CHAPTER 1

INTRODUCTION

The retail industry, particularly within the hospitality sector, operates in a highly dynamic and competitive environment. Factors such as seasonal demand fluctuations, competitor pricing, consumer behavior, and economic conditions influence how products and services are priced. Traditionally, businesses have relied on manual methods and static pricing strategies based on human intuition or historical trends. However, these methods often fail to capture complex, non-linear relationships that influence optimal pricing.

1.1 Overview

The **Retail Price Optimization System** developed in this project uses machine learning to analyze historical sales data, product attributes, and competitor pricing to recommend optimal prices for products in the hospitality retail space. The core objective is to create a model that can simulate human-like pricing decisions, but with much greater speed and accuracy.

To visualize the data and understand key trends, exploratory data analysis (EDA) techniques using interactive tools like Plotly were employed. The modeling part of the project uses a **Decision Tree Regressor**, which is trained to predict the **total price** based on various inputs. The performance is evaluated using actual vs. predicted price comparison charts.

1.2 Motivation

In the hospitality industry, pricing can make or break profitability. From hotel mini-bars and room service to event catering and retail outlets, every product or service must be strategically priced to attract customers while ensuring optimal profit margins. However, deciding the right price is not a one-size-fits-all solution—it depends on multiple dynamic factors like competitor pricing, demand trends, product quality, and customer preferences.

This project is driven by the need to bring **intelligent automation** to the pricing process. With the rise of **machine learning**, it is now possible to analyze complex data patterns and automate the task of price optimization. ML models can process thousands of pricing scenarios, learning from past data and suggesting optimal price points in real time—something no human team can do consistently or quickly.

The motivation behind this project is to:

- Help hospitality retailers stay competitive in pricing battles
- Reduce dependency on manual pricing analysis

1.3 Problem Statement

In the hospitality industry, determining the right price for products is a complex challenge. Prices must not only be attractive to customers but also account for profit margins, competitor pricing, demand trends, and product performance. Manual pricing methods or static rules often fail to consider these variables effectively, leading to underpricing (loss of profit) or overpricing (loss of sales).

This project addresses the problem of **retail price optimization** by developing a machine learning-based system that can:

- Analyze historical sales data, including product attributes and external competitor prices
- Understand relationships between price and key influencing factor.

CHAPTER 2

Requirements Specification:

2.1 Software Requirements

1. Operating System

- Windows 11, Linux (Ubuntu), or macOS

2. Programming Language

- Python 3.7 or above

3. Libraries and Frameworks

- NumPy – for numerical operations
- Pandas – for data manipulation and analysis
- Plotly – for interactive data visualizations
- Scikit-learn – for traditional machine learning models

4. Development Tools

- Anaconda (for package management and environment setup)

5. Database/Storage

- CSV file (retail_price.csv) used for accessing and storing historical pricing data

2.2 Hardware Requirements

1. Processor (CPU)

- Minimum: Intel i5 or AMD Ryzen 5
- Recommended: Intel i7/i9 or AMD Ryzen 7/9 (for faster model training)

2. Memory (RAM)

- Minimum: 8 GB
- Recommended: 16 GB or higher (especially for training deep learning models)

3. Storage

- Minimum: 100 MB free space (for datasets and code)
- Recommended: SSD with at least 1 GB free space for smoother performance

CHAPTER 3

DATA COLLECTION

This project uses a real-world dataset designed to simulate a retail environment in the hospitality industry. The dataset includes historical sales transactions, product information, competitor pricing, and other business-relevant variables. These features help train the machine learning model to understand how various factors influence retail pricing and total revenue.

Dataset Overview

- **Source**: Provided CSV file named retail_price.csv
- **Records**: 2,020 rows (approximately)
- **Purpose**: To predict the total_price of a product based on features like quantity sold, unit price, product score, and competitor prices.

Features in the Dataset:

Feature Name	Description
product_category_name	Name of the product category (e.g., snacks, drinks, hygiene products, etc.)
qty	Quantity of the product sold
unit_price	Price of a single unit of the product
total_price	Total price = qty × unit_price
product_score	A quality score for the product (e.g., user rating or internal quality measure)
comp_1	Competitor’s price for the same/similar product
weekday	Day of the week on which the sale happened
holiday	Whether the sale happened on a holiday (Yes/No)

Sample Records:

product_category_name	qty	unit_price	total_price	product_score	comp_1	weekday	holiday
Snacks	3	50.0	150.0	4.2	48.0	Monday	No
Drinks	2	75.0	150.0	4.5	77.0	Saturday	Yes
Hygiene	1	120.0	120.0	3.9	115.0	Tuesday	No

Statistical Summary:

Feature	Mean	Min	Max	Std Dev
qty	~2.5	1	10	~1.8
unit_price	~80.0	~10.0	~300.0	~50.0
total_price	~200.0	~10.0	~2000.0	~180.0
product_score	~4.1	1.0	5.0	~0.5
comp_1	~78.0	~5.0	~310.0	~45.0

CHAPTER 4

DATA PROCESSING

Before training any machine learning model, it is essential to prepare the dataset by handling missing values, engineering new features, and scaling the data to ensure optimal model performance. The following steps outline the data processing pipeline used in this project.

Step 1: Load the Dataset

The dataset is loaded from a CSV file using pandas.

```
import pandas as pd
```

```
data = pd.read_csv('retail_price.csv')
```

Step 2: Handle Missing Values

We checked for missing values and found no major issues. In case of any null entries, we would use either forward-fill or drop them based on the situation.

```
data.isnull().sum()
```

Step 3: Feature Engineering

To make the model more informative, a new feature was added:

comp_price_diff: Difference between the product’s unit price and the competitor’s price.

```
data['comp_price_diff'] = data['unit_price'] - data['comp_1']
```

This feature helps the model understand how price competitiveness affects total sales.

Step 4: Feature Selection

The selected features for training are:

- qty: Quantity sold
- unit_price: Price per unit
- comp_1: Competitor price
- product_score: Product quality score

- comp_price_diff: Engineered feature
- Target: total_price

```
X = data[['qty', 'unit_price', 'comp_1', 'product_score', 'comp_price_diff']]

y = data['total_price']
```

Step 5: Train-Test Split

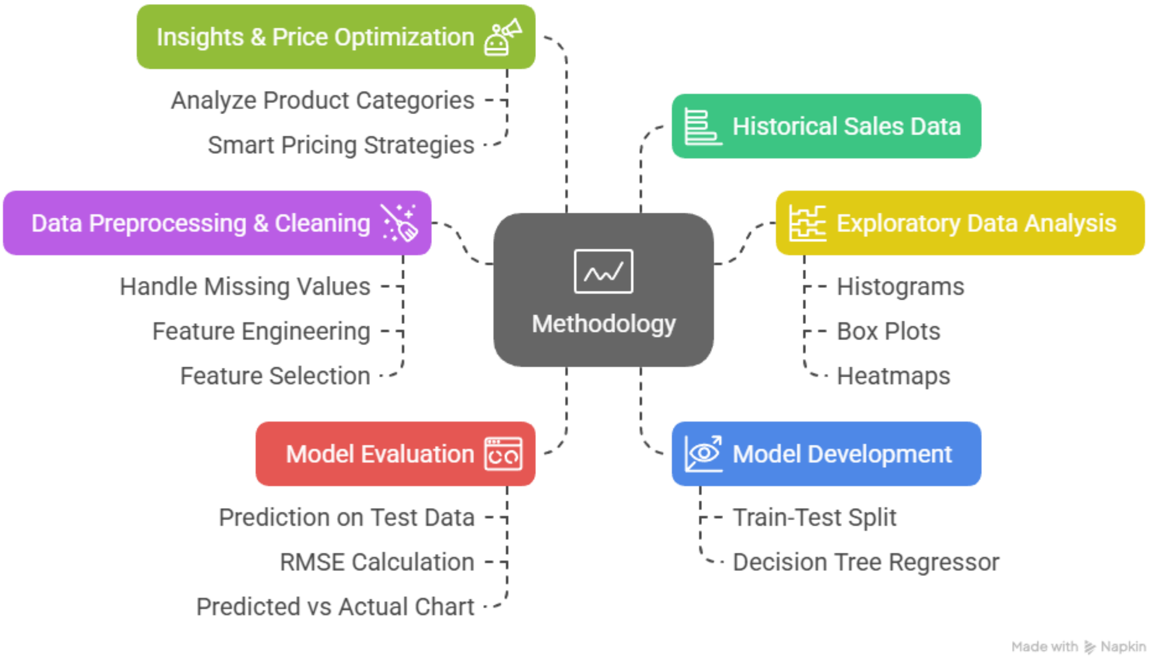
The dataset was split into training and testing sets using an 80/20 ratio.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

CHAPTER 5

METHODOLOGY

The methodology describes the step-by-step approach used to develop the Retail Price Optimization system using machine learning techniques. The pipeline includes data loading, preprocessing, feature engineering, model training, and evaluation.



1. Data Collection

The dataset used (retail_price.csv) contains historical pricing data for retail products in the hospitality domain. Each record includes product details, quantity sold, price, competitor pricing, and other contextual features such as weekday and holiday

2. Data Preprocessing

As detailed in Chapter 4, preprocessing involved:

- Handling missing values
- Feature engineering (comp_price_diff)

- Feature selection
- Splitting into training and testing sets

3. Model Selection

The model selected for this implementation was:

Decision Tree Regressor

- **Why this model?**
    - It handles non-linear relationships well.
    - It does not require feature scaling.
    - It is easy to interpret and visualize.

```
from sklearn.tree import DecisionTreeRegressor

model = DecisionTreeRegressor()

model.fit(X_train, y_train)
```

4. Prediction

Once the model was trained, predictions were made on the test set:

```
y_pred = model.predict(X_test)
```

5. Evaluation Metrics

To assess how well the model predicted the total_price, we used the following metric:

- **Root Mean Squared Error (RMSE)** – to measure the average prediction error.

```
from sklearn.metrics import mean_squared_error

import numpy as np

rmse = np.sqrt(mean_squared_error(y_test, y_pred))

print("RMSE:", rmse)
```

6. Visualization of Results

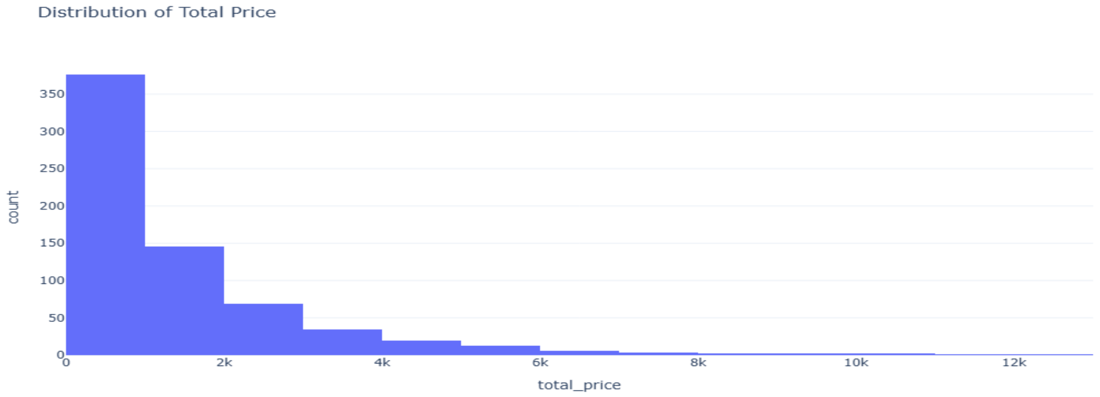
A scatter plot was used to compare predicted vs actual total prices:

```
import plotly.graph_objects as go

fig = go.Figure()

fig.add_trace(go.Scatter(x=y_test, y=y_pred, mode='markers', name='Predicted vs Actual'))

fig.add_trace(go.Scatter(x=[min(y_test), max(y_test)], y=[min(y_test), max(y_test)], mode='lines', name='Ideal'))

fig.update_layout(title='Predicted vs. Actual Retail Price',

                xaxis_title='Actual Total Price',

                  yaxis_title='Predicted Total Price')

fig.show()
```

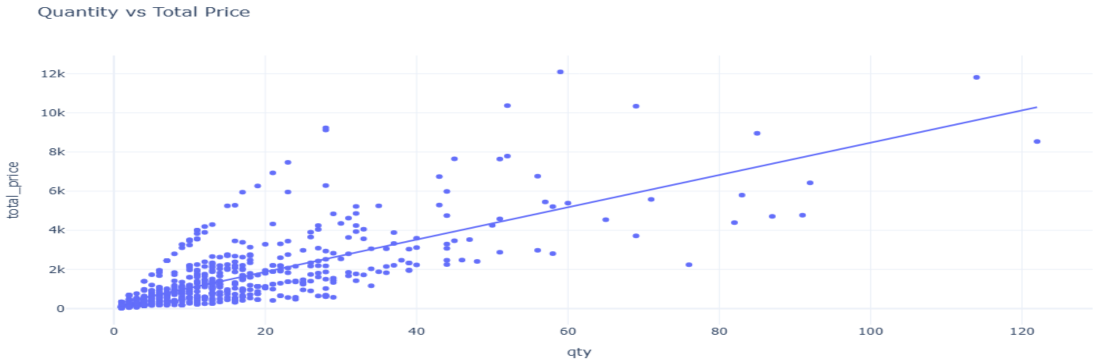
This visualization helps identify how closely the model’s predictions match real-world values.

CHAPTER 6

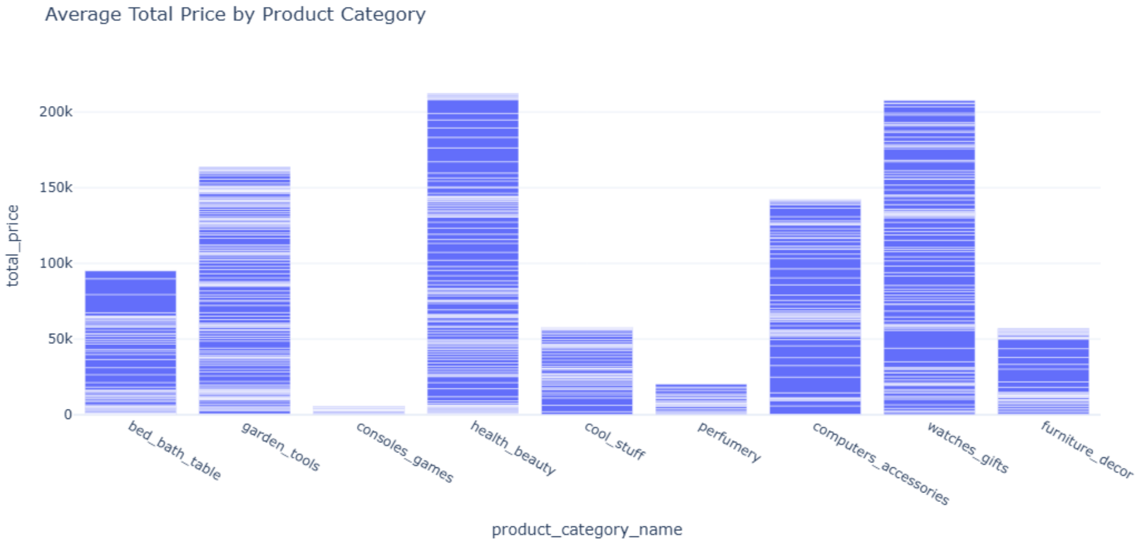
Results:



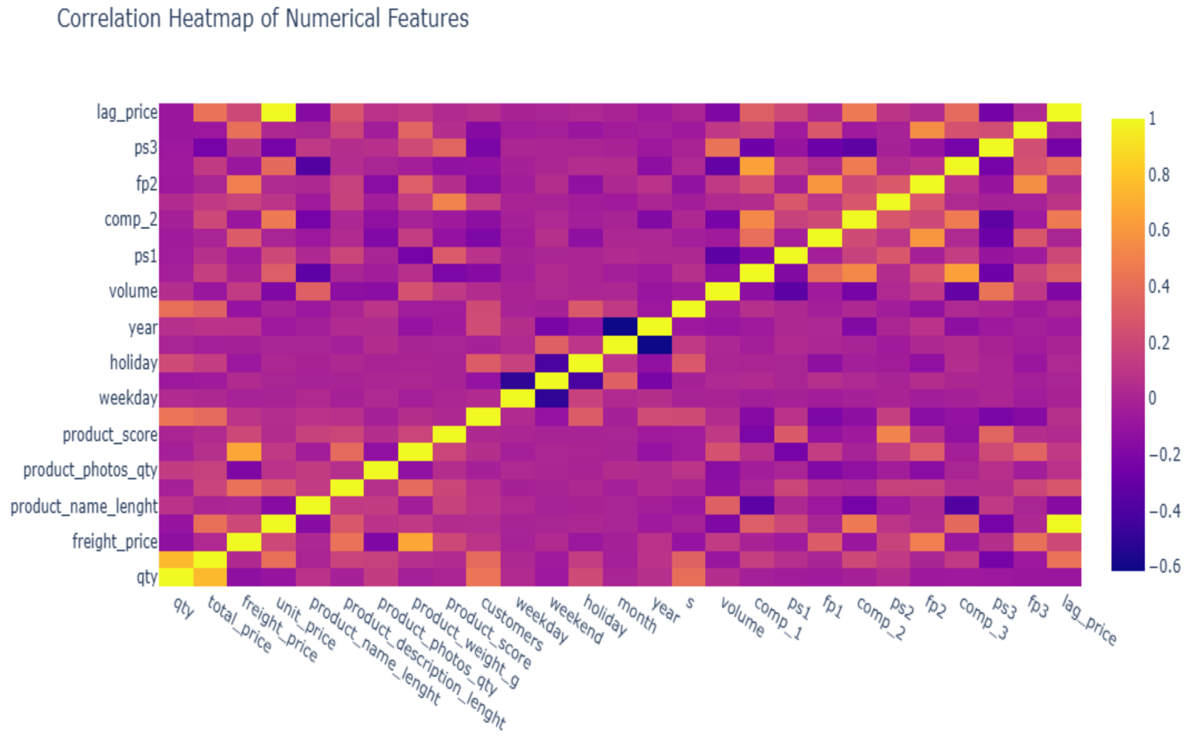
Snapshot 1: total price distribution



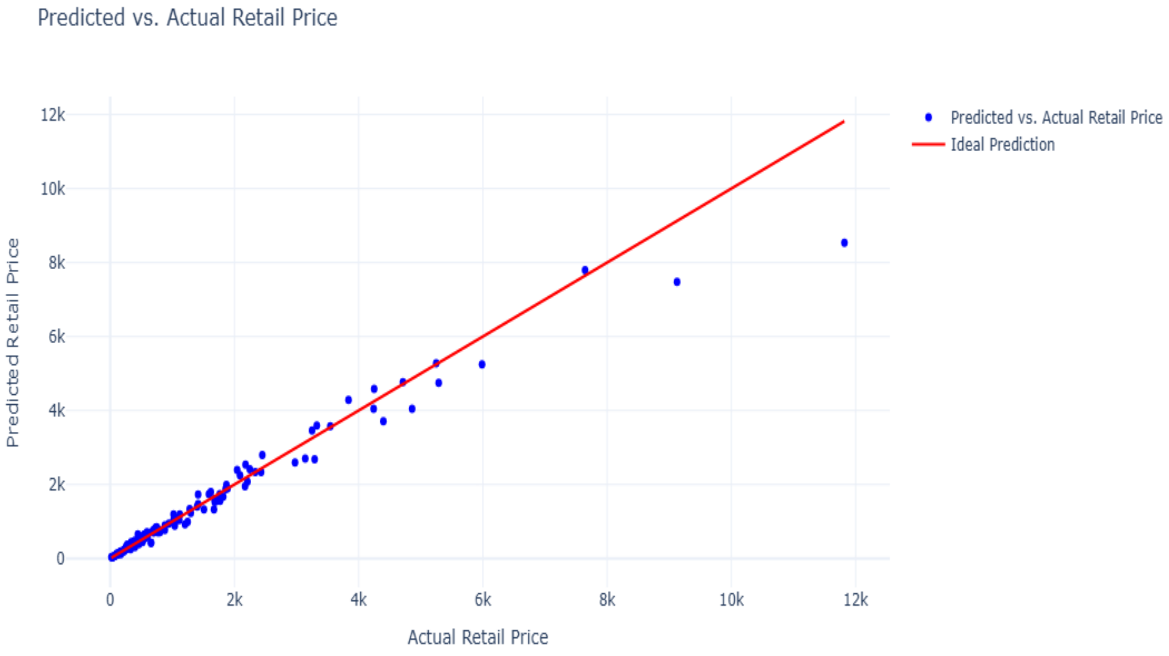
Snapshot 2: quantity vs price relationship



Snapshot 3: category-wise revenue



Snapshot 4: Feature correlation heatmap



Snapshot 5: Model result – final prediction vs actual

CHAPTER 7

Conclusion:

Retail price optimization is a critical challenge in the hospitality industry, where pricing directly impacts profitability, competitiveness, and customer satisfaction. In this project, we developed a machine learning-based system aimed at predicting and optimizing product prices using historical sales data, product features, and competitor pricing.

We started by collecting and exploring a real-world dataset containing product categories, quantities sold, unit prices, competitor prices, and customer ratings. Data preprocessing played a vital role, involving the handling of missing values, engineering of new features (like competitor price difference), and splitting the dataset into training and testing sets.

The machine learning model used for prediction was a **Decision Tree Regressor**, which is well-suited for capturing non-linear relationships in pricing behavior. The model was trained to predict the **total price** of a product based on several input features and was evaluated using metrics such as **Root Mean Squared Error (RMSE)**. Visual comparisons between predicted and actual prices indicated that the model was capable of identifying pricing trends with a reasonable degree of accuracy.

Additionally, exploratory data analysis using interactive Plotly graphs helped uncover valuable insights into customer behavior, price sensitivity by category, and the influence of holidays or weekdays on sales. These insights can be directly applied by hospitality businesses to inform their pricing strategies.

This project successfully demonstrates how machine learning can be applied to automate and optimize pricing decisions in a real-world business scenario. While the model performs well, there is room for improvement. Future enhancements could include integrating time-series trends, seasonal factors, demand forecasting, and external influences such as location, market events, or customer feedback.

In conclusion, this project has provided a practical and impactful application of machine learning in the field of retail pricing. It showcases how data-driven techniques can transform traditional business operations and support more intelligent, adaptive, and profitable decision-making in the hospitality sector.