Senior Design Project For Engineers
Southern Polytechnic College of Engineering and Engineering Technology

Spring 4-24-2023

Reducing Restaurant Inventory Costs Through Sales Forecasting

Tyler Mason
Kennesaw State University

Chris Schoen
Kennesaw State University

Trevor Gilbert
Kennesaw State University

Jonathan Enriquez
Kennesaw State University

Kaizen Consulting

Project
Reducing Restaurant Inventory Costs

Members
Tyler Mason - Project Manager
Chris Schoen - Software Specialist
Trevor Gilbert - Operations Specialist
Jonathan Enriquez - Data Analyst

Advisor
Dr. Adeel Khalid

University
Kennesaw State University

Executive Summary

Family Restaurant is a local restaurant in the greater Atlanta area that serves a variety of dishes that include an assortment of 19 different proteins. Currently, Family Restaurant places protein orders based on business intuition, and tends to over-stock and sometimes under-stock. To minimize inventory costs by reducing over-stocking and preventing under-stocking of proteins, we applied Facebook Prophet (FB Prophet), ARIMA, and XG Boost machine learning models to predict protein demand and then fed these results into a Fixed Time Period inventory model to make an overall order suggestion based on the specified time period. We trained our models on sales data from 2021 and 2022 and tested our models on January 2023 data. Overall, FB Prophet shows a 6% savings per month from actual inventory spending, ARIMA shows a 34% savings, and XG Boost shows a 5% increase in spending for January 2023. ARIMA shows such high savings as it tends to under-stock in periods of high demand, while FB Prophet adequately meets periods of high demand and tends to over-stock during periods of normal demand. The restaurant prefers to over-stock, as under-stocking implies lost sales and thus, the loss of customer good faith, which is unacceptable for their business. Family Restaurant could adapt a hybrid approach of applying FB Prophet during known times of peak sales volume, while applying ARIMA during times of normal sales volume and realizing savings of 30%. The hybrid approach is slightly riskier, as it still relies on intuition. Ultimately, our recommendation is to follow the conservative approach of always applying the FB Prophet model and realizing savings at or around 6%.

Table of Contents

List of Figures

List of Tables

Chapter 1: The Project

1.1 Introduction

Family Restaurant is experiencing issues ordering the incorrect quantity of proteins for their dishes, this results in under-stocking and over-stocking. Under-stocking results in the restaurant running out of a protein item, preventing them from making dishes with that item, which ultimately results in lost sales as well as potential loss of customer loyalty. Over-stocking results in protein being wasted because they cannot use it before it loses freshness. Currently, protein orders are placed on their intuition of what the demand will be. Family Restaurant has asked us to provide them with a more consistent ordering system to reduce their costs and increase profit margins.

1.2 Objective

The goal of this project is to reduce the inventory costs of Family Restaurant by optimizing their protein ordering process by using sales forecasting and inventory optimization models. By aligning their orders with their sales, we aim to minimize over-stocking, which can result in lost revenue due to spoilage, and under-stocking, which can lead to missed sales opportunities. Ultimately, reducing waste and lowering expenses will reduce inventory costs for the restaurant.

1.3 Justification

With most restaurants, one of the largest issues is over-stocking and under-stocking of alcohol and food items. We narrowed our focus after meeting with the company's owners to just the protein items on the menu because they were the most expensive, used in most dishes, and are perishable. We then wanted to use a sales forecast of protein items to help us develop an order quantity to meet demand. With the biggest cost of over-ordering being the potential for loss on wasted products, we hope to have accurate forecasts for these products, so that we can help the restaurant save money.

1.4 Project Background

For our research, we will call our clients Family Restaurant. Family Restaurant is a well-established restaurant in a small town in Georgia. Growing in population and economic status, it is a great place to start a business. Family Restaurant has been established for a little over two decades, building itself up in the community with its charity and community work. Family Restaurant currently experiences protein ordering errors such as overstocking and under-stocking because orders are placed based on intuition rather than business metrics. Fortunately, they keep years' worth of daily sales data. With this information, we saw fit that a sales forecasting model could be created to accurately predict the demand for dishes containing protein. Thus, an accurate inventory ordering process can be implemented using forecasted sales data.

1.5 Problem Statement

Family Restaurant currently performs all its protein orders based on intuition. While this is common in the restaurant industry, it can result in inaccurate orders. Over-ordering can lead to spoiled food, or under-ordering can lead to the restaurant not having ample ingredients to supply a customer with their requested dish. To fix this, we hope to use their daily sales data to forecast the expected sales for the week for all their protein dishes. Using an accurate sales forecast, we will suggest ad hoc orders based on their current delivery schedule and needs.

Chapter 2: Literature Review

A. Parizad and C. J. Hatziadoniu found that electrical power companies are consistently looking to explain patterns of consumption data through the use of forecasting models [1]. Short-term load forecasting (STLF) models are the most used in the industry. This paper explores a two-stage STLF based on the FB Prophet model. The first stage extracts seasonality, trend, and holiday effects from the consumption data and then adds weather and lag data as regressors to make the model more robust. In the second stage, mean absolute percentage error (MAPE) is used to find hyperparameters, which increase the accuracy of the model and create a more accurate prediction. Three case studies were used to predict “day-ahead” consumption using different hyperparameters. Different consumption patterns were identified with this method and each 24-hour consumption case study was forecasted within 3% of the confidence interval.

B. Kumar Jha and S. Pande discuss how the FB Prophet model is a relatively new forecasting model that can be applied in various fields with time-series data that is affected by seasonality and trends [2]. Naturally, it must be determined how the FB Prophet Model compares to traditional forecasting models. In this paper, FB Prophet Model is compared to ARIMA and Holt Winter’s models using supermarket sales data to determine which has the best prediction power. In this use case, it was determined that FB prophet has better prediction capabilities as it had the lowest error and better fitting. Likely, the FB Prophet model is best adapted to retail sales data that is heavily affected by seasonality, trends, and holidays as these are the specialties of the model.

B. Pavlyshenko goes through his process on a Kaggle competition looking at sales forecasting [3]. He is focusing on the use of machine learning and trying out different model types to see which comes up with the least error and proving that a stacked model ends up creating the best result by lowering the overall mean squared error. Pavlyshenko begins by showing an example of how to analyze different data features, explain the relevance, and build out your models, and then moves into a focus on different models, specifically: Neural network, Lasso, Random Forest, ARIMA, and ExtraTrees. The big result from his example shows an increase in at least 1% accuracy shown from a decrease in the validation error. This article shows us another example of something we could try with our different models and see if using a stacking method could help us improve the accuracy of our forecast at the end of the day as well as some good example visualizations that can be generated to explain the state of the sales data.

C. I. P. Mousinho focuses on ARIMA and Facebook's prophet model in the fast-food industry highlighting their differences and capabilities [4]. All models are not perfect, their performance is measured by calculating their errors. This article dives into five main techniques used to measure forecasting error. Mean absolute deviation, mean squared Error, Mean absolute percentage error, the cumulative sum of forecast errors, and standard deviation. A major difference between the two is seasonality, ARIMA makes very few assumptions and struggles to predict changes in trend behavior. The prophet model can detect turning points as well as work when values in a series are missing. Our project will have a seasonality to account for which makes the prophet model the optimal choice.

D. A. Cranage and W. P. Andrew discuss what judgmental, econometric, and time series model types are, and how they are used in practice today [5]. Afterward, they go into a more detailed comparison of time-series and econometric models comparing their forecasting results using statistical standards such as the sum of squared residuals, R-squared, and the Durbin-Watson Statistic. The results indicate that the time-series models displayed a similar, if not better ability, to accurately forecast the data, but even more so, they are significantly cheaper both in data collection and processing than the econometric model. This article helps us to explain our choice of moving towards a time-series forecast model over less data-intensive options like judgmental models, or even more data-extensive models like econometric, with no real assurance that some values might be important to the model.

F. M. Puspita and N. A. Primadani apply the Auto-Regressive Integrated Moving Average (ARIMA) Model to forecast sales for food items in this article [6]. Then, they apply the Fixed Order Quantity inventory model to suggest when new food items should be ordered. The ARIMA model assumes the future will resemble the past and uses a moving average to smooth trends and seasonality. The dependent variable uses past values of itself, sales, in this case, to predict future values. The Fixed Order Quantity model assumes inventory is tracked continuously and orders new inventory in the same quantities once the inventory hits a specified level, the reorder point.

F. R. Jacobs and R. B. Chase propose a more simplistic fixed-time period model with safety stock [7]. It calculates the mean and standard deviation of demand per time period based on annual demand. The equation is outlined below. Since our demand will technically be known for each time period, we could replace the average demand with the expected demand and calculate safety stock as a percentage of expected demand, as determined by the decision maker and recommended by us. A 5-10% safety stock may be appropriate. Lastly, we would subtract whatever inventory is currently on hand and on order from the expected demand plus safety stock.

F. S. Hillier and G. J. Lieberman suggest several possible inventory models that can be used depending on what relates most closely to their inventory system [8]. A continuous review EOQ model can be applied if demand is constant and known and inventory is updated continuously. It can also account for quantity discounts. This model orders the same quantity when inventory reaches reorder point and can account for quantity discounts. The assumptions of constant demand may be violated for this model since demand from period to period will vary despite demand within the period remaining constant. This could be addressed by adjusting the reorder point for each period. The periodic review model allows demand to vary from period to period and will vary order quantity on a consistent ordering schedule. However, this periodic review model only reaches optimality by allowing inventory to reach 0, at which point the new inventory would arrive immediately. This implies a Just-In-Time (JIT) inventory model which may not be realistic for the restaurant. Additionally, it does not account for lead time or safety stock directly.

F. Taigel and J. Meller examine a data set through two different styles of machine learning: supervised Separated Estimation and Optimization (SEO) and Joint Estimation-optimization (JEO) [9]. The first method SEO is a two-function estimation where we first fit a prediction model to the demand and then optimize the inventory. The

JEO improves on the SEO two-step process by directly connecting the features and demand model, eliminating the need for a prediction model.

G. Rafferty's book looks at the development of a Facebook Prophet Model [10]. This book goes into detail on a lot of different subjects in model creation, the first is an explanation of different model types such as ARIMA, ARCH/GARCH, and machine learning/neural networks. Along with these explanations, he explains their strong points, and how the Facebook prophet model ends up outperforming them with the right data. Through the chapters, he explains with python context the implementation of the model including how to account for seasonality and holidays. This is the most important part of the book for our project because we hope to try and implement a Facebook prophet model with the daily sales data we have. Having a resource to reference both the theory behind the model as well as the python code will help us avoid beginner mistakes and waste time finding coding documentation later.

P. Verma, S. V. Reddy, L. Ragha, and D. Datta discuss different forecasting models [12]. ARIMA is a forecasting model that forecasts time-series data by performing regression on past values of the time-series data. The FB prophet model also regresses past values of time-series data and is specially designed to account for trends, seasonality, and holidays. LSTM models are a powerful machine learning tool used for prediction, especially with time-series data. Importantly, LSTM models are often used for weather prediction, including the Air Quality Index (AQI), which is the focus of this study. LSTM outperformed FB Prophet in RMSE and MAE but FB Prophet had a better MAPE. The researchers concluded that LSTM was the superior model in this instance. This may have been because AQI is less subject to seasonality and holidays, where FB

Prophet specializes. The results may have been different for restaurant sales forecasting.

Romero-Gelvez, Jorge Ivan, Edisson Alexander Delgado-Sierra, and Jorge Aurelio discuss the use of open-source tools for material requirement planning in this article. The prophet forecasting model in python is shown combined with this equation [13]. y(t) = g(t) + s(t) + h(t) + error t. G represents the trend with non-periodic changes, s is periodic changes, and h is holidays. Error t is idiosyncratic changes not allowed for by the model. The article shows an example of the Facebook prophet model being used for the fragrance industry minimizing inventory and significant deviation. However, this is in tandem with SARIMA and LSTM using KERAS and Google Collab. The equation and other tools used with the prophet model will enable us to better optimize our model as we minimize wasted food.

S. Siami-Namini, N. Tavakoli and A. Siami Namin look at the differences between ARIMA and LSTM forecasting models [14]. The article goes into detail on the general definition of both models, as well as their setup in comparison using a large financial data set of stocks. The researchers then implement their tests in python and compare the results of the RSME (root mean square error) to judge the accuracy of forecasts on the data set. While the authors found an extreme benefit to using the LSTM model, they do note several times that there is a heavy difference in the processing requirements between the machine learning from the LSTM model compared to a standard ARIMA model. In terms of our project, we plan to test multiple types of forecasting models, including machine learning models and think that the python and algorithm explanation in this article can be used for us to formulate our needs. The power of accuracy in the machine learning model will have to be weighed based on the restaurant’s willingness to spend on the computation on their side in the future. Overall, this article gives a strong argument for the perks of using machine learning-based forecasting to increase overall prediction accuracy.

T. Feng, Z. Zheng, J. Xu, M. Liu, M. Li, H. Jia, and X. Yu evaluated road traffic injuries, one of the deadliest health problems in the world, utilizing forecasting models [16]. The data collected in this article is obtained from hospitals between 2015 to 2020 from people admitted due to car-related complications. The researchers split this data into three different models including SARIMA, LSTM, and Facebook prophet model to compare predictive effectiveness between these. The results of this research concluded that the best model in this scenario was LSTM due to the nature of the data. The other two models handle stable data much better than a random array, and due to the nature of the data, there was no clear seasonality in the number of traffic-related incidents. The prophet model, in particular, works best when the years of data have defined the holiday effect, despite this, it still provided the second-best set of data. With our project showing clear signs of seasonality, we can confidently conclude this model will work best with our data.

T. Tanizaki, T. Hoshino, T. Shimmura, T. Takenaka use the Random Forest Regression model to forecast customer order quantities and inventory forecasting methods [17]. A random forest of decision trees is constructed and performs majority decisions on each decision tree. Then the learning data is extracted at random to eliminate bias. The model overlays projected data with the actual data to show spikes in demand. Overall, both forecasts did not yield high accuracy percentages, the experiment will be continued later in years to see improvements.

W.-X. Fang, P.-C. Lan, W.-R. Lin, H.-C. Chang, H.-Y. Chang, and Y.-H. Wang use the Facebook prophet model and LSTM to help predict the stock prices of the Taiwan stock exchange [18]. These models are extremely effective when we are dealing with data that is changing constantly. After doing the research using both models, the LSTM was the more consistent model, with the empirical results validating the LSTM model. The Facebook prophet model showed upward trends on the weekly models, but the full-year model showed an upward trend for the first half of the year and a downward trend during the second half of the year.

X. Dairu and Z. Shilong explore the use of XGBoost, a popular machine learning algorithm, for a variety of tasks such as classification, regression, and ranking. XGBoost stands for Extreme Gradient Boosting and works by combining multiple decision trees to make more accurate predictions [19]. It is a scalable and efficient algorithm that can handle large datasets with high-dimensional features. The article explains the basic workings of XGBoost, including its objective function and regularization techniques. It also provides practical examples of how XGBoost can be used, such as in fraud detection, stock price prediction, and image classification. Overall, the article highlights the versatility and effectiveness of XGBoost in various machine learning applications.

Chapter 3: Problem Definition

3.1 Project Plan

The models we will use to forecast sales are ARIMA, XGBoost, and FB Prophet. Past sales data from 2021 to 2022 will be used to train the models. Sales will be forecasted for 2023 and compared to the actual sales from 2023 for each model. We will then feed the forecasted demand into a fixed time-period inventory model to create an order suggestion based on the designated review period. We will verify the success of our project by comparing the restaurant's actual spending, cost of goods sold, and the model's predicted costs using our proposed inventory model. The most successful forecasting model will be determined based on the goals of saving on costs as well as meeting expected demand, as those are the priorities for Family Restaurant.

Below, Figure 1 shows a block diagram split into the major three sections of this project. The first is the data. This is the most important part because without accurate data that matches our model's criteria, any sort of forecasting and thus order suggestions become moot and void. After the data, which will be supplied by Family Restaurant, we have the models, both a sales forecasting model and an inventory suggestion model. Lastly, we have testing and validation, which includes verifying our forecasted numbers as well as the financial benefit from the suggested order quantities through comparison with the current 2023 data.

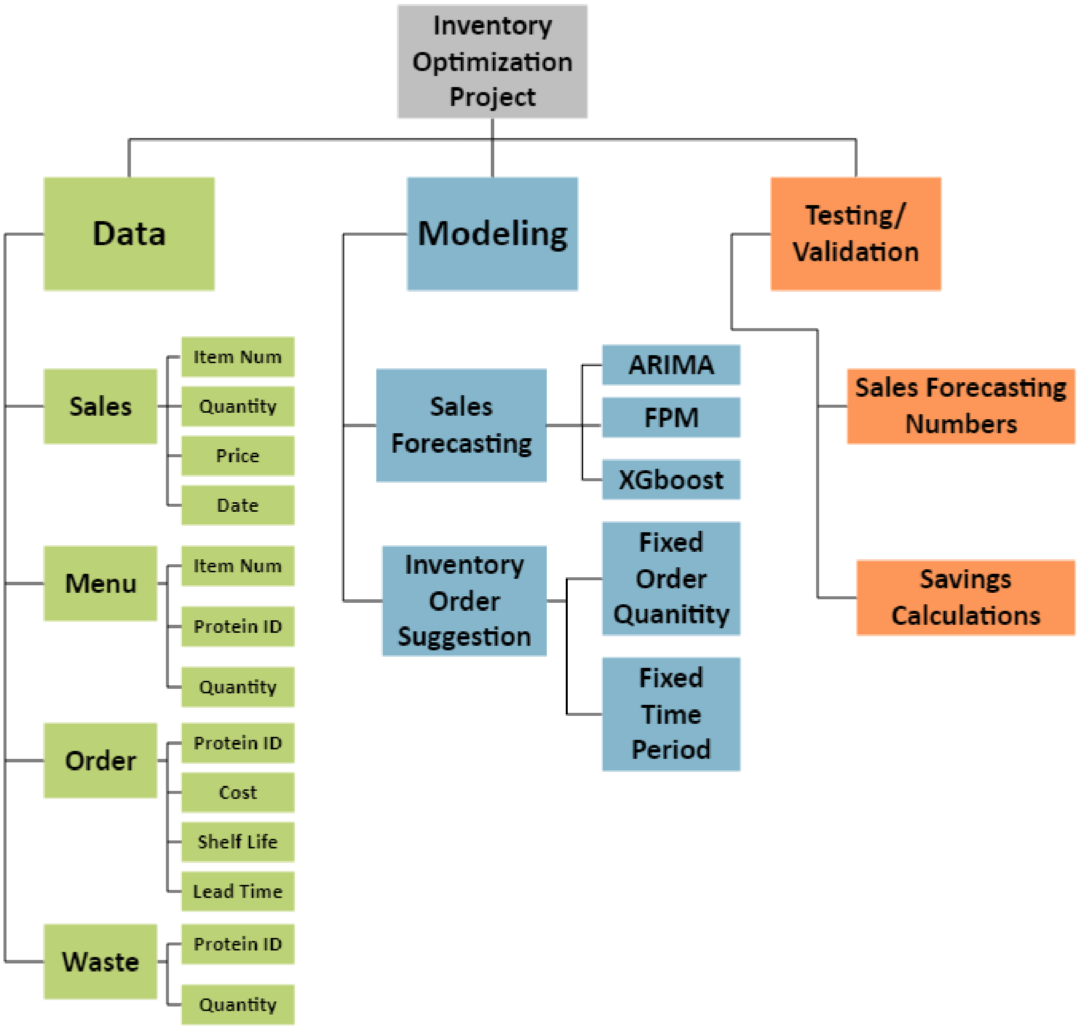


Figure 1: Block Diagram of the System Overview

3.2 Model Overview

Forecasting has been a vital tool used for businesses for many years to help understand their demands and the ever-changing economy of today's society. Understanding the demand for one's products can be a vital part of helping a company produce the right number of products to reduce waste and prevent lost sales. We use the forecast of past sales data to build a model of what future demand will look like for products. This can be essential to saving money for businesses by not overproducing products, leading to a waste of products when these can't be sold. On the other end, producing too little of a product will lead to customers' demand not being fulfilled, which leads to lost profits.

In our project, we are going to be implementing multiple machine learning models to see which works best for the data. But what is machine learning? Machine learning is the creation of artificial intelligence that works to fit models on a data set that can then be used to reach conclusions. This comes from the idea of utilizing supervised learning, the process of feeding a model data with answers to a question, in our case "how much of item A did we sell on date 1?", giving the model a trend to work with and understand, so when we ask it "how much of item A will be sold on a date 2 weeks from now?" it can generate an answer based on past data. In our case, we utilize supervised learning for both the Facebook Prophet and XGBoost Models.

3.2.1 Facebook Prophet

The Facebook Prophet model was designed to be able to forecast certain events happening in the future using daily data. They designed the model around the idea of looking at a data's trend, as well as considering seasonality and holidays, all of which is possible specifically because of the data being on a daily level rather than a higher level. To account for this, the equation below is used:

$$\mathrm{y}\,(\mathrm{t}) = \mathrm{g}(\mathrm{t}) + s(\mathrm{t}) + \mathrm{h}(\mathrm{t}) + \epsilon_t \tag{1}$$

In equation 1, ""(t) is the trend function which models non-periodic changes in the value of the time series, s(t) represents periodic changes (e.g., weekly and yearly seasonality), and h(t) represents the effects of holidays which occur on potentially irregular schedules over one or more days. The error term ϵ_t represents any idiosyncratic changes which are not accommodated by the model" [15]. In essence, the prophet model works extremely well for our data because we have access to daily sales data, as well as the fact that restaurants are heavily affected by seasonal trends and holiday shifts.

3.2.2 XGBoost

XGBoost, or Extreme Gradient Boosting, is a popular machine learning algorithm used for classification, regression, and ranking tasks. It works by creating an ensemble of weak decision trees that are trained sequentially to correct the errors made by the previous tree. Each subsequent tree focuses on the instances that were incorrectly classified by the previous tree. XGBoost uses gradient boosting, which means that it minimizes a loss function by repeatedly adding new models to the whole thing. This approach produces a powerful predictive model that can handle large datasets with high-dimensional features. XGBoost has become a go-to method for forecasting tasks in a variety of fields, such as finance, marketing, and healthcare. Its ability to handle missing data, handle a variety of input types, and capture complex relationships between variables makes it an ideal choice for time series forecasting. All of these features make this model an ideal fit for forecasting the sales of Family Restaurant.

3.2.3 ARIMA

The ARIMA (Autoregressive Integrated Moving Average) model is a popular and widely used time series forecasting technique that is employed to predict future values of a variable based on its past values. The ARIMA model assumes that the time series data has a stochastic or random component that can be modeled using statistical methods. The model is comprised of three main components: the autoregressive (AR) component, the differencing (I) component, and the moving average (MA) component. The AR component models the relationship between the current value of the variable and its past values, while the MA component models the relationship between the current value and past forecast errors. The I component removes any trends or seasonality in the data through differencing. The ARIMA model is widely used in various applications, including finance, economics, and weather forecasting, among others.

3.2.4 Fixed Time Period

The fixed time-period model assumes that inventory is updated only at the end of each review period. For the Family Restaurant, this is at the end of each day. It also assumes that demand and order quantities will vary from period to period, which is the case with the Family Restaurant. The formula and layout for the fixed time-period model is outlined below [7]. The review period is every day, meaning T=1, and for most proteins, the lead time is one day, meaning L=1. Since Family restaurant places orders every two days, the time period in review is two days. Thus, the average demand for two days will be multiplied by 2 since T and L are both equal to 1, meaning the expected demand for the period will be exactly equal to the forecasted demand for the period.

Family Restaurant requested that we apply a safety stock of 10% of total protein demand during the period. Thus, the safety stock formula is adapted to reflect an addition of 10% of total protein demand during the period. Finally, the current inventory is subtracted from the summation of the previous metrics to determine the final order quantity for the period.

$$\begin{matrix} Order \\ quantity \end{matrix} = \begin{matrix} Average\ demand \\ the\ vulnerable\ period \end{matrix} + Safety\ Stock - \begin{matrix} Inventory\ currently \\ on\ hand \end{matrix}$$

$$q = \bar{d}(T + L) + D(10\%) - I$$

q = Quantity to be ordered

T = The number of days between reviews

L = Lead time in days

$\bar{d}$ = Forecast average daily demand

D = Total demand for the period

3.3 Measure of Goodness

In this study, we aim to benchmark and evaluate the effectiveness of three popular forecasting models, namely ARIMA, Facebook Prophet, and XGBoost. To do this, we will use sales data from 2021 and 2022 to predict the future sales of Family Restaurant for 2023 using each of these models. We will then compare the predicted sales data to the actual sales data for 2023 to determine which model is the most accurate and reliable. We measured the accuracy of our models using three key metrics: Root Mean Square Error (RMSE), Normalized Root Mean Square Error (NRMSE), and Mean Absolute Percent Error (MAPE). RMSE is a measure of how far the predicted values are from the actual values. NRMSE facilitates the fraction of the overall range that is typically resolved by the models. MAPE is a measure of the average difference between the predicted and actual values as a percentage of the actual values. Beyond the accuracy of our forecasts, we will also be running forecasts from all three models through our inventory model to suggest the optimal amount of protein for the restaurant to order. By comparing the amount of protein ordered in the past with the suggested amount by our inventory model and the corresponding pricing information to evaluate the effectiveness of our inventory management strategy in terms of inventory costs. We will also track the amount of protein ordered to the actual amount used by the restaurant and consider the cost difference between the two to ensure that we are optimizing inventory management by reducing cost and increasing profitability while maintaining customer satisfaction and meeting demand for protein orders. Overall, this study will provide valuable insights into the effectiveness of different forecasting models and inventory management strategies in the context of a real-world industrial application.

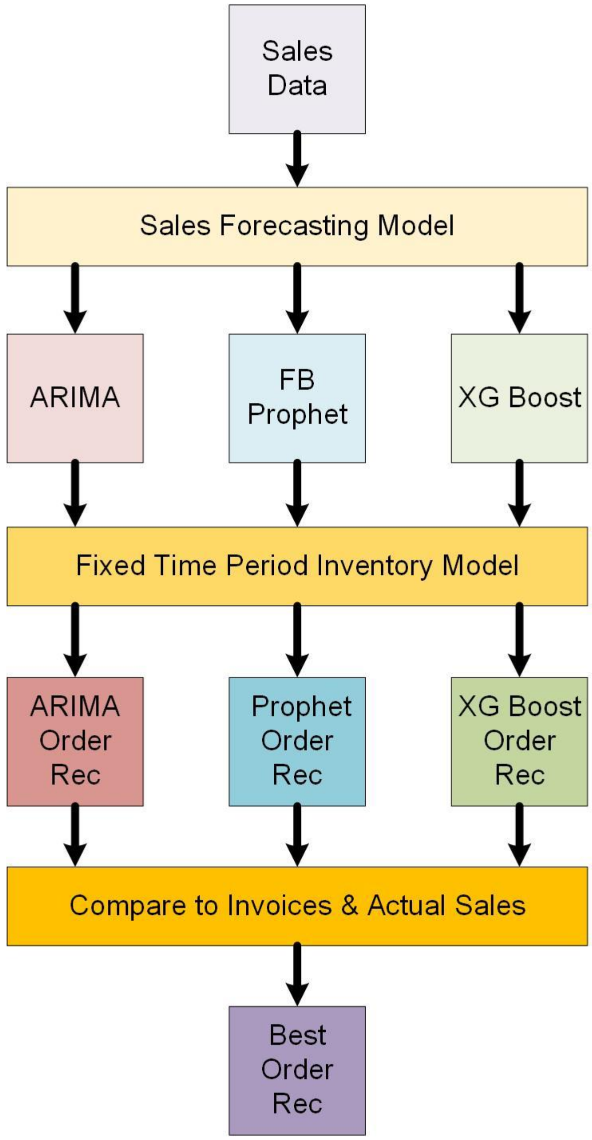


Figure 2: Benchmarking Flow Chart

Figure 2 shows the flow of the benchmarking and validation process for the system. Sales data is used in three different forecasting models which are each fed into the fixed-time period inventory model to create an order suggestion. Then the models are compared to the cost of actual invoices, their current ordering method, and the cost of actual sales to determine the best fit model.

3.4 Design Requirements and Specifications

The design requirements and specifications for the forecasting models and inventory model are listed below.

3.4.1 Forecasting Model

∉ The forecasting model shall predict sales at least one week into the future

∉ The forecasting model shall account for seasonality, holidays, and trends

∉ The forecasting model shall be accurate within 2-3% for all menu items

∉ The forecasting model shall take 5 minutes or less to produce a one-week sales forecast

3.4.2 Inventory Model

- The inventory model shall take in forecasted sales data as well as current inventory numbers
- The inventory model shall suggest order quantities for each protein based on current inventory, projected sales, and estimated waste

3.5 Minimum Success Criteria

∉ The forecast model will be able to predict estimated sales quantities of every protein item up to one day in advance

∉ The Inventory Optimization model will suggest order quantities based on current inventory, projected sales, and estimated waste

- ∉ The forecasting model will be accurate within 10% of the actual 2023 sales
- ∉ We will be able to present the order suggestions in an accessible and understandable format

3.6 Gantt Chart and Schedule

Figure 3 displays the original project schedule with percentage allocations for estimated time spent by each member on those tasks based on an expected five to ten hours a week spent per person.

The weekly meeting task is there to represent our planned weekly in-person meetings on Monday at the very least, with virtual and in-person meetings added in as needed. We assume stakeholder meetings with the restaurant owners and managers to be occurring bi-weekly and will be scheduled around other project milestones, such as design reviews. Milestones are set for the design reviews to guide the completion of necessary tasks and ensure the project is delivered on time. It is implied by the schedule that we will consistently be updating our report as we continue to work on the project.

The work of our project is split into four main sections: data collection, model research, model creation, and testing and validation. It is customary that data collection takes the most time in a data-driven project, so we allotted the most time for collection and research. Before the data can be used, it needs to be rendered in a usable format; thus, 10 hours over 5 days was originally allotted to this task, as seen in Figures 3 & 5. However, scraping the sales data from pdf’s and wrangling it into an excel file that is easily read by python proved to be more difficult and time consuming than we originally anticipated. We ended up spending 25 hours over the course of almost three weeks to complete the data wrangling. This delayed the following tasks by about two weeks. Fortunately, we planned to bring the project in early and we have finished all the technical aspects of the project including model building and testing by Monday, April

10th, the original date for the Critical Design Review, as seen in Figures 4 & 6. The creation of the forecasting models and inventory model are the most labor-intensive aspects of the project. The forecasting model creation was allotted 30 hours to complete, which was very close to the actual time required. The inventory model was originally allotted 10 hours because it was thought to only perform simple algebra. However, we ended up using the inventory model to build out many functions to convert data from the forecasting models into a usable format and display the results in an intuitive way to be used in benchmarking analysis. We allotted 10 more hours to the inventory model for a total of 20. We essentially borrowed the 10 hours of time from the testing and validation phase which was originally allotted 40 hours and was brought down to 30, as seen in Figures 2 & 6. The testing and validation phase was used to troubleshoot our forecasting and inventory models and develop a benchmarking system to compare our order suggestions with the baseline of the restaurant’s current ordering process of using intuition.

We finished the testing and validation phase on Sunday, April 9th, the day before the original Critical Design Review. Now, the Critical Design Review has been rescheduled for Monday, April 17th. The poster was created and finalized the week of April 10th, the video was created and finalized on Monday, April 17th, and the final changes and edits to the report were completed between April 17th and April 23rd.

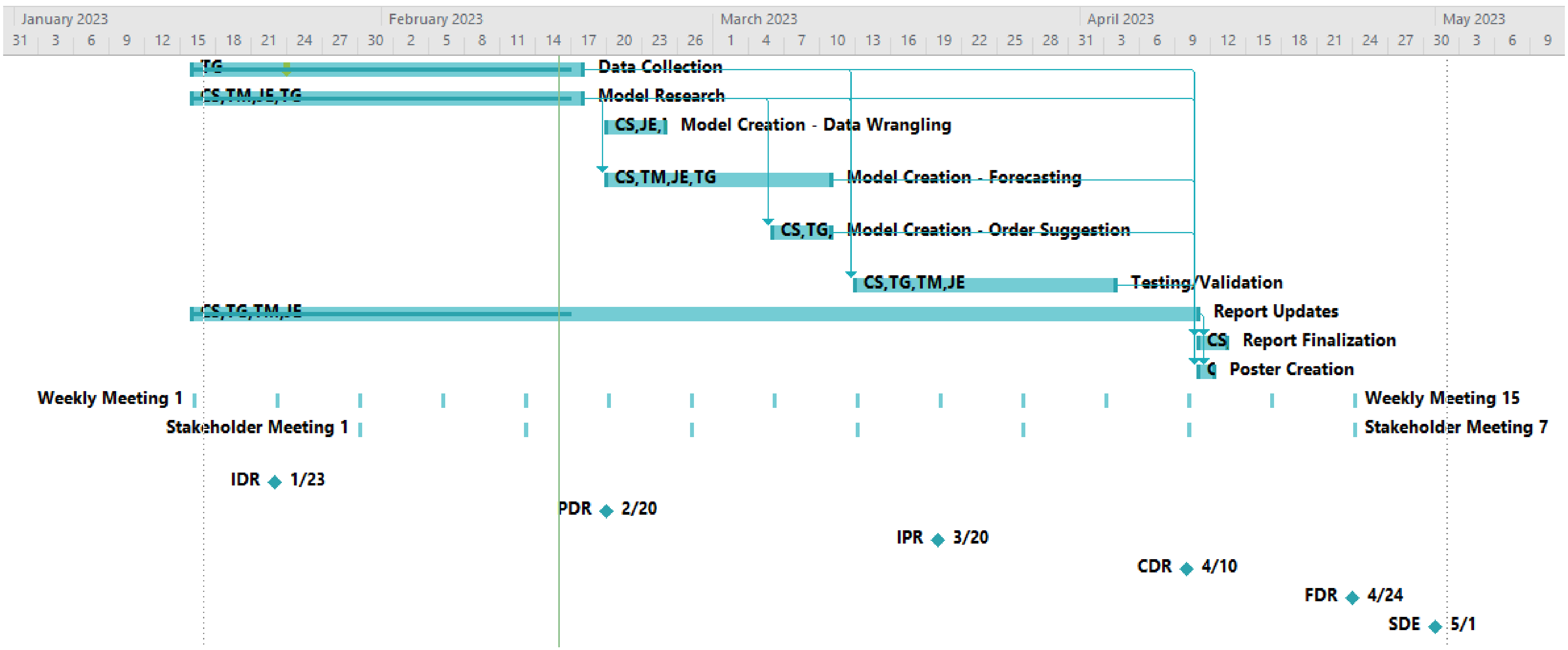


Figure 3: Gantt chart for the project

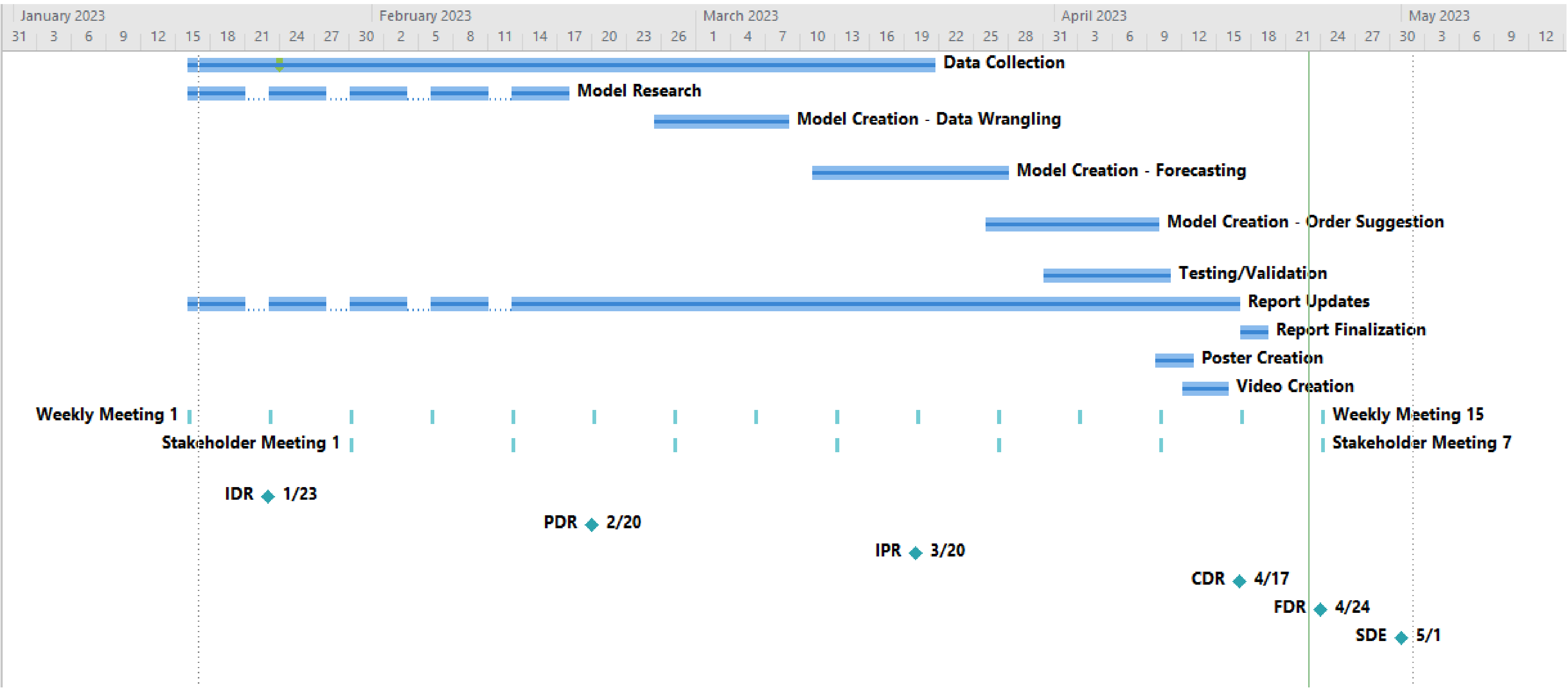


Figure 4: Current Gantt chart for the project

Task Name	Work	Duration	Start	Finish	Predecessors	Resource Names	Cost
Data Collection	15 hrs	25 days	Mon 1/16/23	Fri 2/17/23		Trevor Gilbert	$555.00
Model Research	30 hrs	25 days	Mon 1/16/23	Fri 2/17/23		Chris Schoen,Tyler Mason,Jonathan Enriquez,Trevor Gilbert[50%]	$1,110.00
Model Creation - Data Wrangling	10 hrs	5 days	Mon 2/20/23	Fri 2/24/23		Chris Schoen[75%],Jonathan Enriquez[50%],Tyler Mason[50%],Trevor Gilbert[25%]	$370.00
Model Creation - Forecasting	30 hrs	15 days	Mon 2/20/23	Fri 3/10/23	2	Chris Schoen[75%],Tyler Mason[50%],Jonathan Enriquez[50%],Trevor Gilbert[25%]	$1,110.00
Model Creation - Order Suggestion	10 hrs	5 days	Mon 3/6/23	Fri 3/10/23	2	Chris Schoen[75%],Trevor Gilbert[25%],Tyler Mason[50%],Jonathan Enriquez[50%]	$370.00
Testing/Validation	40 hrs	16 days	Mon 3/13/23	Mon 4/3/23	1,2,4,5	Chris Schoen[50%],Trevor Gilbert[50%],Tyler Mason[50%],Jonathan Enriquez[50%]	$1,480.00
Report Updates	30 hrs	61 days	Mon 1/16/23	Mon 4/10/23		Chris Schoen[10%],Trevor Gilbert[50%],Tyler Mason[30%],Jonathan Enriquez[50%]	$0.00
Report Finalization	5 hrs	2.44 days	Tue 4/11/23	Thu 4/13/23	1,2,4,5,6,7	Chris Schoen,Trevor Gilbert,Tyler Mason,Jonathan Enriquez	$185.00
Poster Creation	3 hrs	1.31 days	Tue 4/11/23	Wed 4/12/23	1,2,4,5,6,7	Chris Schoen,Trevor Gilbert,Tyler Mason,Jonathan Enriquez	$0.00
Weekly Meeting	**60 hrs**	**71 days**	**Mon 1/16/23**	**Mon 4/24/23**			**$2,220.00**
Stakeholder Meeting	**28 hrs**	**60.13 days**	**Mon 1/30/23**	**Mon 4/24/23**			**$1,036.00**
IDR	0 hrs	0 days	Mon 1/23/23	Mon 1/23/23			$0.00
PDR	0 hrs	0 days	Mon 2/20/23	Mon 2/20/23			$0.00
IPR	0 hrs	0 days	Mon 3/20/23	Mon 3/20/23			$0.00
CDR	0 hrs	0 days	Mon 4/10/23	Mon 4/10/23			$0.00
FDR	0 hrs	0 days	Mon 4/24/23	Mon 4/24/23			$0.00
SDE	0 hrs	0 days	Mon 5/1/23	Mon 5/1/23			$0.00

Figure 5: Task layout for schedule and resources

Task Name	Work	Duration	Start	Finish	Predecessors	Resource Names	Cost	Actual Duration
Data Collection	30 hrs	46.88 days	Mon 1/16/23	Tue 3/21/23		Trevor Gilbert	$1,110.00	46.88 days
Model Research	30 hrs	25 days	Mon 1/16/23	Fri 2/17/23		Chris Schoen,Tyler Mason,Jonathan Enriquez,Trevor Gilbert[50%]	$1,110.00	25 days
Model Creation - Data Wrangling	25 hrs	11.94 days	Sat 2/25/23	Wed 3/8/23		Chris Schoen[75%],Jonathan Enriquez[50%],Tyler Mason[50%],Trevor Gilbert[25%]	$925.00	11.94 days
Model Creation - Forecasting	30 hrs	17 days	Sat 3/11/23	Mon 3/27/23		Chris Schoen[75%],Tyler Mason[50%],Jonathan Enriquez[50%],Trevor Gilbert[25%]	$1,110.00	17 days
Model Creation - Order Suggestion	20 hrs	15 days	Sun 3/26/23	Sun 4/9/23		Chris Schoen[75%],Trevor Gilbert[25%],Tyler Mason[50%],Jonathan Enriquez[50%]	$740.00	15 days
Testing/Validation	30 hrs	11 days	Fri 3/31/23	Mon 4/10/23		Chris Schoen[50%],Trevor Gilbert[50%],Tyler Mason[50%],Jonathan Enriquez[50%]	$1,110.00	10 days
Report Updates	30 hrs	83 days	Mon 1/16/23	Sun 4/16/23		Chris Schoen[10%],Trevor Gilbert[50%],Tyler Mason[30%],Jonathan Enriquez[50%]	$0.00	76 days
Report Finalization	5 hrs	2.44 days	Mon 4/17/23	Wed 4/19/23		Chris Schoen,Trevor Gilbert,Tyler Mason,Jonathan Enriquez	$185.00	0 days
Poster Creation	5 hrs	3.06 days	Sun 4/9/23	Wed 4/12/23		Chris Schoen,Trevor Gilbert,Tyler Mason,Jonathan Enriquez	$74.00	0.06 days
Video Creation	5 hrs	4 days	Wed 4/12/23	Sat 4/15/23		Chris Schoen,Trevor Gilbert,Tyler Mason,Jonathan Enriquez	$74.00	0 days
Weekly Meeting	**60 hrs**	**98.13 days**	**Mon 1/16/23**	**Mon 4/24/23**			**$2,220.00**	**78.5 days**
Stakeholder Meeting	**28 hrs**	**84.13 days**	**Mon 1/30/23**	**Mon 4/24/23**			**$1,036.00**	**60.09 days**
IDR	0 hrs	0 days	Mon 1/23/23	Mon 1/23/23			$0.00	0 days
PDR	0 hrs	0 days	Mon 2/20/23	Mon 2/20/23			$0.00	0 days
IPR	0 hrs	0 days	Mon 3/20/23	Mon 3/20/23			$0.00	0 days
CDR	0 hrs	0 days	Mon 4/17/23	Mon 4/17/23			$0.00	0 days
FDR	0 hrs	0 days	Mon 4/24/23	Mon 4/24/23			$0.00	0 days
SDE	0 hrs	0 days	Mon 5/1/23	Mon 5/1/23			$0.00	0 days

Figure 6: Current task layout for schedule and resource

3.7 Responsibilities

The original schedule and budget breakdown by task is shown in Figure 5, but over the course of the project we had to adapt to changing time estimations which can be seen in Figure 6. Figure 6 shows the current tasks, predecessor tasks, their planned dates, resource allocation, and labor cost for the project. We assign team members to tasks and estimate their percentage contribution based on their expertise and responsibility for the task.

As the software lead, Chris will lead the development of the forecasting and inventory models and ensure the accuracy of the models through verification testing. He will also ensure that the stakeholders understand and can use the proposed models.

As the operations lead, Trevor will keep the team informed on current restaurant practices and serve as a direct line of communication to upper management. Additionally, he will collect all the data to be used in the forecast and inventory models.

As the Data Analyst, Jonathan will assist in model development. He will also be the primary writer and editor for the design reports.

As the project manager, Tyler will schedule tasks, assign resources, ensure deadlines are met, and maintain the budget. He will ensure the project is delivered on time and adapt the schedule to any overages. He will also assist in model development and validation.

3.8 Schedule Timeline

The timeline in Figure 7 shows an overall spread of where we are on the project and the major milestones throughout the semester that we need to be working around. The milestones of the design reviews are used to guide our completion of necessary project tasks. The Critical Design Review was pushed back to April 17th. Despite this, we have finished all the technical aspects of the project by the original date of April 10th. Final edits to the report, poster creation, and video creation were all completed before the Final Design Review.

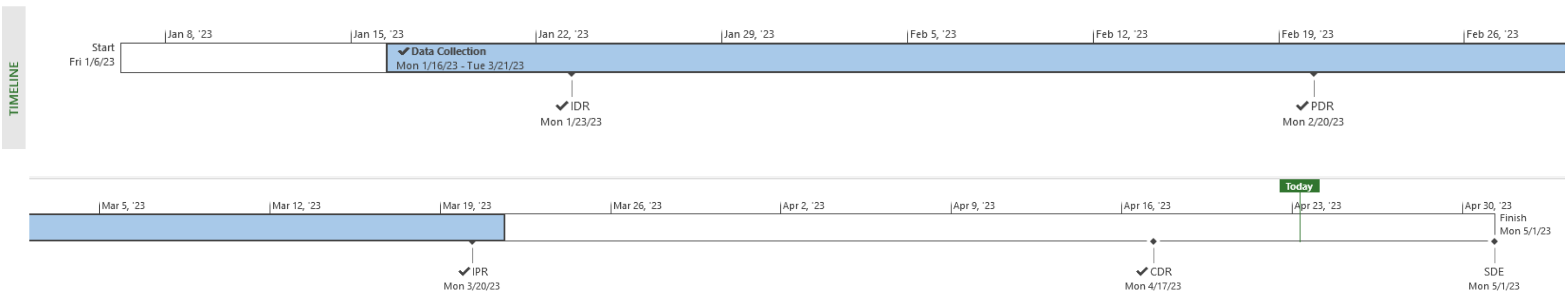


Figure 7: Timeline for project schedule

3.9 Budget

We created a conceptual budget for our project, this was done because the software we used was all free through the school but would not be in a real-world setting. Additionally, Family Restaurant records all its data in-house, so we do not need to pay a third party to access it, this makes labor costs the main costs of the project. We calculated the labor costs using a $37/hr rate for each team member. We estimated the total work hours for each task and assigned team members' time for each task as a percentage of the total work hours. Figure 6 displays the itemized cost for each task. Table 1 displays the total cost of the project with the total duration and work hours of the project. In addition, it displays the duration, work hours, and cost of the project used so far with the remaining.

Software costs were calculated for the necessary software packages for four users over four months. The required software is Microsoft Office, Microsoft Visio, and the python IDE, PyCharm.

Table 1: Current budget based on schedule

	Duration	Work	Cost
Current	115d	298h	$9,694.00
Baseline	0d	0h	$0.00
Actual	111.85d	290h	$9,398.00
Remaining	3.15d	8h	$296.00

Table 2: Software Budget

Software Cost Calculations		
Software	Monthly Cost	Cost
Microsoft Office Package -Basic	$ 6.99	$ 111.84
Microsoft Visio	$ 5.00	$ 80.00
PyCharm	$ 20.75	$ 332.00
Python (Free)	$ -	$ -
Total		$ 523.84

3.10 Available and Required Resources

3.10.1 Available Resources:

- Personnel - Chris Schoen, Tyler Mason, Jonathan Enriquez, and Trevor Gilbert
- Software - Python, PyCharm, Microsoft Excel, Microsoft Project

3.10.2 Required Resources:

- Data - sales numbers, menu ingredient requirements, order schedule and cost, and delivery schedule
- Model - research papers to help create a usable model for the forecasting and order suggestions based on real-world examples
- Personnel - experienced professionals to consult about the feasibility of the model

Chapter 4 Results and Discussions

4.1 Forecast Model Results

4.1.1 Interpreting Data Validity

The first part of our results is our forecasting models. Before checking how well our models perform, we needed to make sure that our data was okay to be used in the models. The first step is making sure that our data is stationary. This represents the fact that the data doesn't have pre-existing bias. To do this, we ran an autocorrelation on the data.

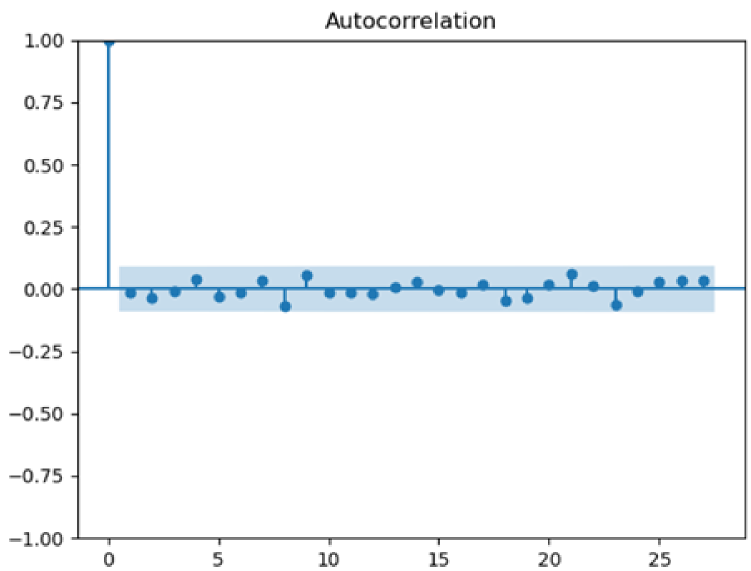
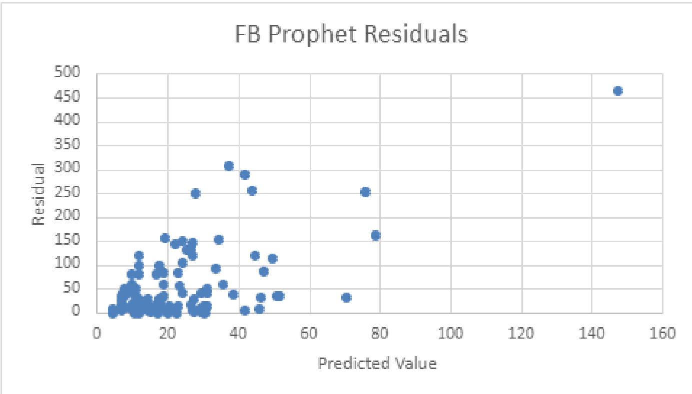
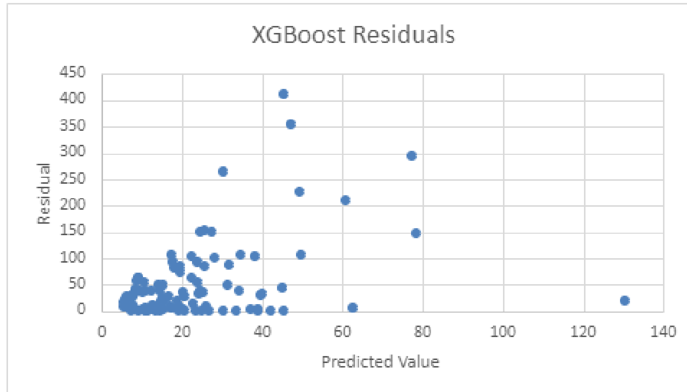


Figure 8: Autocorrelation chart for sales data

We are looking for a large drop off from the 1.00 in autocorrelation to show that there isn't a pattern in the data, which can be seen in Figure 8. Another thing we are looking for in our models is that we are trying to make sure that variance is independent between data points. To do this, we can look at the residual plots of the model.





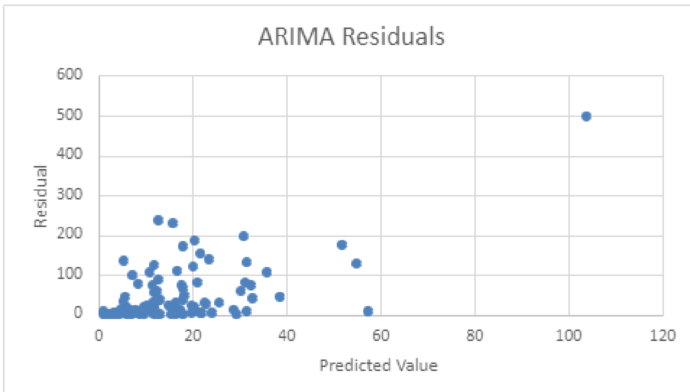


Figure 9: Model Residuals
(Top left – Facebook Prophet, Top right XG Boost, Bottom – ARIMA)

Looking above at Figure 9, we can see that the variance in residuals looks random. There are a few outliers, but overall, we can mark off the variance assumption as being met. After making sure about the validity of our models, we then need to evaluate the model's performance.

4.1.2 Interpreting Model Accuracy

To evaluate our models, we looked at three different statistics: RMSE (Root Mean Square Error), normalized RMSE, and MAPE (Mean Absolute Percent Error). RMSE is a statistic that looks at the difference between the predicted values and the actual values. This is important in showing how accurate your model is. Because RMSE is hard to put a set number on for what is a “good RMSE” vs “bad RMSE”, normalizing the value allows you to put into reference with your data. Because of this, we evaluated the normalized RMSE, which uses the max and min residuals as the endpoints and puts the RMSE in terms of them, effectively a value between 0 and 1, with values closer to 0 being more accurate. Lastly, MAPE is a value that looks at the average difference between the predictions and the actual values, in the form of a percent. It can be interpreted as the average difference between them, showing the consistency of your forecasts.

Table 3: Forecasting Results for January 2023

Month Average Data			
Measures	**FB**	**ARIMA**	**XGB**
AVG Residual	114.19	184.70	218.91
RMSE	10.58	13.57	14.75
Normalized RMSE	0.04	0.07	0.05
MAPE	74.78	32.71	73.96

Table 4: MAPE Interpretations [11]

MAPE	Interpretation
<10	Highly Accurate Forecasting
10-20	Good Forecasting
20-50	Reasonable forecasting
>50	Inaccurate forecasting

4.1.3 Facebook Prophet

Looking at Table 3 we can see the results for each model. The first statistic to look at is RMSE, where the FB Prophet model had a 10.58. While this can be considered a good result, it is easier to judge with the Normalized RMSE. This value is between 0 and 1, so the result of a .04 is extremely telling for the accuracy of our model. The value that becomes more concerning is the MAPE, which resulted in a 74.78. Comparing that value to Table 4, that puts us in the “inaccurate forecasting” range. After investigating the results, we found that it was caused by inaccuracies coming out of seasonal high periods causing certain items to overshoot drastically while the large bulk of predictions were close to the actual value.

4.1.4 ARIMA

We can see the results of the ARIMA model’s forecasting in Table 4. The ARIMA model performed well in terms of RMSE and Normalized RMSE with results of 10.58 and 0.07 respectively. This shows that the accuracy of the of the model is in line with the goals of the project. Looking at MAPE, we can confirm this accuracy, using Table 4 as the scale, we see that an MAPE of 32.71 is along the lower part of the range of “Reasonable Forecasting”. Overall, based on our model criteria, we can say that ARIMA has succeeded in all statistical checks.

4.1.5 XGBoost

In Table 4, we see the results of the XGBoost model, the first statistic being looked at is RMSE. The XGBoost model had an RMSE of 14.75, which was normalized to 0.05. This shows that the model is accurate in terms of its overall data. The value that shows the issues with the model comes in the form of the MAPE, which ended up being 73.96. This occurred due to some heavy outliers in the results causing the MAPE to reflect an inaccurate forecast despite most of the forecasts being close to the actual values.

4.1.6 Combined Model Results

Looking at Table 3, we can see the results of each model for the month of January 2023. We tested the models using the most up-to-date sales and inventory data we had to run our models with. From these results, we can see several big takeaways. The RMSE and normalized RMSE for all the models were close. This can be seen heavily in the form of the normalized RSME being between .04 and .07 in an available range of 0 to 1. The big statistic where we see the difference is MAPE. With MAPE, we see a large gap between the ARIMA model and the Facebook Prophet and XGBoost models, around 40. This signifies that while the Facebook Prophet model might display a slightly higher accuracy, when it is wrong, it is very wrong.

It can be seen in Table 4 that models are aiming for a MAPE in the range of 10-20 for excellent forecasting, where anything over 50 is inaccurate. We originally stated that the forecasting model should be accurate within 10% as a part of our minimum success criteria. Now that we understand MAPE and the different ranges of forecasting accuracy, we understand that accuracy within 10% indicates a highly accurate forecast, as seen in Table 4. This is a highly unrealistic expectation for a minimum success criterion. In reality, a more appropriate minimum success criteria would be a model that is accurate within 20-50%, which is indicative of a reasonable forecast. With our new criteria we can see that the ARIMA model is within the expected range, at 32.71, while the XGBoost and Facebook Prophet models are not. We believe that our Facebook Prophet and XGBoost models' error is being heavily affected by several outliers pushing the model outside of our minimum success criteria.

4.2 Inventory Model Results

We measured the performance of each model by feeding the forecasted sales value through the inventory model to calculate an order suggestion in a number of cases for each protein. Each case contains a set number of pounds of protein depending on the protein and the supplier. The cost of a case for each protein is known and applied to the total number of cases to develop a total cost for each protein. We used four weeks in the month of January in 2023 as the test period for our models. We compared the costs of the respective models to the actual cost from sales and the costs from protein order invoices during that month.

Table 5: Monthly Saving Statistics

Monthly Statistics					
	FB Prophet	**ARIMA**	**XGBoost**	**Hybrid**	**Invoice**
Excess Spending	60%	11%	78%	29%	69%
Percent Savings	6%	34%	-5%	30%	

4.2.1 Facebook Prophet

As seen in Table 5, FB Prophet spends 60% more than the actual cost of sales while realizing 6% savings per month compared to the invoices, Family Restaurant's actual spending. In Week 1 of Table 6, FB Prophet spends slightly more than the actual cost while still being lower than the actual purchasing invoice costs for the restaurant. This is ideal, as the restaurant prefers to slightly overstock to ensure all demand is met. FB Prophet spends less than the restaurant in Week 3, but more than the restaurant in Weeks 2 & 4. FB Prophet spends more in the weeks following Week 1 because it is more sensitive to spikes in sales. Thus, it can accurately account for sales spikes, but will typically overspend during following periods of normal sales activity.

4.2.2 ARIMA

ARIMA only spends 11% more than the actual cost of sales while realizing a staggering savings of 34% per month compared to the invoices, Family Restaurant's actual spending, as seen in table 5. At first glance, ARIMA seems to be ideal. However, looking at Week 1 in Table 6, we see that ARIMA spends significantly less than the actual cost of sales. This implies that sales would be lost due to under-stocking; meaning, customers would not be able to order their desired dish which would result in the loss of customer good faith. This is unacceptable for the restaurant, as the loss of sales and customer good faith would not be worth the inventory cost savings. In the following weeks, ARIMA's spending is very close to the actual sales. ARIMA is less sensitive to spikes in sales, thus on a high-sales volume week such as Week 1, it will spend less than is required, but on normal sales volume weeks, it is highly accurate.

4.2.3 XGBoost

XGBoost spends 78% more than the actual cost of sales while realizing a 5% increase in spending compared to the restaurant, as seen in Table 5. Likewise, in Table 6, we can see that XBBoost spends more than the restaurant during each week of the month. Clearly, this is unacceptable as it performs worse than the baseline of the restaurant's invoices on every metric. As such, we do not recommend that the restaurant applies XGBoost.

Table 6: Monthly Spend Break Down (Numbers are transformed for anonymity)

Protein Inventory Costs							
Source	**Week 1**	**Week 2**	**Week 3**	**Week 4**	**Total Spending**	**Savings**	**Sales Residuals**
FB Prophet	$ 338.89	$ 402.57	$ 399.48	$ 429.09	$ 1,570.02	$ 107.07	$ (666.29)
ARIMA	$ 200.89	$ 273.45	$ 270.33	$ 284.40	$ 1,029.08	$ 648.01	$ (125.35)
XG Boost	$ 372.09	$ 454.83	$ 469.56	$ 473.23	$ 1,769.71	$ (92.62)	$ (865.98)
Hybrid	$ 338.89	$ 273.45	$ 270.33	$ 284.40	$ 1,167.07	$ 510.01	$ (263.34)
Actual Sales	$ 293.52	$ 247.31	$ 204.65	$ 158.26	$ 903.73		
Invoices	$ 508.84	$ 360.37	$ 406.77	$ 401.11	$ 1,677.09		

4.2.4 Combined Results

As seen in Figure 10, the actual sales during the period follow a negative linear trend. The forecasting models all follow the same trend of a relatively steep positive linear increase from Week 1 to Week 2 and then a gradual increase in the following weeks. FB Prophet is the closest to the actual on Week 1, which would accurately meet demand while still maintaining an overall savings for the month, as seen in Figure 11.

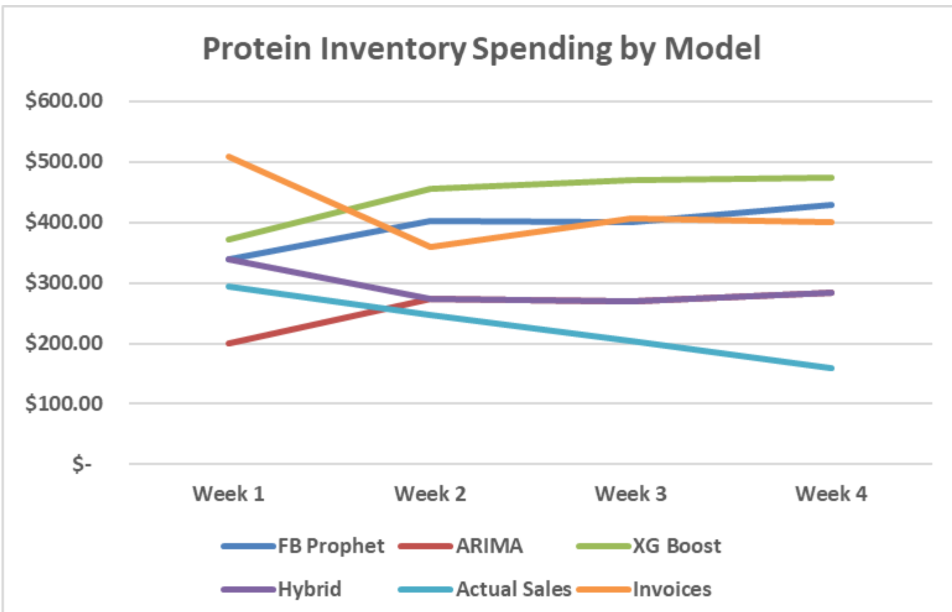


Figure 10: Line Chart of Inventory Costs

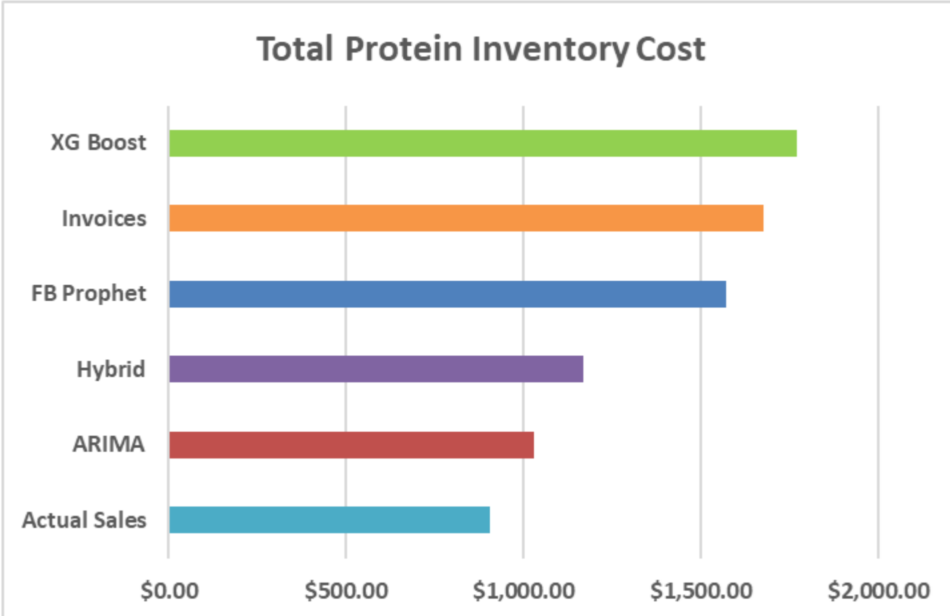


Figure 11: Bar Chart of Total Inventory Costs (Numbers are transformed for anonymity)

ARIMA is significantly under the actual sales in Week 1 but is generally closer to the actual for the following weeks, as seen in Figure 10. From Family Restaurant's perspective, the loss of sales and customer good faith in Week 1 would not be worth the cost savings in the following weeks. By this standard, FB Prophet would be the most conservative recommendation as it will meet all sales while still earning a 6% inventory cost savings over the month, as seen in Table 5. The restaurant also has the option of using a hybrid approach in model selection by applying FB Prophet during periods of known high sales volume, like during the holidays, as is the case in the first Week in January, and applying the ARIMA model during periods of normal sales volume. The hybrid approach would allow the restaurant to realize a savings of 30% per month while ensuring all demand is met, as seen in Table 5. It is seen by the invoices line in Figure 10, that the restaurant has a good intuition of the general trend of demand; however, the specific amount of their orders is historically unnecessarily high. Thus, by combining their domain knowledge with the accuracy of our models in the hybrid approach, they would be able to realize the highest savings while ensuring all demand is met.

4.3 Results Summary

The FB Prophet model performed best in terms of Normalized RMSE, with a rating of 0.05, while ARIMA performed substantially better than FB Prophet in MAPE, with a rating of 32.71. Ultimately, the ARIMA model has the highest accuracy, as it has the lowest MAPE, which is in the range of a reasonable model. As such, ARIMA realizes this most savings at 34% per month. Despite this, there are periods where the ARIMA model under-stocks protein inventory. The restaurant stated this is unacceptable, as under-stocking results in lost sales which ultimately cause loss of customer good faith. To avoid this, the restaurant can apply the FB Prophet model to realize modest savings of 6% per month while meeting all the demand. By combining their domain knowledge with our models, the restaurant can adopt a hybrid approach by applying the FB Prophet model during periods of known high demand, and ARIMA during periods of known normal demand to realize savings of 30% per month while meeting all demand.

4.4 Challenges

We originally planned to use the Long Short-Term Memory (LSTM) machine learning model along with the ARIMA and Prophet models. After doing extensive research on how to execute the LSTM model, we found that there is no Python package that easily allows you to build out the model. Instead, the model needs to be created from “scratch”. Given our set project schedule and budget, we decided the LSTM model would take too much time to build out, which would delay following tasks in the project and inflate the budget. Per recommendation of one of our professional advisors, Dr. Regenstein, we used the XG Boost model as there is a package in Python that allows one to easily build out the model and it has become an industry standard in machine learning due to its powerful predictive power and ease of use.

Another challenge faced was accurately determining the portion of proteins 3 and 5 that should be allotted to two different dishes that contain these proteins. In the restaurant management software for Family Restaurant. When either of these two entrees, which contain these proteins, are entered into the system, a button is clicked that records the size, of which there are three options to choose from. Once the portion size is selected, a protein modifier button is clicked to determine whether protein 3 or 5 will be used. The restaurant management software records the portion of the dish and the protein modifiers under two separate item numbers in the sales reports. Thus, there is no direct way to know exactly which proteins are allotted to the different portions of the two dishes. To fix this issue, we used the process of getting a weighted average on all the different size modifiers for the two types of dishes. By compiling the quantity sold of those, and then dividing the sum of each dish sold in the forecasted date range, then dividing it by the sum for the corresponding dish type, we got percentages that added up to one. From there, we took the quantity of each of the protein dishes sold in the time period and multiplied it by the weighted average for the sizes and then converted that into the protein usage amounts specific to those sizes of each dish. Finally, that number was added on to the rest of the protein usage used by other dishes in the forecast.

The ARIMA model requires daily sales data and cannot interpret missing days like the Facebook Prophet model. Thus, for days that the restaurant is closed, there is no recorded sales data. This significantly reduced the prediction power of the model as it functions on a rolling average. Without incorporating days with zero sales, the prediction is heavily skewed. To fix this, we auto-filled missing dates with a value of zero sales. With that, the accuracy of the model was significantly increased. When dealing with highly sensitive data that the business was not comfortable letting out to the public, we had to be extra cautious about how we handled receiving our data. This causes some issues when we really needed data for a deadline since we had to get information but had to go through steps to receive access, which could take days to receive. One thing we should have done to help avoid this issue was to have a clear idea of all the data we would need before the project started to get the business a list of all data to let them collect and distribute to us, this way we could have helped alleviate some of the stress of waiting for data to be approved later in the semester.

Another challenge our group dealt with is the experience level of the staff at Family Restaurant. Staff for the business are inexperienced with Python and advanced Excel tools; this will present our team with challenges as we move to install the software and tools necessary for the restaurant to use the models after the study has concluded. We will set up the model so that the staff can open the files for the models and input a set of dates that they plan on ordering for, and the model will run the models for the specified dates to give an order quantity for all protein items. The other issue is the data that needs to be input into the model to maintain accuracy. We have designed a spreadsheet for the staff to inventory all protein on a weekly basis and then upload these Excel files to a folder for the model to input.

Another challenge our team faced in this project was scheduling. We planned out a rough schedule initially in the project, trying to consider all the different parts we would need to complete. This led us to assume the coding, or model construction, periods of our project would only take around two to three weeks. Early on in that part of the project, we realized that was not the case, forcing us to extend the period to around five to six weeks, and thus moving back the following steps of our project. While we initially planned to be done ahead of the overall course schedule, these delays put us on track to complete the project right on time. This challenge just forced us to have an agile mindset and continue to update each other and the project manager on the status of our tasks and project overall.

As industrial and system engineers with entry level coding experience, we faced significant challenges while developing our three forecasting models using machine learning techniques. Developing these models required extensive research and implementation, including coding and troubleshooting, to ensure that they produced accurate and reliable data. Thankfully, python is generally considered one of the easier programming languages to learn to work with. It has simple and intuitive syntax that is easy to read and write, making it accessible to those with little or no programming experience. Python is also a high-level language, meaning that it abstracts away from many of the low-level details of computer systems, allowing developers to focus on solving problems. There is a large active community of users that provide packages and libraries to help beginners learn and work with the language. Overall, these tasks required a considerable amount of time, effort, and collaboration to ensure that our code and models were functional and met the needs of the restaurant. Despite these challenges, we persevered and successfully developed a powerful set of tools to optimize the restaurant’s ordering system. This project highlights the importance of rigorous research and testing in machine learning, as well as the challenges of developing complex models without a strong background in programming.

Chapter 5

5.1 Conclusion

With the original goal of saving the company money by reducing their spending on protein orders, we can say we were successful. With the development of effective forecasting models, and the use of that data into a built-out inventory optimization model, we can recommend a protein ordering solution to the Family Restaurant. Our recommendation is that the company uses the Facebook Prophet model as the base for their forecast. The reasoning behind this stems from our conversations with the owners and managers of the restaurant and their belief that saving more money should not come at the cost of meeting demand. While the ARIMA model has more consistency in its forecasts, due to it not considering seasonal trends or holidays, it is the only model that undercuts the sales at those times, which restaurants consider to be the most important. With the result of still saving the restaurant on average 6% compared to their current intuitive ordering, we feel comfortable that our recommendation will meet the Family Restaurant's goals of customer good faith as well as increased savings. If the company wants to maximize its savings, we recommend a hybrid of the two models. During seasons of high sales, we recommend using the Facebook Prophet model, which helps in accounting for these seasonal trends. During seasons of low sales, we recommend using the Arima model to ensure that they won't under-order and lose out on sales.

5.2 Final Recommendations

- Apply the FB Prophet model always to guarantee demand is met and savings are realized at a conservative rate.
- Use domain knowledge to apply FB Prophet during periods of known high demand and apply ARIMA during periods of known normal demand to realize the highest savings.

References

1) A. Parizad and C. J. Hatziadoniu, “Using prophet algorithm for pattern recognition and short term forecasting of load demand based on seasonality and exogenous features,” *2020 52nd North American Power Symposium (NAPS)*, 2021.

2) B. Kumar Jha and S. Pande, “Time Series forecasting model for supermarket sales using FB-prophet,” *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021.

3) B. Pavlyshenko, “Machine-Learning Models for Sales Time Series Forecasting,” *Data*, vol. 4, no. 1, p. 15, Jan. 2019, doi: 10.3390/data4010015.

4) C. I. P. Mousinho, “Forecasting sales and transactions of fast-Food Stores: A proof of concept,” *RUN*, 28-Jan-2022.

5) D. A. Cranage and W. P. Andrew, “A comparison of time series and Econometric Models for Forecasting restaurant sales,” *International Journal of Hospitality Management*, vol. 11, no. 2, pp. 129–142, 1992.

6) F. M. Puspita, N. A. Primadani, and E. Susanti, “Application of material requirement planning with Arima forecasting and fixed order quantity method in optimizing the inventory policy of raw materials of sederhana restaurant in Palembang,” *Proceedings of the 5th Sriwijaya Economics, Accounting, and Business Conference (SEABC 2019)*, 2020.

7) F. R. Jacobs and R. B. Chase, “Ch 20: Inventory Management,” in *Operations and Supply Chain Management*, 14th ed., New York, NY: McGraw Hill, 2021, pp. 513–538.

8) F. S. Hillier and G. J. Lieberman, “Ch 18: Inventory Theory,” in *Introduction to operations research*, New York, NY, NY: McGraw-Hill, 2021, pp. 800–862.

9) F. Taigel and J. Meller, “Machine Learning for Inventory Management: Analyzing Two Concepts to Get From Data to Decisions,” *SSRN Electronic Journal*, 2018.

10) G. Rafferty*, Forecasting time series data with Facebook Prophet: Build, improve, and optimize time series forecasting models using the advanced forecasting tool.* Birmingham: Packt Publishing, 2021.

11) J. J. M. Moreno , A. P. Pol, A. S. Abad, and B. C. Blasco, *Using the R-MAPE index as a resistant measure of forecast accuracy*.

12) P. Verma, S. V. Reddy, L. Ragha, and D. Datta, “Comparison of time-series forecasting models,” *2021 International Conference on Intelligent Technologies (CONIT)*, 2021.

13) Romero-Gelvez, Jorge Ivan, Edisson Alexander Delgado-Sierra, and Jorge Aurelio. “Demand Forecasting and Material Requirement Planning Optimization using Open Source Tools,” 2019.

14) S. Siami-Namini, N. Tavakoli and A. Siami Namin, “A Comparison of ARIMA and LSTM in Forecasting Time Series,” 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 2018, pp. 1394-1401, doi: 10.1109/ICMLA.2018.00227.

15) S. J. Taylor and B. Letham, “Forecasting at scale,” Forecasting at Scale, 2017.

16) T. Feng, Z. Zheng, J. Xu, M. Liu, M. Li, H. Jia, and X. Yu, “The Comparative Analysis of Sarima, Facebook Prophet, and LSTM for road traffic injury prediction in Northeast China,” *Frontiers in Public Health*, vol. 10, 2022.

17) T. Tanizaki, T. Hoshino, T. Shimmura, T. Takenaka, “Restaurant store management bases on demand forecasting,” *Procedia CIRP, vol. 88, pp. 580-583, 2020.*

18) W.-X. Fang, P.-C. Lan, W.-R. Lin, H.-C. Chang, H.-Y. Chang, and Y.-H. Wang, “Combine facebook prophet and LSTM with BPNN forecasting financial markets: The morgan taiwan index,” *2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2019.

19) X. Dairu and Z. Shilong, “Machine learning model for sales forecasting by using XGBoost,” *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, 2021.

Appendix A: Acknowledgments

We extend our gratitude to:

Dr. Adeel Khalid, Professor & Assistant Chair Graduate Program Coordinator, for his valuable feedback and continual guidance in the successful completion of our project.

Cam Avent, Director of Data Analytics for Union Square Hospitality, for his humor and guidance in model research and data storytelling.

Dr. Jonathan Regenstein, Professor & Executive Director of Research and Analytics for Sylvan Road Capital LLC, for his inspiration and guidance in applying advanced data science principles.

Family Restaurant, for graciously allowing us to use their data and providing valuable insights into their business.

Appendix B: Contact Information

Tyler Mason – Project Manager

Email: tylermason415@gmail.com

Chris Schoen – Software Specialist

Email: christopherhschoen@gmail.com

Trevor Gilbert – Operations Specialist

Email: jtrevorgilbert10@gmail.com

Jonathan Enriquez – Data Analyst

Email: enriquezj641@gmail.com

Appendix C: Reflections

Chris Schoen:

My experience on this project has only been rewarding. Through a lot of struggles in learning about forecasting models to coding in python it helped me grow and learn more about the subject. Some challenges I had throughout this project were managing my time, utilizing GitHub and git to coordinate code, and coming up with solutions and explanations to the different problems we found throughout the project. I currently work 30 hours a week as well as taking five classes, which can make setting aside 2-5 hours a week to meet and work together difficult, luckily our group was on the same page, and this allowed us to keep up with our deadlines and handle when we got behind effectively. Lastly, I had a lot of fun working with my group mates on issues and finding solutions that made sense to all of us, it is probably the most rewarding part of this whole thing. Overall, I wouldn't change anything!

Trevor Gilbert:

Working on this project with my group has helped to expand my knowledge and experience in many fields of engineering. Working in the establishment that we used for the project opened my eyes to a whole new way of dealing with issues in my everyday life and gave me a real-life connection to the material I have been learning. Overall, this project was very intriguing to work on and exciting for future endeavors in the field. One thing that did challenge us was the scheduling, with me working 40+ hours challenge us to be ahead of the game and communicate effectively which helped me in this experience. I had a great time working on this project and with this team to make something that we can be proud of.

Tyler Mason:

I am naturally ambitious and a perfectionist. Going into this project, I had unrealistic expectations for what was possible to achieve given the time and resources at our disposal. Thankfully, with the guidance of Dr. Khalid and my group mates, we were able to choose a project with a reasonable, tangible scope that could be accomplished and implemented. Throughout the project, I had to consistently manage my expectations for what was reasonable to accomplish and ultimately, I am extremely pleased with the result. I think we found the perfect balance of ambition and rationality in making crucial design decisions. Ultimately, I am extremely proud of the end result, of each of my teammates, and of myself. This project proved that I truly learn best from experience. I was able to develop as an engineer, a student, a leader, a teammate, and friend. I believe this project is a great summation of my experience at KSU and it will serve me well as I begin my professional engineering career.

Jonathan Enriquez:

This project will be a defining feature of my college experience. I am very grateful to my group for allowing me to join late because of complications with my original group. Learning that the project was leaning more towards coding and creating forecasting models intimidated me at first but looking back now I'm glad I got to learn how to implement these models in a real-world situation. Also, having to manage my time between this project, my internship, and the rest of my classes helped me with my time management skills. I am very proud that this will be my mark made at KSU, and that I was able to take everything that I have learned into my future endeavors.

Appendix D: Contribution Tables

Table 7: Paper Task Distribution

Paper Contributions		
Section	**Main Contributer**	**Secondary Contributer**
Chapter1		
1.1	Trevor Gilbert	
1.2	Trevor Gilbert	
1.3	Tyler Mason	
1.4	Trevor Gilbert	
1.5	Trevor Gilbert	
Chapter 2	**Equal contribution**	
Chapter 3		
3.1	Jonathan Enriquez	Tyler Mason
3.2	Chris Schoen	Tyler Mason
3.2.1	Chris Schoen	Trevor Gilbert
3.2.2	Jonathan Enriquez	Trevor Gilbert
3.2.3	Tyler Mason	Trevor Gilbert
3.2.4	Tyler Mason	Trevor Gilbert
3.3	Jonathan Enriquez	Trevor Gilbert
3.4	Tyler Mason	Chris Schoen
3.4.1	Jonathan Enriquez	Chris Schoen
3.4.2	Jonathan Enriquez	Chris Schoen
3.5	Tyler Mason	Chris Schoen
3.6	Tyler Mason	
3.7	Tyler Mason	Jonathan Enriquez
3.8	Chris Schoen	Jonathan Enriquez
3.9	Trevor Gilbert	Tyler Mason
3.10	Trevor Gilbert	Jonathan Enriquez
Chapter 4		
4.1	Chris Schoen	
4.1.1	Jonathan Enriquez	Trevor Gilbert
4.1.2	Jonathan Enriquez	Trevor Gilbert
4.1.3	Chris Schoen	Trevor Gilbert
4.1.4	Chris Schoen	Chris Schoen
4.1.5	Chris Schoen	Jonathan Enriquez
4.1.6	Chris Schoen	Jonathan Enriquez
4.2		
4.2.1	Trevor Gilbert	Jonathan Enriquez
4.2.2	Tyler Mason	Jonathan Enriquez
4.2.3	Tyler Mason	Trevor Gilbert
4.2.4	Tyler Mason	Trevor Gilbert
4.3	Jonathan Enriquez	
4.4	**Equal contribution**	
Chapter 5		
5.1	Chris Schoen	Tyler Mason
5.2	Trevor Gilbert	Tyler Mason

Task Name	Work	Duration	Start	Finish	Predecessors	Resource Names
Data Collection	30 hrs	46.88 days	Mon 1/16/23	Tue 3/21/23		Trevor Gilbert
Model Research	30 hrs	25 days	Mon 1/16/23	Fri 2/17/23		Chris Schoen,Tyler Mason,Jonathan Enriquez,Trevor Gilbert[50%]
Model Creation - Data Wrangling	25 hrs	11.94 days	Sat 2/25/23	Wed 3/8/23		Chris Schoen[75%],Jonathan Enriquez[50%],Tyler Mason[50%],Trevor Gilbert[25%]
Model Creation - Forecasting	30 hrs	17 days	Sat 3/11/23	Mon 3/27/23		Chris Schoen[75%],Tyler Mason[50%],Jonathan Enriquez[50%],Trevor Gilbert[25%]
Model Creation - Order Suggestion	20 hrs	15 days	Sun 3/26/23	Sun 4/9/23		Chris Schoen[75%],Trevor Gilbert[25%],Tyler Mason[50%],Jonathan Enriquez[50%]
Testing/Validation	30 hrs	11 days	Fri 3/31/23	Mon 4/10/23		Chris Schoen[50%],Trevor Gilbert[50%],Tyler Mason[50%],Jonathan Enriquez[50%]
Report Updates	30 hrs	83 days	Mon 1/16/23	Sun 4/16/23		Chris Schoen[10%],Trevor Gilbert[50%],Tyler Mason[30%],Jonathan Enriquez[50%]
Report Finalization	5 hrs	2.44 days	Mon 4/17/23	Wed 4/19/23		Chris Schoen,Trevor Gilbert,Tyler Mason,Jonathan Enriquez
Poster Creation	5 hrs	3.06 days	Sun 4/9/23	Wed 4/12/23		Chris Schoen,Trevor Gilbert,Tyler Mason,Jonathan Enriquez
Video Creation	5 hrs	4 days	Wed 4/12/23	Sat 4/15/23		Chris Schoen,Trevor Gilbert,Tyler Mason,Jonathan Enriquez

Figure 12: Work break down by percent contribution