PRICE OPTIMIZATION

Abstract

This project aims to optimize pricing strategies for diverse product categories using historical sales data. By identifying patterns, trends, and key factors influencing pricing and demand, we will develop a pricing optimization model. The model will empower our client to make informed decisions, maximizing revenue and profitability while staying competitive in the market. The deliverables include a data analysis report, a functional pricing optimization model, and comprehensive documentation for seamless implementation.

Group 6

Ahmed Abdulrahim

Bimsara Geethachapa Siman Meru Pathiranage

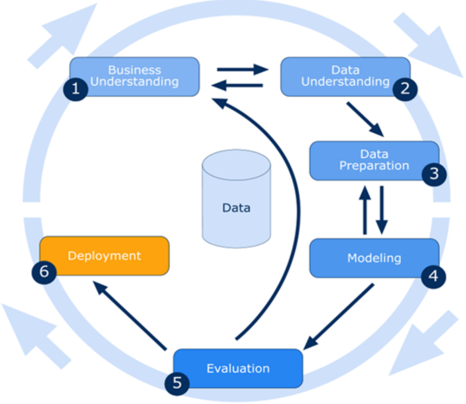
Simranjeet Kaur

Table of Contents

1. Methodology

In this project, we are using the **CRISP-DM** (Cross-Industry Standard Process for Data Mining) framework to guide the data mining process. CRISP-DM (Cross-Industry Standard Process for Data Mining) is a widely used methodology for data mining that provides a structured approach to planning, executing, and evaluating data mining projects. It is a process model that outlines six phases of the data mining process. These phases are:

1. Business Understanding
2. Data Understanding
3. Data Preparation
4. Modeling
5. Evaluation
6. Deployment



We started with the Business Understanding phase, where we identified the problem and defined the project goal. Next, in the Data Understanding phase, we gathered and explored retail and historical sales data . Then, we moved to the Data Preparation phase, where we cleaned, transformed, and prepared the data for modeling. The Modeling phase involved building and testing several machine learning models to optimize prices.

We then evaluated the performance of the different models in the Evaluation phase and selected the best one. At the completion of this project, the following deliverables will be provided:

- **Data analysis report**: A report summarizing the findings, insights, and recommendations based on the analysis of the historical sales data
- **Pricing optimization model**: A functional model that predicts optimal prices for different product categories, taking into account the identified factors and goals.
- **Documentation**: Detailed documentation outlining the methodology, assumptions, limitations, and recommendations for implementing the model in the client's system.

1.1 Business Understanding

During this initial phase, we thoroughly comprehend the problem at hand, which in our case is pricing optimization for different product categories. We define our project goal, ensuring that it aligns with the project scope.

1.2 Data Understanding

In the Data Understanding phase, we collect and explore the "Retail Price Optimization" dataset from Kaggle. This dataset will provide us with historical pricing data across various product categories, and the data contains the demand and corresponding average unit price at a product month-year level. By familiarizing ourselves with the data's structure and characteristics, we gain insights necessary for the subsequent steps.

1.3 Data Preparation

In this phase, we clean, transform, and preprocess the "retail_price" data to make it suitable for modeling. Tasks like handling missing values, negative values, and grouping columns are performed to ensure data quality and integrity.

1.4 Modeling

During the Modeling phase, we construct and evaluate multiple pricing optimization models using the preprocessed data. Various algorithms and techniques will be employed to predict optimal prices for different product categories.

1.5 Evaluation

The Evaluation phase involves comparing the performance of the different pricing optimization models. Metrics like revenue, profitability, and other relevant factors will be used to identify the model that best meets our objectives.

1.6 Deployment

Upon selecting the most effective pricing optimization model, we will proceed to the Deployment phase. Here, we will hope to integrate the model into the client's systems, enabling them to make informed pricing decisions based on the insights gained from the analysis. This integration will empower the client to maximize their revenue and profitability while considering the various factors influencing their product categories.

2. Business Understanding

Businesses that operate in competitive markets must overcome the crucial challenge of price optimization. Setting the right prices for products or services can significantly impact revenue, market share, and customer satisfaction. Inaccurate pricing strategies can lead to lost sales opportunities, reduced profitability, and potential customer churn. To address these issues, we aim to develop a sophisticated machine learning model for price optimization.

The goal of this project is to create a data-driven pricing strategy that maximizes revenue and profitability while considering market dynamics, customer preferences, and cost structures. By leveraging historical sales data, market trends, customer behavior, and competitor pricing information, we can identify optimal price points for each product or service.

We identified that the project's objectives are multi-faceted and divided into In-scope and out-of-scope goals.

In Scope:

- Analyzing historical sales data for different product categories
- Identifying factors that impact pricing and demand
- Developing a pricing optimization model
- Conducting statistical analysis and modeling techniques.
- Evaluating the effectiveness of the pricing optimization model

Out of Scope:

- External market factors such as competitor pricing and economic conditions
- Implementation of the pricing optimization model into a system
- Testing the model with real-time data

By implementing the proposed pricing optimization model, our client will be able to make informed pricing decisions that optimize revenue and profitability while considering the various factors influencing their product categories. This project will empower the client to stay competitive and maximize their market potential in an ever-evolving business environment. To achieve these objectives, we will leverage various machine learning techniques, including regression models, ensemble methods, and neural networks.

3. Data Understanding

Data understanding is a critical step in our project. It involves obtaining a comprehensive understanding of the data that will be used, which includes sales data, competitor pricing data, market demand, and trends. During this phase, we diligently collect the "Retail Price Optimization" dataset from Kaggle. Then we generated synthetic data using data augmentation, ensuring we obtain a substantial and representative sample of historical pricing data, which is vital to ensuring the accuracy and completeness of our analysis.

Once we have the data, we perform data cleaning and preprocessing. This step ensures the data is accurate, consistent, and in a usable format. Removing duplicates, handling missing values, and correcting errors are part of this process. Additionally, we may need to transform or normalize the data to ensure compatibility with other data sources.

Data understanding also involves exploring and analyzing the data. By doing so, we gain valuable insights into its patterns and trends. This exploration is crucial to identifying any relevant relationships between variables and understanding how they impact our predictions.

The insights obtained from this data-understanding process serve as the foundation for developing our predictive model. By thoroughly understanding the data and its nuances, we can build a robust model that maximizes the accuracy and usefulness of our predictions, leading to better decision-making and successful outcomes for the project.

3.1 Major factors considered to select dataset

Historical Sales Data: The dataset should include historical sales data for the products or services under consideration. This data will provide insights into past pricing strategies, customer purchasing behavior, and revenue trends, serving as a foundation for building the price optimization model.

Product Attributes and Features: Information about the products or services, such as specifications, features, and categories, is essential. Different products may have varying price sensitivities and demand patterns, and including these attributes will enable the model to identify pricing patterns specific to each product category.

Competitor Pricing Data: Incorporating data on competitor pricing is crucial for understanding the market dynamics and competitive landscape. By analyzing how competitors set their prices, the model can recommend competitive pricing strategies to maintain market share.

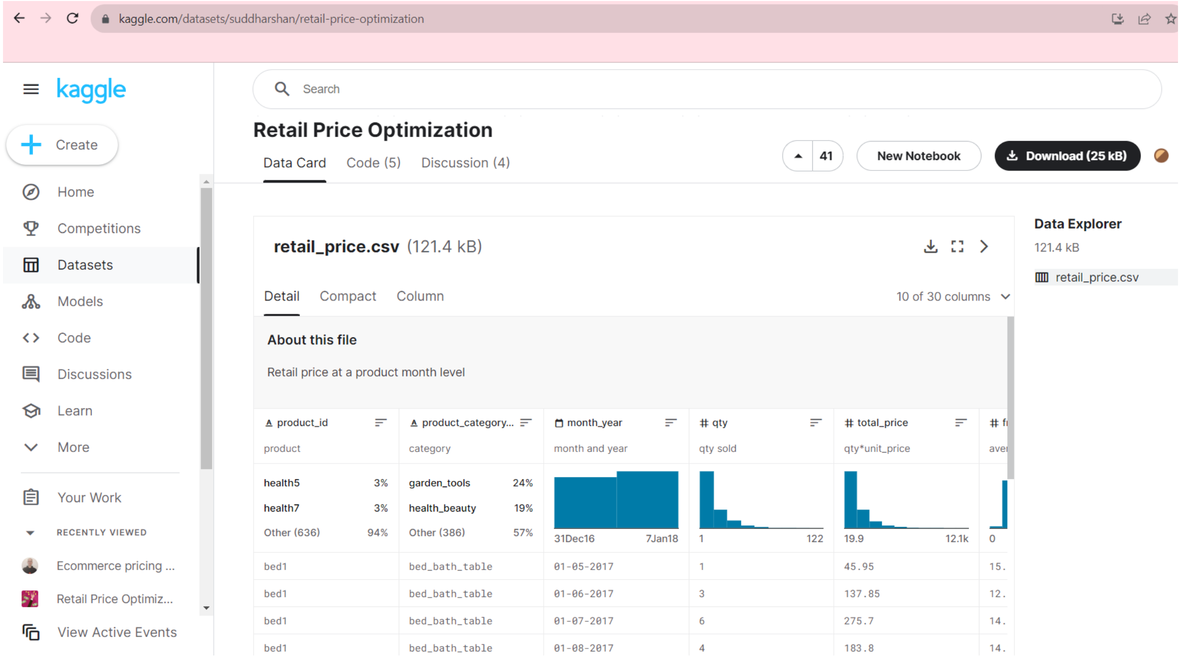
Market Demand and Trends: Data related to market demand and trends, such as seasonal fluctuations, economic indicators, and industry-specific factors, will help the model adapt pricing based on changing market conditions and customer preferences.

Customer Demographics and Behavior: Understanding customer demographics, preferences, and purchasing behavior is vital for personalized pricing strategies. This data will assist in identifying customer segments with distinct price sensitivities and tailoring prices accordingly.

Cost Structure: Information about production costs, operational expenses, and other relevant cost factors should be included in the dataset. Accurate cost data will enable the model to recommend prices that maintain profit margins while remaining competitive.

3.2 Overview of price optimization dataset from Kaggle

Our data gathering technique involves utilizing the Kaggle platform as our primary source of data collection. For those who are interested in data science and machine learning, Kaggle is a reputable online community where members share and make various datasets accessible to the general public. To accomplish the project objective, we need to gather historical sales data for different product categories, including information such as product attributes, pricing, quantities sold, and timestamps. In this project, we use the following dataset from Kaggle:



Retail Price Optimization

retail_price.csv (121.4 kB)

About this file

Retail price at a product month level

product_id	product_category...	month_year	qty	total_price	fi
product	category	month and year	qty sold	qty*unit_price	aver
health5 3% health7 3% Other (636) 94%	garden_tools 24% health_beauty 19% Other (386) 57%				
bed1	bed_bath_table	01-05-2017	1	45.95	15.
bed1	bed_bath_table	01-06-2017	3	137.85	12.
bed1	bed_bath_table	01-07-2017	6	275.7	14.
bed1	bed_bath_table	01-08-2017	4	183.8	14.

Here is a brief description of each column present in the Retail_price_ptimization dataset:

- ➔ customers (integer): monthly demand for a given subcategory of goods
- ➔ freight_price (float): freight price of the company goods
- ➔ fp1, fp2, fp3 (float): freight price of competitors 1, 2, and 3 goods, respectively
- ➔ product_category_name (categorical) broad group category name
- ➔ product_id (categorical): detailed group subcategory name
- ➔ product_description_length (integer) number of words in the subcategory description
- ➔ product_score (float): user rating for subcategories of the goods
- ➔ ps1, ps2, ps3 (float): user rating for subcategories of competitors 1,2,3 respectively
- ➔ product_photos_qty (integer): number of photos for each subcategory (product_id)
- ➔ product_weight_g (integer): unit weight in grams
- ➔ total_price (float): monthly revenue, which can be calculated using formula: total_price = unit_price * qty
- ➔ month_year (string): data in the format (dd-mm-yyyy) within the range between 01-01-2017 and 01-08-2018. Only months and years are important here.
- ➔ year (integer): year which was taken from the 'month_year'
- ➔ month (integer): month which was taken from the 'month_year'
- ➔ qty (integer): monthly sales per subcategory
- ➔ unit_price (float): monthly unit price of subcategory goods of company goods
- ➔ comp_1, comp_2, comp_3 (float): unit price of within the subcategory of competitors 1,2,3 goods, respectively
- ➔ lag_price (float): unit price on the previous month
- ➔ weekend (integer): number of weekends per month
- ➔ weekday (integer): number of weekdays per month
- ➔ holiday (integer): number of holidays per month
- ➔ s (float): yet unknown parameter

4. Data Preparation

Data preparation is a critical step in any data analysis project. In the case of retail price data, it is essential to ensure that the data is clean, accurate, and formatted correctly before tuning a machine learning model. This process involves several steps, such as identifying missing or erroneous data, removing duplicates, and transforming data types and values as necessary.

4.1 Data Cleaning and Preprocessing

Once the data is consolidated, we shift our focus to the essential tasks of data cleaning and preprocessing. Our meticulous approach during this stage involves sifting through the dataset to identify any inconsistencies, missing values, negative values, outliers, and sorting values. We diligently address these issues, as data accuracy is paramount to ensuring the reliability of our subsequent analysis.

4.1.1 Handling Zero or Negative Values

We first check for any zero or negative values in the 'qty', 'total_price', and 'unit_price' columns using the lambda function and 'apply' method. If any of these columns contain zero or negative values, it indicates potential data issues that need to be resolved. Specifically, we identify if any of the values in each column are less than or equal to zero. If such values are found, we replace negative 'qty' values with 1, assuming that negative quantities might be erroneous or represent a different meaning (e.g., returns), and they are set to a minimum value of 1 for consistency.

4.1.2 Calculating the Expected Total Price

Next, We calculate the 'expected_total_price' column based on the 'unit_price' and 'qty' columns. We multiply the 'unit_price' by the 'qty' to obtain the expected total price for each transaction.

4.1.3 Checking Consistency between 'total_price' and 'expected_total_price'

We then check if the newly calculated 'expected_total_price' column aligns with the original 'total_price' column. This check is essential to verify whether the data is logically consistent in terms of the relationship between 'unit_price', 'qty', and 'total_price'. If the 'expected_total_price' matches the 'total_price' for all records, it indicates that the data is consistent in terms of pricing and quantities. If there are discrepancies, it could suggest potential data issues or inaccuracies in either the 'unit_price', 'qty', or 'total_price' columns.

4.1.4 Extracting Year and Month from 'month_year'

We then proceed to extract the 'year' and 'month' from the 'month_year' column. The 'month_year' column appears to contain date information in the format 'dd-mm-yyyy'. Using the 'pd.to_datetime' method and specifying the format '%d-%m-%Y', we extract the year and month components separately into new columns 'year' and 'month'. This step is essential for further analysis that might require grouping or aggregating data based on time periods.

At this stage, the data is now thoroughly prepared, with missing values addressed, zero or negative values handled, and new columns created to facilitate time-based analysis. The dataset is now ready for the subsequent steps, such as exploratory data analysis, feature engineering, and building predictive models for price optimization.

4.1.5 Log Transformations

We perform log transformations on the 'total_price', 'unit_price', and 'qty' columns and store the results in new columns 'total_price_log', 'unit_price_log', and 'qty_log', respectively. The reason for applying a logarithmic transformation is to stabilize the variance and make the relationship between variables more linear. Price optimization models typically work on relative changes rather than absolute values. If we apply a logarithmic transformation to our price or target variable, it will have the effect of compressing the range of values. As a result, the model will learn the relative changes in prices, which is often more appropriate for price optimization tasks

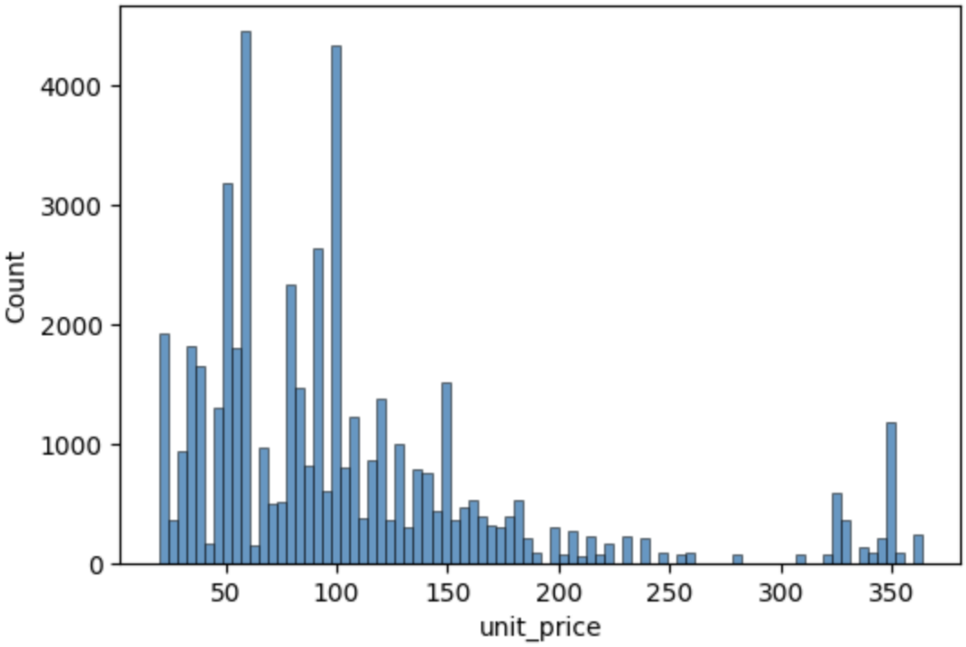
4.2 Exploratory Data Analysis (EDA)

With a clean and refined dataset, we embark on the Exploratory Data Analysis (EDA) journey. EDA is a fundamental aspect of the Data Understanding phase, enabling us to uncover hidden patterns, trends, and relationships within the data.

Through univariate analysis, we examine the distribution and characteristics of individual variables, gaining insights into their central tendencies and variations. Bivariate and multivariate analysis further deepen our understanding by exploring connections and correlations between different variables. By scrutinizing the relationships between demand, average unit price, and product categories, we aim to unearth factors that influence pricing and demand patterns.

Visualization techniques such as scatter plots, histograms, box plots, and time series plots enhance our exploration. These visual representations provide intuitive insights, enabling us to identify any apparent trends, anomalies, or seasonal effects.

★ Target Variable Distribution



★ Unit price vs quantity

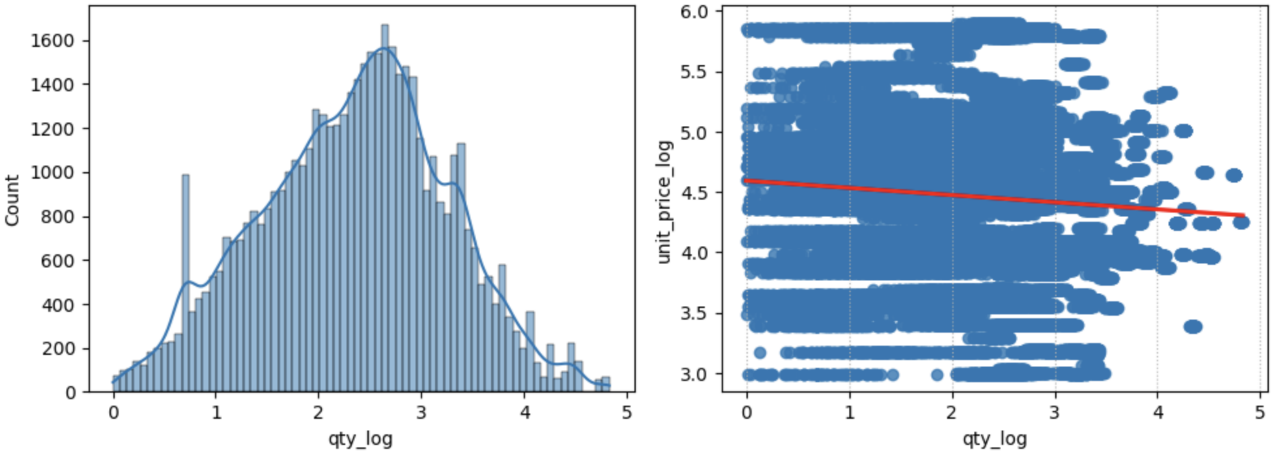


Fig.1 - Unit price vs quantity

★ **Unit price distribution vs competitors**

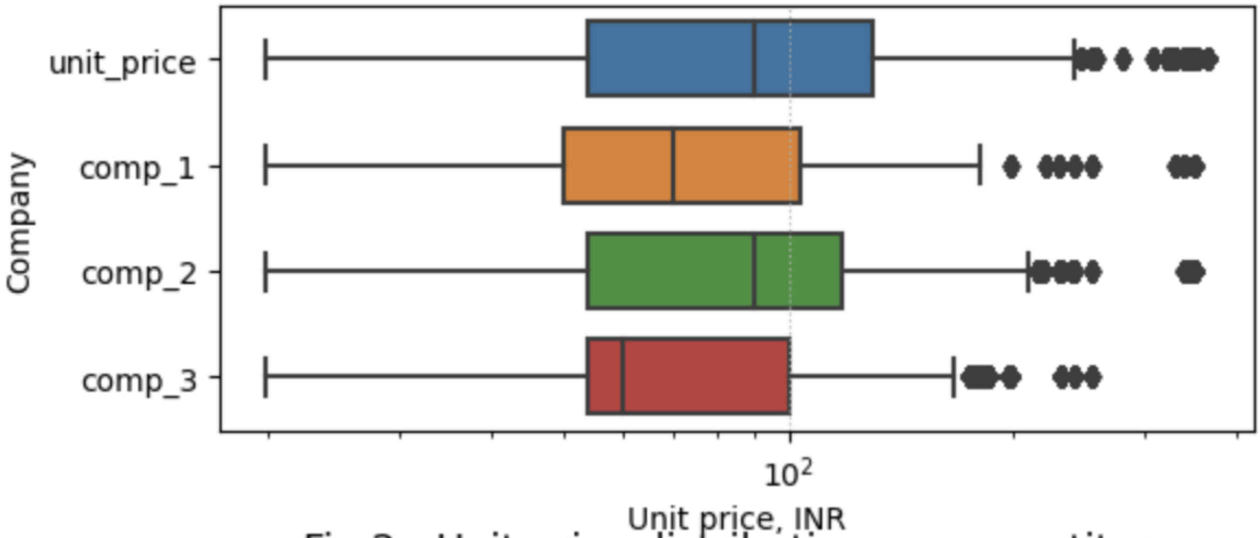


Fig.2 - Unit price distribution vs competitors

★ **Freight price comparison vs competitors**

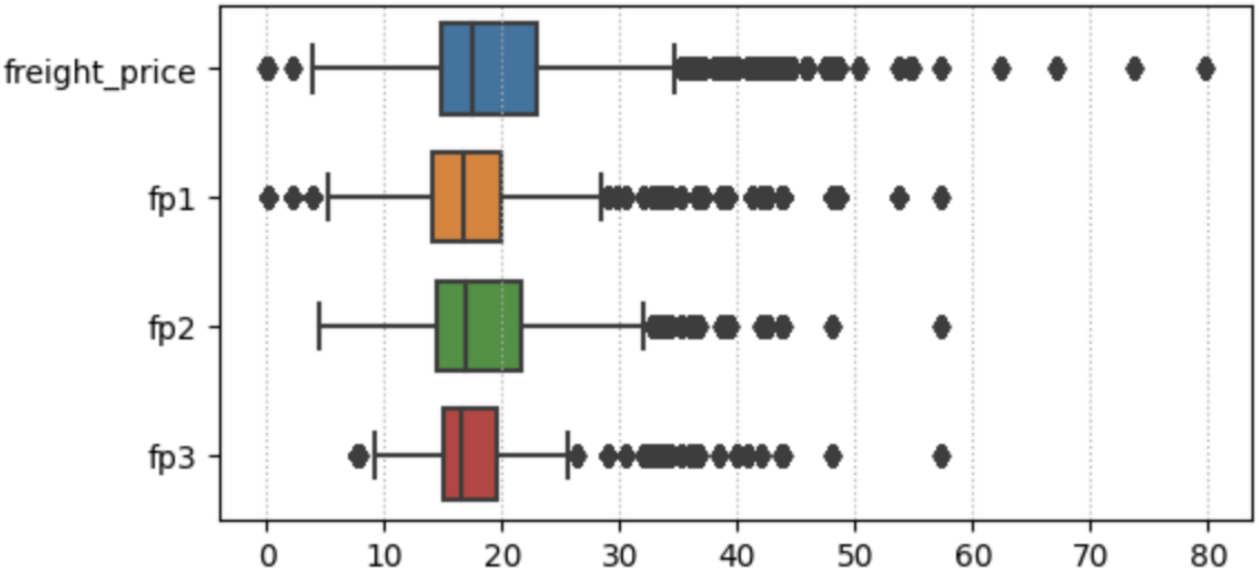


Fig.3- Freight price comparison vs competitors

★ Freight price by weight

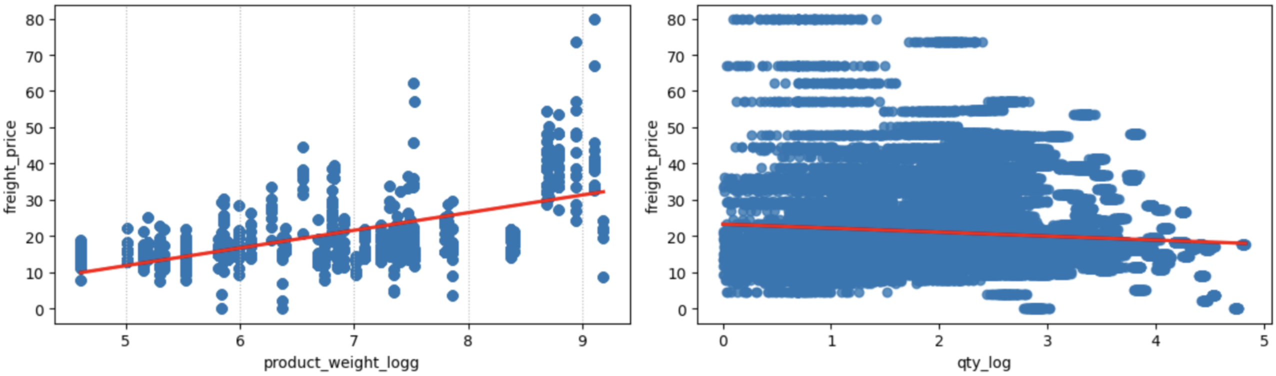
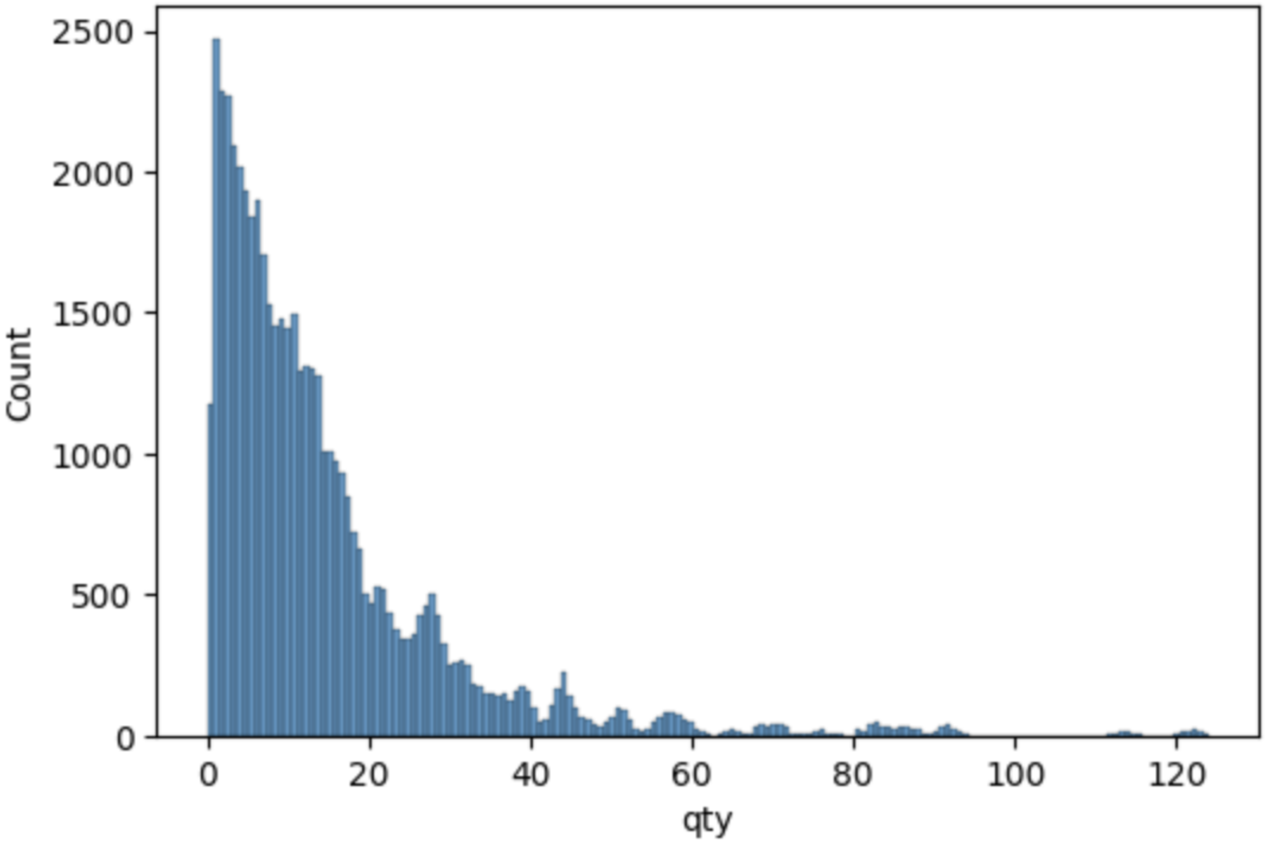
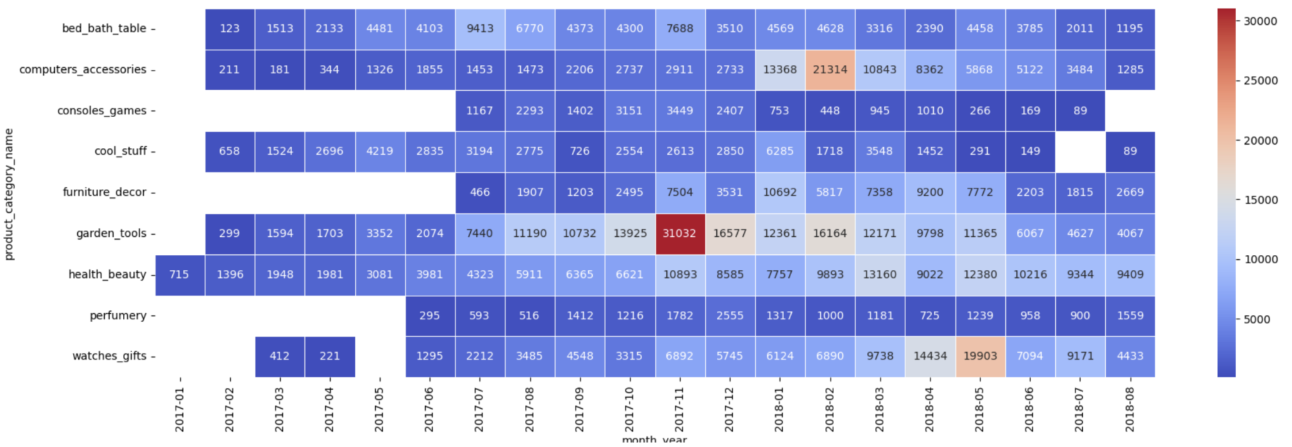


Fig.4 - Freight price by weight

★ Qty Distribution



★ Sum of monthly sales by category



★ Unit weight vs price

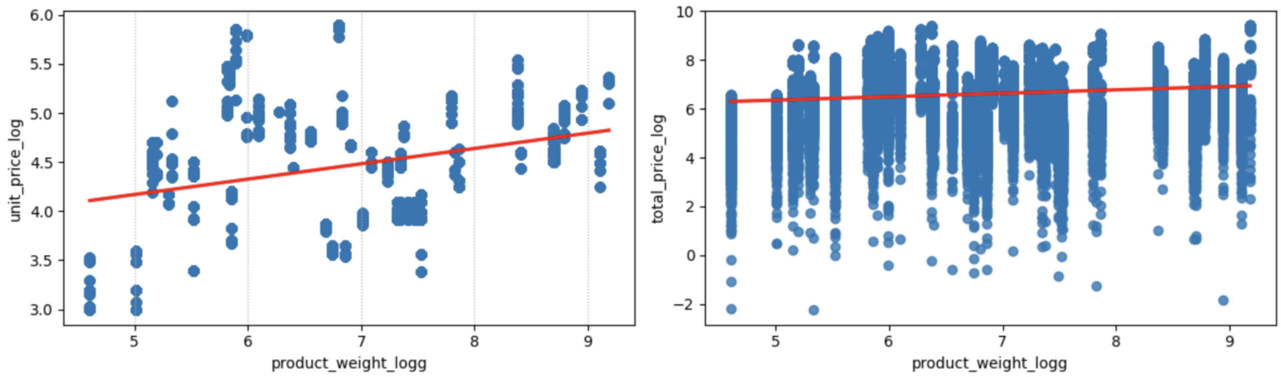


Fig.6 - Unit weight vs price

★ Product score distribution vs competitors

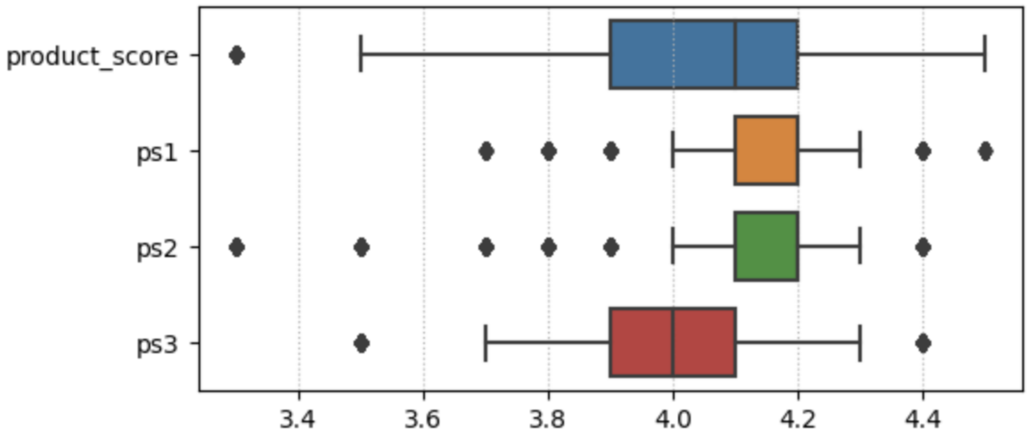
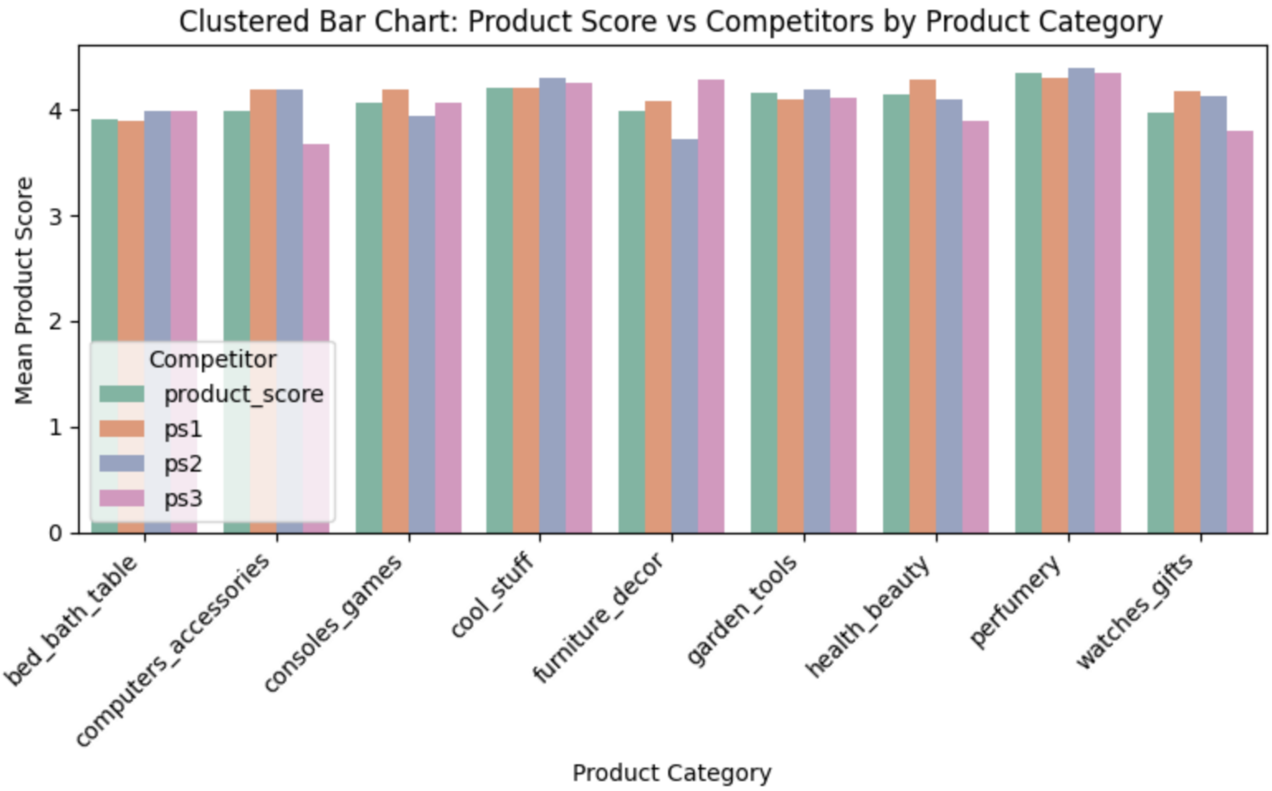


Fig.7 - Product score distribution vs competitors

★ Product Score vs Competitors by Product Category



5. Modeling and Evaluation

Once we have analyzed the data and identified key patterns and trends, we can then use this information to build a predictive model using Python and related tools. This model will utilize historical sales data and various factors such as product attributes, promotional activities, and seasonal trends to predict optimal prices for different products. By employing machine learning algorithms such as regression and neural networks, we can train the model to recognize patterns and make predictions based on the input data. The goal of building this predictive model is to help businesses optimize their pricing strategies and maximize revenue while considering market demand, customer behavior, and competitive landscape.

By predicting the optimal prices for products, businesses can dynamically adjust their pricing to meet changing market conditions, maximize sales during peak demand, and remain competitive during low-demand periods. This price optimization model can provide valuable insights to businesses, enabling them to make data-driven pricing decisions that lead to improved profitability and customer satisfaction.

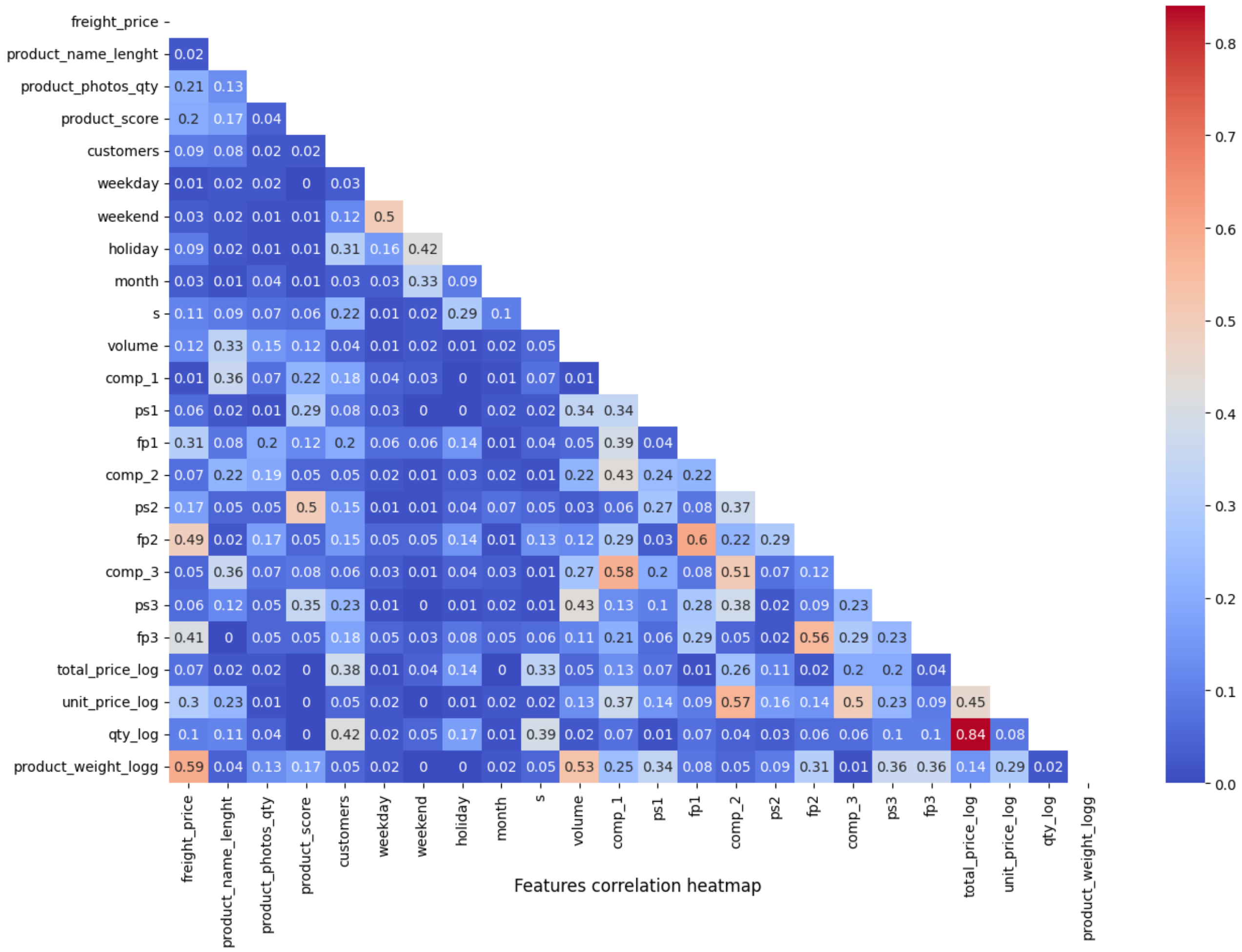
Overall, the process of analyzing sales data and building a price optimization model involves a combination of data analysis and machine learning techniques. By leveraging these tools, businesses can gain insights into pricing dynamics and develop models that help them achieve their revenue and profitability objectives.

5.1 Feature Engineering

Feature engineering is a crucial step in preparing data for analysis and building predictive models. In this process, we aim to transform and select relevant features that can improve the performance of our models and provide valuable insights. Here are the key steps involved:

1. **Data Preparation**: First, we make a copy of the original dataset and remove certain columns that are not needed for the analysis, such as 'product_id', 'month_year', 'year', 'qty', and others, as they may not contribute significantly to predicting the target variable, 'unit_price_log'.

2. **Correlation Analysis**: We then visualize the correlation between the remaining features using a heatmap. This helps us understand the relationships between variables and identify any multicollinearity that might impact the model's performance.

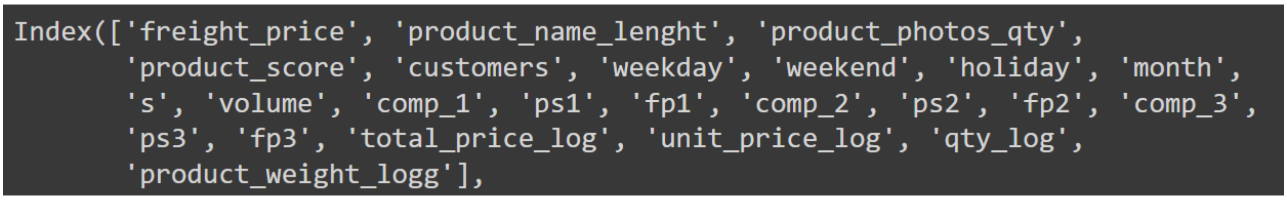


3. **Numeric and Categorical Features**: We categorize the remaining features into numeric and categorical types. Numeric features contain numerical data, while categorical features consist of non-numeric values like 'object' or 'category'.

4. **Feature Selection**: In feature selection, we choose the most relevant features that have the highest impact on predicting the target variable. This step is essential to avoid overfitting and improve model interpretability.

5. **Feature Transformation**: We applied encoding techniques to convert categorical variables into numerical representations that algorithms can work with.

Selected columns for modeling



```
Index(['freight_price', 'product_name_lenght', 'product_photos_qty',
       'product_score', 'customers', 'weekday', 'weekend', 'holiday', 'month',
       's', 'volume', 'comp_1', 'ps1', 'fp1', 'comp_2', 'ps2', 'fp2', 'comp_3',
       'ps3', 'fp3', 'total_price_log', 'unit_price_log', 'qty_log',
       'product_weight_logg'],
```

By conducting thoughtful feature engineering, we can enhance the quality of our data, uncover hidden patterns, and build predictive models that are more accurate and effective in solving real-world problems.

5.2 Benchmark Model using Linear Regression

In the modeling phase of our price optimization project, we aimed to establish a benchmark predictive model to serve as a reference point for evaluating the performance of more complex models. We chose the linear regression model as our benchmark due to its simplicity, interpretability, and ease of implementation. Linear regression allows us to understand the relationships between the target variable 'unit_price_log' and the various features in a straightforward manner.

5.2.1 Interpretation of Results

The coefficients obtained from the linear regression model offer valuable insights into the impact of each feature on the predicted 'unit_price_log'. A positive coefficient suggests a positive relationship, indicating that an increase in the corresponding feature will lead to higher predicted prices. Conversely, a negative coefficient indicates a negative relationship, where an increase in the feature leads to lower predicted prices.

The intercept term, **0.6856**, represents the baseline predicted value for 'unit_price_log' when all features have zero values.

Coefficient Interpretation:

- **'comp_2' (Competitor 2 Store):**

Coefficient: ***0.1447***
Interpretation: The unit price of Competitor 2's goods within the subcategory has a substantial positive impact on the predicted 'unit_price_log'. For every one-unit increase in Competitor 2's unit price, the predicted unit price tends to increase by 0.1447. This suggests that when facing competition from Competitor 2, the company might adjust its prices slightly upward to maintain competitiveness or differentiate based on perceived value.

- **'qty_log' (Log-Transformed Monthly Sales):**

Coefficient: ***-0.9808***
Interpretation: The log-transformed monthly sales per subcategory has the most substantial negative impact on the predicted 'unit_price_log'. For every one-unit increase in the log-transformed sales, the predicted unit price tends to decrease by 0.9808. This suggests that as the sales quantity increases, the company might offer volume discounts or lower prices to incentivize larger purchases and maintain a competitive edge.

- **'total_price_log' (Log-Transformed Monthly Revenue):**

Coefficient: ***0.7879***
Interpretation: The log-transformed monthly revenue (total_price) has the most significant positive impact on the predicted 'unit_price_log'. For every one-unit increase in the log-transformed revenue, the predicted unit price tends to increase by 0.7879. This implies that as the monthly revenue generated from a subcategory increases, the company might adjust its unit prices upward, potentially indicating a pricing strategy to optimize profitability.

- **'comp_3' (Competitor 3 Unit Price):**

Coefficient: ***0.0658***
Interpretation: The unit price of Competitor 3's goods within the subcategory has a moderate positive impact on the predicted 'unit_price_log'. For every one-unit increase in Competitor 3's unit price, the predicted unit price tends to increase by 0.0658. This indicates that competition

from Competitor 3 might influence the company to adjust its prices upward to remain competitive in the market.

➢ **'product_photos_qty' (Number of Photos per Subcategory):**

Coefficient: ***0.0268***
Interpretation: The number of photos for each subcategory has a moderate positive impact on the predicted 'unit_price_log'. For every one-unit increase in the number of photos, the predicted unit price tends to increase by 0.0268. This suggests that providing more photos for a subcategory might positively influence customer perception, leading to slightly higher unit prices as customers may perceive the products as having higher quality or more detailed information.

➢ **ps3' (User Rating for Competitor 3's Subcategories):**

Coefficient: ***-0.0961***
Interpretation: The user rating for Competitor 3's subcategories has a moderate negative impact on the predicted 'unit_price_log'. For every one-unit increase in the user rating for Competitor 3's subcategories, the predicted unit price tends to decrease by 0.0961. This implies that higher user ratings for Competitor 3's products might influence the company to price its goods slightly lower, as higher ratings could indicate a higher perceived value compared to the competition.

5.2.2 Model Evaluation

The linear regression model demonstrates promising performance on both the training and test datasets, as indicated by the Mean Squared Error (MSE) and R-squared (R2) metrics.

➢ Training Set Evaluation:

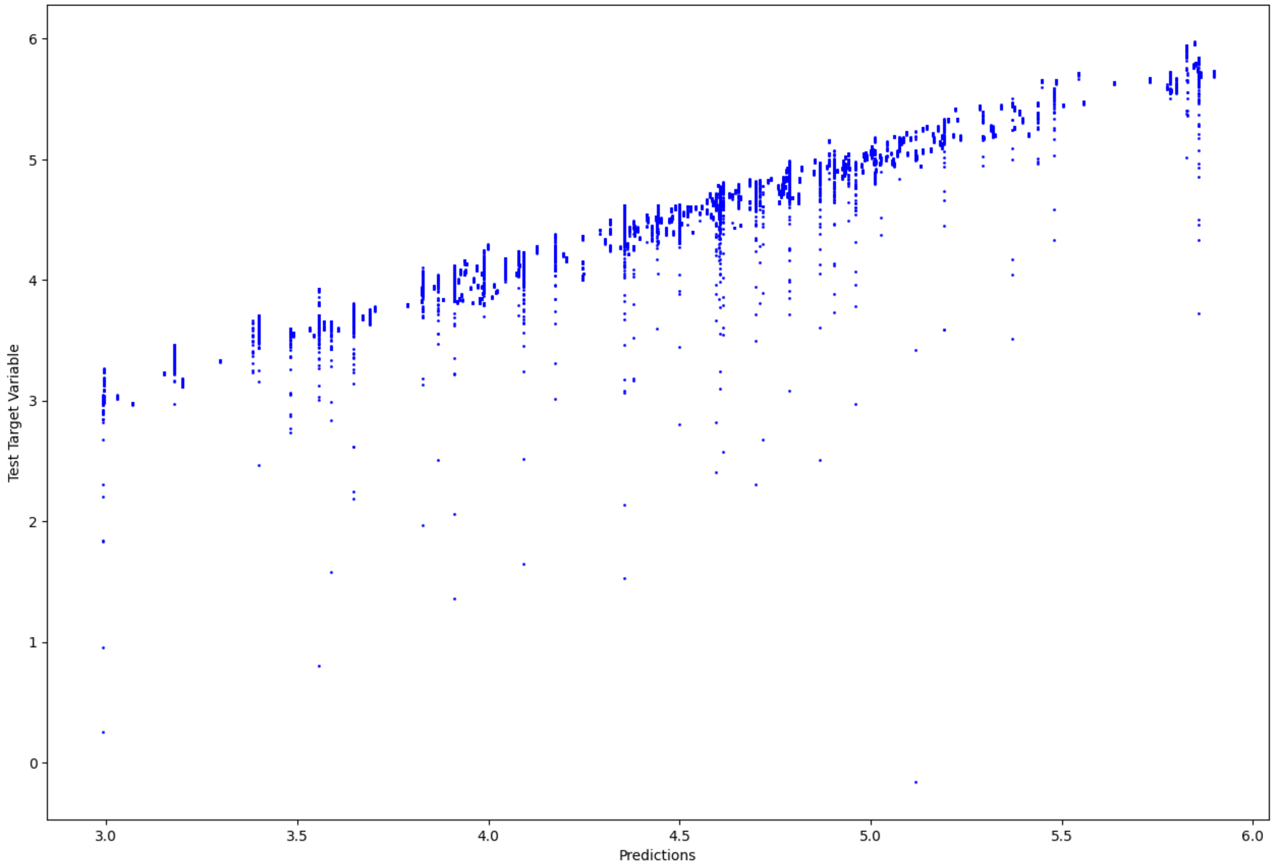
Mean Squared Error (MSE): 0.0288
R-squared (R2): 0.9320

➢ Test Set Evaluation:

Mean Squared Error (MSE): 0.0305
R-squared (R2): 0.9264

The R-squared values, which gauge how much of the variance the model explains, are similar for both the training and test sets, showing that the model generalizes well to new data. The MSE values are relatively low, suggesting that the model's predictions are close to the actual 'unit_price_log' values.

Prediction and Target variable



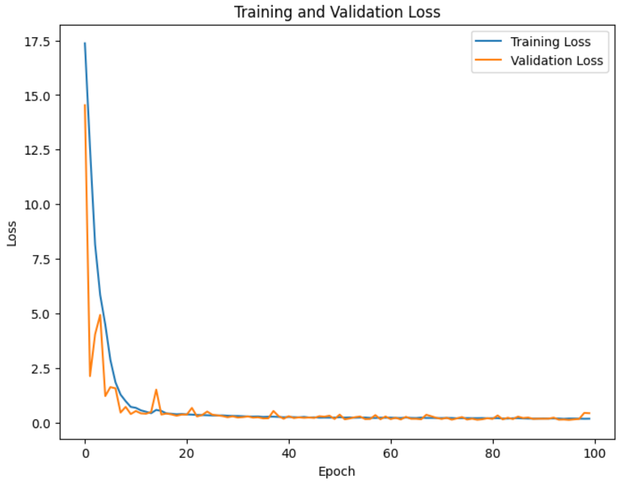
5.3 First model: Neural Network

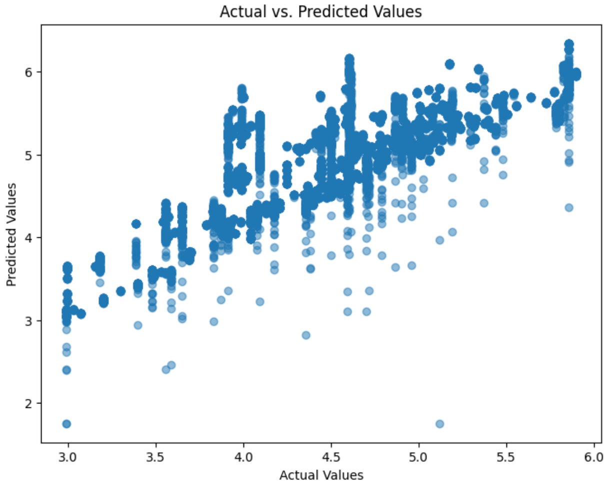
For our next step in the price optimization project, we decided to implement a deep neural network (DNN) model. Neural networks are well-suited for complex, non-linear relationships in the data, and can capture intricate patterns that may be challenging for traditional linear models. The DNN model architecture we created consists of three hidden layers with 64, 32, and 16 neurons, respectively, using the ReLU (Rectified Linear Unit) activation function. The output layer contains one neuron for the final prediction. The model was compiled with the Mean Absolute Error (MAE) loss and optimized using the 'adam' optimizer.

5.3.1 Model Training and Evaluation Results:

After training the DNN model on the training dataset, we evaluated its performance on the test dataset. The model achieved the following results:

- Test Loss (MAE): **0.4423**
- R-squared: **-30988.2863**





5.3.2 Interpretation of Results

- **Test Loss (MAE):**

The test loss (Mean Absolute Error) measures the average absolute difference between the actual 'unit_price_log' values and the predicted values produced by the neural network model. In this case, the test loss of **0.4423** indicates that, on average, the model's predictions have a deviation of approximately **0.4423** from the true 'unit_price_log' values. Lower values of MAE indicate better model performance, and this result suggests that the DNN model has made reasonably accurate predictions.

- **R-squared:**

The R-squared value quantifies the proportion of variance in the target variable ('unit_price_log') that the model can explain. A negative R-squared value indicates that the model's predictions do not perform better than a simple horizontal line (constant prediction) through the mean of the target variable. This result, with an R-squared of **-30988.2863**, indicates that the model's predictions have extremely poor performance and do not capture the variance in the 'unit_price_log' values.

5.3.3 Interpretation of Poor Performance

The DNN model's poor performance, as indicated by the negative R-squared value, raises concerns about the model's ability to capture meaningful patterns in the data. Several possible reasons could lead to this suboptimal performance:

1. **Insufficient Data**: Neural networks typically require a large amount of data for effective training. If the dataset is small, the model may struggle to generalize well to unseen data, resulting in poor predictions.
2. **Overfitting**: The model may have overfitted the training data, learning noise and idiosyncrasies rather than general patterns. Overfitting can lead to poor generalization to new data, resulting in a negative impact on model performance.
3. **Model Complexity**: The chosen architecture may be too complex for the given dataset, leading to difficulties in learning meaningful patterns and causing the model to perform poorly.

5.4 Implementing Gradient Boosting Regressor

For our second model, we decided to utilize the Gradient Boosting Regressor, a powerful ensemble learning technique that combines the predictions of multiple weak learners (typically decision trees) to create a strong predictive model. Gradient Boosting Regressor is known for its ability to handle complex relationships and produce accurate predictions. We trained the model on the training dataset and evaluated its performance on both the test and training datasets.

5.4.1 Model Evaluation Results

Test Evaluation:

- Mean Squared Error (MSE): **0.008824234901947614**
- R-squared (R2): 0**.9787133350263721**
- Mean Absolute Percentage Error (MAPE): **6.877781132032563**

Training Evaluation:

- Mean Squared Error (MSE): **0.008551962500882014**
- R-squared (R2): **0.9797937759091521**
- Mean Absolute Percentage Error (MAPE): **6.837487451357786**

5.4.1 Interpretation of Results

- **Mean Squared Error (MSE):**

In both test and training evaluations, the MSE values are relatively low, indicating that the model's predictions are close to the true 'unit_price_log' values. Lower MSE values suggest more accurate predictions.

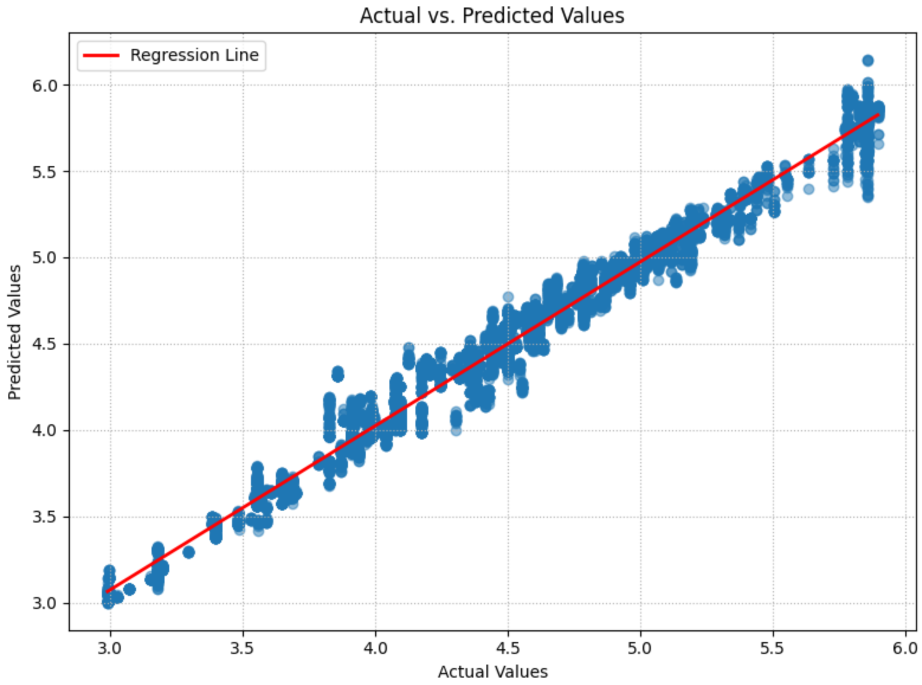
- **R-squared (R2):**

In both test and training evaluations, the R-squared values are high (around **0.98**), indicating that the model captures a large portion of the variance in the 'unit_price_log' values. Higher R-squared values suggest a better fit of the model to the data.

- **Mean Absolute Percentage Error (MAPE):**

The MAPE values in both test and training evaluations are relatively low (around **6.88%**), indicating that the model's predictions have a small percentage error on average. Lower MAPE values suggest more accurate predictions.

- **Actual vs. Predicted Values**



➢ Feature Importance for Gradient Boosting Regressor

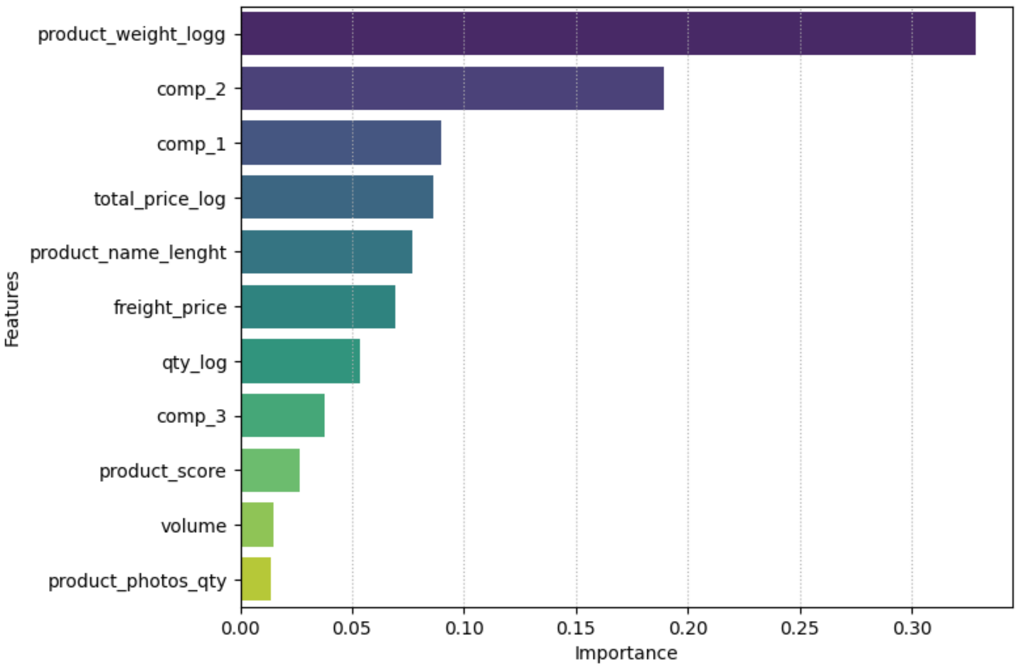


Fig.10 - Feature Importance for Gradient Boosting Regressor

6. Findings

Our price optimization project aimed to develop an effective model to predict the 'unit_price_log' and optimize prices for enhanced profitability. After conducting a comprehensive analysis, we have gathered essential insights to inform our pricing strategies. Below are the key findings from our study:

1. Competitor Price Analysis:

The Interquartile Range (IQR) for competitor 1 and competitor 3 prices is slightly smaller than that of the company's unit_price and freight_price. Additionally, both competitor 1 and competitor 3 prices show a slight left skewness. These observations suggest that competitor prices may play a significant role in shaping our pricing decisions, and we must carefully analyze our pricing strategies relative to competitors to remain competitive in the market.

2. Monthly Sales by Category:

Our analysis of the heat map of monthly sales by category revealed interesting patterns. In November, the highest sales were observed in the 'Garden_tools' category, while in February, 'Computer_accessories' recorded the highest sales. In May, the 'Watches_gifts' category dominated in terms of monthly sales. Understanding the seasonal variations in product categories can help us develop targeted pricing strategies to capitalize on high-demand periods.

3. Product Score Analysis:

A box plot comparing our product's score distribution with that of competitors (1 and 2) showed that our product scores range from 3.5 to 4.5. Meanwhile, both competitors 1 and 2 exhibit a smaller Interquartile Range (IQR) ranging from 4.0 to 4.5. This finding highlights an area for improvement, as enhancing our product scores might positively influence pricing strategies and customer perception.

4. Gradient Boosting Regressor Performance:

Our implementation of the Gradient Boosting Regressor model has yielded highly promising results. The model achieved low Mean Squared Error (MSE) values of 0.0088 for both test and training datasets. Additionally, the R-squared (R2) values were close to 0.98 for both evaluations, indicating that the model effectively captures a large portion of the variance in the 'unit_price_log' values. The Mean Absolute Percentage Error (MAPE) values were relatively low, around 6.88%, suggesting that the model's predictions have a small average percentage error.

5. Feature Importance:

The feature importance analysis of the Gradient Boosting Regressor highlighted several crucial features. 'Product_weight' was the most influential feature, followed by 'competitor 1 price' and 'total_price'. Understanding the significance of these features allows us to focus on their respective impacts and make informed pricing decisions.

7. Feature Enhancements

In the context of price optimization, developing an effective predictive model is crucial for businesses to maximize profits, enhance customer satisfaction, and remain competitive in the market. By leveraging machine learning techniques, we can build price optimization models that analyze various factors affecting pricing decisions, such as product demand, competitor prices, customer behavior, and market trends. Following the CRISP-DM process, we can enhance our price optimization model with the following suggestions:

1. **Increase Dataset Size**: To improve the accuracy of the price optimization model, it is essential to gather a more extensive and diverse dataset. Collecting additional data on historical pricing, sales volumes, customer preferences, and external market indicators can enable the model to capture more intricate patterns and make better pricing predictions.

2. **Feature Engineering**: Identifying and creating relevant features is critical for effective price optimization. In addition to the standard features like product attributes and time-related variables, we can engineer new features such as pricing history, customer segmentation, and promotional periods. These engineered features can provide deeper insights into price-demand relationships and aid in making more informed pricing decisions.

3. **Incorporating External Data Sources**: Integrating external data sources can enrich the price optimization model further. Data from economic indicators, social media sentiment analysis, and competitor pricing trends can offer valuable contextual information to refine pricing strategies and respond to market dynamics more effectively.

4. **Dynamic Model Deployment**: Deploying the price optimization model into production requires careful planning and consideration. A robust deployment pipeline is necessary to ensure that the model can handle real-time pricing scenarios, adapt to changing market conditions, and scale to meet the demands of a dynamic pricing environment.

By implementing these techniques, businesses can fine-tune their pricing strategies, anticipate customer behavior, and optimize prices in real-time, leading to increased revenue, improved customer loyalty, and a more competitive market position. Price optimization models provide valuable insights that empower businesses to strike the right balance between profitability and customer value, ultimately driving long-term success in their respective industries.

7. References

https://www.kaggle.com/datasets/suddharshan/retail-price-optimization

https://en.wikipedia.org/wiki/Price_optimization

https://www.qualtrics.com/experience-management/product/product-price-optimization/

https://www.projectpro.io/article/price-optimization-machine-learning/838