Project

"Determining Optimal Pricing for Retail Products"

Submitted By,

Mavani Krishnkumar Manjibhai

INDEX

1. Project Introduction

1.1. Domain

The domain of this project is **Retail and E-Commerce Analytics**, with a specific focus on price optimization using machine learning techniques. This domain involves analyzing historical sales data, competitor pricing, and product attributes to determine optimal price points. By leveraging predictive modeling and data-driven insights, businesses can enhance profitability, maintain competitive positioning, and refine their pricing strategies to align with market dynamics.

This project encompasses various aspects of **data science and machine learning** to develop an intelligent pricing model. It integrates **time-series forecasting, feature engineering, and model interpretability** to provide actionable recommendations. By employing regression models and explainability techniques, the solution enables retailers to make informed pricing decisions, ensuring revenue maximization while adapting to customer demand patterns and seasonal trends.

1.2. Definition

The project aims to develop a machine learning-based price optimization model that analyzes historical sales data, competitor pricing, and product attributes to determine optimal pricing strategies for retail products. By leveraging predictive analytics and regression models, the solution will identify price points that maximize profitability while maintaining market competitiveness.

1.3. Problem

The problem is a **supervised regression** problem aimed at optimizing retail product pricing. The goal is to predict the optimal price for a product based on historical sales data, competitor pricing, and other relevant factors. This problem falls under supervised learning, as the model is trained on labeled data containing past pricing and sales performance, allowing it to learn patterns and relationships. By leveraging this trained model, the system can generate accurate price recommendations for new or existing products to maximize profitability while maintaining market competitiveness.

1.4. Benefits

- **Profit Maximization**: The model will identify optimal price points that maximize revenue and profit margins while maintaining competitive pricing.

- **Market Competitiveness**: By analyzing competitor pricing trends, the company can strategically position its products to attract customers without compromising profitability.
- **Data-Driven Decision Making**: Leveraging historical sales data, customer behavior, and seasonal trends, the model will provide actionable insights for informed pricing strategies.
- **Demand Forecasting**: Predicting optimal pricing based on past sales performance enables better inventory and sales planning, reducing overstock and lost sales opportunities.
- **Operational Efficiency**: Automating the pricing optimization process minimizes manual effort, ensuring consistent and scalable pricing strategies across product categories.

1.5. Issues

1. **Data Quality and Completeness**: Inconsistent, missing, or inaccurate sales and pricing data can affect the model's performance and reliability.
2. **Competitor Pricing Volatility**: Rapid and unpredictable changes in competitor pricing can make it difficult for the model to adapt in real time.
3. **Consumer Behavior Variability**: Fluctuations in customer demand due to seasonal trends, promotions, or external economic factors can impact pricing effectiveness.
4. **Model Interpretability**: Complex machine learning models, such as ensemble methods, may lack transparency, making it challenging for businesses to trust and implement pricing recommendations.
5. **Regulatory and Ethical Considerations**: Price optimization must comply with fair pricing laws and ethical standards to avoid price discrimination or anti-competitive practices.

1.6. Challenges

1. **Feature Selection and Engineering**: Identifying the most influential features, such as competitor pricing, demand elasticity, and product attributes, requires extensive experimentation and domain expertise.
2. **Balancing Profitability and Competitiveness**: Setting optimal prices that maximize profits without losing customers to competitors is a delicate balance.

3. **Handling Dynamic Market Conditions**: The model must be adaptable to changing market trends, new product launches, and unexpected economic shifts.
4. **Scalability of the Model**: Ensuring that the model can handle large datasets across multiple product categories while maintaining efficiency is critical.
5. **Real-Time Pricing Adjustments**: Implementing real-time or near-real-time price adjustments based on model predictions requires robust infrastructure and automation capabilities.

1.7. Objective

The objective of this project is to develop a machine learning-based price optimization model that leverages historical sales data, competitor pricing, and market trends to determine optimal product pricing. The model aims to maximize profitability while ensuring competitive positioning through data-driven insights. Additionally, it enhances decision-making by providing accurate price recommendations that adapt to changing market conditions.

1.8. Summary

This project focuses on developing a machine learning-driven price optimization model to enhance retail pricing strategies. By analyzing historical sales data, competitor pricing, and market trends, the model predicts optimal price points to maximize profitability while maintaining a competitive edge. Through data-driven insights, the solution enables businesses to make informed pricing decisions, improving revenue generation and operational efficiency.

2. Design Details

2.1. Architecture

The architecture of the **Retail Price Optimization System** follows a structured, modular approach to ensure an efficient and systematic workflow for data analysis and model development. The key stages include:

2.1.1. Data Flow Architecture

1. **Import Necessary Libraries:**
   - **Objective:** Import essential libraries for data manipulation, preprocessing, visualization, model training, and evaluation.
   - **Libraries:** pandas, numpy, matplotlib, seaborn, sklearn, shap, joblib, warnings, etc.
2. **Load the Dataset:**
   - **Objective:** Load the dataset into a pandas DataFrame and create a copy for further exploration and processing.
   - **Implementation:** Use pd.read_csv() to load the dataset '**retail_price.csv**' and store it in the old_df variable. Copy data from old_df to df for further processing.
3. **Data Exploration:**
   - **Objective**: Understand the data structure in the dataset.
   - **Tasks**:
     I Display shape of the dataset (df.shape).
     II Dataset in find all column name (df.,columns)
     III Show the first 5 rows (df.head()).
     IV Summarize data (df.info()).
4. **Data Preprocessing (Data Preparation):**
   - **Objective**: Clean and preprocess the data to make it suitable for machine learning.
   - **Tasks**:
     I Handle Missing Values
       - Identify columns with missing values and their count.
       - Visualize missing values using a heatmap.

     II Handle Duplicates (Identify and count duplicate rows).

III Identify Outliers in Every Columns.

IV Identify Unique columns in the data.

V Identify categorical features & numerical features columns.

VI Perform Statistical Operation in numeric columns

2.1.2. Exploratory Data Analysis (EDA) & Visualization

- Distribution of Total Price, Unit Price, Freight Price, Product Weight, Customers per Product, Product Ratings, Product Prices
- Trend Over Time (Sales Volume, Total Price)
- Top 10 Product Categories by Sales Volume
- Product Category vs. Average Total Price
- Monthly Trend of Freight Price & Product Score
- Heatmap of correlation matrix ('product_weight_g', 'qty', 'total_price', 'freight_price', 'unit_price', 'product_score', 'customers', 'weekday', 'weekend', 'holiday')
- Pairplot ('product_weight_g', 'qty', 'total_price', 'freight_price', 'unit_price', 'product_score', 'customers', 'weekday', 'weekend', 'holiday')

2.1.3. Feature Engineering

1. Calculate Revenue, Profit, Margin
2. Price Ratio
3. Price Differences
4. Market Demand Indicators
5. Time-related Features
6. Temporal features
7. Product attribute ratios
8. Define input features (X) and target variable (y).
9. Split data into train and test sets (80% train, 20% test) using train_test_split().
10. Standardize the features using StandardScaler.

2.1.4. Model Flow Architecture

1. **Model Selection**
    - **Objective:** Train and evaluate multiple regression models to determine the best-performing one for price optimization.
    - **Models Considered**:

    - Random Forest Regression
    - Ridge
    - Linear Regression
    - Lasso
    - Gradient Boosting
    - XGBoost
- **Model Training**:
  - Fit each model on the dataset using the model.fit() method.

2. **Prediction and Performance Evaluation**

- **Objective:** Use trained models to predict optimal product prices and assess their accuracy.
- **Performance Metrics:**
  - **R^2 Score (R-Squared):** Measures how well the model explains variance in pricing.
  - **Mean Absolute Error (MAE):** Indicates the average absolute error in predicted prices.
  - **Root Mean Squared Error (RMSE):** Captures the standard deviation of residuals (prediction errors).
- **Prediction Process:**
  - Use model.predict() to generate price predictions for the test dataset.
  - Compute R^2, MAE, and RMSE using the actual vs. predicted values.

3. **Model Performance Comparison**

- **Table Representation**: Performance metrics for each model will be compared in a structured table.

4. **Visualization of Model Performance**

- **Bar Charts**: Compare the performance of different models based on R^2 Score, MAE, and RMSE.
- **Color Gradient Representation**:
  - **Green** → Better performance (higher R^2, lower MAE & RMSE)
  - **Red** → Worse performance (lower R^2, higher MAE & RMSE)

2.1.5. Model Explainability

- To ensure transparency in pricing predictions, we will use SHAP (SHapley Additive Explanations) and Permutation Importance to interpret the model's decision-making process.

  I. SHAP (SHapley Additive Explanations)

  - **Objective**: Understand the contribution of each feature to the model's predictions.
  - **Methodology**:
    - Compute SHAP values for each feature.
    - Identify the most influential factors affecting product price predictions.
    - Visualize feature importance using summary plots, dependence plots, and force plots.
  - **Outcome:** Provides insights into how historical sales data, competitor pricing, and product attributes impact optimal pricing decisions.

  II. Permutation Importance

  - **Objective**: Assess feature importance by measuring the decrease in model performance when a feature's values are randomly shuffled.
  - **Methodology**:
    - Shuffle one feature's values while keeping other features unchanged.
    - Measure the drop in R^2 score and increase in MAE/RMSE.
    - Rank features based on their impact on model performance.
  - **Outcome:** Identifies key pricing factors by evaluating how much each feature contributes to prediction accuracy.

- **Visualization of Model Explainability**
  - **SHAP Summary Plot**: Highlights the overall impact of each feature on pricing predictions.
  - **Permutation Importance Bar Chart**: Ranks features based on their influence on model performance.

2.1.6. Visualization Flow Architecture

1. **Pricing Trends**
   - **Objective:** Track the fluctuations in product prices over time, including competitor pricing and our own price trends.

- **Visualization:** A line plot comparing the monthly pricing trends of our product and three competitors.

2. **Competitor Analysis**
   - **Objective:** Analyze the relationship between competitor pricing and total sales, as well as compare our product prices with competitor prices.
   - **Visualization:** A scatter plot representation for:
     - Competitor Prices vs. Total Sales
     - Our Prices vs. Competitor Prices

3. **Model Predictions**
   - **Objective:** Compare actual vs. predicted product prices for different regression models.
   - **Visualization:** Scatter plots for each model's predictions.

2.1.7. Model Deployment

- Scaler Model
- Random Forest Regressor Model
- Rodge Model
- Linear Regression Model
- Lasso Model
- Gradient Boosting Model
- XGBoost Model

2.1.8. Complete Flow Diagram

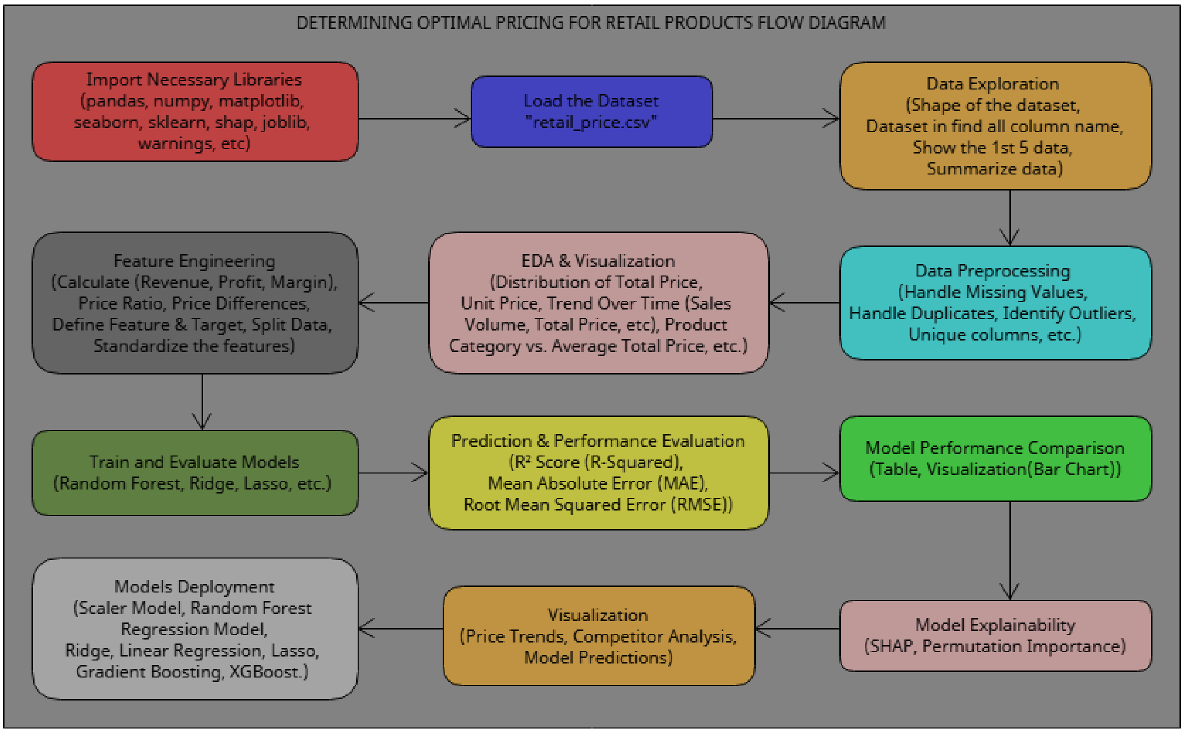


Fig 1 - Complete Model Flow Diagram

2.1.9. Implementation Workflow

1. **Import Libraries:** Load essential Python libraries for data manipulation, visualization, and model training, including Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, SHAP, and Joblib.
2. **Load Dataset:** Read "**retail_price.csv**" into a Pandas DataFrame (old_df) and create a copy (df) for further processing.
3. **Data Exploration**
   - Display dataset shape (df.shape), first 5 rows (df.head()), and column details (df.info()).
   - List all column names (df.columns).
4. **Data Preprocessing:**
   - Handle missing values by identifying columns with missing data and visualizing them using a heatmap.
   - Detect and remove duplicate rows.
   - Identify outliers in numerical columns using statistical analysis.

- Separate categorical and numerical features.
- Perform statistical operations on numerical columns.

5. **Feature Engineering**
    - Compute additional features like revenue, profit, and price margins.
    - Calculate price ratios and differences across competitors.
    - Generate market demand indicators and temporal features (seasonality, month, and year effects).
    - Define input features (X) and target variable (y).
    - Split the dataset into training (80%) and testing (20%) using train_test_split().
    - Standardize numerical features using StandardScaler().
6. **Model Training & Evaluation**
    - Train multiple regression models, including Random Forest, Ridge, Linear Regression, Lasso, Gradient Boosting, and XGBoost.
    - Make predictions using model.predict().
    - Evaluate performance using **R^2 Score, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE)**.
    - Compare model performance using a structured table and bar charts (green for better performance, red for worse).
7. **Model Explainability**
    - Use **SHAP values** to identify key features influencing price predictions.
    - Apply **Permutation Importance** to measure the impact of each feature on model performance.
    - Visualize feature importance using SHAP summary plots and permutation importance bar charts.
8. **Visualization Flow**
    - **Pricing Trends Over Time**: Line plot comparing price trends for our product vs. competitors.
    - **Competitor Analysis**: Scatter plot showing the relationship between competitor pricing and sales.
    - **Predicted vs. Actual Prices**: Scatter plots for different models comparing predictions to actual values.

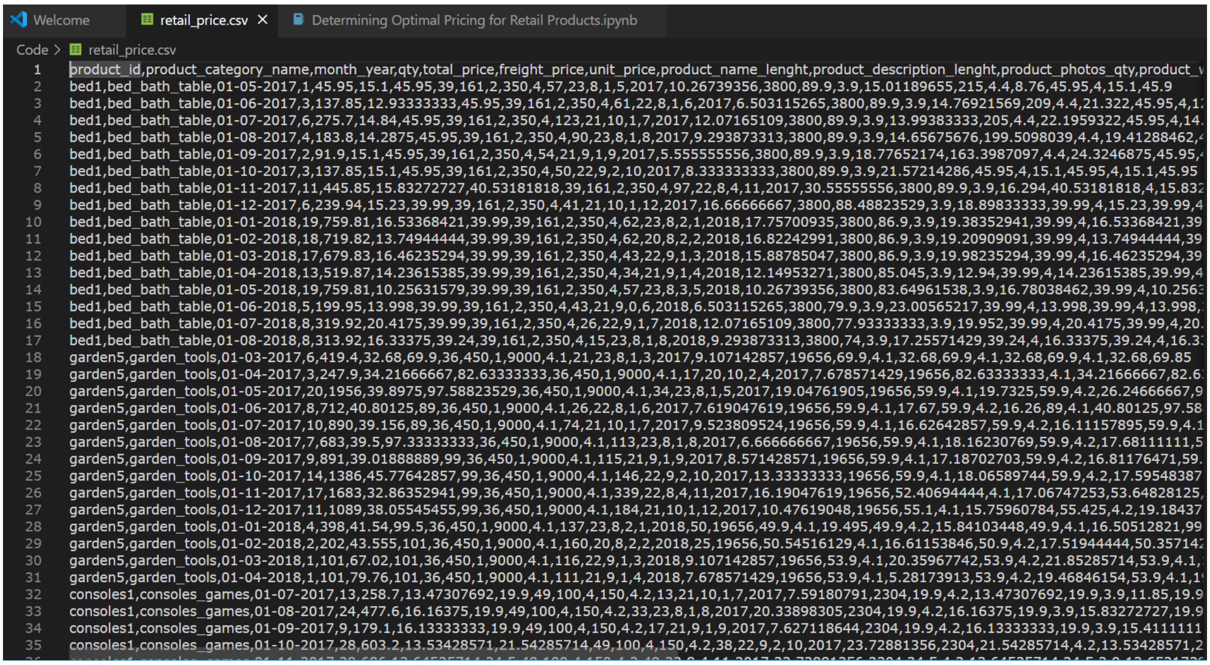
9. **Model Deployment**
   - Save trained models (joblib.dump(model, "model.pkl")).
   - Deploy the best-performing model for real-time price recommendations.

2.2. Dataset Details

Field Name	Description
product_id	Unique identifier for each product
product_category_name	Category of the product
month_year	Date of the sales record
qty	Quantity of product sold
total_price	Total price of the sale
freight_price	Shipping cost
unit_price	Price per unit of the product
product_name_lenght	Length of the product name
product_description_lenght	Length of the product description
product_photos_qty	Number of photos for the product
product_weight_g	Weight of the product in grams
product_score	Rating or score of the product
customers	Number of customers who purchased the product
weekday	Number of weekdays in the month
weekend	Number of weekend days in the month
holiday	Number of holidays in the month
month	Month of the sale
year	Year of the sale
s	Likely a calculated field, possibly sales rate or similar metric
volume	Possibly the volume of the product
comp_1	Price of competitor 1's product
ps_1	Product score of competitor 1's product
fp_1	Freight price of competitor 1's product
comp_2	Price of competitor 2's product
ps2	Product score of competitor 2's product
fp2	Freight price of competitor 2's product
comp_3	Price of competitor 3's product

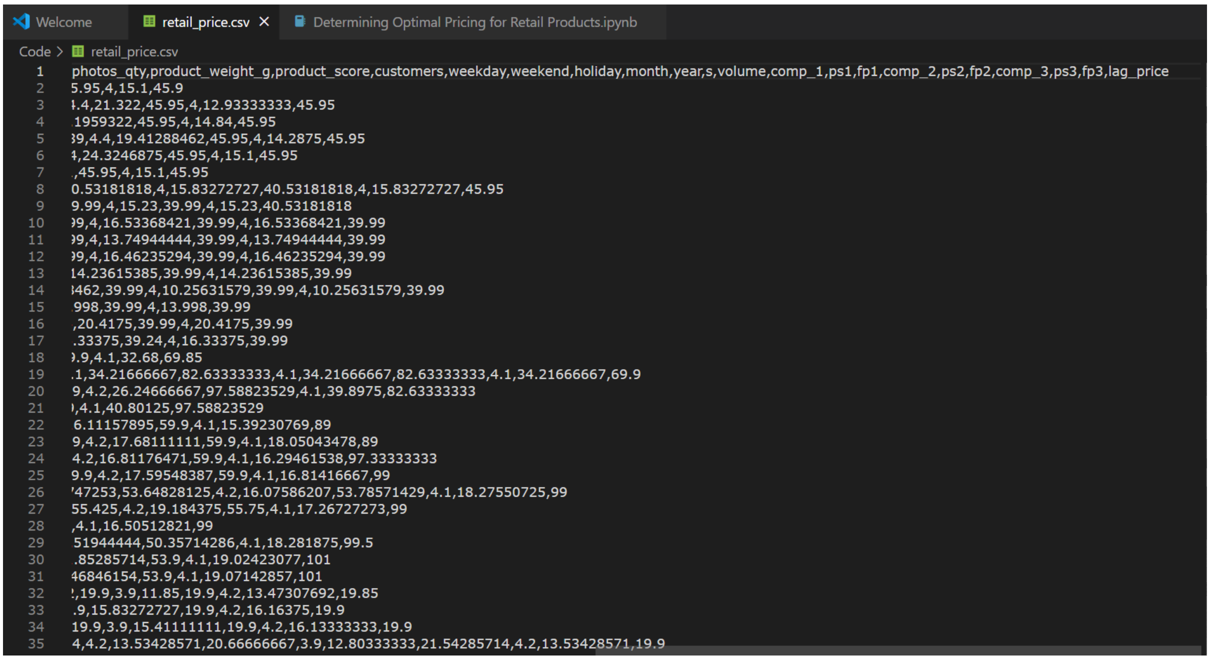
ps3	Product score of competitor 3's product
fp3	Freight price of competitor 3's product
(Target) lag_price	Price of the product in the previous time period

Table 1 - Dataset Details



```
Welcome    retail_price.csv ×    Determining Optimal Pricing for Retail Products.ipynb
Code > retail_price.csv
product_id,product_category_name,month_year,qty,total_price,freight_price,unit_price,product_name_lenght,product_description_lenght,product_photos_qty,product_v
bed1,bed_bath_table,01-05-2017,1,45.95,15.1,45.95,39,161,2,350,4,57,23,8,1,5,2017,10.26739356,3800,89.9,3.9,15.01189655,215,4.4,8.76,45.95,4,15.1,45.9
bed1,bed_bath_table,01-06-2017,3,137.85,12.93333333,45.95,39,161,2,350,4,61,22,8,1,6,2017,6.503115265,3800,89.9,3.9,14.76921569,209,4.4,21.322,45.95,4,1
bed1,bed_bath_table,01-07-2017,6,275.7,14.84,45.95,39,161,2,350,4,123,21,10,1,7,2017,12.07165109,3800,89.9,3.9,13.99383333,205,4.4,22.1959322,45.95,4,14.
bed1,bed_bath_table,01-08-2017,4,183.8,14.2875,45.95,39,161,2,350,4,90,23,8,1,8,2017,9.293873313,3800,89.9,3.9,14.65675676,199.5098039,4.4,19.41288462,
bed1,bed_bath_table,01-09-2017,2,91.9,15.1,45.95,39,161,2,350,4,54,21,9,1,9,2017,5.555555556,3800,89.9,3.9,18.77652174,163.3987097,4.4,24.3246875,45.95,
bed1,bed_bath_table,01-10-2017,3,137.85,15.1,45.95,39,161,2,350,4,50,22,9,2,10,2017,8.333333333,3800,89.9,3.9,21.57214286,45.95,4,15.1,45.95,4,15.1,45.95
bed1,bed_bath_table,01-11-2017,11,445.85,15.83272727,40.53181818,39,161,2,350,4,97,22,8,4,11,2017,30.55555556,3800,89.9,3.9,16.294,40.53181818,4,15.832
bed1,bed_bath_table,01-12-2017,6,239.94,15.23,39.99,39,161,2,350,4,41,21,10,1,12,2017,16.66666667,3800,88.48823529,3.9,18.89833333,39.99,4,15.23,39.99,4
bed1,bed_bath_table,01-01-2018,19,759.81,16.53368421,39.99,39,161,2,350,4,62,23,8,2,1,2018,17.75700935,3800,86.9,3.9,19.38352941,39.99,4,16.53368421,39
bed1,bed_bath_table,01-02-2018,18,719.82,13.74944444,39.99,39,161,2,350,4,62,20,8,2,2,2018,16.82242991,3800,86.9,3.9,19.20909091,39.99,4,13.74944444,39
bed1,bed_bath_table,01-03-2018,17,679.83,16.46235294,39.99,39,161,2,350,4,43,22,9,1,3,2018,15.88785047,3800,86.9,3.9,19.98235294,39.99,4,16.46235294,39
bed1,bed_bath_table,01-04-2018,13,519.87,14.23615385,39.99,39,161,2,350,4,34,21,9,1,4,2018,12.14953271,3800,85.045,3.9,12.94,39.99,4,14.23615385,39.99,4
bed1,bed_bath_table,01-05-2018,19,759.81,10.25631579,39.99,39,161,2,350,4,57,23,8,3,5,2018,10.26739356,3800,83.64961538,3.9,16.78038462,39.99,4,10.2563
bed1,bed_bath_table,01-06-2018,5,199.95,13.998,39.99,39,161,2,350,4,43,21,9,0,6,2018,6.503115265,3800,79.9,3.9,23.00565217,39.99,4,13.998,39.99,4,13.998,
bed1,bed_bath_table,01-07-2018,8,319.92,20.4175,39.99,39,161,2,350,4,26,22,9,1,7,2018,12.07165109,3800,77.93333333,3.9,19.952,39.99,4,20.4175,39.99,4,20.
bed1,bed_bath_table,01-08-2018,8,313.92,16.33375,39.24,39,161,2,350,4,15,23,8,1,8,2018,9.293873313,3800,74,3.9,17.25571429,39.24,4,16.33375,39.24,4,16.3
garden5,garden_tools,01-03-2017,6,419.4,32.68,69.9,36,450,1,9000,4.1,21,23,8,1,3,2017,9.107142857,19656,69.9,4.1,32.68,69.9,4.1,32.68,69.9,4.1,32.68,69.85
garden5,garden_tools,01-04-2017,3,247.9,34.21666667,82.63333333,36,450,1,9000,4.1,17,20,10,2,4,2017,7.678571429,19656,82.63333333,4.1,34.21666667,82.6
garden5,garden_tools,01-05-2017,20,1956,39.8975,97.58823529,36,450,1,9000,4.1,34,23,8,1,5,2017,19.04761905,19656,59.9,4.1,19.7325,59.9,4.2,26.24666667,9
garden5,garden_tools,01-06-2017,8,712,40.80125,89,36,450,1,9000,4.1,26,22,8,1,6,2017,7.619047619,19656,59.9,4.1,17.67,59.9,4.2,16.26,89,4.1,40.80125,97.58
garden5,garden_tools,01-07-2017,10,890,39.156,89,36,450,1,9000,4.1,74,21,10,1,7,2017,9.523809524,19656,59.9,4.1,16.62642857,59.9,4.2,16.11157895,59.9,4.1
garden5,garden_tools,01-08-2017,7,683,39.5,97.33333333,36,450,1,9000,4.1,113,23,8,1,8,2017,6.666666667,19656,59.9,4.1,18.16230769,59.9,4.2,17.68111111,5
garden5,garden_tools,01-09-2017,9,891,39.01888889,99,36,450,1,9000,4.1,115,21,9,1,9,2017,8.571428571,19656,59.9,4.1,17.18702703,59.9,4.2,16.81176471,59.
garden5,garden_tools,01-10-2017,14,1386,45.77642857,99,36,450,1,9000,4.1,146,22,9,2,10,2017,13.33333333,19656,59.9,4.1,18.06589744,59.9,4.2,17.59548387
garden5,garden_tools,01-11-2017,17,1683,32.86352941,99,36,450,1,9000,4.1,339,22,8,4,11,2017,16.19047619,19656,52.40694444,4.1,17.06747253,53.64828125,
garden5,garden_tools,01-12-2017,11,1089,38.05545455,99,36,450,1,9000,4.1,184,21,10,1,12,2017,10.47619048,19656,55.1,4.1,15.75960784,55.425,4.2,19.18437
garden5,garden_tools,01-01-2018,4,398,41.54,99.5,36,450,1,9000,4.1,137,23,8,2,1,2018,50,19656,49.9,4.1,19.495,49.9,4.2,15.84103448,49.9,4.1,16.50512821,99
garden5,garden_tools,01-02-2018,2,202,43.555,101,36,450,1,9000,4.1,160,20,8,2,2,2018,25,19656,50.54516129,4.1,16.61153846,50.9,4.2,17.51944444,50.357142
garden5,garden_tools,01-03-2018,1,101,67.02,101,36,450,1,9000,4.1,116,22,9,1,3,2018,9.107142857,19656,53.9,4.1,20.35967742,53.9,4.2,21.85285714,53.9,4.1,
garden5,garden_tools,01-04-2018,1,101,79.76,101,36,450,1,9000,4.1,111,21,9,1,4,2018,7.678571429,19656,53.9,4.1,5.28173913,53.9,4.2,19.46846154,53.9,4.1,1
consoles1,consoles_games,01-07-2017,13,258.7,13.47307692,19.9,49,100,4,150,4.2,13,21,10,1,7,2017,7.59180791,2304,19.9,4.2,13.47307692,19.9,3.9,11.85,19.9
consoles1,consoles_games,01-08-2017,24,477.6,16.16375,19.9,49,100,4,150,4.2,33,23,8,1,8,2017,20.33898305,2304,19.9,4.2,16.16375,19.9,3.9,15.83272727,19.9
consoles1,consoles_games,01-09-2017,9,179.1,16.13333333,19.9,49,100,4,150,4.2,17,21,9,1,9,2017,7.627118644,2304,19.9,4.2,16.13333333,19.9,3.9,15.4111111
consoles1,consoles_games,01-10-2017,28,603.2,13.53428571,21.54285714,49,100,4,150,4.2,38,22,9,2,10,2017,23.72881356,2304,21.54285714,4.2,13.53428571,2
```

Fig 2 - Dataset



```
Welcome    retail_price.csv ×    Determining Optimal Pricing for Retail Products.ipynb
Code > retail_price.csv
photos_qty,product_weight_g,product_score,customers,weekday,weekend,holiday,month,year,s,volume,comp_1,ps1,fp1,comp_2,ps2,fp2,comp_3,ps3,fp3,lag_price
5.95,4,15.1,45.9
.4,21.322,45.95,4,12.93333333,45.95
.1959322,45.95,4,14.84,45.95
39,4.4,19.41288462,45.95,4,14.2875,45.95
,24.3246875,45.95,4,15.1,45.95
,45.95,4,15.1,45.95
0.53181818,4,15.83272727,40.53181818,4,15.83272727,45.95
9.99,4,15.23,39.99,4,15.23,40.53181818
99,4,16.53368421,39.99,4,16.53368421,39.99
99,4,13.74944444,39.99,4,13.74944444,39.99
99,4,16.46235294,39.99,4,16.46235294,39.99
14.23615385,39.99,4,14.23615385,39.99
462,39.99,4,10.25631579,39.99,4,10.25631579,39.99
998,39.99,4,13.998,39.99
,20.4175,39.99,4,20.4175,39.99
.33375,39.24,4,16.33375,39.99
.9,4.1,32.68,69.85
.1,34.21666667,82.63333333,4.1,34.21666667,82.63333333,4.1,34.21666667,69.9
9,4.2,26.24666667,97.58823529,4.1,39.8975,82.63333333
,4.1,40.80125,97.58823529
6.11157895,59.9,4.1,15.39230769,89
9,4.2,17.68111111,59.9,4.1,18.05043478,89
4.2,16.81176471,59.9,4.1,16.29461538,97.33333333
9.9,4.2,17.59548387,59.9,4.1,16.81416667,99
47253,53.64828125,4.2,16.07586207,53.78571429,4.1,18.27550725,99
55.425,4.2,19.184375,55.75,4.1,17.26727273,99
,4.1,16.50512821,99
51944444,50.35714286,4.1,18.281875,99.5
.85285714,53.9,4.1,19.02423077,101
46846154,53.9,4.1,19.07142857,101
,19.9,3.9,11.85,19.9,4.2,13.47307692,19.85
.9,15.83272727,19.9,4.2,16.16375,19.9
19.9,3.9,15.41111111,19.9,4.2,16.13333333,19.9
4,4.2,13.53428571,20.66666667,3.9,12.80333333,21.54285714,4.2,13.53428571,19.9
```

Fig 3 - Dataset

3. Implementation Details of Project

https://qr.me-qr.com/mobile/pdf/7eaefde9-239a-41e0-bef0-8ff4de6e6dd9

Fig 4 – Implementation Code PDF & Link