionary and sEntiment Reasoner

The sentiment score produced by VADER represents the overall emotional tone of a piece of text, as follows:

- Positive score (greater than 0): The text is considered to have a positive sentiment. The closer the score is to +1, the more positive the sentiment.

- Neutral score (equal to 0): The text is considered neutral, lacking any significant positive or negative sentiment.
- Negative score (lower than 0): The text is considered to have a negative sentiment. The closer the score is to -1, the more negative the sentiment.

The sentiment score is assigned here using a combination of qualitative and quantitative methods. It starts with a list of lexical features (e.g., words) which are labeled according to their semantic orientation as either positive or negative. Each word in the lexicon has a score that ranges from -4 to +4, with negative scores indicating negative sentiment and positive scores indicating positive sentiment. VADER analyzes the text to see which words it contains and aggregates the scores of those words. It also takes into account aspects like word order, punctuation, and capitalization, which can amplify or reduce the sentiment intensity. The compound score is a normalized score that takes into account the sum of all the lexicon ratings that have been normalized between -1 and 1.

As can be seen in Figure 5, the sentiment analysis performed by VADER was fairly successful. For example 'Beautiful Day' was assigned a score of 60%, while 'Register Problems' was assigned a score of -40%. However, many notes were assigned a neutral score of 0%, thus failing to detect the sentiment there. As a result, only 2.92% of the observations were assigned a sentiment score with VADER (256 observations). Even if limited, these scores were added as a variable to the final dataset.

Transformers

Subsequently, a set of pre-trained transformers specifically fine-tuned for sentiment analysis were tested. All of them were variations of *BERT (Bidirectional Encoder Representations from Transformers)*, a revolutionary model in natural language processing that enhances sentiment scoring by understanding the context of each word in a text from both directions (left and right). This allows BERT to capture nuanced meanings and the relationships between words, thereby improving its ability to determine the sentiment expressed in the text (see the architecture in Figure 4). Specifically, the three models used are:

- *BERT-base-multilingual-uncased-sentiment model*, pre-trained on 150,000 product reviews.
- *BERTweet-sentiment-analysis*, pre-trained on approximately 40,000 tweets and optimized for short, informal text.
- *RoBERTa-targeted-sentiment-classification-newsarticles*, pre-trained on 10,000 sentences from news articles and designed to capture more formal and nuanced expressions of sentiment.

The results of the sentiment scores obtained can be observed in Figure 5. Unfortunately, the scores assigned by all BERT architectures were not effective

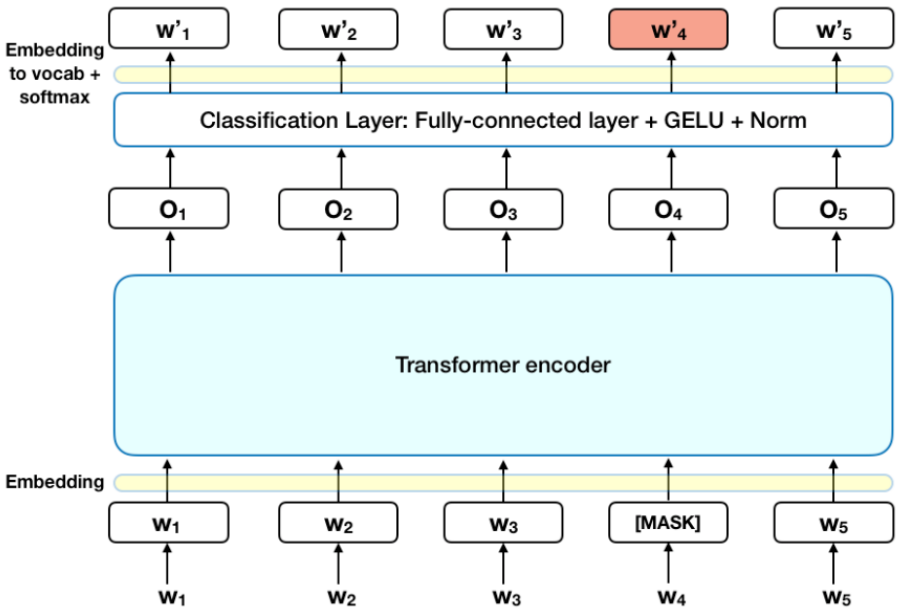

Figure 4: BERT Architecture

Date	Total	Notes	Notes_Sentiment	BERTbase_Sentiment	BERTtweet_Sentiment	Roberta_Sentiment
2000-01-01	93.0	Open 11-5 Slow Start	0.0000	0.312978	0.955318	0.505801
2000-01-03	82.0	Warm	0.2263	0.350797	0.591683	0.512750
2000-01-08	298.0	Erica Wheeler 15	0.0000	0.434273	0.956542	0.510082
2000-01-10	60.0	$850 Catering	0.0000	0.265881	0.958745	0.504859
2000-01-11	87.0	$800.00 Catering	0.0000	0.273852	0.960665	0.503449
2000-01-13	42.0	Snow all day	0.0000	0.258034	0.511187	0.502925
2000-01-21	97.0	Very Cold and Windy	0.0000	0.437474	0.697162	0.511511
2000-01-22	192.0	Louise Taylor 12 people	0.0000	0.472878	0.964392	0.525527
2000-01-23	224.0	Howie Bursen +	0.0000	0.336244	0.957861	0.505400
2000-01-25	87.5	6"-8" Sonw Yucky	-0.4215	0.237619	0.981940	0.510306
2000-01-26	90.0	Lingering snow	0.0000	0.308442	0.768903	0.512730
2000-01-29	232.0	Maria Sangiolo - 25 people	0.0000	0.309293	0.970597	0.517379

Figure 5: Sentiment Scoring Results

Date	Day	Breakfast	Lunch	Dinner	Total	Revenue	Avg Spend	Weather_Cloudy	Weather_Cold	Weather_Hot	Weather_Neutral	Weather_Rain	Weathe
2000-01-01	Saturday	2	91	0	93.0	1064.21	11.443118	0	0	0	0	0	
2000-01-02	Sunday	55	161	29	245.0	2259.14	9.220980	0	0	0	1	0	
2000-01-03	Monday	3	79	0	82.0	840.04	10.244390	0	0	0	0	0	
2000-01-04	Tuesday	5	92	0	97.0	1077.91	11.112474	0	0	0	1	0	
2000-01-05	Wednesday	8	110	20	138.0	1320.27	9.567174	0	0	0	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	

Figure 6: Final Dataset

in capturing the real connotations of the annotations in the dataset. This was likely due to the short length, vague meaning, and inconsistent writing style in them. To successfully leverage the potential of transformers on note sentiment scoring, a more complete record would be required. As a result, the obtained sentiment scores were not included in the final dataset.

After adding all the knowledge obtained from weather and special events annotations, the final dataset was formed (see Figure 6).

5 Results

In this section, the models used for short-term demand forecasting will be analyzed in detail. It is important to note that classical methods (ARIMA, Holt-Winters) only allow for univariate analyses. Conversely, deep learning models are more flexible and allow for multivariate forecasts.

5.1 Classic Forecasting

Classic forecasting methods require the stationarity of time series data before training. To visually check for this, the time series process of the target variable (representing meal counts) was decomposed into its trend, seasonal component, and residuals using additive decomposition (see Figure 7).

Figure 7 shows a varying trend —with values over 200,000 in the years between 2004 and 2008, and below 150,000 in 2020—, discarding the hypothesis of the process being stationary. Additionally, from the decomposition plot, a clear seasonality could be identified. To confirm the observed non-stationarity of the process, the *Augmented Dickey-Fuller* test was conducted, returning a critical value of -1.58 (which was greater than the t-values at 1%, 5%, and 10% confidence intervals) and a p-value of 0.49. As a result, the null hypothesis of the process being non-stationary could not be rejected, thus confirming the process was non-stationary.

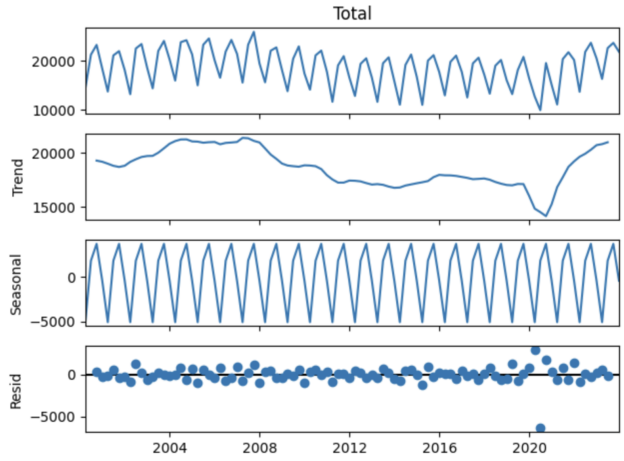

Figure 7: Classical Additive Decomposition of the Meal Servings Evolution

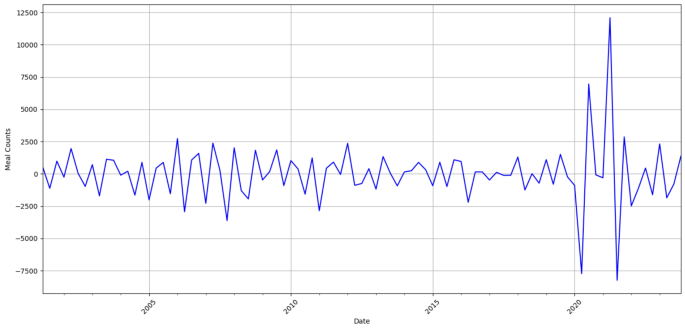

Figure 8: Stationary Data (After Differencing)

With the intent of providing seasonality to the process, it was differentiated once (see Figure 8). After that, the critical value was -3.75, which was lower than the t-value at the 1% confidence interval (-3.51), thus guaranteeing the process' stationarity. The resulting process shows a constant trend and a fairly constant variance —except for the observations in 2021 and 2022, which will need to be accounted for separately as a temporary intervention (likely due to the COVID-10 pandemic)—.

5.1.1 ARIMA: Autoregressive Integrated Moving Average

To get some basic forecasts, an ARIMA model was first fitted. ARIMA requires setting a specific order, as such, both the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) were examined (see Figure 9). From the plots, it seemed reasonable to consider a low-order ARMA model. The

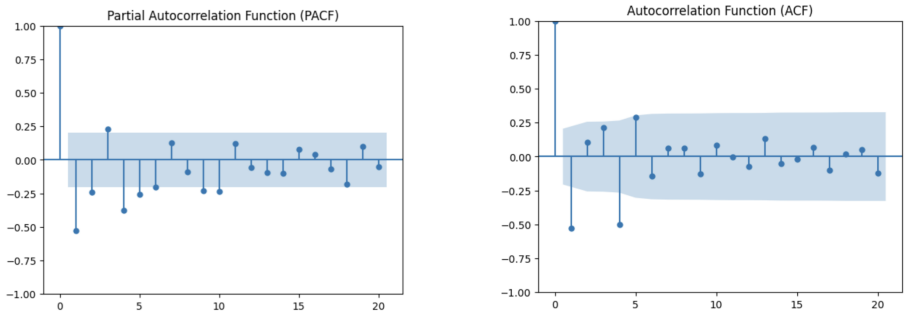

Figure 9: PACF and ACF Plots

PACF cuts off near lag 2 decaying to 0 and the ACF cuts off after lag 2 decaying to 0. However, although there is an early cutoff, the lag significance returns around lag 4, confirming the seasonality.

For the initial model, I chose an ARMA (2,1,2) x SARMA (1,1,1)[4] model. First, for the non-seasonal ARMA(p, d, q) model, there is a decaying pattern in the ACF and PACF. Second, for the seasonal ARMA(P, D, Q) model, there is strong autocorrelation and strong partial autocorrelation around lag 4, and both cutoffs decay to 0. As there are multiple interpretations to take, I generated six additional variations and compared performance. Although the different models had roughly comparable AIC (Akaike Information Criterion) values (see Table 1), the initial model had the smallest one and thus was used for forecasting the time series. To get the final predictions, the results of the forecast had to be integrated, as the model had been fitted on differentiated data (see resulting forecasts in Figure 10).

Model	**AIC**
ARMA (2,1,2) × SARMA (1,1,1)[4]	1551.71*
ARMA (2,1,1) × SARMA (1,1,1)[4]	1552.54
ARMA (2,1,0) × SARMA (1,1,1)[4]	1573.81
ARMA (1,1,2) × SARMA (1,1,1)[4]	1567.62
ARMA (0,1,2) × SARMA (1,1,1)[4]	1554.08
ARMA (2,1,2) × SARMA (0,1,1)[4]	1565.92

Table 1: AIC Scores for Different ARMA Models

5.1.2 Holt Winter's Method

Another widely used and successful forecasting method is the *Seasonal Holt-Winters Method*. The smoothing-based method uses the pattern of the data to extrapolate the forecast using double exponential smoothing. I performed the additive version, as the seasonal pattern remained roughly the same for the

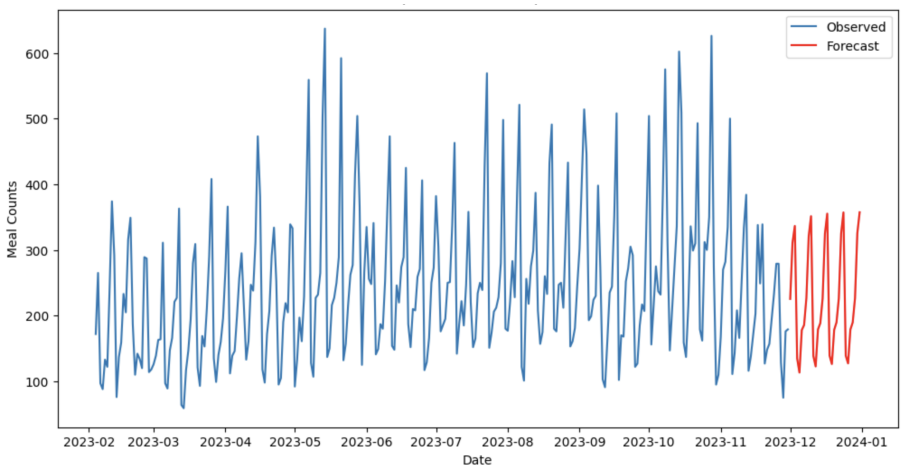

Figure 10: ARIMA Forecast

range of the data. Additionally, I applied the *Basin-Hopping* optimizer, as it returned the most accurate results during validation when compared against other optimizers, like *Powell's Method*. The resulting forecasts can be observed in Figure 11.

5.2 Deep Learning

Deep learning architectures, such as NeuralProphet and N-BEATS, offer extensive flexibility through hyperparameter tuning. Each hyperparameter —from learning rates and batch sizes to the configuration of layers and seasonal adjustments— can dramatically affect the outcomes of the forecasts. This study involved experimenting with various configurations, assessing their impact on forecast accuracy, and systematically refining the parameters to enhance model performance. The specific configurations detailed below are the result of meticulous tuning and validation tailored for this precise problem (forecasting the Vanilla Bean Café's daily meal counts). However, different configurations may be more suitable in different scenarios.

5.2.1 NeuralProphet

The NeuralProphet model was set up to capture the significant seasonal effects observed in the data, with yearly and weekly seasonality enabled, but daily seasonality turned off to avoid overfitting to noise that does not significantly impact the overall trends. The model was configured to use neural networks for future regressors, enhancing its ability to adapt to new data. Key parameters were carefully selected to balance model complexity and training efficiency: seasonality regularization was set at 50 to control overfitting, and the model was

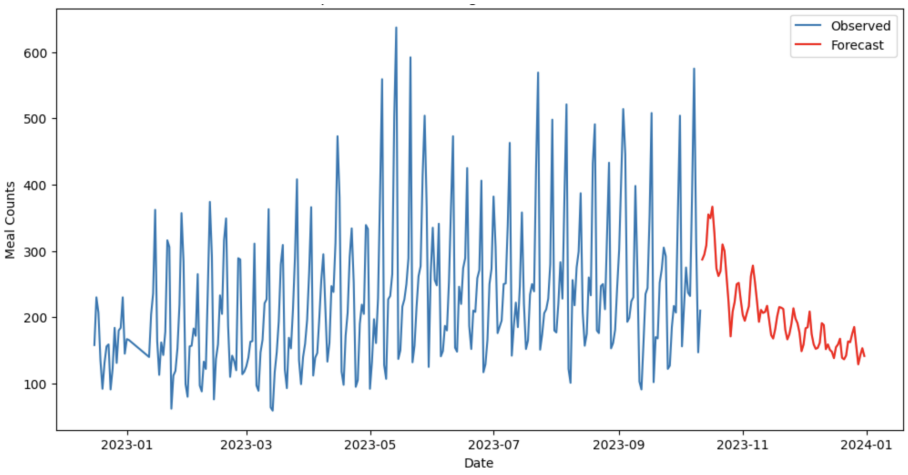

Figure 11: Seasonal Holt-Winter's Method Forecast

instructed to make 14 forecasts ahead, aligning with the two-week operational planning needs of the café.

The training involved a high learning rate of 0.5 to accelerate convergence, with 60 epochs to allow sufficient learning without overfitting, and a large batch size of 2000 to ensure robust gradient estimates. The model utilized Stochastic Gradient Descent (SGD) as the optimizer, with Mean Absolute Error (MAE) as the loss function to focus on the median performance metric, which is often more robust to outliers. The training process was executed with these settings on the daily data, and validation was performed simultaneously to monitor overfitting and adjust the training process when necessary.

The resulting NeuralProphet forecasts can be observed in Figure 12.

5.2.2 N-BEATS: Neural Basis Expansion Analysis Time Series

Tthe N-BEATS model was configured with an 81-day forecast length and a 7-day backcast length, providing a compact yet effective learning window for capturing the essential dynamics in the data. The model employed a 2000 batch size for efficient computation and used 100 hidden units in its layers to capture the intricate patterns in the time series without excessively complicating the model structure.

The fitting process utilized the *AdamW* optimizer, a variant of the classic Adam optimizer that includes weight decay to help prevent overfitting —a crucial feature when dealing with time series data that can exhibit subtle changes in patterns over time. The learning rate was set at 0.001, with standard beta parameters for momentum adjustments. This setup aimed to refine the model's

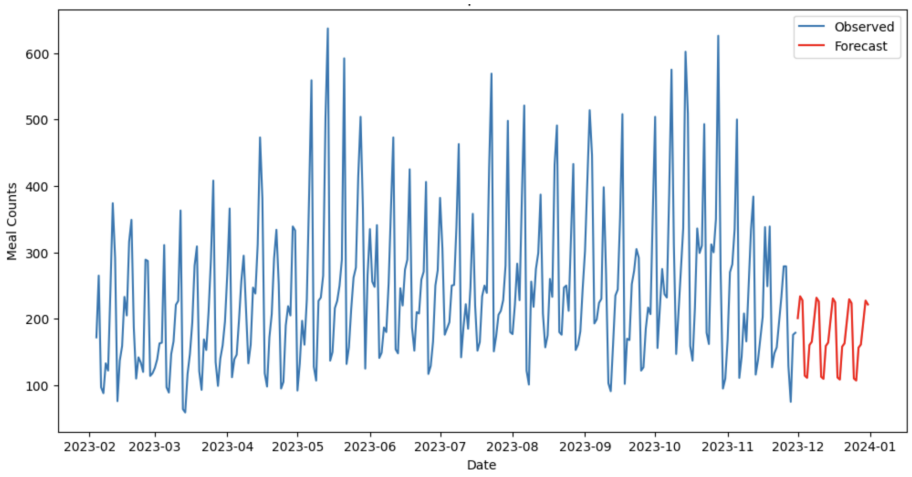

Figure 12: NeuralProphet Forecast

ability to generalize from historical data to future predictions effectively. The average forecast from the model was calculated to smooth out any erratic predictions and provide a stable forecast trend.

The resulting N-BEATS forecasts can be observed in Figure 13.

6 Evaluation

To test for the accuracy of the fitted models, a hold-out testing approach was used, with observations from 01/01/2006 to 11/30/2023 encompassing the training dataset, and observations from 12/01/2023 to 12/31/2023 encompassing the test dataset (these correspond to the 31 most recent samples). All four forecasting models —ARIMA, Exponential Smoothing, NeuralProphet, and N-BEATS— were evaluated based on the Mean Absolute Percentage Error (MAPE) metric (with A corresponding to actual values and F corresponding to forecasts):

$$\text{MAPE} = \left(\frac{1}{n} \sum_{i=1}^{n} \left| \frac{A_i - F_i}{A_i} \right| \right) \times 100 \tag{1}$$

From Table 2, it can be seen that NeuralProphet achieved the lowest MAPE of 0.1676, implying that it was the most effective model. This superior performance might be attributed to its ability to automatically detect and adjust to seasonality and to conduct a multivariate analysis. On the other hand, the classical models, ARIMA and Holt-Winters, yielded MAPEs of 0.2837 and 0.4259, respectively, with ARIMA performing significantly better. This indicates that traditional models are fairly useful, but their effectiveness may be limited due to their inability to deal with multivariate datasets.

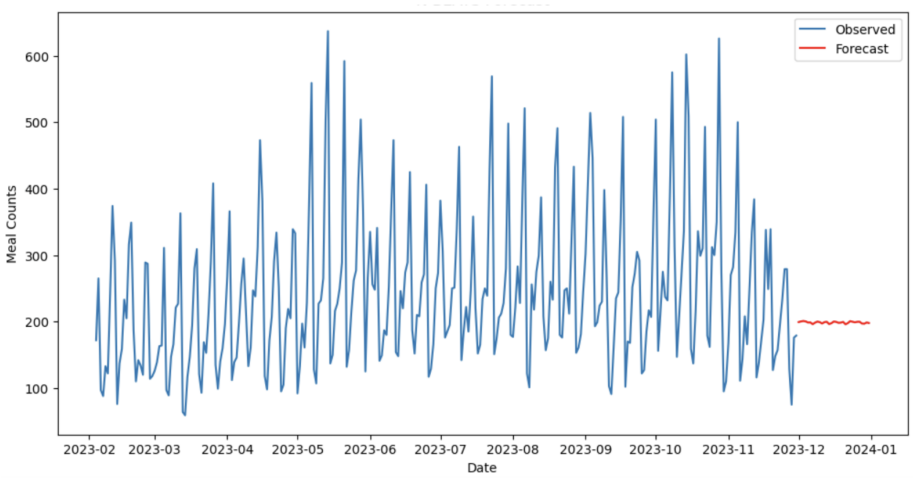

Figure 13: N-BEATS Forecast

	ARIMA	**Holt Winter's**	**NeuralProphet**	**N-BEATS**
Method	Classic	Classic	Deep Learning	Deep Learning
MAPE	0.2837	0.4259	0.1676	0.2308

Table 2: Comparison of Forecasting Methods

7 Conclusions

This study thoroughly evaluated the performance of both traditional and deep learning forecasting models in predicting the daily meal counts for a local café, providing essential insights into the capabilities of these models in practical business applications.

The findings confirm that deep learning models, particularly NeuralProphet, significantly outperform traditional methods like ARIMA and Holt-Winters in forecasting accuracy. NeuralProphet, with the lowest Mean Absolute Percentage Error (MAPE), demonstrated a benefit in automatically capturing complex patterns and seasonality without manual specification.

Furthermore, the study explored the impact of integrating additional variables such as weather and special events into the forecasting models. The inclusion of these variables in a multivariate forecasting approach proved to enhance the accuracy of the predictions.

This research also highlighted the importance of leveraging basic datasets collected with limited resources, showcasing how qualitative data, such as simple activity annotations, can be effectively incorporated into the models. This ap-

proach underlines the potential of simple, cost-effective data collection methods in enhancing the forecasting capabilities of local businesses.

Overall, the reliability of AI in guiding operational decisions for local businesses is promising, especially when advanced models like NeuralProphet are employed. These models not only provide precise forecasts but also offer the flexibility to adapt to the specific needs of a business. As AI technology continues to evolve, its integration into daily business operations could become more streamlined, offering even greater benefits in terms of efficiency and accuracy.

References

[1] Carmenpelayo. (2024). Local Business Short-Term Demand Forecasting Exploration Leveraging Neural Network Architectures. GitHub. https://github.com/carmenpelayo/DLDemandForecast/tree/main

[2] Jones, M. (n.d.). 5-OUT — Demand Forecasting Importance for restaurants. 5Out.io. https://www.5out.io/post/demand-forecasting-importance-for-restaurants

[3] Box, G.E.P., Jenkins, G.M. (1970). Time Series Analysis: Forecasting and Control.

[4] Holt, C.E. (1957). "Forecasting seasonals and trends by exponentially weighted moving averages."

[5] Taylor, S.J., Letham, B. (2018). "Forecasting at scale." The American Statistician.

[6] Triebe, O., Hewamalage, H., Pilyugina, P., Laptev, N., Bergmeir, C., and Rajagopal, R. (2021, November 29). NeuralProphet: Explainable Forecasting at scale.

[7] The Vanilla Bean Café. https://thevanillabeancafe.com.

[8] GeeksforGeeks. (2023, December 26). How to deal with missing values in a Timeseries in Python? GeeksforGeeks. https://www.geeksforgeeks.org/how-to-deal-with-missing-values-in-a-timeseries-in-python/

[9] Bevans, R. (2023, June 22). Akaike Information Criterion — When and How to use it (Example). Scribbr. https://www.scribbr.com/statistics/akaike-information-criterion/

[10] Oreshkin, B.N., Carpov, D., Chapados, N., Bengio, Y. (2019). "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting."

[11] AdamW — PyTorch 2.3 documentation. (n.d.). https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html