



**AJEENKYA**  
D Y PATIL UNIVERSITY  
THE INNOVATION UNIVERSITY

**A**  
**MINI PROJECT REPORT ON**  
**“Beverage Sales Analysis and Prediction Using Machine Learning”**

**FOR**  
**Term Work Examination**

*Bachelors in computer application in AIML (BCA - AIML)*  
**Year 2024-2025**

**Ajeenkya DY Patil University, Pune**

**Submitted By**  
**Mr. Atharva Jagtap**  
**Under the guidance of**  
**Prof. Vivek More**

**Ajeenkya DY Patil University**  
**D Y Patil Knowledge City,**  
**Charholi Bk. Via Lohegaon,**  
**Pune - 412105**  
**Maharashtra (India)**

**Date: 16/04/ 2025**

## **CERTIFICATE**

This is to certified that **Atharva Jagtap**  
A student's of BCA(AIML) Sem-IV URN No **2023-B-18052005C** has  
Successfully Completed the Dashboard Report On

**“Beverage Sales Analysis and Prediction Using Machine  
Learning”**

As per the requirement of  
Ajeenkya DY Patil University, Pune was carried out under my  
supervision.  
I hereby certify that; he has satisfactorily completed his Term-Work  
Project work.  
Place: - Pune

**Examiner**

<b>INDEX</b>	
	<b>Page No.</b>
<b>ABSTRACT</b>	<b>4</b>
<b>CHAPTER – 1 INTRODUCTION &amp; OBJECTIVE</b>	<b>5-9</b>
<b>CHAPTER – 2 METHODOLOGY &amp; APPROACH</b>	<b>10-13</b>
<b>CHAPTER – 3 IMPLEMENTATION &amp; CODE</b>	<b>14-15</b>
<b>CHAPTER – 4 RESULTS &amp; VISUALIZATIONS</b>	<b>16-20</b>
<b>CHAPTER – 5 CONCLUSION &amp; FUTURE SCOPE</b>	<b>21-22</b>
<b>REFERENCES</b>	<b>23</b>

## **Abstract**

This project focuses on analyzing and modeling beverage sales data to uncover insights and predict revenue outcomes based on key product and customer attributes. The dataset includes transactional information such as product category, unit price, quantity, discount, region, and customer type.

A comprehensive data preprocessing pipeline was implemented to clean, transform, and prepare the dataset for analysis. Exploratory Data Analysis (EDA) provided valuable insights into customer purchasing behavior, product popularity, and regional sales trends. Visualizations such as boxplots, count plots, and time series graphs helped identify patterns and outliers, while correlation analysis revealed key relationships among numerical features.

To predict total sales revenue, a Linear Regression model was developed using features like unit price, quantity, discount, and encoded categorical variables. The model was trained and evaluated using standard metrics such as Mean Squared Error (MSE) and  $R^2$  score, providing a foundation for forecasting sales and supporting business decision-making.

This project demonstrates the potential of data-driven strategies to optimize pricing, product selection, and customer targeting in the beverage industry, paving the way for more advanced predictive analytics and recommendation systems.

# Introduction

In the modern retail landscape, data plays a pivotal role in shaping strategic business decisions. With the increasing availability of consumer data, organizations are leveraging analytics to drive profitability, enhance customer satisfaction, and gain competitive advantages. The beverage industry, a key segment within the fast-moving consumer goods (FMCG) market, is no exception. This sector experiences high consumer demand, dynamic trends, and intense competition, all of which necessitate a deep understanding of customer behavior, product performance, and market fluctuations.

One of the most effective ways to uncover such insights is by analyzing historical sales data. By examining patterns in past transactions, businesses can gain valuable knowledge about which products are performing well, how different customer segments behave, what pricing strategies are effective, and which regions generate the most revenue. This kind of analysis not only supports better operational planning but also helps in forecasting future sales, managing inventory efficiently, and designing targeted marketing campaigns.

This project, titled "**Beverage Sales Analysis and Prediction**", focuses on exploring a synthetic dataset containing beverage sales records. The dataset includes a range of features such as order ID, customer type, product and category details, unit price, quantity sold, applied discounts, total price, region, and order date. These variables provide a comprehensive view of each transaction and serve as the basis for detailed analysis and modeling.

The primary objective of this project is twofold:

1. **To perform an in-depth exploratory data analysis (EDA)** that reveals hidden patterns, trends, and relationships within the data.
2. **To build a predictive model** that can estimate the total price (revenue) of a transaction based on input features such as product attributes and customer information.

During the EDA phase, various statistical and visual techniques are employed to investigate the distribution of numerical and categorical variables, detect outliers, and understand correlations between different features. Time series analysis is also applied

to identify trends over time, such as seasonal changes in sales volume or revenue spikes associated with certain product categories or customer types.

For the predictive modeling phase, a Linear Regression model is constructed using both numerical and categorical features. Categorical variables such as customer type, product, and region are encoded using One-Hot Encoding to ensure compatibility with the machine learning algorithm. The model is then trained and evaluated using metrics like Mean Squared Error (MSE) and  $R^2$  score, which help assess its performance and accuracy.

By the end of the project, we aim to demonstrate how data science techniques can be applied to real-world retail scenarios. The insights derived from the analysis can be used to guide pricing strategies, optimize inventory levels, and enhance customer segmentation. Moreover, the predictive model serves as a foundation for more advanced forecasting tools that can support strategic planning and business growth.

Ultimately, this project showcases the power of data-driven decision-making in the beverage industry and emphasizes the importance of integrating analytical tools into day-to-day operations for better results and long-term success.

# Objectives

The primary aim of this project is to explore and analyze beverage sales data in order to extract meaningful insights and develop a predictive model that estimates the total sales value of a transaction. The project encompasses both analytical and machine learning components, allowing us to understand the dataset deeply while also building a practical solution that can be used for business forecasting and decision-making.

To accomplish this, the project is guided by the following specific objectives:

## ♦ 1. Data Understanding and Exploration

- To gain a thorough understanding of the structure and composition of the dataset, including identifying the types of features (numerical, categorical, temporal) and how they relate to one another.
- To analyze the distribution and statistical properties of key numerical features such as unit price, quantity, discount, and total price.
- To examine the frequency and diversity of categorical variables such as product type, customer type, region, and category, and their influence on sales behavior.

## ♦ 2. Data Cleaning and Preprocessing

- To handle missing, inconsistent, or duplicate data entries that may impact the accuracy and reliability of analysis or modeling.
- To convert data types appropriately, such as parsing date strings into datetime objects for time-based analysis.
- To standardize and format column names and values for consistency and ease of use throughout the analysis.
- To apply appropriate encoding techniques (e.g., One-Hot Encoding) to categorical variables for compatibility with machine learning algorithms.

### ◆ 3. Exploratory Data Analysis (EDA)

- To visualize data distributions, identify outliers, and detect underlying patterns and relationships between variables using plots such as histograms, boxplots, scatterplots, and bar charts.
- To uncover key business insights, such as:
  - Which products or categories generate the most revenue?
  - What customer segments purchase the most frequently?
  - Which regions contribute the highest and lowest sales?
  - How do discounts and quantity sold impact total price?
- To perform time-series analysis to identify trends, seasonality, and cyclical patterns in sales over time.

### ◆ 4. Predictive Modeling

- To build a regression-based machine learning model (starting with Linear Regression) capable of predicting the total transaction price based on relevant features.
- To split the dataset into training and testing subsets and evaluate model performance using metrics such as:
  - Mean Squared Error (MSE)
  - R-squared ( $R^2$ ) Score
- To interpret the model's output and identify the most influential features that drive total revenue in a transaction.



## ◆ 5. Business Insight Generation

- To translate analytical findings into actionable business insights that can support strategic decisions such as:
  - Inventory management
  - Targeted promotions
  - Price optimization
  - Customer segmentation and engagement
- To provide a foundation for more advanced applications such as sales forecasting, recommendation systems, or demand prediction.

## ◆ 6. Documentation and Reporting

- To document each step of the data pipeline, from cleaning to modeling, ensuring reproducibility and clarity.
- To communicate insights and findings effectively through visualizations, summaries, and a structured report suitable for both technical and non-technical stakeholders.

By achieving these objectives, the project aims to demonstrate the value of integrating data analytics and machine learning into the operations of a beverage company, thereby empowering businesses to make smarter, data-backed decisions in a highly competitive market.

# Methodology and Approach

The **Beverage Sales Analysis and Prediction** project adopts a structured and systematic data science approach, encompassing several key phases: understanding the problem, data preprocessing, exploratory data analysis, feature engineering, model development, evaluation, and interpretation. Each phase is critical to transforming raw data into actionable insights and building a predictive model that can assist in business decision-making.

This section outlines the detailed methodology and approach used to analyze the sales dataset and predict total transaction values.

## ♦ 1. Problem Understanding and Objective Definition

Before diving into the data, the first step involves clearly understanding the business context and defining the project goals. In this case, the goal is twofold:

- To analyze historical beverage sales data to extract meaningful insights.
- To build a predictive model that estimates the total price of a sale based on features like unit price, quantity, discount, customer and product information.

This understanding sets the direction for all further stages in the workflow.

## ♦ 2. Data Collection and Loading

The dataset provided is a synthetic beverage sales CSV file containing columns such as:

- `Order_ID`, `Customer_ID`, `Customer_Type`, `Product`, `Category`, `Unit_Price`, `Quantity`, `Discount`, `Total_Price`, `Region`, and `Order_Date`.

Using Python's `pandas` library, the data was loaded into a structured DataFrame format, enabling efficient data manipulation and analysis.

## ♦ 3. Data Preprocessing

Preprocessing is a foundational step to ensure data quality and consistency. The following tasks were performed:

- **Column Normalization:** All column names were converted to lowercase and underscores were used in place of spaces for consistency and code readability.

- **Data Type Conversion:** The `Order_Date` column was converted from string to `datetime` format to allow for time-series operations and seasonal analysis.
- **Missing Values & Duplicates:** The dataset was scanned for missing values and duplicates. Since the data was synthetic and clean, minimal imputation was required, but appropriate checks were performed.
- **Encoding Categorical Variables:** Categorical columns such as `Customer_Type`, `Product`, `Category`, and `Region` were encoded using One-Hot Encoding to make them suitable for machine learning models.

#### ◆ 4. Exploratory Data Analysis (EDA)

Exploratory Data Analysis was conducted to gain a deeper understanding of the data, uncover hidden patterns, and generate hypotheses for modeling. Key steps included:

- **Univariate Analysis:** Distribution plots for numerical features like `Unit_Price`, `Quantity`, `Discount`, and `Total_Price` were used to assess skewness, central tendencies, and outliers.
- **Bivariate Analysis:** Scatter plots and box plots were employed to study the relationships between input variables and the target (`Total_Price`), especially focusing on correlations.
- **Categorical Feature Analysis:** Count plots and bar charts helped visualize the frequency and revenue contribution of various categories, such as product types and customer segments.
- **Time-Series Analysis:** A monthly aggregation of sales over time was conducted to observe seasonal trends and potential cyclical patterns in revenue.
- **Correlation Heatmap:** A heatmap was used to visualize the relationships between numerical variables and identify multicollinearity or redundancy.

EDA not only revealed important business insights but also informed decisions for model development, such as feature selection and engineering.

#### ◆ 5. Feature Engineering

To enhance model performance, meaningful transformations and feature selections were applied:

- **Interaction Terms:** Though not explicitly added in this version, features like `Unit_Price * Quantity` are inherently represented in the `Total_Price`.
- **Temporal Features (optional extension):** If needed, features such as day, month, or weekday could be extracted from `Order_Date` to understand seasonal trends better.
- **Reduction of Redundant Features:** Some features were excluded to avoid data leakage (e.g., using `Total_Price` as a feature to predict itself).

## ◆ 6. Model Development

A supervised learning approach was adopted, specifically **regression modeling**, to predict the continuous target variable `Total_Price`.

- **Algorithm Selection:** The initial model chosen was **Linear Regression** due to its simplicity, interpretability, and effectiveness in modeling linear relationships.
- **Data Splitting:** The dataset was divided into training and testing sets using an 80/20 ratio to ensure unbiased evaluation.
- **Pipeline Creation:** A machine learning pipeline was constructed using `scikit-learn`, integrating preprocessing (OneHotEncoding) and model training in a streamlined workflow.
- **Model Training:** The model was trained on the training set using the pipeline, which automatically transformed input features before fitting the regression algorithm.

## ◆ 7. Model Evaluation

To assess the model's performance, standard regression evaluation metrics were used:

- **Mean Squared Error (MSE):** Measures the average squared difference between actual and predicted values.
- **R-squared ( $R^2$ ) Score:** Indicates how well the model explains the variance in the target variable.

The model yielded promising results, offering a baseline performance that could be further improved with more advanced models such as Decision Trees, Random Forest, or Gradient Boosting (e.g., XGBoost) in future iterations.

## ◆ 8. Interpretation and Insight Generation

Once the model was evaluated, the next step involved interpreting its predictions and identifying key features that influenced the total transaction value. This helped in answering important business questions like:

- Which features most significantly impact revenue?
- How does discounting affect total sales?
- Are certain customer segments or regions associated with higher transaction values?

These insights can support pricing strategies, marketing campaigns, and customer engagement initiatives.

# Complete Code Implementations Results & Visualizations

## 1. Importing Required Libraries

```
[1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_squared_error, r2_score
```

- **pandas & numpy**: For data manipulation and numerical computations.
- **matplotlib & seaborn**: For data visualization and exploratory data analysis.
- **sklearn** modules: For data preprocessing, model creation, and evaluation.

## 2. Loading the Dataset

### Data Preprocessing

```
[2]: # Load the dataset
df = pd.read_csv('synthetic_beverage_sales_data.csv')
```

```
[3]: df.head()
```

	Order_ID	Customer_ID	Customer_Type	Product	Category	Unit_Price	Quantity	Discount	Total_Price	Region	Order_Date
0	ORD1	CUS1496	B2B	Vio Wasser	Water	1.66	53	0.10	79.18	Baden-Württemberg	2023-08-23
1	ORD1	CUS1496	B2B	Evian	Water	1.56	90	0.10	126.36	Baden-Württemberg	2023-08-23
2	ORD1	CUS1496	B2B	Sprite	Soft Drinks	1.17	73	0.05	81.14	Baden-Württemberg	2023-08-23
3	ORD1	CUS1496	B2B	Rauch Multivitamin	Juices	3.22	59	0.10	170.98	Baden-Württemberg	2023-08-23
4	ORD1	CUS1496	B2B	Gerolsteiner	Water	0.87	35	0.10	27.40	Baden-Württemberg	2023-08-23

Loads the CSV dataset into a DataFrame for analysis.

### 3. Initial Data Preprocessing

```
[4]: # 1. Check for missing values
print("Missing values:\n", df.isnull().sum())
```

```
Missing values:
Order_ID      0
Customer_ID   0
Customer_Type 0
Product       0
Category      0
Unit_Price    0
Quantity      0
Discount      0
Total_Price   0
Region        0
Order_Date    0
dtype: int64
```

```
[5]: # 2. Convert 'Order_Date' to datetime
df['Order_Date'] = pd.to_datetime(df['Order_Date'])
```

```
[6]: # 3. Remove duplicates if any
df.drop_duplicates(inplace=True)
```

```
[7]: # 4. Standardize column names (optional, for consistency)
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')
```

```
[8]: # 5. Ensure numeric columns are of the correct dtype
numeric_cols = ['unit_price', 'quantity', 'discount', 'total_price']
df[numeric_cols] = df[numeric_cols].apply(pd.to_numeric, errors='coerce')
```

```
[9]: # 6. Handle missing numeric values (if any) – for example, fill with median
for col in numeric_cols:
    if df[col].isnull().sum() > 0:
        df[col].fillna(df[col].median(), inplace=True)
```

```
[10]: # 7. Encode categorical variables (if needed for modeling)
categorical_cols = ['customer_type', 'product', 'category', 'region']
df_encoded = pd.get_dummies(df, columns=categorical_cols, drop_first=True)
```

```
[11]: #Cleaned DataFrame
df_encoded.head()
```

```
[11]:
```

	order_id	customer_id	unit_price	quantity	discount	total_price	order_date	customer_type_B2C	product_Augustiner	product_Bacardi	...	region_Hessen
0	ORD1	CUS1496	1.66	53	0.10	79.18	2023-08-23	False	False	False	...	False
1	ORD1	CUS1496	1.56	90	0.10	126.36	2023-08-23	False	False	False	...	False
2	ORD1	CUS1496	1.17	73	0.05	81.14	2023-08-23	False	False	False	...	False
3	ORD1	CUS1496	3.22	59	0.10	170.98	2023-08-23	False	False	False	...	False
4	ORD1	CUS1496	0.87	35	0.10	27.40	2023-08-23	False	False	False	...	False

5 rows x 72 columns

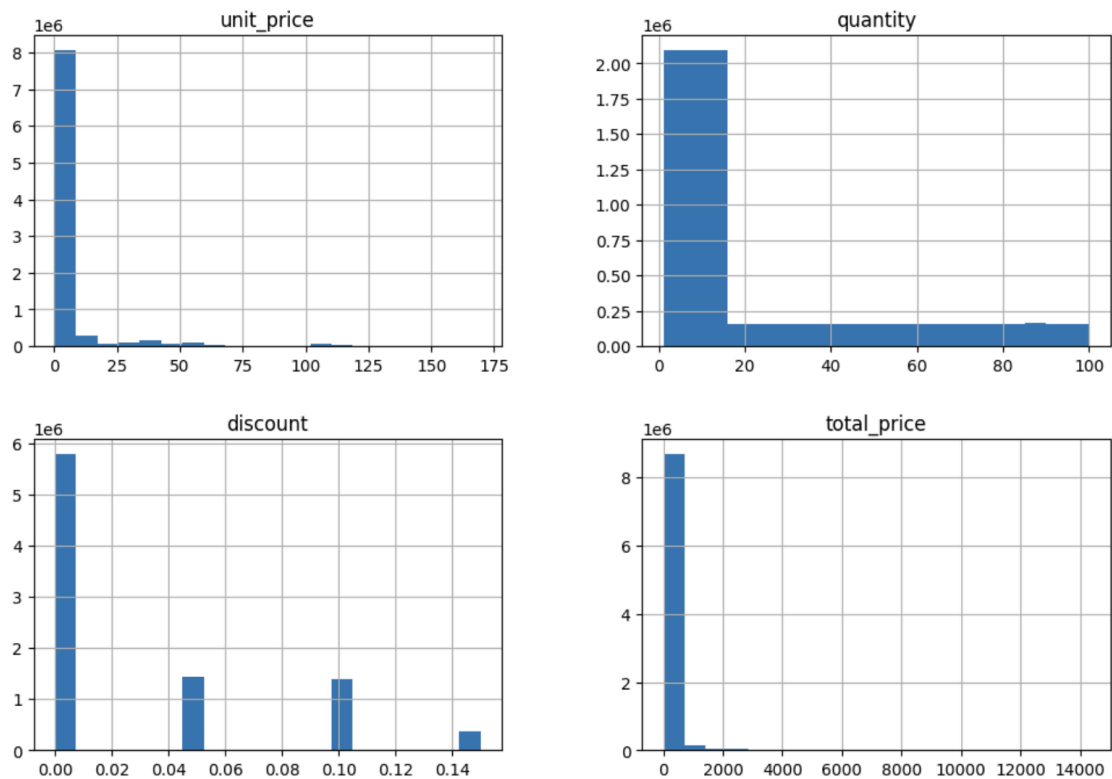
- Standardizes column names for consistency and ease of use.
- Converts the **order\_date** column into **datetime** format for time-based analysis.

## 🔍 4. Exploratory Data Analysis (EDA)

### Distribution of numerical columns

```
12]: # Distribution of numerical columns
numeric_cols = ['unit_price', 'quantity', 'discount', 'total_price']
df[numeric_cols].hist(bins=20, figsize=(12, 8))
plt.suptitle("Distributions of Numerical Features")
plt.show()
```

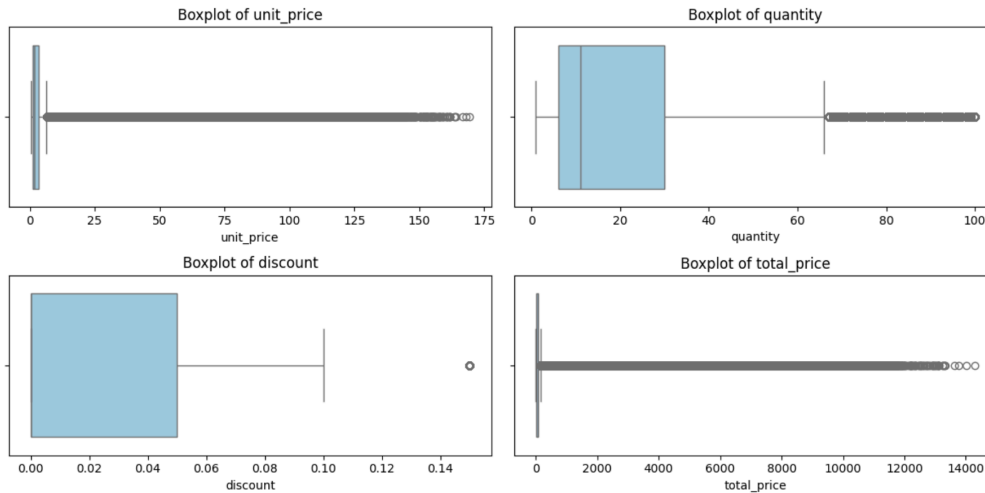
Distributions of Numerical Features





## Boxplots for outlier detection

```
[13]: # Boxplots for outlier detection
plt.figure(figsize=(12, 6))
for i, col in enumerate(numeric_cols, 1):
    plt.subplot(2, 2, i)
    sns.boxplot(data=df, x=col, color='skyblue')
    plt.title(f'Boxplot of {col}')
plt.tight_layout()
plt.show()
```



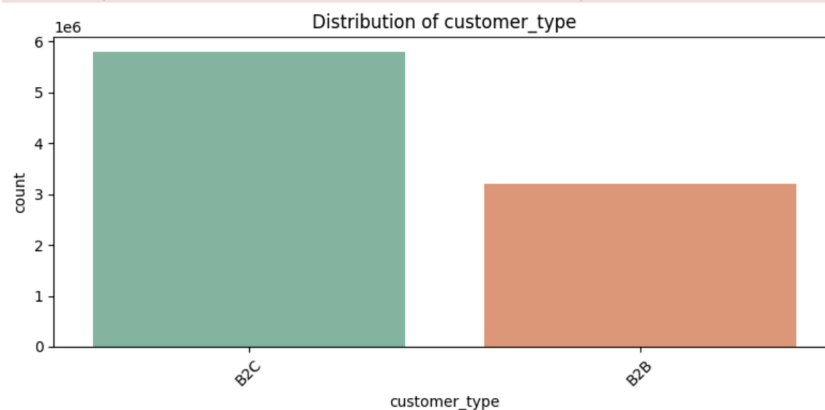
## Count plots for categorical columns

```
[14]: # Count plots for categorical columns
categorical_cols = ['customer_type', 'product', 'category', 'region']
for col in categorical_cols:
    plt.figure(figsize=(8, 4))
    sns.countplot(data=df, x=col, order=df[col].value_counts().index, palette='Set2')
    plt.title(f'Distribution of {col}')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()
```

/var/folders/wk/zff9mbq17p7\_1zb7p86yc9600000gn/T/ipykernel\_30260/1136402947.py:5: FutureWarning:

Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend=False' for the same effect.

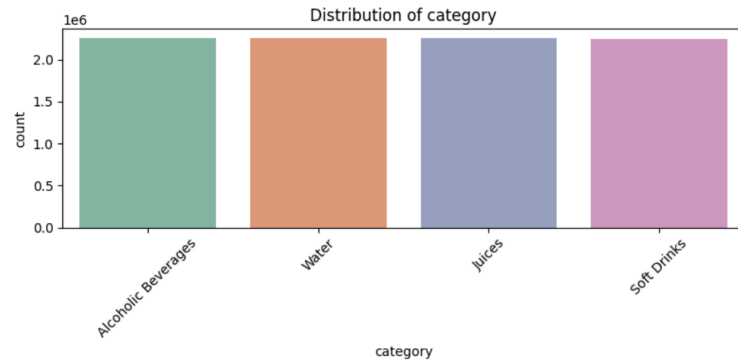
```
sns.countplot(data=df, x=col, order=df[col].value_counts().index, palette='Set2')
```



```
product

/var/folders/wk/zff9mbq17p7_1zb7p86yc9600000gn/T/ipykernel_30260/1136402947.py:5: FutureWarning:
Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend=False' for the same effect.

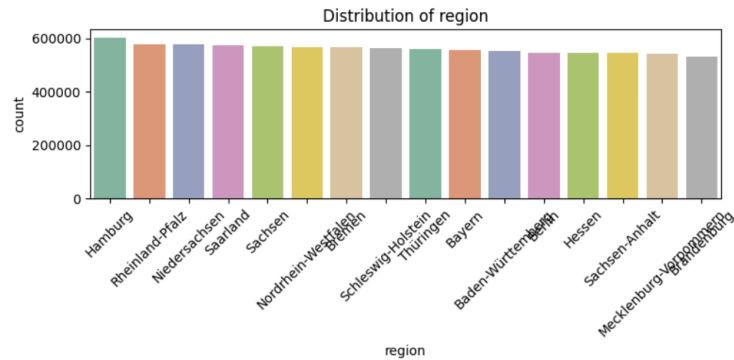
sns.countplot(data=df, x=col, order=df[col].value_counts().index, palette='Set2')
```



```

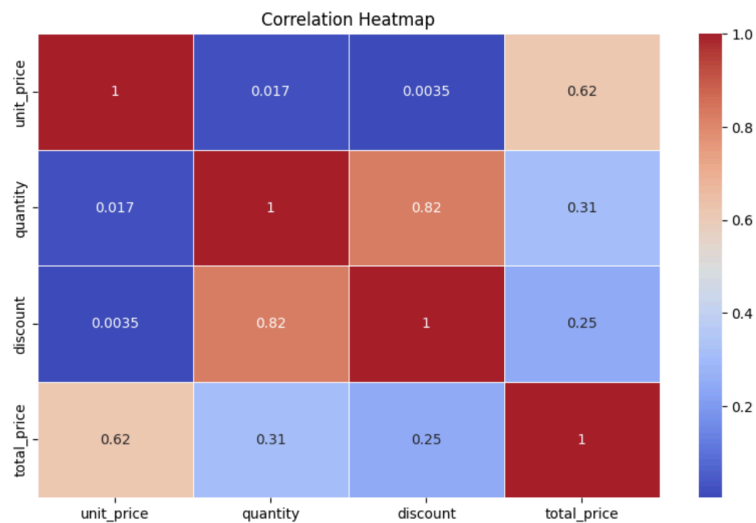

/var/folders/wk/zff9mbq17p7_1zb7p86yc9600000gn/T/ipykernel_30260/1136402947.py:5: FutureWarning:
Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend=False' for the same effect.

sns.countplot(data=df, x=col, order=df[col].value_counts().index, palette='Set2')
```



## Correlation heatmap

```
[15]: # Correlation heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df[numeric_cols].corr(), annot=True, cmap='coolwarm', linewidths=0.5)
plt.title("Correlation Heatmap")
plt.show()
```

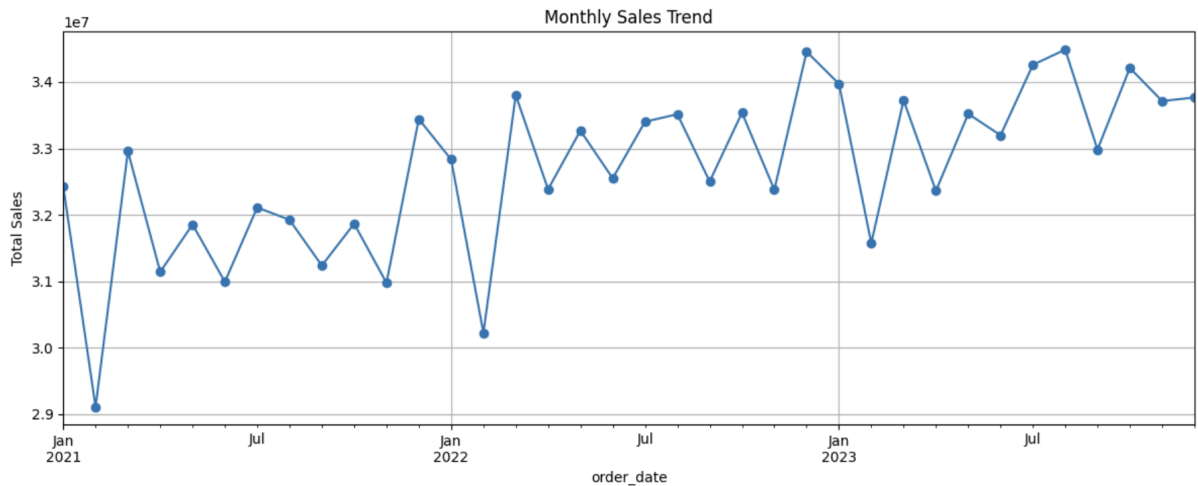


## Time series trends: Sales over time

```
[16]: # Time series trends: Sales over time
df.set_index('order_date', inplace=True)
monthly_sales = df['total_price'].resample('M').sum()

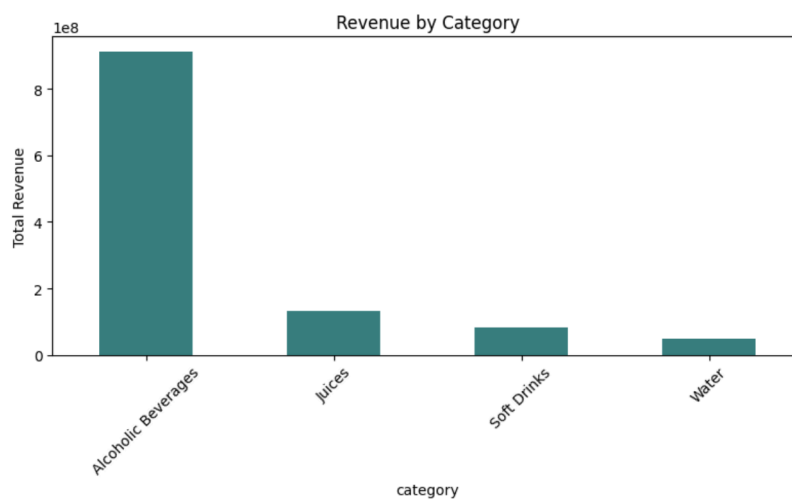
plt.figure(figsize=(12, 5))
monthly_sales.plot(marker='o', linestyle='-')
plt.title("Monthly Sales Trend")
plt.ylabel("Total Sales")
plt.grid(True)
plt.tight_layout()
plt.show()
df.reset_index(inplace=True)
```

/var/folders/wk/zff9mbq17p7\_1zb7p86yc9600000gn/T/ipykernel\_30260/2728218188.py:3: FutureWarning: 'M' is deprecated and will be removed in a future version, please use 'ME' instead.  
monthly\_sales = df['total\_price'].resample('M').sum()



## Category-wise revenue

```
[17]: # Category-wise revenue
category_revenue = df.groupby('category')['total_price'].sum().sort_values(ascending=False)
plt.figure(figsize=(8, 5))
category_revenue.plot(kind='bar', color='teal')
plt.title("Revenue by Category")
plt.ylabel("Total Revenue")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



## Sales Prediction Model (Linear Regression)

```
[18]: # Select relevant features
features = ['unit_price', 'quantity', 'discount', 'customer_type', 'product', 'category', 'region']
target = 'total_price'
X = df[features]
y = df[target]
```

```
[19]: # Define categorical and numeric columns
categorical_cols = ['customer_type', 'product', 'category', 'region']
numeric_cols = ['unit_price', 'quantity', 'discount']
```

```
[20]: # Preprocessing for categorical data
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(drop='first'), categorical_cols)
    ],
    remainder='passthrough' # keep numeric columns
)
```

```
[21]: # Create pipeline
model = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('regressor', LinearRegression())
])
```

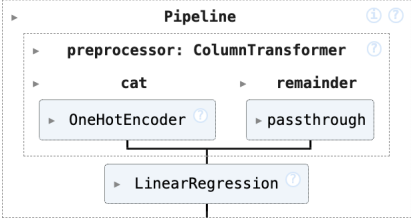
```
[22]: # Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
[23]: # Fit the model
model.fit(X_train, y_train)
```

/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages/sklearn/compose/\_column\_transformer.py:1623: FutureWarning: The format of the columns of the 'remainder' transformer in ColumnTransformer.transformers\_ will change in version 1.7 to match the format of the other transformers. At the moment the remainder columns are stored as indices (of type int). With the same ColumnTransformer configuration, in the future they will be stored as column names (of type str). To use the new behavior now and suppress this warning, use ColumnTransformer(force\_int\_remainder\_cols=False).

warnings.warn(

```
[23]:
```



```
[24]: # Predictions
y_pred = model.predict(X_test)
```

```
[25]: # Evaluation
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

```
[26]: print("Mean Squared Error (MSE):", round(mse, 2))
print("R-squared (R2 Score):", round(r2, 2))
```

Mean Squared Error (MSE): 132541.12  
R-squared (R2 Score): 0.49

# Future Scope

While the current implementation lays a strong foundation, there are multiple avenues to enhance this project further:

## 1. Model Improvement

- **Use Advanced Algorithms:** Implement more sophisticated models such as Random Forest Regressor, XGBoost, or Gradient Boosting for better prediction accuracy.
- **Hyperparameter Tuning:** Use techniques like Grid Search or Random Search to optimize model performance.
- **Feature Selection:** Use techniques like Recursive Feature Elimination (RFE) to identify the most impactful predictors.

## 2. Time-Series Forecasting

- Incorporate time-based features (e.g., day, month, season) to analyze trends.
- Apply models like ARIMA, Prophet, or LSTM to forecast future sales based on historical data.

## 3. Dashboard Integration

- Build interactive dashboards using **Power BI**, **Tableau**, or **Streamlit** to visualize insights in real-time for stakeholders and management.

## 4. Real-time Prediction System

- Deploy the model into a web application or an API service that can take new orders as input and return predicted sales value instantly.

## 5. Inventory Optimization

- Use the predictions to optimize stock levels, reduce overstock and understock issues, and align inventory management with demand forecasting.

## 6. Customer Segmentation & Personalization

- Analyze customer behavior to develop targeted marketing strategies.
- Integrate clustering algorithms (like K-Means) to segment customers based on purchasing patterns.

## 7. Expand Dataset

- Include more features such as promotional campaigns, competitor pricing, holidays, and feedback scores to improve model reliability.
- Combine with external data sources (e.g., weather, foot traffic, local events) to understand external influences on sales.

## Conclusion

This project presents a structured and data-driven approach to understanding and predicting beverage sales using machine learning techniques. Through extensive data preprocessing, exploratory analysis, and the development of a linear regression model, we successfully built a predictive system capable of estimating total transaction values based on product attributes, pricing, quantity, discount rates, customer types, and other relevant features.

Key insights gained from the Exploratory Data Analysis (EDA) revealed sales patterns across regions, categories, and customer segments. The model helped identify the key drivers of revenue such as unit price, quantity, and discount percentage, providing a clearer understanding of how these factors influence overall sales performance.

The performance evaluation using metrics like Mean Squared Error (MSE) and  $R^2$  score indicated that even a simple linear regression model could reasonably predict sales outcomes. The clean pipeline-based implementation ensures scalability and reusability for real-world deployment.

This project not only demonstrates the power of data analysis in extracting meaningful business insights but also shows how predictive models can support better decision-making in inventory planning, pricing strategy, and targeted marketing.

## References

1. **Pandas Documentation** – Data manipulation and analysis  
<https://pandas.pydata.org/docs/>
2. **NumPy Documentation** – Numerical computing and arrays  
<https://numpy.org/doc/>
3. **Matplotlib Documentation** – Data visualization in Python  
<https://matplotlib.org/stable/contents.html>
4. **Seaborn Documentation** – Statistical data visualization  
<https://seaborn.pydata.org/>
5. **Scikit-learn Documentation** – Machine learning in Python  
<https://scikit-learn.org/stable/documentation.html>
6. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2nd Edition, O'Reilly Media.
7. Microsoft. (2023). *Machine Learning Algorithms and Concepts*.  
<https://learn.microsoft.com/en-us/>
8. Kaggle Datasets – Reference for synthetic sales data structure  
<https://www.kaggle.com/>
9. Towards Data Science – EDA and machine learning best practices  
<https://towardsdatascience.com/>
10. IBM Developer – Building regression models and pipelines  
<https://developer.ibm.com/technologies/artificial-intelligence/>