

BELGIUM CAMPUS

MLG382

PROJECT CYO

Student	Student Number
Karl Christiaan Schmutz	577511
Noah Blaauw	601195
Gordon Mullin	600248
Eduard Jacobus Engelbrecht	600237

23/04/2025

Contents

Introduction.....	2
Research Problem	2
Dataset Details	4
Dataset (Cleaned).....	5
Hypothesis Generation.....	6
H_0 (Null Hypothesis):.....	6
H_1 (Alternative Hypothesis):	6
Correlation Data.....	6
Features that increase profits:	6
Features that decrease profits:.....	6
ML Model(s).....	7
Objective of Model Development.....	7
Models Implemented	7
Model Evaluation (Metrics).....	8
Model Performance.....	8
Web Application.....	9
Deployment Steps	10
Development Process.....	10
Conclusion	11
References.....	12

GitHub Link:

[MLG382_ProjectCYO_GroupX](#)

Render Link:

<https://mlg382-cyo-proj-groupx.onrender.com>

Introduction

For small to medium businesses Inventory Management has become a hard requirement (Praveen et al., 2020), the importance of effective inventory management cannot be underestimated. It allows for operational efficiency and improving customer satisfaction whilst, traditional approaches frequently prove to be lacking in addressing the issues associated with fluctuating demand, variable supplier lead times, and the associated risks of overstocking or stockouts. The application of machine learning presents a comprehensive solution by harnessing historical sales data, analysing market trends, and evaluating supply chain variables. Using these predictions organizations can mitigate risks. According to (Praveen et al., 2020) high competition, labour unrest and changes in governmental laws can be addressed with predictions for products and services. This produces precise demand forecasts and optimizes inventory levels. Applying an inventory management methodology gives predictive insights and encourages better decision making (Chaudhary et al., 2023). Inventory management then changes from a reactive measure to a proactive strategy.

Research Problem

Vertex PC Supply faces multiple internal inventory management challenges that significantly impact both operational efficiency and customer satisfaction. Despite collecting vast amounts of sales and inventory data, the company struggles to understand how to change this information into a proactive response to stop these issues from happening. Below are the Key issues discussed:

- Frequent Stockouts of High-Demand Items:
 - Incorrect demand forecasting often leads to empty shelves during peak demand period, resulting in lost sales and a decline in customer trust (Chaudhary et al., 2023).
- Excess Inventory and Elevated Carrying Costs:
 - Over purchasing of certain items causes surplus inventory, and a demand for capital incurring unnecessary storage expenses (Chaudhary et al., 2023).

- Low Profitability in Certain Regions or Categories:
 - According to (Sudirjo, 2023), global markets show that companies adapt their strategies to align with the unique consumer preferences, legal requirements and market conditions. This explains why a product will have different sales worldwide.
- Cost and Complexity:
 - Designing machine learning models can take a long time before it is effective and beneficial to the business. Costs are impacted not only by the time required for the model to be developed and deployed but also the expertise required to design it (Chaudhary et al., 2023).

The aim of this research is to design and implement a machine learning framework that predicts profitability based on key product, regional, and pricing attributes. By identifying what makes profit increase and decrease, Vertex PC Supply can optimize inventory decisions, regional marketing and distribution, reduce excess inventory and minimize stockouts.

Dataset Details

The dataset utilized for this research contains various data from Vertex PC Supply, which provides essential variables that significantly impact inventory management decisions.

Following are attributes within the dataset:

Attribute	Description
RegionName	Region of operation
CountryName	Country of transaction
State	State within country
City	City of transaction
WarehouseName	Warehouse fulfilling order
CategoryName	Product category (e.g., Video Card)
ProductName	Specific product name
ProductStandardCost	Standard unit cost
Profit	Profit per unit
ProductListPrice	Product list price
CustomerCreditLimit	Customer's credit limit
Status	Order status
OrderDate	Date of order
OrderItemQuantity	Quantity per order item
PerUnitPrice	Price paid per unit
TotalItemQuantity	Total items in order

The dataset provided by [Kaggle](#) captures essential details across geographic locations, product specifications, financial metrics and order details. It includes attributes such as region, country, and warehouse to track the flow of inventory, alongside product categories and specific component names for precise analysis. Pricing elements like standard cost, list price, and per-unit price help in financial evaluation, whilst quantities and order dates enable demand forecasting and trend analysis. Additional fields such as customer credit limits and order status support customer management and operational efficiency. Combining these attributes provides a comprehensive view to effectively apply machine learning for inventory optimization and decision making.

Dataset (Cleaned)

Attribute	Description
WarehouseName	Warehouse fulfilling order
ProductName	Specific product name
ProductStandardCost	Standard unit cost
Profit	Profit per unit
ProductListPrice	Product list price
CustomerCreditLimit	Customer's credit limit
Status	Order status
OrderItemQuantity	Quantity per order item
PerUnitPrice	Price paid per unit
TotalItemQuantity	Total items in order
Year	Derived feature from OrderDate
Month	Derived feature from OrderDate
Day	Derived feature from OrderDate
DayOfWeek	Derived feature from OrderDate
CostPriceRatio	Derived feature from ProductStandardCost/ProductListPrice
RegionName_Australia	One-hot encoded region name
RegionName_North America	One-hot encoded region name
RegionName_South America	One-hot encoded region name
RegionName_Canada	One-hot encoded region name
RegionName_China	One-hot encoded region name
RegionName_India	One-hot encoded region name
RegionName_Mexico	One-hot encoded region name
CountryName_United States of America	One-hot encoded country name
State_California	One-hot encoded state
State_Distrito Federal	One-hot encoded state
State_Maharashtra	One-hot encoded state
State_New Jersey	One-hot encoded state
State_New South Wales	One-hot encoded state
State_Ontario	One-hot encoded state
State_Texas	One-hot encoded state
State_Washington	One-hot encoded state
City_Bombay	One-hot encoded city
City_Mexico City	One-hot encoded city
City_Seattle	One-hot encoded city
City_South Brunswick	One-hot encoded city
City_South San Francisco	One-hot encoded city
City_Southlake	One-hot encoded city
City_Sydney	One-hot encoded city
City_Toronto	One-hot encoded city
CategoryName_Mother Board	One-hot encoded product category
CategoryName_RAM	One-hot encoded product category
CategoryName_Storage	One-hot encoded product category
CategoryName_Video Card	One-hot encoded product category

Hypothesis Generation

H_0 (Null Hypothesis):

There is no significant correlation between product pricing, product category, regional factors and overall profitability.

H_1 (Alternative Hypothesis):

Profitability is significantly affected by factors such as product list price, standard cost, regional factors. An example of this is that certain products and specific markets tend to have an increase in profits, where other products and regions would have a negative effect on profitability.

Correlation Data

Top correlations with TotalItemQuantity:

Feature	Correlation	Description
Order Item Quantity	+0.77	Strong positive correlation, more of a specific item leads to higher total order volume.
Profit	+0.46	Moderate positive correlation, larger orders generate more profit.
Product List Price	+0.37	Higher priced products tend to include larger orders.
Product Standard Cost	+0.36	Higher priced products tend to include larger orders.
RegionName_North America	+0.25	Some regional influence on larger orders

Lowest correlations with TotalItemQuantity:

Feature	Correlation	Description
Month	0.01	Very weak correlations; no strong seasonal trend.
Year	0.03	Very weak correlations; no strong yearly trend.
Warehouse Name	-0.06	This shows that the warehouse has little to no influence on the total items ordered.

ML Model(s)

To effectively address the inventory management and total items ordered forecasting objectives, multiple supervised machine learning models were developed and used during this project.

Objective of Model Development

The primary objective was to generate accurate predictions of total order amounts based on key features such as product pricing, category, regional data and individual order quantities. These insights allow for Vertex PC Supply to identify and prioritize high-demand product and region combinations. This approach supports more effective inventory planning, reduces risks in terms of stockouts or overstocking.

Models Implemented

- Random Forest Regressor
 - An ensemble learning technique renowned for its robustness in modelling complex data interactions and its calculated resistance to overfitting data.
- Gradient Boosting Regressor
 - An iterative approach that sequentially improves prediction accuracy. This is achieved by correcting prior errors, suitable for our dataset for use.
- AdaBoost Regressor
 - An ensemble method that emphasizes difficult to predict instances. Very effective in managing noisy data.
- XGBoost Regressor
 - An advanced optimized boosting algorithm. It is recognized for its speed and high predictive accuracy.
- Deep Learning (Neural Network)
 - A feedforward neural network was employed to model nonlinear relationships between features and profit metrics.

Model Evaluation (Metrics)

Each model's performance was assessed by employing standard regression evaluation metrics:

- R^2 Score:
 - Provides the proportion of the variance in the target variable defined by the model.
- Mean Absolute Error (MAE):
 - Represents the average absolute difference between predicted and actual values.
- Mean Squared Error (MSE):
 - Measures the average squared difference, penalizing larger errors to a greater extent.

Model Performance

Model	R^2 Score	MAE	MSE
AdaBoost	0.7832	0.33	0.16
Deep Learning	0.8543	0.27	0.11
Gradient Boosting	0.9279	0.17	0.05
Random Forest	0.9125	0.19	0.06
XGBoost	0.9267	0.16	0.05

With the results above the best model for our dataset was **Gradient Boosting Regressor**. This training model achieved the highest performance across all metrics.

Web Application

Purpose

The Total Item Quantity Prediction Web App is designed to provide Vertex PC Supply with a user-friendly interface that allows for the prediction of total order quantities using multiple advanced machine learning models.

Styling and Layout

The web application emphasises clarity and simplicity. It uses Dash and Dash Bootstrap Components for the UI. Form inputs are neatly segmented for different types of input data. Key UI components such as drop downs and input fields are styled with rounded edges, soft shadows, and intuitive spacing to improve readability and user experience.

Overview of functionality

Loading models

The application loads five pre-trained models:

- Deep Learning (Keras-based Neural Network)
- Random Forest Regressor
- AdaBoost Regressor
- Gradient Boosting Regressor
- XGBoost Regressor

It also loads a reference dataset to dynamically generate input fields and set default values. This allows for quick and easy model and integration testing.

Input fields

The web app supports a comprehensive set of features split between product details (e.g., category and pricing) and order details (e.g., region, month, day, and quantities). Each feature is a numeric input field and a dropdown menu is used for model selection.

Technologies Used

- Dash
- Dash Bootstrap Components
- NumPy / Pandas
- Scikit-learn
- TensorFlow / Keras
- XGBoost
- Pickle

Deployment Steps

1. Set up project structure with modular code components.
2. Train and store models in the `artifacts` directory.
3. Build the interactive Dash web app and test locally.
4. Generate `requirements.txt` using `pip freeze > requirements.txt`.
5. Push project repository to GitHub.
6. Create and configure a web service on Render.
7. Deploy the application with environment paths and scaling configurations.

Development Process

The development of this project was conducted through an iterative and collaborative process that kept improving substantially over time. Initially, all code, visualizations, and models existed within a single Jupyter notebook. This method proved challenging to maintain as the project lifespan continued.

Following team discussions and strategic planning, the project's structure was redesigned into a more structured and organized framework. Each machine learning model was moved to its own dedicated notebook, and a central notebook was established for data analysis and visualization interpretation. This restructuring improved clarity, ease of maintenance, and overall presentation quality.

Furthermore, reusable data preparation, preprocessing and modelling functionalities were encapsulated into standalone Python scripts. These are discussed below:

- A python script file `'prepare_data.py'` exists of functions that split columns of the dataset to target the column that is used for prediction as well as splitting train & test data.
- Another python script file `'preprocess_data.py'` uses functions for data cleaning, encoding and feature engineering.
- The last python file `'train_models.py'` includes reusable functions for training various machine learning models, such as Random Forest, AdaBoost and Deep Learning architectures.

This object-oriented programming-inspired modular design allowed notebooks to remain focused and easily interpretable, allowed for consistent logic reuse, and simplified debugging processes. The resulting directory structure presents the project in a professional manner, ensuring scalability and maintainability.

Throughout development, issues such as data transformation errors, encoding issues, and model performance limitations were identified and addressed through rigorous testing. Leading to restructuring of data pipelines and continuous parameter tuning.

After multiple development cycles and collaborative refinements, the project successfully achieved high accuracy and fulfilled the team's initial performance and usability objectives.

Conclusion

This project proved that machine learning can significantly improve inventory management by predicting total item quantities based on key features like pricing, category, and region. Gradient Boosting emerged as the most accurate model. The modular code structure, combined with a user-friendly web app, enabled practical development of predictions. Overall, the project successfully transformed raw data into actionable insights, helping Vertex PC Supply reduce stockouts, control inventory costs and improve decision-making.

References

Chaudhary, V. et al. (2023) 'Exploring the Use of Machine Learning in Inventory Management for Increased Profitability', *New Zealand Herpetology*, 12(1), pp.658-66.

Praveen, K.B. et al. (2020) 'Inventory Management using Machine Learning', *International Journal of Engineering Research & Technology (IJERT)*, 9(06), pp.866-69.

Sudirjo, F. (2023) 'Marketing Strategy in Improving Product Competitiveness in the Global Market', *Journal of Contemporary Administration and Management (ADMAN)*, 1(2), pp.63-69.