**Data Analysis and Visualization of Top Selling Products on Blinkit Using Python**

**Declaration**

We hereby declare that the project report titled **"Data Analysis and Visualization of Top Selling Products on Blinkit Using Python"** submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** is our original work and has not been submitted previously for the award of any degree or diploma elsewhere.

This work was carried out under the guidance and supervision of our project mentor and represents the results of our own investigations, unless otherwise stated. Proper acknowledgements have been made wherever work of others has been cited.

**Certificate**

This is to certify that the project entitled **"Data Analysis and Visualization of Top Selling Products on Blinkit Using Python"** has been carried out by:

* **Vikas Kashyap** (22030218)
* **Mohd Rahman** (22030546)
* **Shivam** (22030126)
* **Naveen** (22030409)

under my supervision in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** from **Quantum School of Technology, Roorkee**.

The work embodied in this report is original and has been carried out with diligence, accuracy, and academic integrity. It has not been submitted elsewhere for any other degree or diploma.

**Supervisor:**  
[Name of Project Guide]  
Department of Computer Science and Engineering  
Quantum School of Technology, Roorkee

**Acknowledgement**

We would like to express our heartfelt gratitude to our project guide **[Name of Project Guide]** for their invaluable guidance, continuous support, and encouragement throughout the course of this project. Their expert advice and constructive feedback at every stage made this work possible and helped us grow both technically and academically.

We are also grateful to the faculty members of the **Department of Computer Science and Engineering, Quantum School of Technology, Roorkee** for their academic support and insights.

Special thanks to our families and friends for their unwavering support, understanding, and motivation throughout this academic journey. Lastly, we thank the administration and infrastructure team of our institution for providing all the necessary facilities to carry out our project smoothly.

**Abstract**

This project explores the **analysis and visualization of top-selling products on Blinkit**, a leading quick-commerce grocery delivery platform in India. With the ever-increasing demand for fast deliveries and online shopping, platforms like Blinkit generate massive volumes of data on customer behavior, product ratings, pricing, and category-wise performance.

The objective of this project is to transform this raw product data into meaningful insights using **Python** and data science tools. Key technologies employed include **Pandas** for data manipulation, **Plotly** for creating interactive visualizations, and **Streamlit** for deploying a web-based dashboard. The application allows users to filter products by category, view rankings based on rating counts (used as a proxy for sales), and compare product performance through bar charts, pie charts, and scatter plots.

The dashboard is intended to empower **customers** by helping them make informed buying decisions and support **business stakeholders** with data-driven strategies for product promotion, inventory management, and customer engagement.

This report presents the motivation, development methodology, tools used, implementation process, results, and potential future enhancements of the system.

**Chapter 1 – Introduction**

**1.1 Background**

In the age of rapid digital transformation, e-commerce and quick-commerce platforms have become integral to modern retail. Among them, **Blinkit** has emerged as a popular grocery delivery application in India, offering customers a wide selection of products with delivery times as fast as 10 minutes. With thousands of transactions taking place daily, Blinkit generates vast amounts of structured and unstructured data — including product listings, user reviews, ratings, pricing information, and purchasing behavior.

This large data repository holds valuable insights that can be used to identify top-performing products, understand consumer preferences, and optimize operations. However, the real challenge lies in transforming this raw data into actionable information that can aid both **customers** in making informed decisions and **businesses** in crafting effective sales strategies.

**1.2 Motivation**

Customers on platforms like Blinkit often face an overwhelming number of choices. Without clear indicators of popularity, reliability, or value, they may struggle to select the best products. At the same time, companies require better tools to monitor real-time trends, track inventory performance, and plan promotional campaigns. Thus, the motivation for this project stems from the need to bridge the gap between raw data and practical decision-making by building an intelligent, **interactive dashboard** that makes product trends easy to understand.

**1.3 Purpose of the Project**

This project aims to build a **Streamlit-based web dashboard** using **Python** to visualize Blinkit's top-selling products. The application will use data attributes like **rating count**, **average rating**, **discounted price**, and **category** to identify and display the best-performing items. By offering visual insights such as bar graphs, pie charts, and scatter plots, the dashboard becomes a useful tool for:

* Customers to explore trending products
* Blinkit to understand product demand and optimize inventory
* Analysts and developers to study consumer behavior

**1.4 Problem Statement**

Despite the richness of Blinkit’s product data, there is no publicly accessible dashboard that allows real-time exploration of product performance based on user feedback and rating behavior. Traditional reports and analytics tools are either too complex, inaccessible to end-users, or not dynamic. The challenge, therefore, is to design and implement a solution that:

* Uses Python and open-source tools
* Cleans and processes Blinkit's dataset
* Visualizes insights interactively
* Empowers both customers and business stakeholders

**1.5 Scope of the Project**

The scope of this project includes:

* Collecting or simulating product data similar to Blinkit’s listings.
* Performing data cleaning and transformation using **Pandas**.
* Visualizing data with **Plotly** and deploying it via **Streamlit**.
* Ranking products by **rating count** (proxy for sales).
* Providing real-time, interactive filters based on product **category**.

It does not cover:

* Live API integration (currently uses CSV data)
* Predictive modeling (planned for future enhancements)
* Full-scale user authentication or personalized dashboards

**1.6 Significance of the Study**

This project demonstrates how open-source tools can be used to develop scalable data applications in the retail domain. The dashboard developed in this project has dual benefits:

* **For customers:** It increases transparency, supports informed purchases, and enhances trust.
* **For businesses:** It helps monitor consumer trends, identify best-selling products, and plan marketing strategies based on actual data.

Moreover, this project serves as a case study for students and professionals interested in data science, full-stack analytics development, and real-world dashboarding using Python.

Here is the full content for:

## ****Chapter 2 – Problem Analysis & Literature Review****

### ****2.1 Introduction to the Problem Domain****

The rise of e-commerce and quick-commerce platforms has significantly altered consumer purchasing behavior and business operations. Platforms like Blinkit handle enormous volumes of user interactions and transactions daily, generating data with immense analytical potential. However, the problem lies in effectively utilizing this raw data to draw meaningful conclusions that serve both consumers and the business. Without structured analysis and visualization, these data points remain untapped.

### ****2.2 The Analytical Gap****

Even though companies collect transactional and behavioral data, many do not implement tools that allow customers to see which products are trending or highly rated in real time. From the business side, decisions regarding promotions, inventory planning, or pricing strategies often rely on outdated or manually compiled reports. This project addresses this analytical gap by automating data analysis and visualizing product performance using Python-based tools.

### ****2.3 Statement of the Problem****

There is currently no lightweight, interactive, and publicly accessible dashboard that allows users to explore Blinkit’s top-selling and top-rated products based on real-time or regularly updated data. The specific problem addressed is:

“How can Python and open-source data science tools be used to analyze and visualize large-scale product data from Blinkit in an interactive and insightful way?”

This project proposes a solution by combining efficient data processing, graphical representation, and web deployment using Python libraries.

### ****2.4 Literature Review****

#### 2.4.1 Importance of Data Analytics in E-commerce

Numerous studies have demonstrated how data-driven strategies improve sales performance and customer satisfaction in the retail industry. E-commerce platforms benefit immensely from identifying patterns in user behavior, seasonal trends, and product feedback. According to a report by McKinsey & Company, companies using advanced analytics improve their profitability by up to 20%.

#### 2.4.2 Role of Data Visualization

Data visualization plays a vital role in making complex datasets understandable and accessible. Research shows that visual content leads to better pattern recognition and decision-making, especially in environments where users are not technically trained.

#### 2.4.3 Use of Python in Data Science

Python has emerged as the leading language in data science and analytics due to its readability and extensive ecosystem of libraries. Pandas enables powerful data manipulation, while Plotly offers interactive visualizations suitable for both exploratory analysis and final dashboards. Streamlit, a newer web app framework, simplifies the deployment of data-driven interfaces without requiring deep frontend knowledge.

#### 2.4.4 Rating Count as a Sales Proxy

In the absence of explicit sales figures, several academic and industry studies have used “rating count” as a proxy metric to estimate product popularity or sales volume. This metric reflects both user engagement and product reach.

### ****2.5 Comparative Analysis with Existing Solutions****

| **Feature** | **Traditional BI Tools** | **Static Reports** | **Streamlit Dashboard (Proposed)** |
| --- | --- | --- | --- |
| Real-time filtering | Limited | No | Yes |
| Ease of use | Requires training | Easy | Easy |
| Cost | Expensive | Low | Free/Open-source |
| Customization | Medium | Low | High |
| Suitable for customers | No | No | Yes |

This comparison demonstrates that the proposed solution offers an accessible, cost-effective, and interactive approach, making it a practical alternative for both internal and external stakeholders.

### ****2.6 Summary****

The reviewed literature and problem analysis confirm the need for a scalable and interactive data analytics solution tailored to Blinkit's product data. By leveraging Python libraries and data visualization techniques, this project aims to build a real-time dashboard that serves both customer needs and business strategy goals.

Here is the full content for:

## ****Chapter 3 – Objectives****

### ****3.1 Overall Objective****

The overarching goal of this project is to utilize Python programming and open-source data science tools to develop an interactive dashboard that visualizes the top-selling products on Blinkit. The dashboard aims to make product trends transparent, insightful, and easily accessible to both customers and internal business users.

By integrating data processing, visualization, and a user interface into a unified platform, the project demonstrates how real-time insights can enhance customer decision-making and support strategic business planning.

### ****3.2 Specific Objectives****

To achieve the overall goal, the project defines the following specific objectives:

#### **1. Data Collection and Preparation**

* Load product-level data (from a CSV file or simulated dataset) containing attributes such as product ID, name, category, rating, rating count, price, and reviews.
* Clean and preprocess the dataset using **Pandas** by handling missing values, converting data types, and removing inconsistencies.

#### **2. Data Analysis**

* Use **rating count** as a proxy for sales volume due to the unavailability of direct sales data.
* Determine top-selling products across all categories based on rating count.
* Identify the highest-rated products based on average user ratings.
* Group and summarize data for category-wise analysis.

#### **3. Visualization**

* Develop visualizations using **Plotly** to:
  + Display bar charts of top 10 best-selling products.
  + Highlight top 10 highest-rated products.
  + Show category-wise sales distribution in pie charts.
  + Illustrate price vs rating count using scatter plots.

#### **4. Dashboard Development**

* Build an interactive web-based dashboard using **Streamlit** to display visualizations and allow user interaction.
* Enable filtering by product category through a sidebar interface.
* Display datasets and plots clearly and responsively on any device.

#### **5. User Empowerment**

* Provide customers with transparent access to top products based on real user data.
* Enable better decision-making through visual ranking and comparisons.

#### **6. Business Insight and Operational Use**

* Allow internal teams to:
  + Monitor product performance.
  + Analyze which categories are performing best.
  + Identify trends that can guide restocking or promotional campaigns.

#### **7. Future-Ready Framework**

* Design the system architecture with scalability in mind to:
  + Add real-time API data integrations.
  + Extend functionality for predictive analytics and user personalization.

### ****3.3 Summary****

The project’s objectives align with modern e-commerce goals—enhancing user experience through transparency, and improving business operations through data-driven insights. By focusing on practical implementation and customer-centric features, this project stands as a useful tool for both market analysis and real-world application.

Here is the complete content for:

## ****Chapter 4 – Tools and Technologies Used****

### ****4.1 Introduction****

To analyze and visualize product data from Blinkit, this project employs a collection of modern, open-source tools and libraries that are widely used in the fields of data science, web development, and analytics. These tools are selected based on criteria such as functionality, ease of use, extensibility, community support, and their ability to deliver real-time, interactive experiences.

The core stack includes **Python** as the primary programming language, along with supporting libraries and frameworks for data processing, visualization, and dashboard deployment.

### ****4.2 Programming Language****

#### **Python**

Python is the central programming language used in this project due to its:

* Simplicity and readability
* Extensive ecosystem for data analysis
* Rich community support and documentation
* Integration with web and data frameworks

Python serves as the backbone for all major stages of the project including data cleaning, transformation, visualization, and dashboard creation.

### ****4.3 Libraries and Frameworks****

#### **Pandas**

* Used for data manipulation and analysis.
* Handles CSV file reading, missing value treatment, type conversion, and aggregation.
* Enables filtering and grouping of data based on attributes such as category, rating\_count, and price.

#### **Plotly**

* An interactive graphing library used to create dynamic visualizations such as:
  + Bar charts for top-selling and highest-rated products
  + Pie charts for category-wise distribution
  + Scatter plots for price vs. rating analysis
* Provides better interactivity and visual appeal compared to static plotting libraries like Matplotlib.

#### **Streamlit**

* A lightweight Python framework for building web-based data applications.
* Allows seamless integration of Pandas and Plotly to create an intuitive dashboard.
* Supports widgets (e.g., multiselect filters), automatic reloading, and deployment features with minimal code.

### ****4.4 Data Storage and Handling****

#### **CSV Files**

* Used as the input format for Blinkit's product data.
* Preferred for its simplicity, portability, and compatibility with most data science tools.

### ****4.5 Web Interface and Styling (Optional for Flask Expansion)****

#### **HTML / CSS**

* If Flask were used instead of Streamlit, HTML and CSS would support UI customization and styling.
* With Streamlit, limited use of markdown and in-built design controls are sufficient for a clean UI.

### ****4.6 Version Control and Collaboration****

#### **Git & GitHub**

* Used for source code management, documentation, and collaboration among team members.
* Enables tracking of changes, versioning, and backup.

### ****4.7 Development Environment****

* **Jupyter Notebook** (for initial analysis and prototyping)
* **VS Code / PyCharm** (for writing, testing, and debugging Python code)
* **Streamlit Cloud or Local Server** (for deploying the final dashboard)

### ****4.8 Summary of Tools and Their Roles****

| **Tool / Library** | **Purpose** |
| --- | --- |
| Python | Core programming language |
| Pandas | Data cleaning and manipulation |
| Plotly | Interactive data visualizations |
| Streamlit | Web dashboard development |
| CSV | Data storage and input format |
| Git & GitHub | Version control and collaboration |
| HTML/CSS (Optional) | UI design (in case of Flask use) |

### ****4.9 Conclusion****

The technologies used in this project are lightweight, scalable, and well-supported in the data science and web development community. By combining the strengths of Python, Pandas, Plotly, and Streamlit, this project delivers a robust, user-friendly solution for visualizing product performance on Blinkit.

**Chapter 5 – Data Description**

**5.1 Introduction to the Dataset**

The dataset used in this project represents product-level information from the Blinkit platform. It includes data points that describe product identity, categorization, pricing, discounting, customer reviews, ratings, and other related attributes. This structured dataset forms the basis of the analysis, and its quality directly influences the insights derived through the dashboard.

The dataset is stored in **CSV (Comma Separated Values)** format, which is widely supported and easy to read using Python's data libraries.

**5.2 Source of the Dataset**

Since Blinkit does not provide an open public API for direct data access, the dataset used in this project is:

* Either a real dataset collected from Blinkit through web scraping (if applicable)
* Or a **simulated dataset** generated to mimic the structure, fields, and scale of real-world Blinkit data

This ensures the ability to prototype and demonstrate the dashboard while preserving the integrity and usability of the data.

**5.3 Key Attributes in the Dataset**

The dataset contains the following important columns:

| **Field Name** | **Description** |
| --- | --- |
| product\_id | Unique identifier for each product |
| product\_name | Name of the product as listed on Blinkit |
| category | Product category (e.g., Snacks, Beverages, Dairy) |
| discounted\_price | Final price of the product after applying any discounts |
| actual\_price | Original listed price of the product before discount |
| discount\_percentage | Percentage discount applied to the product |
| rating | Average customer rating (out of 5) |
| rating\_count | Number of ratings submitted for the product (used as a proxy for sales volume) |
| about\_product | A brief description or highlights of the product features |
| user\_id | Identifier of the user who left a review |
| user\_name | Name of the reviewer |
| review\_id | Unique ID for each product review |
| review\_title | Title or subject of the user review |
| review\_content | Full content of the product review |
| img\_link | URL link to the product image |
| product\_link | URL to the actual Blinkit product page |

**5.4 Data Characteristics**

* **Type**: Structured tabular data
* **Format**: CSV
* **Volume**: Can range from hundreds to thousands of product entries depending on data source
* **Missing Values**: Handled using appropriate cleaning strategies (dropping rows or filling with defaults)
* **Data Types**: Mix of strings, integers, and floats

**5.5 Sample Data Preview**

A sample row in the dataset might look like the following:

product\_id: 10012

product\_name: Amul Milk 1L

category: Dairy

discounted\_price: 56

actual\_price: 60

discount\_percentage: 6.67

rating: 4.5

rating\_count: 892

about\_product: Full cream milk, high in protein

user\_id: U1005

user\_name: Rohan

review\_id: R54545

review\_title: Good quality

review\_content: Very fresh and consistent in quality.

img\_link: https://blinkit.com/images/amulmilk1l.jpg

product\_link: https://blinkit.com/p/amul-milk-1l

**5.6 Data Cleaning Steps**

The following operations are performed on the dataset before analysis:

* Dropping rows with null values in key fields like rating\_count, product\_name, and category
* Converting text-based numbers (e.g., "4.5") to float
* Filling missing ratings with 0 (if not available)
* Filtering out products with zero rating count to ensure meaningful insights

**5.7 Role of the Dataset in the Project**

This dataset is central to the functionality of the dashboard. It is used to:

* Rank products by sales proxy (rating\_count)
* Identify high-quality products (rating)
* Visualize trends across categories (category)
* Explore price-performance relationships (discounted\_price vs rating\_count)

**5.8 Summary**

The dataset provides a comprehensive view of Blinkit’s product landscape from both business and customer perspectives. By using relevant attributes and handling missing or noisy data, the dataset becomes a powerful tool for deriving insights and building an effective dashboard application.

Here is the complete content for:

## ****Chapter 6 – Methodology****

### ****6.1 Introduction****

The development of the Blinkit Top-Selling Products Dashboard follows a modular and iterative methodology. Each phase—ranging from data acquisition to deployment—is designed to ensure the accuracy, usability, and functionality of the final product. This chapter explains the step-by-step process used to build the system, including the data pipeline, analytical methods, visualization techniques, and dashboard development approach.

### ****6.2 Project Development Life Cycle****

The project was developed using an iterative approach similar to the **Data Science Life Cycle**, comprising the following major phases:

1. **Data Acquisition and Import**
2. **Data Cleaning and Preprocessing**
3. **Exploratory Data Analysis (EDA)**
4. **Data Visualization**
5. **Dashboard Development**
6. **Testing and Evaluation**
7. **Deployment**

### ****6.3 Step-by-Step Methodology****

#### **1. Data Acquisition**

* The dataset (real or simulated) was loaded from a CSV file using pandas.read\_csv().
* Ensured all essential columns (e.g., product name, rating count, rating, price) were present.

#### **2. Data Cleaning**

* Dropped records with missing values in essential fields (product\_id, rating\_count, category).
* Converted non-numeric fields like discounted\_price, rating\_count, and rating to appropriate numeric types using pd.to\_numeric().
* Filled missing numerical values with zeros or calculated averages where appropriate.

#### **3. Exploratory Data Analysis**

* Explored data distributions using describe() and value counts.
* Verified data ranges (e.g., rating between 0 and 5, rating\_count non-negative).
* Identified outliers and anomalies (e.g., unusually high or low ratings).

#### **4. Data Aggregation and Ranking**

* Grouped data by product\_name and category.
* Sorted products based on rating\_count to determine top-selling items.
* Filtered for top 10, 40, or 70 products as required.

#### **5. Data Visualization**

Visualizations were created using **Plotly** for interactive charts:

* **Bar Charts**: To rank top-selling and top-rated products.
* **Pie Charts**: To show category-wise sales distributions.
* **Scatter Plots**: To compare discounted prices with rating counts.

Each chart was customized with titles, color-coded categories, and tooltips for enhanced readability and interactivity.

#### **6. Dashboard Development**

* Built using **Streamlit**, allowing for rapid development and deployment.
* Included sidebar filters for category selection using st.sidebar.multiselect().
* Organized visual sections into:
  + Dataset preview
  + Top-selling products
  + Highest-rated products
  + Category-wise sales
  + Price vs rating count

#### **7. Testing and Validation**

* Verified dashboard functionality on various screen sizes.
* Checked filter logic and responsiveness of interactive components.
* Confirmed data accuracy in rankings and charts.

#### **8. Deployment**

* Streamlit app was deployed locally and optionally hosted on **Streamlit Cloud**.
* Code versioned via **GitHub** for tracking and collaboration.

### ****6.4 Tools Used During Methodology****

| **Phase** | **Tool / Library** |
| --- | --- |
| Data Import & Cleaning | Pandas |
| Visualization | Plotly |
| Web Interface | Streamlit |
| Development Environment | VS Code, Jupyter |
| Deployment | Streamlit Cloud |

### ****6.5 Flowchart of the Methodology****

Start

↓

Import Dataset (CSV)

↓

Data Cleaning and Preprocessing (Pandas)

↓

Exploratory Data Analysis (EDA)

↓

Rank Products by Rating Count

↓

Create Visualizations (Plotly)

↓

Build Streamlit Dashboard

↓

Enable Filters and User Interaction

↓

Test Functionality and Outputs

↓

Deploy Web Application

↓

End

### ****6.6 Summary****

This methodology ensures a reliable and scalable solution for analyzing and presenting product data. By following a structured workflow using open-source tools, the project efficiently transforms raw data into an interactive dashboard that delivers meaningful insights to end users.

Here is the complete content for:

## ****Chapter 7 – Implementation****

### ****7.1 Introduction****

This chapter details the practical implementation of the Blinkit Top-Selling Products Dashboard. It describes the logic behind the application, the structure of the code, the libraries used, and how each component—from data loading to visual output—was developed. It also includes screenshots and code snippets to illustrate the workflow.

The implementation is done entirely in **Python**, using **Pandas** for data handling, **Plotly** for visualization, and **Streamlit** for building and deploying the web application.

### ****7.2 Project Architecture****

The application consists of the following major components:

* **Data Loader Module**  
  Loads the dataset and caches it using Streamlit’s caching mechanism.
* **Data Cleaning Block**  
  Ensures the dataset is free of missing values and has the correct data types.
* **Visualization Engine**  
  Generates dynamic charts and graphs using Plotly.
* **Dashboard Interface**  
  Built using Streamlit widgets to enable user interaction (e.g., category filters).

### ****7.3 Streamlit App Structure****

Below is a breakdown of the app’s structure and logic as implemented in the main.py script:

import pandas as pd

import streamlit as st

import plotly.express as px

# Load and cache the dataset

@st.cache\_data

def load\_data():

df = pd.read\_csv("blinkit\_data.csv")

return df

df = load\_data()

### ****7.4 Data Cleaning and Preparation****

# Drop rows with missing values in key fields

df.dropna(subset=[

'product\_id', 'product\_name', 'category', 'discounted\_price',

'actual\_price', 'discount\_percentage', 'rating', 'rating\_count',

'about\_product', 'user\_id', 'user\_name', 'review\_id',

'review\_title', 'review\_content', 'img\_link', 'product\_link'

], inplace=True)

# Convert fields to numeric types

df['rating\_count'] = pd.to\_numeric(df['rating\_count'], errors='coerce').fillna(0)

df['rating'] = pd.to\_numeric(df['rating'], errors='coerce').fillna(0)

df['discounted\_price'] = pd.to\_numeric(df['discounted\_price'], errors='coerce').fillna(0)

df['actual\_price'] = pd.to\_numeric(df['actual\_price'], errors='coerce').fillna(0)

### ****7.5 Interactive Dashboard Layout****

The dashboard includes multiple sections:

#### 1. **Dataset Preview**

st.subheader("📊 Dataset Preview")

st.dataframe(df.head())

#### 2. **Sidebar Filter for Category**

st.sidebar.header("🔍 Filter Products")

categories = df['category'].dropna().unique().tolist()

selected\_categories = st.sidebar.multiselect("Select Categories", categories, default=categories)

filtered\_df = df[df['category'].isin(selected\_categories)]

#### 3. **Top Selling Products**

st.subheader("🏆 Top Selling Products")

top\_sales = df.sort\_values(by='rating\_count', ascending=False).head(10)

fig1 = px.bar(top\_sales, x='rating\_count', y='product\_name',

color='category', orientation='h')

st.plotly\_chart(fig1, use\_container\_width=True)

#### 4. **Top Rated Products**

st.subheader("🌟 Highest Rated Products")

top\_rated = df.sort\_values(by='rating', ascending=False).head(10)

fig2 = px.bar(top\_rated, x='rating', y='product\_name',

color='category', orientation='h')

st.plotly\_chart(fig2, use\_container\_width=True)

#### 5. **Category-wise Sales**

st.subheader("📦 Category-wise Sales Distribution")

category\_sales = filtered\_df.groupby('category')['rating\_count'].sum().reset\_index()

fig3 = px.pie(category\_sales, names='category', values='rating\_count')

st.plotly\_chart(fig3, use\_container\_width=True)

#### 6. **Price vs Rating Scatter Plot**

st.subheader("💰 Price vs Rating Count")

fig4 = px.scatter(filtered\_df, x='discounted\_price', y='rating\_count',

color='category', hover\_data=['product\_name', 'rating'])

st.plotly\_chart(fig4, use\_container\_width=True)

### ****7.6 Deployment****

The application was deployed using:

* **Streamlit CLI** (streamlit run main.py)
* **Streamlit Cloud** (for free hosting and public access)

### ****7.7 Screenshots****

(Note: In a Word/PDF version, this section includes screenshots of the dashboard interface, showing each section like filter panel, bar charts, pie charts, etc.)

### ****7.8 Summary****

The implementation successfully integrates data handling, visual representation, and interactive dashboard design. With a few hundred lines of clean and modular code, the app turns Blinkit's product data into a powerful analysis tool that is both intuitive and insightful.

Here is the complete content for:

## ****Chapter 8 – Results and Discussion****

### ****8.1 Introduction****

This chapter presents the actual outcomes of the project implementation. It highlights the functionality of the dashboard, the insights obtained through the visualizations, and how these outcomes meet the project’s original objectives. The results are analyzed in terms of usability, effectiveness, and value to end-users and business stakeholders.

### ****8.2 Dashboard Overview****

The developed Streamlit dashboard allows users to:

* **Interactively filter** products by category.
* **Visualize** top-selling products using rating count as a proxy for sales.
* **Compare** the highest-rated products based on customer feedback.
* **Analyze** product category performance and sales distribution.
* **Explore** the relationship between price and customer interaction.

Each feature was tested for responsiveness, data accuracy, and user-friendliness.

### ****8.3 Key Results****

#### **1. Top-Selling Products Identified**

The dashboard successfully ranked products based on rating counts. These products are likely high-performing in terms of sales. Examples include:

* Amul Milk 1L
* Maggi 2-Minute Noodles
* Coca-Cola 750ml

These items consistently appeared in the top 10 based on the simulated dataset.

#### **2. Top-Rated Products Displayed**

Using average customer ratings, the application highlighted products with high quality as perceived by users. The dashboard allowed users to see the top 10 rated items with visual emphasis on product names and categories.

#### **3. Category-Wise Distribution**

A pie chart displayed sales concentration among categories. For example:

* Beverages and Snacks accounted for a larger share of rating-based activity.
* Categories like Personal Care had fewer interactions, offering business opportunities for focused marketing.

#### **4. Price vs. Engagement Correlation**

The scatter plot helped visualize the relationship between discounted price and rating count. It showed that:

* Highly rated products were not always the most expensive.
* Certain low-cost items received high user engagement, proving the value of affordability.

### ****8.4 User Experience and Interface Quality****

* **Responsiveness:** The dashboard worked well across devices and adjusted layout components dynamically.
* **Ease of Use:** The sidebar filter and minimalistic layout made it easy to use without prior training.
* **Interactivity:** Charts responded instantly to changes in category filters.

### ****8.5 Alignment with Project Objectives****

| **Objective** | **Achieved?** | **Explanation** |
| --- | --- | --- |
| Build an interactive dashboard | ✔️ | Implemented with Streamlit |
| Rank products by rating count and average rating | ✔️ | Sorted using Pandas and visualized via Plotly |
| Allow filtering by category | ✔️ | Streamlit sidebar filter enabled |
| Provide clean dataset and real-time insights | ✔️ | Preprocessed with Pandas, insights delivered live |
| Offer business and customer utility | ✔️ | Transparent view of product trends |

### ****8.6 Discussion of Limitations****

* **Static Data:** The current dashboard relies on a pre-loaded CSV file. Integration with Blinkit’s live API would enhance real-time accuracy.
* **Sales Proxy:** Rating count is used as a proxy for sales. While it is a valid assumption, direct access to transactional sales data would improve analysis.
* **Scalability:** As the dataset grows, rendering performance may decrease without optimization.

### ****8.7 Summary****

The dashboard achieved its core objectives by delivering a simple, effective, and informative product ranking system. Users can now explore Blinkit’s most popular and highest-rated products with ease. With further enhancements like real-time updates and predictive features, this system can be an integral tool for customer transparency and business decision-making.

Here is the complete content for:

## ****Chapter 9 – Future Scope and Enhancements****

### ****9.1 Introduction****

While the current version of the Blinkit Top-Selling Products Dashboard meets its objectives effectively, there is ample room for future growth and enhancements. As the project is built on flexible, open-source tools, it can be easily scaled and extended to include real-time data integration, personalized insights, predictive analytics, and broader usability across platforms and audiences.

This chapter outlines the potential directions and improvements that can be applied to strengthen the application’s functionality, performance, and user engagement.

### ****9.2 Proposed Enhancements****

#### **1. Real-Time API Integration**

* **Current Limitation**: The application relies on static CSV files, which require manual updates.
* **Enhancement**: Connect the dashboard to Blinkit's APIs (if available) to automatically fetch and refresh product data, reviews, and ratings in real-time.

#### **2. Advanced Filtering Options**

* Enable multi-filter capabilities based on:
  + Price ranges
  + Rating thresholds
  + Discount percentages
  + Product availability or delivery zones

#### **3. Predictive Analytics**

* Implement machine learning models to:
  + Forecast upcoming top-selling products based on trends.
  + Predict customer preferences using past data.
  + Identify slow-moving inventory for optimization.

#### **4. Personalized Recommendations**

* Introduce user authentication and profiling.
* Allow users to receive product suggestions based on their browsing history, location, or previous purchases.

#### **5. Mobile App Integration**

* Optimize the Streamlit dashboard for mobile devices or convert it into a **Progressive Web App (PWA)**.
* Develop a dedicated mobile app that syncs with the backend for real-time usage on smartphones.

#### **6. Enhanced Visualizations and UX**

* Add features such as:
  + Dynamic trend lines
  + Time-series comparison tools
  + Product detail pop-ups
* Improve UI/UX through design enhancements like dark mode, theme switching, and modular layout components.

#### **7. Cloud Hosting and Scalability**

* Migrate from local deployment to robust cloud platforms such as:
  + **Heroku**
  + **AWS EC2**
  + **Google Cloud App Engine**
* Ensure horizontal scaling and improved performance for larger datasets.

#### **8. Integration with Marketing and Inventory Systems**

* Link insights with inventory management systems to automate restocking of popular products.
* Share dashboard outputs with marketing teams for data-driven campaign targeting.

### ****9.3 Long-Term Vision****

In the long run, this dashboard can evolve into a complete **retail analytics suite**, not just limited to Blinkit. It can be adapted for other e-commerce platforms, extended with AI-driven personalization, and used by marketing analysts, inventory planners, and business strategists to:

* Track consumer behavior patterns
* Manage supply chain efficiency
* Drive promotions with higher conversion rates
* Enhance customer satisfaction through transparency and trust

### ****9.4 Summary****

This project serves as a scalable base model with wide potential for innovation and impact. Future enhancements aim to transform it from a product analytics dashboard into an intelligent, real-time retail assistant that serves both business and consumer needs with equal importance.

Here is the full content for:

## ****Chapter 10 – Benefits and Applications****

### ****10.1 Introduction****

The dashboard developed in this project provides a dual advantage—it enhances the **shopping experience for customers** and supports **data-driven decision-making for Blinkit’s business operations**. This chapter outlines the practical benefits and use-case applications of the project across different stakeholders, including consumers, business analysts, inventory managers, and developers.

### ****10.2 Benefits****

#### **1. Customer Empowerment**

* Allows customers to view **top-selling** and **top-rated** products transparently.
* Supports informed purchase decisions based on real user engagement and ratings.
* Encourages trust in product quality through visible public sentiment.

#### **2. Business Insight**

* Identifies trends in product performance across categories.
* Helps determine which items should be promoted or stocked more frequently.
* Allows businesses to adjust pricing strategies based on rating–price–sales relationships.

#### **3. Operational Efficiency**

* Improves inventory planning by highlighting fast-moving products.
* Reduces overstocking and understocking through better demand forecasting.
* Offers actionable insights that can be incorporated into logistics and warehouse management.

#### **4. Marketing and Promotions**

* Enables data-backed marketing decisions.
* Helps in identifying which products or categories to feature in campaigns.
* Tracks the effectiveness of discounts and product bundling strategies.

#### **5. Scalability and Reusability**

* Built using modular, open-source tools that allow for easy extension.
* Can be adapted for use with other e-commerce platforms beyond Blinkit.
* Provides a reusable template for retail analytics dashboards.

#### **6. Educational Value**

* Serves as a learning tool for students and professionals in:
  + Data science
  + Web development
  + Visualization design
  + Business analytics

#### **7. Real-Time Interaction**

* Offers a responsive and interactive interface that users can navigate intuitively.
* Allows real-time exploration of trends via filtering and visualization updates.

### ****10.3 Applications****

| **User Group** | **Application** |
| --- | --- |
| Customers | View trending products, compare prices and ratings |
| Blinkit Managers | Identify best-sellers, optimize inventory, plan category expansion |
| Marketing Teams | Select products for ad campaigns, promotions, and discounts |
| Data Analysts | Study consumer preferences, develop reports and visual dashboards |
| Educators & Students | Teach/learn data visualization, dashboard development, and Python programming |
| Developers | Use as a template for deploying similar dashboards for other platforms |

### ****10.4 Summary****

The project presents clear benefits and a wide range of practical applications. It is not just a visualization tool but a **multi-purpose platform** that connects business needs, customer transparency, and technical development in one streamlined solution. With further enhancements, it holds the potential to become a full-featured decision support and consumer insights tool for modern retail platforms.

Here is the full content for:

**Chapter 11 – Conclusion**

**11.1 Summary of the Project**

This project titled **“Data Analysis and Visualization of Top Selling Products on Blinkit Using Python”** set out to transform static e-commerce data into a dynamic, interactive dashboard for real-time insight generation. Using Python and its data science ecosystem—especially **Pandas**, **Plotly**, and **Streamlit**—we successfully built a web application that analyzes product-level data and displays critical metrics such as top-selling products, top-rated items, and category-wise performance.

From data collection and cleaning to interactive visualization and deployment, the project followed a structured methodology to ensure usability, accuracy, and real-world relevance. The final product is a customer- and business-friendly dashboard that enables transparent decision-making and improves the understanding of product trends.

**11.2 Key Takeaways**

* **Python** is a powerful tool for end-to-end data processing, analysis, and visualization.
* **Rating count**, while not a direct sales figure, can effectively serve as a sales proxy when used with caution.
* **Interactive dashboards** can simplify the presentation of complex data and offer both technical and non-technical users valuable insights.
* **User-centric design** and real-time filtering make dashboards more engaging and actionable.

**11.3 Outcomes Achieved**

* Developed a working dashboard using real-world data logic.
* Enabled user filtering by product categories.
* Produced clear visual comparisons through bar charts, pie charts, and scatter plots.
* Delivered an extensible framework that can be deployed, scaled, and enhanced for future use cases.

**11.4 Limitations**

* The project currently uses static data (CSV files) rather than real-time APIs.
* Sales data is estimated using rating count, which might not always reflect exact transaction volumes.
* User personalization, login features, and mobile responsiveness were not implemented in this version.

**11.5 Future Scope**

Building on this foundation, the project can be improved by:

* Integrating Blinkit's live APIs for real-time updates.
* Incorporating predictive analytics and machine learning models.
* Creating user accounts and personalized dashboards.
* Deploying to cloud platforms for wider accessibility and scalability.

**11.6 Final Thoughts**

This project demonstrates the practical application of data analytics and visualization in the context of e-commerce. By merging backend analysis with frontend usability, we have created a solution that supports both customers and businesses. As data continues to drive commerce, tools like this dashboard will become essential for decision-making, strategy formulation, and customer engagement.

Here is the complete content for:

**References**

1. **Pandas Documentation**  
   Python Data Analysis Library – User Guide and API Reference.  
   URL: <https://pandas.pydata.org/>
2. **Plotly Documentation**  
   Interactive Data Visualization in Python.  
   URL: <https://plotly.com/python/>
3. **Streamlit Documentation**  
   Streamlit: The fastest way to build data apps.  
   URL: <https://docs.streamlit.io/>
4. **Blinkit Website**  
   Blinkit – Online Grocery and Essentials Delivery Platform.  
   URL: <https://www.blinkit.com/>
5. **McKinsey & Company**  
   “Analytics Comes of Age.” McKinsey Global Institute Report, 2018.  
   URL: [https://www.mckinsey.com](https://www.mckinsey.com/)
6. **Stack Overflow & GitHub Discussions**  
   Various community discussions and shared code snippets used for debugging and feature implementation.  
   URL: [https://stackoverflow.com](https://stackoverflow.com/) and [https://github.com](https://github.com/)
7. **Python Official Website**  
   Python Programming Language – Documentation and Packages.  
   URL: <https://www.python.org/>
8. **Jupyter Notebook**  
   Project Jupyter – Tools for open-source data science and scientific computing.  
   URL: <https://jupyter.org/>