WGU C964

Task 2

COMPUTER SCIENCE CAPSTONE

Anthony Munyan

Student ID # 001566287

December 12, 2023

Table of Contents

Letter of Transmittal...................................................................................................................pg 3 **Project Proposal (Section A)** .................................................................................................. **pg 5** Hypothesis and Goals……….....................................................................................................pg 3 Project Methodology ..................................................................................................................pg 6 Funding Requirements................................................................................................................pg 7 Project Benefits...........................................................................................................................pg 7 Data Security/ Protection............................................................................................................pg 7 **Executive Summary (Section B)**............................................................................................. **pg 8** Problem Statement..................................................................................................................... pg 8 Data Description……………………….. ................................................................................. pg 8 Project Methodology………..................................................................................................... pg 9 Deliverables…………................................................................................................................pg 9 Implementation Plan................................................................................................................ pg 10 Evaluation Plan.........................................................................................................................pg 10 Success Criteria………………………..................................................................................... pg10 Required Resources and Costs..................................................................................................pg 11 Project Timeline………........................................................................................................... pg 12 **Post Implementation (Section D)**...........................................................................................**pg 13** Project Purpose.........................................................................................................................pg 13 Datasets.....................................................................................................................................pg 13 Prescriptive Method………......................................................................................................pg 17 Hypothesis Verification............................................................................................................pg 19 Accuracy Analysis………........................................................................................................pg 19 Visualizations............................................................................................................................pg 21 Application Files.......................................................................................................................pg 24 Quick Start Guide……………………….................................................................................pg 25

**Letter of Transmittal**

December 6, 2023

W. Wendel Windell, CEO

Windell’s Widgets

123 Southeast Street NE

Columbus, OH 44123

Dear W. Wendel Windell:

Every company strives for growth, but it doesn’t always come trouble-free. Unfortunately, these growing pains can cause problems that decrease employee productivity, cause inventory storage issues, or make it harder to fill orders. Many companies that experience rapid or sustained growth come to realize the need to bolster their existing information infrastructure. During our tour of your distribution center, we witnessed some of these issues first-hand.

Your customers order different quantities of different finished goods, which gives your inventory a very dynamic quality. As a result, storage ‘bins’ in your facility often become under-utilized, which lead to more occupied bins than actually necessary, which in turn makes it difficult to find storage for different products off of the production lines. With your company’s forecasted sustained growth over the next several years, the problem has the potential to grow significantly. Thus, we propose a solution that will provide your personnel with the tools they need to tackle the problems and alleviate finished goods storage issues at your distribution center.

Over the years, our firm has successfully developed hundreds of applications that utilize different methods of machine learning for various clients in a wide array of business areas. These applications have provided our clients with specialized, problem-specific solutions. The proposed application will utilize machine learning to help optimize your finished goods inventory by analyzing the product, quantity, capacity, and product status for each bin. The application will find like-products from different bins that can be combined, locate expired product that can be scrapped, and calculate the current and potential bin utilization. Development of the proposed application is expected to take three to four months, and the associated costs should be roughly $60,000. We are confident that this solution will alleviate your inventory issues and help your company obtain its goals of sustained, steady growth. This is a very exciting opportunity for both of our organizations, and we look forward to discussing it further in the near future.

Sincerely,

Anthony Munyan, Lead Machine Learning Developer

**Project Proposal (A)**

Windell’s Widgets is a widget manufacturer with facilities in the Columbus, Ohio area. They manufacture eight unique products and utilize an on-site distribution center for finished goods storage. Sales growth has been increasing, with steady business growth forecasted to continue for the next several years. Their currently daily production averages 320 total units of finished goods, which typically consists of varying numbers of several different products. The company has the actual capacity to produce up to 500 units a day; they expect to reach maximum production sometime in the next decade. As previously noted, the company utilizes a small, onsite warehouse to store their finished goods, where space comes at a premium due to increased production and sales. Their clients place orders of varying quantities of varying products, which often leaves storage locations under-utilized by being not completely full after orders are filled. To help relieve storage space issues, they have commissioned our firm to develop an application that help will them optimize product storage in their warehouse.

The proposed stand-alone application will be designed for use by inventory management personnel to help optimize their finished goods storage. It will connect to their inventory database, obtain the client’s inventory data, and analyze important attributes such as: the bin number, product number, description, quantity, product status, manufacture date, and bin capacity. The application will identify bins of expired products that can be removed or like products that can be consolidated. The proposed application will utilize a direct optimization algorithm that will make decisions on tasks to be completed that will optimize current inventory, which will help create empty bins that can then be used to store other products. The application will be stand-alone, available to inventory managers via their desktop or accessible on the company’s server.

**Project Hypothesis and Goals**

Hypothesis:

This project will produce an application that will obtain and process the client’s current finished goods inventory data and provide recommendations for consolidating or emptying product storage bins to help optimize storage space.

Goals:

* The application will be 99.5% accurate in obtaining and processing the client’s inventory data.
* The application will make accurate inventory recommendations at a minimum rate of 85% for each data set and at an overall average of 94%.

**Project Methodology**

Project development will follow the CRISP-DM methodology, which stands for Cross-Industry Standard Process for Data Mining. There will be five major phases: Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment. These phases will not be sequentially strict as it is often necessary to move between phases to develop a top-quality product. The general development of the project will follow the phases below:

* **Business Understanding**: The project requirements and objectives will be set according to the needs of our client. A detailed plan will be defined for each phase and for the overall product.
* **Data Understanding**: The quality and completeness of the client’s existing data will be examined to determine if additional field identities or reformatting is required.
* **Data Preparation**: Acquire additional data or reformat if necessary. Criteria for data categorization criteria will be set, as well as the importance of each attribute. During this phase, the required tables and queries will be developed.
* **Modelling**: The algorithm will be developed. A separate, mock dataset will be prepared for algorithm training and testing purposes. The GUI for the application will be developed. Application testing will be conducted.
* **Evaluation**: The results from testing will be evaluated to determine if the application meets or exceeds the project requirements. The need or desire for additional objectives will be determined.
* **Deployment**: When approved, the application will be released to the client. Training of the appropriate personnel will occur.

**Funding Requirements**

The estimated project costs are required resources are as follows:

* + Application Development: $29,000
  + Data Preparation: 7,000
  + Training: 5,000
  + Testing: 29,000
  + TOTAL: $70,000

**Project Benefits**

The proposed application will benefit our client by helping them optimize their limited storage space. Combining bins of like-products will help increase the accuracy and productivity of warehouse employees by reducing the need to visit several bins to obtain the required quantity of a particular product and making more room for employees to move product from production to inventory.

**Security/Data Precautions**

The data being accessed and used does not contain any financial, health, or proprietary information. To protect the client’s inventory data, the application will feature a log-in screen. The user will have to enter the proper credentials to log in to the program to see any inventory information or to use the application to generate any reports.

**Executive Summary (B)**

**Problem Statement**

Windell’s Widgets is a widget manufacturer based in the Columbus, Ohio area. The organization has seen a steady increase in business over the last several years and growth is expected to continue for the foreseeable future. Their currently daily production averages 320 total units of finished goods, with an overall daily capacity of 500 units. They expect to reach maximum production sometime in the next decade. Daily output typically consists of varying numbers of several different products. The company utilizes a small, onsite distribution center to store their finished goods, where they are collected and shipped to fill customer orders. Their clients place orders of varying quantities of varying products, which often leaves storage locations under-utilized by being not completely full after orders are filled. To help relieve storage space issues, they have commissioned our firm to develop an application that will help them optimize product storage in their warehouse.

The proposed stand-alone application will be available to inventory management personnel on their desktop computer or on the company’s server. Data regarding current finished goods inventory will be retrieved from the company’s inventory database, organized into a hash table, and analyzed by a direct optimization algorithm. The algorithm will help to optimize finished goods storage by finding bins that can be consolidated and expired product that should be removed. Additionally, it will calculate the percentage of current utilized bins and the percentage of potential utilized bins.

**Data Description**

The client currently utilizes an on-site data server that stores inventory data and employs Oracle’s MySQL as their database management system. Because of the dynamic nature of their inventory, the proposed application will retrieve inventory data every time the application is opened. Data will be processed and loaded into a hash table that will feature attributes such as: bin number, product number, product description, quantity, manufacture date, status, and bin capacity.

Approximately 125 mock sets of inventory data will be created for training and testing purposes. The dataset will be split into two groups: 100 inventory scenarios for training the algorithm and 25 will for testing it.

**Project Methodology**

Project development will follow the CRISP-DM methodology, which stands for Cross-Industry Standard Process for Data Mining. There will be five major phases: Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment. These phases will not be sequentially strict as it is often necessary to move between phases to develop a top-quality product. The general development of the project will follow the phases below:

* Business Understanding: The project requirements and objectives will be set according to the needs of our client. A detailed plan will be defined for each phase and for the overall product.
* Data Understanding: The quality and completeness of the client’s existing data will be examined to determine if additional field identities or reformatting is required.
* Data Preparation: Acquire additional data or reformat if necessary. Criteria for data categorization criteria will be set, as well as the importance of each attribute. During this phase, the required tables and queries will be developed.
* Modelling: The algorithm will be developed. A separate, mock dataset will be prepared for algorithm training purposes. The prepared real inventory data will be used for testing purposes. The GUI for the application will be developed. Application testing will be conducted.
* Evaluation: The results from testing will be evaluated to determine if the application meets or exceeds the project requirements. The need or desire for additional objectives will be determined.
* Deployment: When approved, the application will be released to the client.

**Deliverables**

The proposed project will produce the following deliverables:

* + A stand-alone application that helps optimize inventory.
  + A user manual for the provided application
  + A quick start guide to assist with installation.

**Implementation Plan**

Upon acceptance by the client, the application will be released to the client for implementation. Our firm will support implementation by assisting the client with installation, training, and any additional debugging that is necessary. Implementation will consist of:

Installation:

Our firm will work with the client’s IT personnel to install the application either on the inventory manager’s desktop PC or onto the company’s server.

Training:

Our firm will train the appropriate personnel on the usage of the application and logging bugs. The training team will be available for one week.

Additional Debugging:

The testing process is expected to find and fix the vast majority of bugs that may exist in the application. However, it is impossible to find every bug that exists during testing. It’s not uncommon for additional bugs to be found after product is put to use. The client will be required to log and report any bugs that are found after implementation.

**Evaluation Plan**

During development, testing will consist of three methods: unit testing, usability testing, and acceptance testing. Unit testing will be employed throughout the development process to ensure proper function of all parts of the application. Usability testing will be conducted at the end of each development unit. Acceptance will be conducted by the client in association with our firm to ensure it meets the project requirements.

The criteria to evaluate the success of this project will be based on its goals and deliverables. At implementation, the application will be 99.5% accurate in obtaining and processing the client’s inventory data and will make accurate inventory recommendations at a minimum rate of 94% (Table B.1).

Table B.1: Success criteria

|  |  |
| --- | --- |
| **Objective** | **Success Criteria** |
| Accuracy | Makes appropriate recommendations to consolidate inventory at a rate of 85% individually and 94% collectively. |
| Data Retrieval Accuracy | The data is retrieved from the client’s database and processed with 99.5% accuracy |

**Required Resources and Costs**

The projected costs are required resources are as follows in table B.2:

Table B.2: Projected costs and resources

|  |  |  |
| --- | --- | --- |
| **Resource** | **Description** | **Cost** |
| MySQL  (Data Analytics) | Existing database management system | $0 |
| Recommendation Application  (Developers) | Development of ML algorithm and data-wrangling method, associated GUI | 48,000 |
| Data Preparation  (Data Analytics) | Evaluation and preparation of existing data, create mock data set for training algorithm | 24,000 |
| Inventory Management Training  (Management) | Training appropriate personnel to use application | 5,000 |
| Testing  (Developers) | Training and testing the application using mock data | 42,000 |
|  | **Total** | ($119,000) |

**Project Timeline**

Table B.3: Project timeline

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Event** | **Start** | **End** | **Dependencies** | **Tasks** |
| Evaluation Phase | January 3, 2024 | January 12, 2024 | None | Evaluate existing inventory data |
| Data Preparation | January 15, 2024 | February 2, 2024 | Completion of data evaluation phase | Cleanse and prepare existing inventory data, generate mock data for training and testing |
| Machine Learning Algorithm Development | February 2, 2024 | March 8, 2024 | Completion of data preparation phase | Develop the machine learning algorithm and GUI |
| ML Training | February 12, 2024 | March 8, 2024 | Basic ML algorithm must be developed | Train the machine learning algorithm |
| Application Testing | March 8, 2024 | March 22, 2024 | Machine learning training phase must be complete | Testing the application |
| Deployment | March 25, 2024 | March 29, 2024 | Testing complete, application accepted by client | Installation, training of appropriate personnel |

**Post Implementation (D)**

**Project Purpose**

This project was created to develop an application to assist inventory personnel at Windell’s Widgets in making finished goods inventory optimization decisions. The application obtains the company’s current finished goods inventory data and makes recommendations to consolidate like products from different bins or to remove inventory that has expired.

**Datasets**

For evaluation purposes, the application will access a .csv file to load inventory data. However, the application is designed to obtain the company’s inventory data from their on-site server that employs Oracle’s MySQL as their database management system (Fig D.1). When used as intended, a new dataset (the company’s current inventory) is obtained each time the application is opened. Data is processed and loaded into a hash table with the attributes bin number, product number, product description, quantity, manufacture date, status, and bin capacity (Fig. D.2). An additional attribute that measures the fullness of each bin, ‘percent full’, is calculated by the application and added to the hash table (Fig. D.3). The data is displayed in the ‘Current Inventory’ field in the application upon startup (Fig. D.4)

Fig. D.1 MySQL table featuring the company’s inventory data.

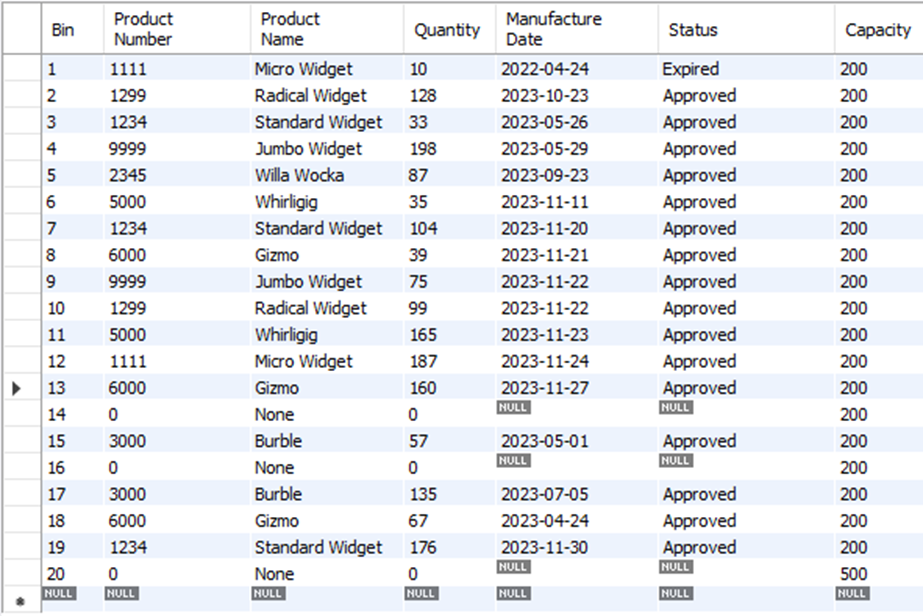


Fig. D.2 Python code for creating and loading inventory data into Hash Table.

class HashTable:  
 def \_\_init\_\_(self, initial\_size=21):  
 self.list = []  
 for i in range(initial\_size):  
 self.list.append([])  
  
 def insert(self, key, item):  
 bucket = hash(key) % len(self.list)  
 bucket\_list = self.list[bucket]  
 for keyval in bucket\_list:  
 if keyval[0] == key:  
 keyval[1] = item  
 return True  
 key\_value = [key, item]  
 bucket\_list.append(key\_value)  
 return True  
  
 def remove(self, key):  
 slot = hash(key) % len(self.list)  
 dest = self.list[slot]  
  
 for keyValue in dest:  
 if keyValue[0] == key:  
 dest.remove([keyValue[0], keyValue[1]])  
  
 if key in dest:  
 dest.remove(key)  
  
 def search(self, key):  
 bucket = hash(key) % len(self.list)  
 bucket\_list = self.list[bucket]  
  
 for pair in bucket\_list:  
 if key == pair[0]:  
 return pair[1]  
 return None

# Constructor for Inventory bin objects

class InventoryAttributes:  
 def \_\_init\_\_(self, bin, productnum, productname, quantity, mandate, status, capacity, percentfull):  
 self.bin = bin  
 self.productnum = productnum  
 self.productname = productname  
 self.quantity = quantity  
 self.mandate = mandate  
 self.status = status  
 self.capacity = capacity  
 self.percentfull = percentfull

def \_\_str\_\_(self):  
 return "%s, %s, %s, %s, %s, %s, %s, %s" % (self.bin,  
 self.productnum,  
 self.productname,  
 self.quantity,  
 self.mandate,  
 self.status,  
 self.capacity,  
 self.percentfull)

# Inserts bin objects into the hash table if connected to a database

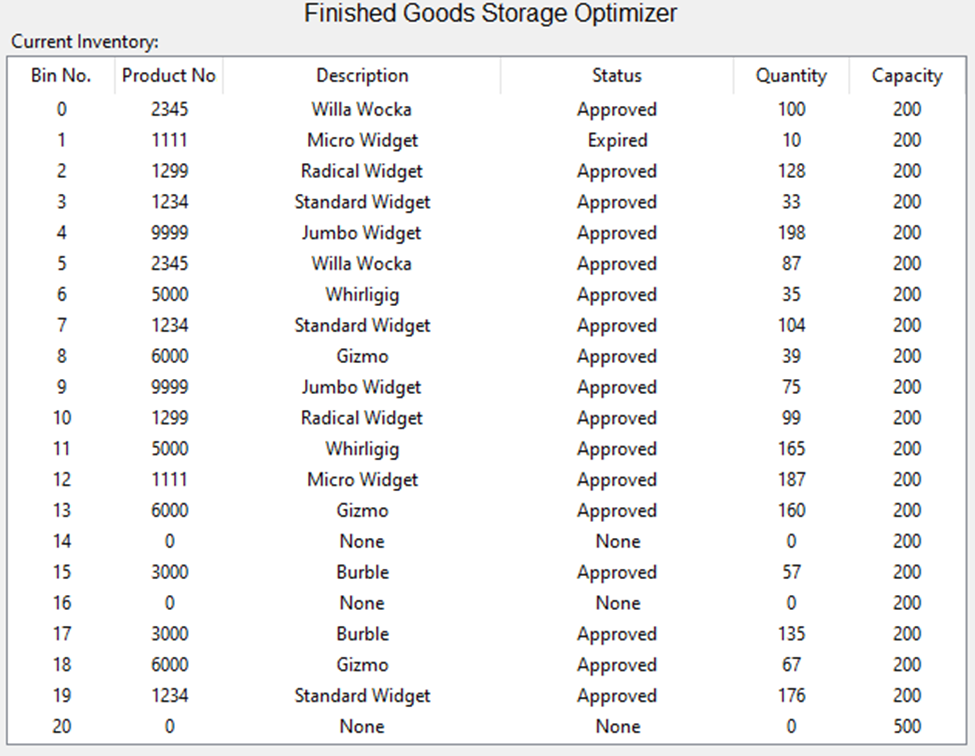
def loadInventoryData(listname):  
 for d in listname:  
 ibin = int(d[0])  
 iproductnum = int(d[1])  
 iproductname = d[2]  
 iquantity = int(d[3])  
 imandate = d[4]  
 istatus = d[5]  
 icapacity = int(d[6])  
 ipercentfull = 0  
 d = InventoryAttributes(ibin, iproductnum, iproductname, iquantity, imandate, istatus, icapacity, ipercentfull)  
 inventoryHash.insert(ibin, d)

# Loads bin objects into the hash table using a .csv file  
def use\_csv\_file(inventory\_data, bins):  
 with open(inventory\_data) as items:  
 inventory\_info = csv.reader(items)  
 for d in inventory\_info:  
 ibin = int(d[0])  
 iproductnum = int(d[1])  
 iproductname = d[2]  
 iquantity = int(d[3])  
 imandate = d[4]  
 istatus = d[5]  
 icapacity = int(d[6])  
 ipercentfull = 0  
 d = InventoryAttributes(ibin, iproductnum, iproductname, iquantity, imandate, istatus, icapacity,  
 ipercentfull)  
  
 bins.insert(ibin, d)

Fig. D.3 Python code for calculating and populating ‘Percent Full’ attribute.

# Calculates and adds bin fullness percentage to hash table  
class CalculatePercentFull:  
 for i in range(len(inventoryHash.list)):  
 status\_search = inventoryHash.search(i)  
 curquan = status\_search.quantity  
 cap = status\_search.capacity  
 percent = (curquan / cap)  
 setattr(inventoryHash.search(i), 'percentfull', percent)

Fig. D.4 Current Inventory field in finished application



To train the algorithm and test the application, 125 mock scenarios of unique current inventory data were created. 100 scenarios were used for training the algorithm and the remaining 25 were used for testing and verification.

**Prescriptive Method**

The ‘rewarehouse’ algorithm (Fig. D.5) utilized by the application is a direct optimization algorithm to make decisions, which is an effective algorithm for solving for real-world, low dimensional problems. The algorithm is designed for situations with bound constraints and few variables, a scenario that describes inventory optimization nicely. For this application, the bounds are determined by the 0-quantity value of an empty bin for the lower boundary, and by the user when they select an ‘optimization factor’ value, which sets the upper boundary. The variables are the inventory values themselves: the quantity in the current bin the algorithm is trying to move, and the quantity in the potential bin the algorithm wants to move the current quantity to.

Fig. D.5: ’rewarehouse’ algorithm

# Direct Optimization Algorithm  
def rewarehouse(self):  
 # Sets the algorithm's upper limit. The algorithm will not recommend moving inventory to a bin with a fullness   
 # percentage above this limit.  
 if rewarehouseFactor.get() == '75':  
 setRwFactor = 0.75  
 if rewarehouseFactor.get() == '85':  
 setRwFactor = 0.85  
 if rewarehouseFactor.get() == '95':  
 setRwFactor = 0.95  
 if rewarehouseFactor.get() == '100':  
 setRwFactor = 1.0  
  
 # Gets user's selection for determining the fullness percentage required for a bin to be considered 'utilized'  
 if utilizationFactor.get() == '50':  
 setUtFactor = 0.50  
 if utilizationFactor.get() == '75':  
 setUtFactor = .75  
 if utilizationFactor.get() == '100':  
 setUtFactor = 1.0  
  
 # Deletes information currently printed in rewarehousing text field   
 Rew.delete(1.0, END)  
   
 # Sets values of variables to 0 or the most appropriate value  
 totalBins = 0  
 totalUtilization = 0  
 newUtilization = 0  
 suggestedQuantity = 0  
 utilizedBins = 0  
 suggestedUtilizedBins = 0  
 potentialEmptyBin = 0  
 moves = 0  
 textLine = 3.0  
  
 # Prints preliminary information to text field  
 Rew.insert(1.0, "Suggested moves: \n")  
   
 # Gathers information from hash table   
 for i in range(len(inventoryHash.list)):  
 prodNum = getattr((inventoryHash.search(i)), 'productnum')  
 binNum = getattr((inventoryHash.search(i)), 'bin')  
 quantity = getattr((inventoryHash.search(i)), 'quantity')  
 capacity = getattr((inventoryHash.search(i)), 'capacity')  
 status = getattr((inventoryHash.search(i)), 'status')  
 binUtilization = quantity / capacity  
   
 # Keeps count of utilized bins  
 if binUtilization > setUtFactor:  
 utilizedBins += 1  
 if binUtilization == 0:  
 utilizedBins += 1  
 totalBins += 1  
 totalUtilization += binUtilization  
   
 # Checks for and suggests removal of expired inventory, ups move count, ups empty bin count  
 if status == 'Expired':  
 suggestedQuantity -= quantity  
 Rew.insert(textLine, 'Scrap expired inventory from bin %s \n' % binNum)  
 potentialEmptyBin +=1  
 moves += 1  
 textLine += 1  
   
 # Gathers information regarding potential bins   
 for j in range(len(inventoryHash.list)):  
 nextProdNum = getattr((inventoryHash.search(j)), 'productnum')  
 nextBinNum = getattr((inventoryHash.search(j)), 'bin')  
 nextQuantity = getattr((inventoryHash.search(j)), 'quantity')  
 nextCapacity = getattr((inventoryHash.search(j)), 'capacity')  
 nextStatus = getattr((inventoryHash.search(j)), 'status')  
 nextUtilization = getattr((inventoryHash.search(j)), 'percentfull')  
 potentialUtilization = quantity + nextQuantity  
  
 # Conditions required for the algorithm to suggest an inventory move  
 if nextUtilization < setRwFactor and prodNum == nextProdNum and binNum != nextBinNum and \  
 (potentialUtilization < nextCapacity) and status == 'Approved' and nextStatus == 'Approved':  
 if quantity < nextQuantity:  
 Rew.insert(textLine, 'move %s units from bin %s to bin %s \n' % (quantity, binNum, nextBinNum))  
 newUtilization = potentialUtilization / nextCapacity  
 potentialEmptyBin += 1  
 textLine += 1  
 moves += 1  
 if newUtilization > setUtFactor:  
 suggestedUtilizedBins += 1  
  
 if nextQuantity > quantity:  
 Rew.insert(textLine, 'move %s units from bin %s to bin %s \n' % (quantity, nextBinNum, binNum))  
 newUtilization = potentialUtilization / nextCapacity  
 potentialEmptyBin += 1  
 textLine +=1  
 moves += 1  
 if newUtilization > setUtFactor:  
 suggestedUtilizedBins += 1  
  
 # Inserts suggested move into text field  
 Rew.insert(1.0, 'Number of suggested moves: %s \n' % moves)  
  
 # Calculations for potential improvement  
 totalPercentUtilization = utilizedBins / totalBins  
 totalNewUtilization = newUtilization / totalBins  
 potentialUtilizedBins = potentialEmptyBin + suggestedUtilizedBins  
 potentialUtilization = potentialUtilizedBins / totalBins  
 improvementFactor = potentialUtilization - totalPercentUtilization  
  
 # Inserts current utilization, potential utilization, and potential improvement statistics into text field  
 Rew.insert(textLine, "Current Utilization: %s \nPotential Utilization: %s \nPotential Utilization Improvement: %s "  
 "\nPotential Utilized Bins %s \n\*\*\*End Report\*\*\*" % (round(totalPercentUtilization, 2),  
 round(potentialUtilization, 2),  
 round(improvementFactor, 2),  
 potentialUtilizedBins))

**Hypothesis Verification**

The hypothesis that an application could be developed that uses current inventory data to help optimize storage space in a distribution center has proven to be correct. Testing has verified that the application correctly retrieves and processes the client’s current inventory data in over 99.5% of cases. Appropriate recommendations to consolidating or remove inventory were made for more than 94% of opportunities to do so.

As always, there is room for future improvements. The algorithm, as currently designed, will make the first recommendation for consolidating inventory, which may or may not be the most optimal. Future versions should analyze all possible consolidations and make a recommendation regarding the most optimal task.

**Accuracy Analysis**

A cross-validation method was used to test the accuracy of the application’s data collection and preparation methods. 25 unique, mock datasets were constructed and loaded onto the data server. Evaluators were then asked to run the program and compare the results from the ‘current inventory’ window of the application to the dataset that was being accessed on the data server. Because highly accurate data is required for proper function of the machine learning algorithm, a stringent baseline of 99.5% was set for the baseline for a passing test. The results from collection and preparation testing can be found below in table D.6. Each of the 25 tests yielded results superior to the requirements.

The machine learning algorithm also employed cross-validation for testing. Using each of the same 25 mock datasets that were used to test data collection and preparation, testers ran the ‘Inventory Optimization’ function for each possible setting. The recommendations made by the application were compared to a key of predetermined recommendations that were prepared for each set of data. To pass, accuracy of at least 85% was required for each individual setting and an overall average of at least 94% was required. The results from testing the machine learning algorithm can be found below in table D.7. Tests results were superior to the requirements for all performed testing.

Table D.6: Data Collection and Preparation Testing Results:

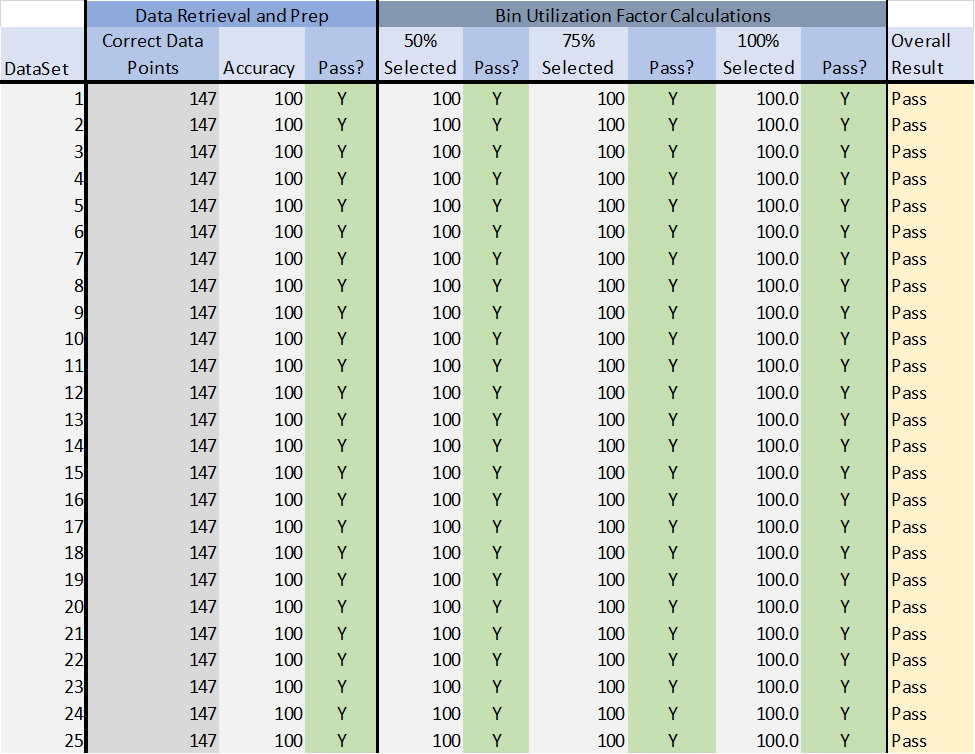


Table D.7: Machine Learning Algorithm Accuracy Testing Results



**Visualizations**

The goal of the application is to increase storage bin utilization. As shown in Figure D.8 below, the user has the option of determining the fullness a bin must have to be considered utilized. For instance, if the user selects 75% for the utilized bin percentage, any bin that is more than 75% full will be considered utilized. Empty bins are also considered a utilized bin, so if the application recommends a move that will make a bin become empty, that bin will be considered a potential empty bin. The optimization factor is also user-selected (Figure d.8). This setting determines which bins the application will consider for optimization. If the user selects a 95% optimization factor, any bin that us more than 95% full will not be available for additional inventory from a suggested move. Below are several graphs and charts pertaining to the application’s performance for the various user settings (Figures D.9 and D.10).

Fig. D.8: Utilization factor and optimization factor settings

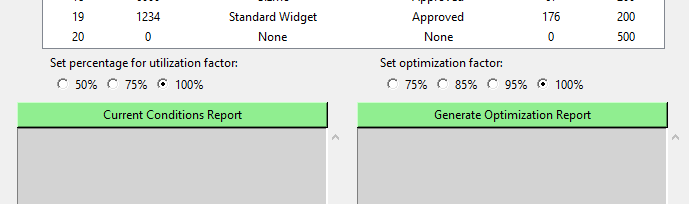


Fig. D.9: Overall possible utilized bin improvements recognized by the application.

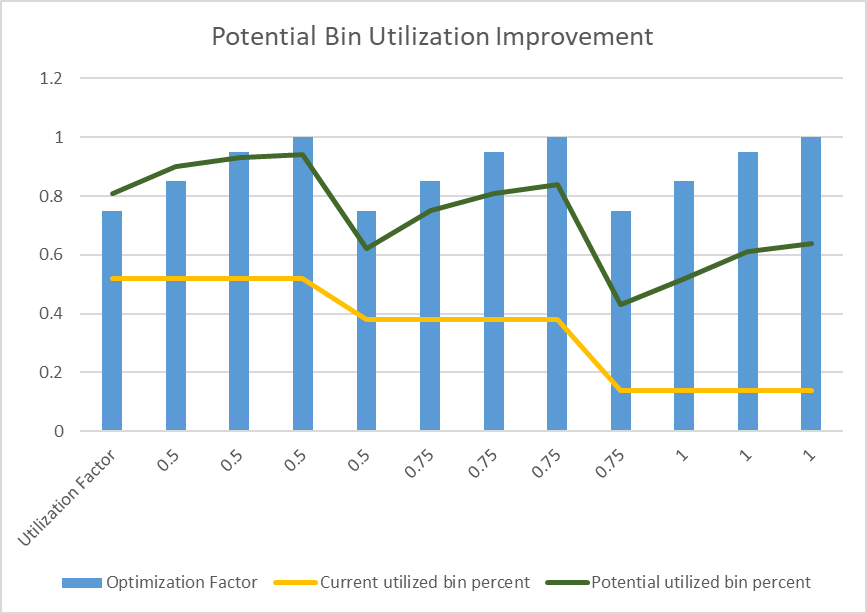
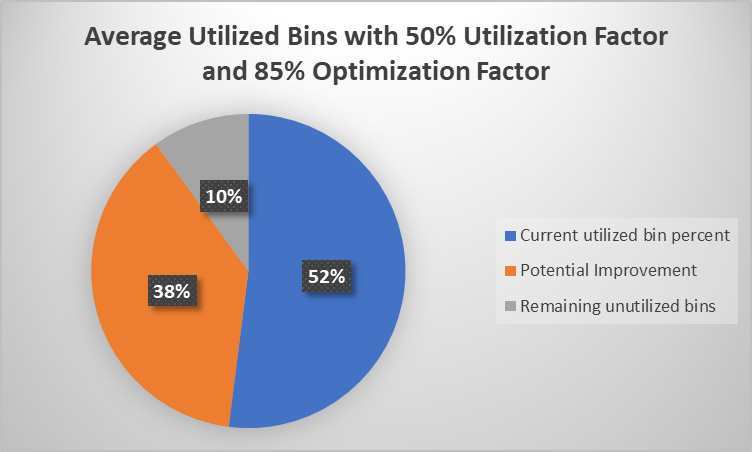
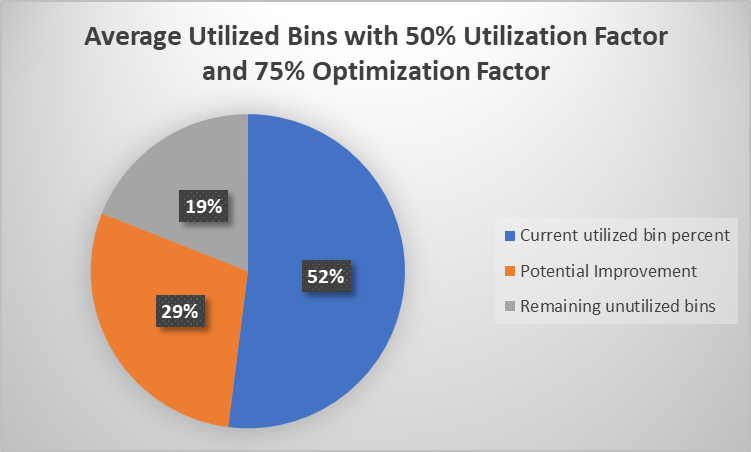
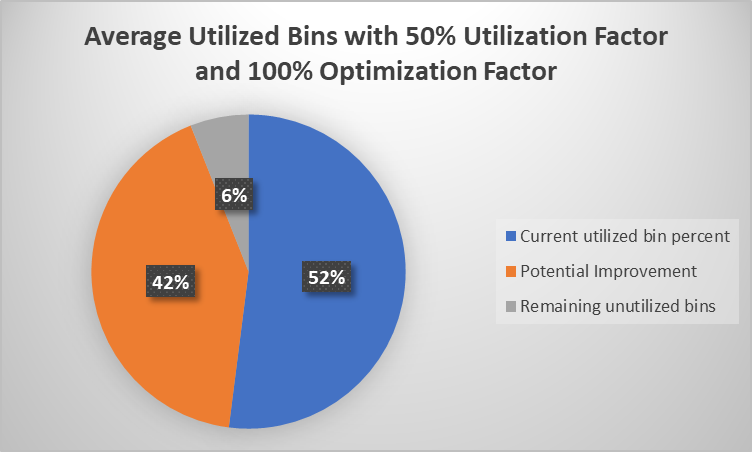
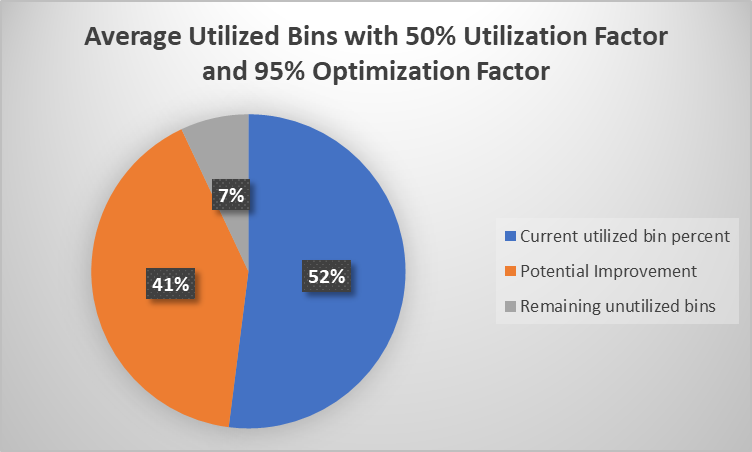
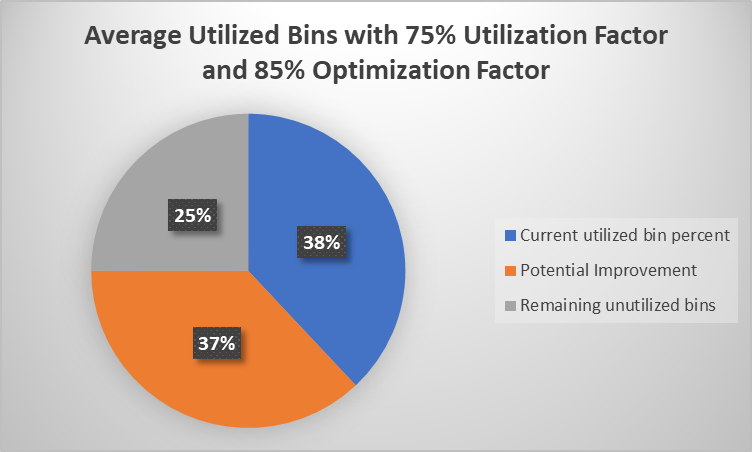
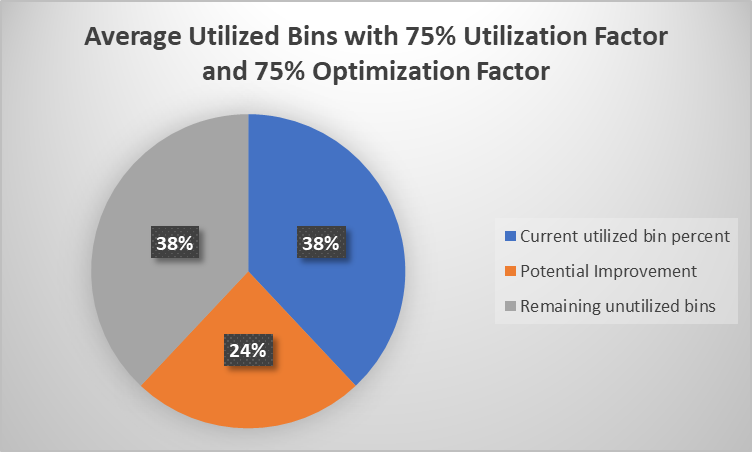
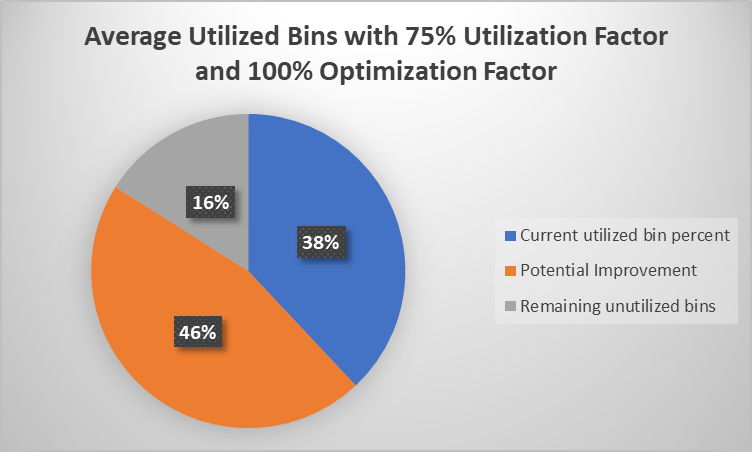
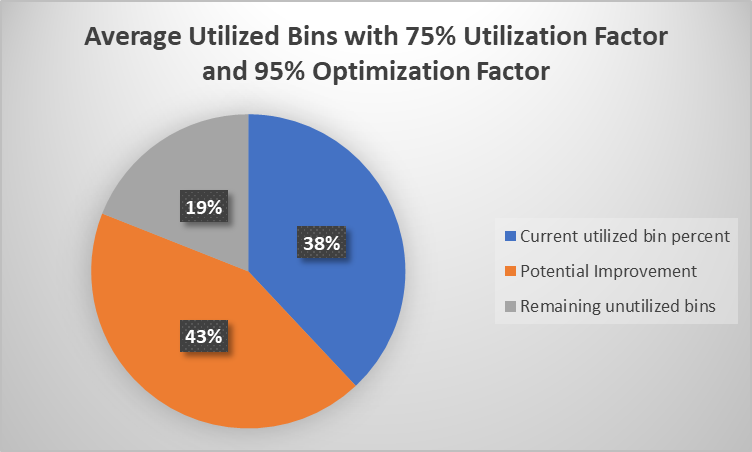


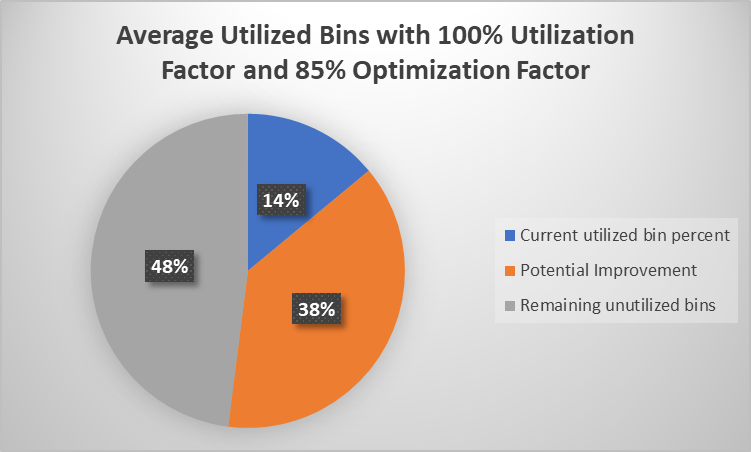
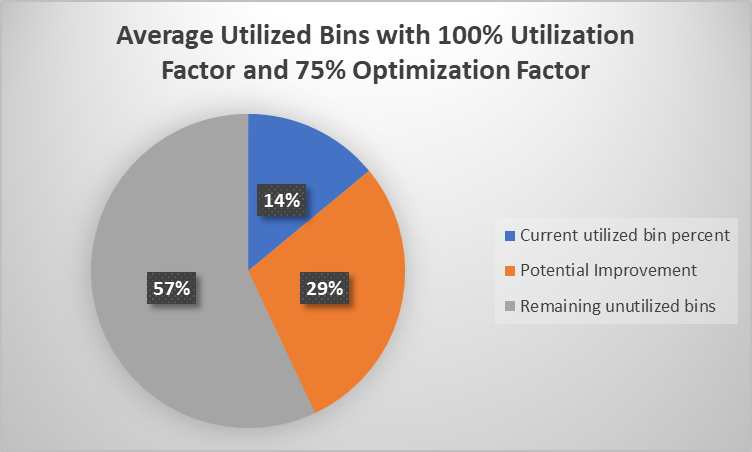
Fig. D.10: Possible utilized bin improvements for each setting.

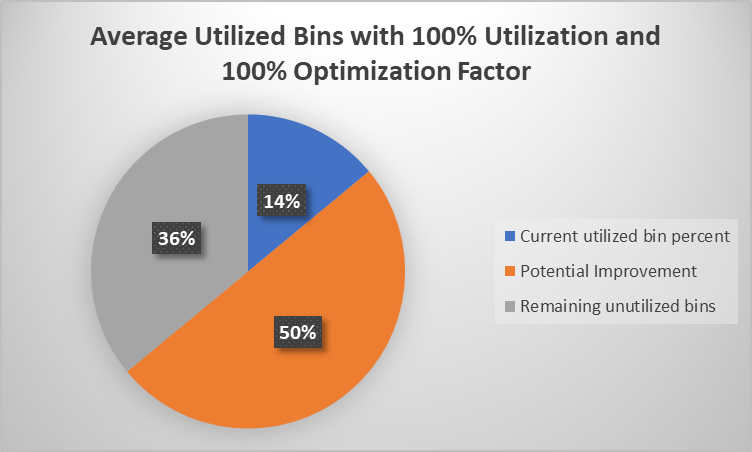
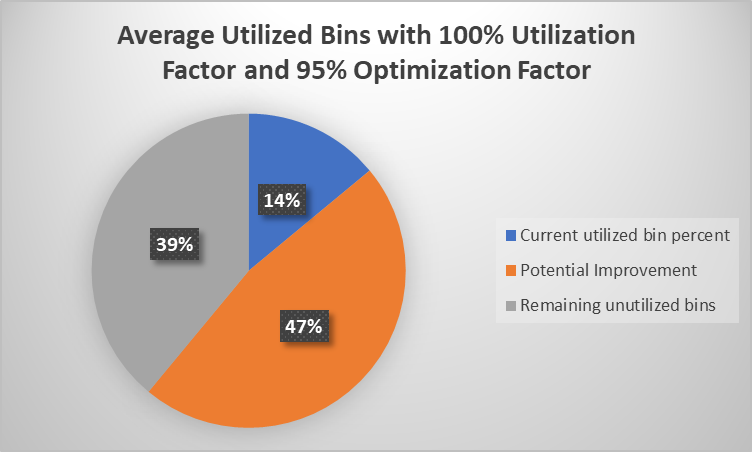












**Application Files**

The following files are included:

* c964a.main.py – The python application file
* inventory.csv – .csv file containing inventory data
* Task2.doc – Microsoft Word file containing:
  + Letter of transmittal
  + Section A: Project Proposal
  + Section B: Executive Summary
  + Section D: Post Implementation Report

**Quick Start Guide**

For evaluation of the application,

1. The application was developed using Python Community Edition (2022, 3.2) with JetBrains Pycharm. It is recommended that a similar environment be used for evaluating the application.
2. Create a new directory and unzip the ‘InventoryOptimizer’ zipped folder into the new directory.
3. Open the file named ‘main.py’ from the c964a file in the newly created directory.
4. The GUI was developed using Tkinter. Most python installations come bundled with Tkinter. If the Tkinter package is not already installed on your machine, it can be installed by right-clicking on the ‘from tkinter import \*’ statement in the application code and following the prompts to install the package.
5. Click on the ‘run’ button in PyCharm. This will start the application.
6. A log in window will appear. Enter ‘associate’ for the username and ‘ww123’ for the password and click the ‘Log In’ button.
7. Select the desired utilization factor and optimization factor from the radio button options below the current inventory table. By default, a utilization factor of 100% and an optimization factor of 100% are selected.
8. To generate a report pertaining to the current utilization conditions of the distribution center, click the ‘Current Conditions Report’ button.
9. To generate a report of tasks that could be completed to increase distribution center optimization, click the ‘Generate Optimization Report’ button.
10. When ready to exit the application, click and release the ‘Exit’ button
11. If testing of the functionality of the application and algorithm under different inventory conditions is desired, navigate to the .csv file in the newly created directory and open it as an excel file. Make the desired changes to the inventory levels and save the file (as a csv file). Re-open the Finished Goods Storage Optimizer and repeat steps 1-11.