July 15, 2025

C964 SIM3 Task 2: Design and Development

John Riekena, SID#[removed]

Contents

[Part A: Letter of Transmittal 2](#_Toc203473491)

[Part B: Project Proposal Plan 4](#_Toc203473492)

[Project Summary 4](#_Toc203473493)

[Data Summary 4](#_Toc203473494)

[Implementation 5](#_Toc203473495)

[Timeline 5](#_Toc203473496)

[Evaluation Plan 7](#_Toc203473497)

[Costs 8](#_Toc203473498)

[Part C: Application 9](#_Toc203473499)

[Industry-Appropriate Security Features 9](#_Toc203473500)

[Part D: Post-implementation Report 10](#_Toc203473501)

[Solution Summary 10](#_Toc203473502)

[Data Summary 10](#_Toc203473503)

[Machine Learning 11](#_Toc203473504)

[Validation 12](#_Toc203473505)

[Visualizations 14](#_Toc203473506)

[User Guide 17](#_Toc203473507)

# Part A: Letter of Transmittal

July 11th, 2025

Steven Stevenson, Chief Marketing Officer

FakeShop.com Online Clothing Sales

4001 S 700 E

Millcreek, UT 84107

Dear Mr. Stevenson,

As we learned from the All-Hands Meeting covering the last quarter, overall sales are up, but the Summer and Winter seasons are consistently underperforming. During the meeting, the Marketing Team created the idea of adding the mailing of physical catalogs to our customers, a first in our company’s history, for those two underperforming seasons, which met almost unanimous approval. I have come up with a solution that leverages our existing internal sales records and the skill of our Digital Operations Team to support that initiative.

This proposed solution is the Seasonal Catalog Mix Predictor. The software is an application of Machine Learning to our existing sales data to create a predicted optimal ratio of item categories in the catalogs being sent out. Our proposed solution simply asks a team member for a season to predict, namely the underperforming seasons already identified by the Marketing Team, then recommends an optimal ratio of item category mix to use when creating the catalog based on past sales trends.

As the Senior Marketing Director, you well know the value of our database of existing shopping data. The leveraging of this data is crucial to keeping us competitive in the modern market, where AI and Machine Learning are becoming more and more powerful tools in our industry. Our proposed solution offers the first step into using these tools to streamline marketing. Our solution can improve the allocation of marketing budget dollars by providing direct data-driven predictions, cutting out the need for the costly investment of human-involved interpretation of historical sales data. This will lead to an improved return-on-investment in our physical catalog initiative as we commit marketing budget dollars to the creation and mailing of physical media. Inter-departmentally, we will be able to provide predictions to provide to our Inventory and Merchandising Teams of which categories of merchandise to stock ahead of the upcoming catalog push, allowing for more efficient order fulfillment for our customers. And we all know that fast fulfillment is a key indicator of customer retention.

While our Season Catalog Mix Predictor solution is specifically tailored to the physical catalog initiative, the underlying technology also represents our first foray into the Machine Learning tools that e-commerce giants have used to drive their success. Beyond the physical catalog initiative, further development of the platform could provide the Marketing Team with even more tools. Proactive marketing based on customer segment behavior, tailored communications based on individual customer behavior, and predicting upcoming trends for everything from seasons to years, or even down to weeks and months. Further development can also aid in aligning interdepartmental strategies for optimizing inventory and supply chain based on predicted trends, helping with the mitigation of overstock and understock. The Season Catalog Mix Predictor is a solution directly tied to our physical catalog initiative, but it is also an investment in the beginning of a powerful AI tool for our company’s future.

Due to the specific scope of our solution, our investment costs will be low. The sales data we will access is already the property of our company and presents no cost beyond the access of our team members and proper maintenance of team information security compliance. The timeline for development is likewise low-cost as it is short. Upon acceptance of this proposal, the completed delivery of the Season Catalog Mix Predictor to the Marketing team is projected to take only 6 weeks. For a final dollar estimate, the cost of the professional user license for the development software of $275.00 a year and the 70 labor hours dedicated to development and quality testing will total $1750. Altogether $2025.00 is a small sum for an investment in keeping our company competitive in the ever-evolving realm of e-commerce.

Since I joined the FakeShop.com team in 2020 as a Digital Operations Team member, I’ve seen this company grow and develop. My experience with software development and our company’s directive has helped me to become the Digital Operations Team Lead. In the DO Team, we are always looking to move the company forward, and the physical catalog initiative is something I look forward to contributing to. My prior experience with machine learning will aid in the development of not only our Season Catalog Mix Predictor but also allow me to offer my support and leadership in further integrating machine learning solutions into our business model, helping us achieve our team goals.

Sincerely,

John W. Riekena

John W. Riekena, Digital Operations Team Lead

# Part B: Project Proposal Plan

## Project Summary

The Marketing Team will be leading the physical catalog initiative to drive sales during our two lowest-performing seasons, Summer and Winter. One of their tasks will be to develop the contents of those catalogs. To aid this, our Seasonal Catalog Mix Predictor will provide a historically informed model’s prediction of the optimal ratio of categories to promote the most popular items in those seasons. Our solution will also provide the optimized ratios for any of the four seasons of the year, should the physical catalog initiative be expanded in the future.

For the final deliverables, we will provide the completed Seasonal Catalog Mix Predictor and a User Guide tailored to non-technical users to be supplied to all Marketing Team Members at the Marketing Team leadership’s discretion. During development, we will provide the Project Review Team with a Proof-of-Concept version of our application with a completed functioning predictive model and command line interface. Internally, our development team will provide the deliverables of the cleaned and transformed data, and Quality Assurance Testing Documentation, including a test plan and evaluation metrics. The Project Review Team will provide the deliverables of all Approval sign-off documentation at each relevant milestone during the development timeline.

Upon deployment, the Seasonal Catalog Mix Predictor will provide the Marketing Team with a simple-to-use, data-driven application to aid in catalog development. This will benefit the Marketing Team by streamlining and lowering labor costs during said development. During the maintenance phase of the lifecycle, the yearly updating of the sales data and retraining of the model will help keep the Marketing Team informed of any changing trends. Beyond the solution’s deployment, it will provide a platform that could be iterated upon to make other predictions such as breaking down categories further to generate more specific optimal ratios by more creating even more granular categories, or even expanded upon to predict other trends beyond the scope of seasons, or provided to other departments such as the Inventory Team to aid in the projected supply line needs.

## Data Summary

For our solution, we will use the already existing customer data from our extensive online sales history. The most recent 5 years of sales data will be pulled from the company server as a full spreadsheet and converted into a comma-separated value file. 5 years will be chosen for training to provide a large enough sample for training while avoiding historical bias from trends and relationships that may be out of date. These 5 years will meet the needs of our solution by providing the necessary datapoints for the gradient boosting model of our solution to generate the requested predictions.

The data will then be cleaned by removing all irrelevant features. For our specific application, the only relevant features will be Sales Year, Sales Month, and Item Category. After removing irrelevant features, all remaining data entries with incomplete features will be dropped. Any outliers will be retained to provide a more robust dataset for the model to work with. The data will then be transformed into a format of a features matrix and matching targets for training. This formatted data will be split into 4 years for training and 1 year for testing, as opposed to a simple 80/20 split to maintain seasonal variability and any other time-sensitive trends or relationships.

During the design and development phases of the project lifecycle, this cleaned data will remain static. After completed deployment the data will be updated yearly. The 5 most recent years of data will again be pulled from the server and transformed and split as before. The model will then be retrained with this updated data to keep the predictions accurate, reflecting any changes in trends and relationships as the company moves forward.

The data we will be using is from our internal servers and is authorized for any internal use by our standard User Agreement. This data will remain internal, never leaving the company network. All data will be completely anonymized through the transformation process and feature development. These conditions will satisfy any legal or ethical concerns about data usage.

## Implementation

For implementation, we will be following the SEMMA methodology. The Sample, Explore, Modify, Model, and Assess iterative steps apply very well to the development of our solution and model selection. We will apply each step of the process during our timeline. First, we will Sample our data by pulling the existing sales data from the company’s server. We will then Explore it through human review, looking at different statistics and relationships. With the data explored, we will Modify it by removing all irrelevant data and any data elements that are missing features. Any outliers will be retained, as looking ahead, the current outliers might become future trends as the solution is maintained. The modify step will also include the splitting of the data into training and testing sets. The Model step will come next, where we will iteratively test different available regression models, such as Random Forest Regression and Gradient Boosting Regression, to find an effective model. After the model is chosen, hyperparameters will be tuned to find the best support for our goal. The last step, Assess, will include using the testing data on our model to measure the improvements over baseline. We will also assess the functionality of the user interface. The Project Review Team will be integral to the Assess step as they will be representative of the internal end user.

While this is how the SEMMA methodology will apply to developing the project, something powerful about the SEMMA model is that it can also be applied to its own steps. For example, during the Model step, the SEMMA steps will be applied as such: Search for different models, Explore those models’ strengths and weaknesses when applied to our problem, Modify the chosen model through tuning, apply the chosen Models to our data, and Assess each explored model’s performance against each other.

## Timeline

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Milestone/Deliverable** | **Dependencies** | **Resources** | **Start/End Date(s)** | **Duration** |
| Project Approval | N/A | Senior Management Review | 7/11 to 7/16 2025 | 1 week |
| IDE Installation | IDE Subscription- $250.00 | IT Support for installation  Existing company workstation | 7/16- to 7/17 | 1 day |
| Project Review Team Assignment | Project Approval | Management assignment of department representatives to the Project Review Team | 7/16 to 7/17 | 1 day  Concurrent with IDE installation |
| Data Pulled and Cleaned | Project Approval | Sales History Database Access | 7/17 to 7/17 | 1 day  (5 hours)  Concurrent with the first day of Proof-of-Concept Development |
| Proof-of-Concept and Documentation | Project Approval Signoff Documentation  Labor Allocation Documentation  Pulled and Cleaned Data | Developer Labor Hours (10) - $250.00est  IDE access from existing company workstation | 7/17 to 7/18 2025 | 2 days  (10 hours) |
| Proof-of-Concept Approval | Functional Proof-of-Concept  Approval Signoff Documentation | Project Review Team Evaluation | 7/21 to 7/23 2025 | 2 days  (Team Meeting) |
| QA Testing Documentation | Proof-of-Concept Documentation | QA developer Labor Hours (10) -  $250.00est | 7/29 to 7/31  2025 | 2 days  (10 hours)  Concurrent with Solution Development |
| Complete Solution and Documentation (including User Guide) | Proof-of-Concept Code/Documentation | Developer Labor Hours (40) - $1000.00est  IDE Access from existing company workstation | 7/24 to 7/31 2025 | 1 week  (40 hours) |
| Solution Installation for Testing | Complete Solution and Documentation | Existing company workstation  Complete Solution | 8/4  2025 | Concurrent with the beginning of QA Testing |
| Solution QA Testing Completion | Complete Solution  Solution Documentation  QA Testing Documentation | Installed Complete Solution on existing company workstation  QA developer Labor Hours (10) - $250.00est | 8/4 to 8/7  2025 | 2 days  (10 hours) |
| QA Testing Completion Approval | Completed QA Testing | Project Review Team Evaluation | 8/7 to 8/9  2025 | 1 day  (Team Meeting) |
| Final Review Approval (Go/No Go) | All prior milestones/deliverables complete | Project Review Team Evaluation | 8/9 to 8/12  2025 | 1 day  (Team Meeting) |
| Final Deployment  Distribution of User Guide | All prior milestones/deliverables complete | IT Support for installation and providing access to relevant team members | 8/13 to 8/15  2025 | 3 days |
| Quarter Review | Solution in use for 3 months | Project Review Team Evaluation | 10/20  2025 | 1 day  (Team Meeting) |

Projected Project Start: 7/11/2025

Projected Project Completion: 8/15/2025

Total (Development) Labor Hours: 75

All meeting-based milestones include multiple days to allow incorporation into existing workflow commitments.

## Evaluation Plan

Verification and validation will be done during the relevant phases and at the relevant milestones following our implementation timeline. After pulling and cleaning the data, human verification of the data will be done by a development team member.

The proof-of-concept will be verified by the Project Review Team during the approval meeting. The department representatives will view a presentation by the development team of the working proof of concept with questions and feedback. The proof of concept will be validated to meet or exceed a 50% Mean Squared Error improvement over baseline. Upon the Project Review Team’s verification that the proof of concept is meeting the approved specifications, standards, and business needs, the Project Review Team will sign off on the Proof-of-Concept Approval form.

The Quality Assurance phase will verify that the completed solution executes all planned QA test cases as expected. The testing will also include validation that the completed solution meets or exceeds the 50% MSE improvement over baseline. The QA review will also include a human review of the expected documentation and user guide.

The Final Review Approval will consist of the Project Review team using a finalized live deployment of the solution. The Team will verify that the solution meets all project specifications. The Team will also review and approve the MSE metric validation.

## Costs

The costs of our solution will be very low. Included here is an itemized list of projected costs:

Hardware: Existing company hardware will be used; no cost will be incurred.

Software: Pycharm Professional Python IDE Subscription: $249.00/year, 273.90 with tax.

Google Colab: free to use.

Existing intracompany software (DocuSign, Gmail, MS Word) will be used at no additional cost.

Labor: 70 additional labor hours: $1750.00. (estimated without access to confidential Human Resources pay information)

Approval and Review meeting times will be integrated into the existing company workflow at no additional cost.

**Total:** $2025.00(rounded up)

# Part C: Application

The application is contained in the ‘Seasonal Catalog Mix Predictor.ipynb’ file, and the dataset is the ‘sales\_dat.csv’ file included in the .zip file with this document.

## Industry-Appropriate Security Features

As the application is for internal use only, no password or security access is required as the user will by necessity be using the appropriate information security features of a corporately maintained system, including but not limited to the principle of least privilege(only the requisite users will have access to the application) and both physical (keycard access to the offices) and proper infosec training (PCI DSS). The included dataset is completely anonymized of both company and customer data. It will also be hosted only on the company system and will be protected by these same safeguards.

# Part D: Post-implementation Report

## Solution Summary

The business problem to be solved was to provide the Marketing Team’s physical catalog initiative with a simple-to-use, data-driven application to aid the development of the catalog by predicting an optimal ratio of items to be included. Our solution, the Seasonal Catalog Mix Predictor, was designed to solve this, delivering the Marketing Team the application to meet those needs.

The solution uses the company’s existing sales data to train a regression model, which can then, upon user (Marketing Team member) request of an upcoming season, predict the ratio of category mix. The information is then provided in both a text/dataframe format and an easy-to-read pie chart for use in Team meetings and presentations.

The implementation plan also accounts for the maintenance of the relevant sales data and retraining of the model yearly to encompass shifting trends. The development of the solution also incurs a low cost by leveraging existing team members and resources, with the only external cost being a subscription to the PyCharm IDE.

Beyond the initial scope of the physical catalog initiative, the provided solution offers data that can be shared among other departments, such as Inventory and Merchandising, upon Management Approval. The solution also provides a platform that can be further developed to provide more predictive insights.

## Data Summary

The data used was synthesized based on multiple open-source data sets available on Kaggle.com. No data from any of these sets was used; however, 3 different sets were explored to find a human-interpreted view of trends. This was then used to inform the development of a tailored dataset for this project.

The data was developed using a weighted randomness using the NumPy library’s random.choice() method. The sample was set to include 10,000 individual entries. Only relevant features were used to simulate the cleaned data that would be used in the real-world application of the solution.

During the transformation phase, the raw data was aggregated by year, month, and category. This was then used to calculate sales by both year and month. These two dataframes were then merged to calculate category percentage ratios by month and pivoted into the final dataframe for preprocessing, giving us the year and month features and the targets of each category’s percentage contribution to the mix of total sales.

For training, the data was split based on time as opposed to the common 80/20% split. The first 4 years (2020-2023) were used for training, and the final year (2024) was reserved for testing. The data was fed into the pipeline method, where the preprocessing functions formatted the data into all numerical elements for training.

After deployment, during the maintenance phase, the data will be updated yearly as described in the project proposal. The 5 most recent years will be extracted, transformed, and loaded as in the initial training. The existing, deprecated data will be deleted.

## Machine Learning

The Machine Learning Algorithm chosen was the SciKitLearn library’s GradientBoostingRegressor() method. The model used an ensemble of decision trees. The first tree made predictions based on each category’s data. The second tree made predictions based on the errors of the first tree (the negative gradient of loss). Each subsequent tree made predictions based on all the preceding trees’ errors in the same way. Finally, the weighted sum of all the individual trees were added to the final model. As gradient boosting works only on a single variable (category in our case), the GradientBoostingRegressor() was then wrapped in a MultiOutputRegressor() from the same library was then applied to combine all the models. The final output results were set into a Pandas dataframe and appeared as follows:

A screenshot of a computer

AI-generated content may be incorrect.

The gradient boosting used is available in the SciKitLearn library. As such, the underlying algorithm was already in place. For our application, development involved developing a Pipeline method called model\_pipeline\_reg() to encompass the multiple steps to effectively use the algorithm. The Pipeline consisted of a preprocessing method, then the gradient boosting regressor wrapped in the multi-output regressor. Preprocessing included a StandardScaler() method from the SKLearn library to normalize the columnar data. A OneHotEncoder() method was included in the preprocessing component of the pipeline to transform any non-numerical data into the requisite numerical format for machine learning. The multi-output regressor then coordinated the data being fed to the gradient boosting regressor that did the actual ‘learning’. The multi-output regressor then combined the individual outputs of the gradient booster into the meaningful results for our solution. Starting with the default hyperparameters, 100 estimators and a learning step of 0.1 were used. Through iterative testing, the estimators were brought to 200 to achieve the desired 50% improvement over baseline MSE. Increasing the estimators did not create a significant enough improvement to justify their use.

The core algorithm, Gradient Boosting Regression, was chosen over other algorithms like Random Forest Regression due to its focus on reducing overfitting. Due to the ever-evolving changes in sales trends the total data set was limited to 5 years, with 4 years selected for training and 1 year split for the test case. Due to this business requirement of the solution overfitting was a major consideration when selecting and testing models.

Sometimes a customer will buy a parka in July or a swimsuit in December. Sales data contains outliers. The model chosen needed to account for those outliers. This is another consideration that led to the selection of Gradient Boosting; the algorithm’s sequential inclusion of errors and/or outliers (through each subsequent tree) which provides robustness when considering a dataset that will include many outliers.

## Validation

The Seasonal Catalog Mix Predictor uses Gradient Boosting Regression. Regression models are a form of Supervised Learning. The target variables are continuous numerical data, and the entire dataset is comprised of labeled feature/target pairs. As the Gradient Boosting Regressor iterates through each successive tree, the use of the errors from the previous trees represents learning from feedback on labeled data, which is the major characteristic of Supervised Learning.

The validation method used is a version of Time-Series Validation. The predictions are based on historical trends and applied to future predictions. By splitting the chronologically ordered data into the training and testing sets based on specific time windows (the first 4 years for training, the final 1 year for testing), as opposed to a random split. By continuing to use this method and comparing past performance when the data is updated yearly, a series of windows will be created, allowing for cross-validation to watch for concept drift or to further tune hyperparameters as the model obtains more data to learn from.

The most valuable metric used is the Mean Squared Error. The MSE is the square of the average difference between the predicted values and the actual values. The MSE was chosen as the most appropriate metric as it allows the actual quantification of the proposed ‘50% improvement’ over simply guessing using the simple average of the historical sales data.

In measuring a baseline MSE was calculated using the training data to represent a simple ‘prediction’ based on extrapolating the existing averages without any ML model. After training, we calculated the model’s predictions’ MSEs to compare the performance of the trained model versus simply applying the baseline averages to predict category ratios.

A screenshot of a graph

AI-generated content may be incorrect.

Table: MSE comparison dataframe

Our validation showed a reduction in MSE from 30.66 to 13.52. When applying R2, we found an R2 value of approximately 0.5589 or 55% after rounding. This means that overall our trained model explained over 55% of the predicted target variations than simple averaging (the baseline) would have. This means that our trained model exceeded our proposed goal of 50% MSE improvement over the baseline.

## Visualizations

The visualizations may be found in the notebook under their corresponding headings. They are included here for reference.

A chart with different colors and numbers

AI-generated content may be incorrect.

Visualization 1:

A heatmap of category sales counts broken down by month using the full dataset.

A pie chart with different colors

AI-generated content may be incorrect.

Visualization 2:

A series of pie charts representing the seasonal ratios of category sales across the entire dataset.

A graph of a bar graph

AI-generated content may be incorrect.

Visualization 3:

A bar chart comparing the MSE of the baseline dataset vs. the MSE of the trained model.

Visualization 4: *Not Pictured*

A fourth visualization, a pie chart of the catalog ratio, will be provided to the user based on the requested season’s predictions.

## User Guide

Please follow these steps to install and run the Seasonal Catalog Mix Predictor.

1. Prerequisites:
   1. The included file, ‘SCMP Folder’
   2. Access to your Google Drive
   3. An open instance of Google Colab through any Chromium-based browser (Chrome, Edge)

<https://colab.research.google.com/>

1. Upload folder to Google Drive/My Drive.
   1. In your chosen browser, sign in to your Google Drive
   2. Drag the ‘SCMP Folder’ onto the ‘My Drive’ pane and wait for the files to upload.
2. Navigate into the SCMP Folder.
3. Double click on the ‘Seasonal Catalog Mix Predictor.ipynb’ file. A new tab will open in Google Colab with the .ipynb notebook open.
4. Click the ‘Run all’ button on the Commands bar.
5. Allow Google Colab to access your Google Drive.
6. Wait for the code to execute. This should bring you to the bottom of the notebook to the text interface. If not, scroll to the bottom to find the “Start Here (User Interface)” heading.
7. Type in a season to receive a catalog ratio.
8. To try another season, click the play button on the cell under the ‘START HERE’ heading (run the ‘seasonal\_prediction’ again)

References:

No sources were quoted, paraphrased, or summarized.

Professional Communication:

This document was run through Grammarly on July 15, 2025.