SmartCart – Machine Learning Powered Shopping Recommendation System

Course Name: Computational Intelligence(2307311L)

Group Members:

1. Namrata Ghule (202402070010)
2. Asmita Lonkar (202402070015)
3. Stuti Orpe (202402070019)

1. Abstract

SmartCart is an intelligent shopping system that integrates Machine Learning with a user-friendly interface to provide real-time product recommendations. The system allows users to browse products, add items to their cart, and instantly receive smart suggestions based on the price category of the recently added item.

The project demonstrates a complete ML workflow including data preprocessing, model training, model deployment, and web application integration using Streamlit. It replicates a simplified version of the recommendation engines used in modern e-commerce platforms.

2. Introduction

SmartCart is a machine learning–based shopping recommendation system created to improve the user's shopping experience by suggesting relevant products. The system predicts product price categories and recommends similar items, mimicking the behavior seen in modern e-commerce platforms like Amazon and Flipkart.

The project integrates a trained Random Forest Regressor model inside a Streamlit web application, demonstrating a complete ML workflow from data preprocessing to deployment. It is designed as an educational prototype showing how ML models can support decision-making in retail systems.

3. Objectives

The main objectives of SmartCart are:

- To design a user-friendly shopping interface
- To integrate a trained ML model with a Streamlit application
- To generate product recommendations based on user selections
- To preprocess and encode product data for ML training
- To demonstrate an end-to-end ML workflow with deployment

4. Dataset Description

This project uses two different datasets, each serving a specific purpose in the SmartCart system: one for machine learning model training and one for the Streamlit user interface.

2.1 Advanced SmartCart Dataset (Google Colab – ML Training Dataset)

This dataset is used exclusively in Google Colab to train the machine learning model.
It contains detailed and processed information such as:

- Product names
- Product prices
- Price-range labels created using predefined bins
- Encoded values generated by LabelEncoder
- Scaled numeric features generated using StandardScaler

This dataset allows the Random Forest Regressor to learn how product names relate to price patterns and ranges. It is not used directly in the Streamlit app, but it produces the final joblib model files required by the app.

2.2 products.csv Dataset (VS Code – Streamlit UI Dataset)

This is a simpler dataset created specifically for the SmartCart interface.
It contains:

- Product name
- Price
- Image filename

These values are used for displaying products, adding them to the cart, and calculating totals.
Inside the Streamlit app, price-based labels are generated dynamically to match the ML model's logic. This dataset keeps the application lightweight and user-friendly.

Relationship Between the Two Datasets

Advanced Dataset (Colab)	**VS Code Dataset (products.csv)**
Used only for ML training	Used only for the Streamlit UI
Includes encoded + scaled values	Includes simple product details
Contains price labels	UI adds labels dynamically
Generates model joblib files	Used for display and recommendation selection

5. Methodology

5.1 Data Collection

Two datasets were collected and used:

- The Advanced SmartCart dataset in Google Colab for ML model training
- The products.csv dataset in VS Code for UI display and recommendation mapping

This separation ensures smooth functioning of both the ML pipeline and the user interface.

5.2 Data Preprocessing

The advanced dataset was fully cleaned and prepared before model training:

- Column names were standardized
- Prices were converted to numeric format
- Missing or invalid values were corrected
- Price categories (labels) were created using pd.cut()
- Product names were encoded into numbers using LabelEncoder
- Encoded values were scaled using StandardScaler

This created a structured ML-ready dataset.

5.3 Encoding and Scaling

To convert product names into usable numerical features:

- LabelEncoder assigned each product a unique number
- StandardScaler normalized the encoded values
  Both encoder and scaler were saved using joblib so they could be reused during predictions in the Streamlit application.

5.4 Model Training (Random Forest Regressor)

The machine learning model used in this project is a Random Forest Regressor. The model was trained in Google Colab using the advanced dataset. It learned relationships between encoded product names and their price ranges.

After successful training, the following files were saved:

- best_model.joblib – the trained Random Forest model
- encoder.joblib – LabelEncoder used during training
- scaler.joblib – StandardScaler used during training

These files are essential for producing real-time predictions inside the SmartCart application.

5.5 Streamlit Integration

The Streamlit app loads the products.csv dataset and displays items to the user. When the user adds a product to the cart:

1. The product name is encoded
2. Encoded value is scaled
3. The Random Forest model predicts the price category
4. The app recommends another product from the same predicted category

This creates a simple but effective price-based recommendation system.

5.6 Testing and Validation

The system was tested to ensure:

- Accurate loading of datasets
- Proper model predictions
- Logical and consistent recommendations
- Smooth shopping flow in the UI
- Correct calculation of total price and cart functionality

6. Software and Tools Used

- Python
- Streamlit – for frontend UI
- Pandas – data handling
- Scikit-Learn – training ML model
- Joblib – saving/loading models
- Jupyter Notebook – experimentation
- GitHub – project submission

7. Results

The SmartCart application successfully:

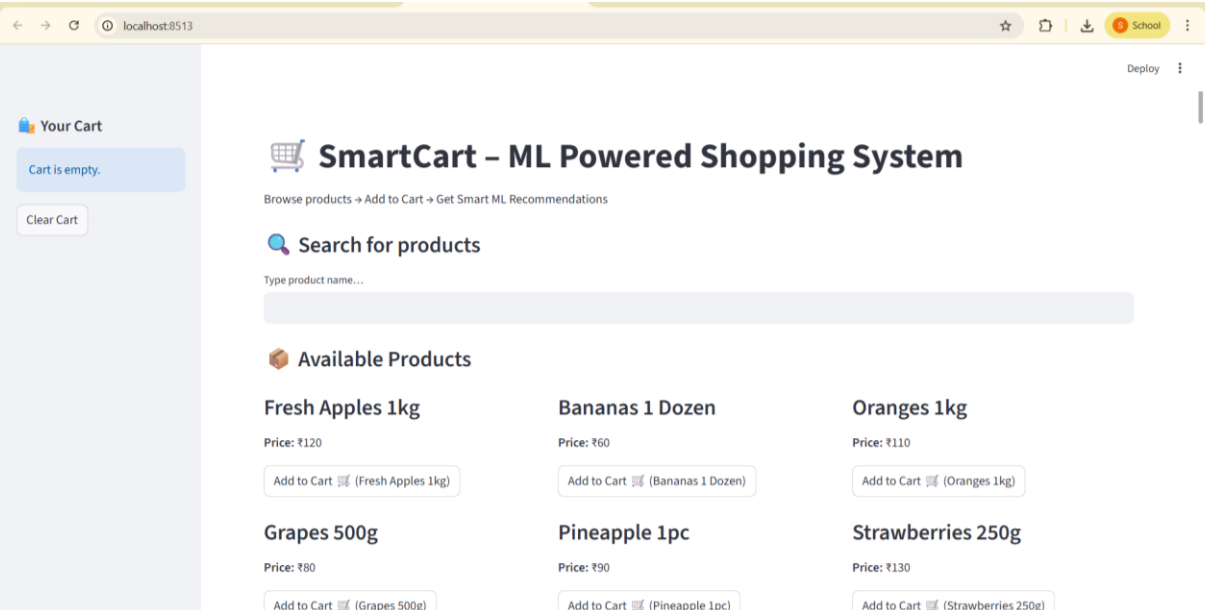
- Loads and displays product items
- Allows real-time adding to cart
- Displays total bill calculation
- Provides meaningful recommendations based on last item added
- Uses a trained ML model to classify product categories
- Responds instantly with suggestions

Example:

- Last item: *Eraser Pack* (low price category)
- Recommendation: *Sharpener / Glue Stick / Pen* (same price category)

8. Screenshots

Home Page & Product list



Your Cart

Cart is empty.

Clear Cart

SmartCart – ML Powered Shopping System

Browse products → Add to Cart → Get Smart ML Recommendations

Search for products

Type product name...

Available Products

Fresh Apples 1kg

Price: ₹120

Add to Cart (Fresh Apples 1kg)

Bananas 1 Dozen

Price: ₹60

Add to Cart (Bananas 1 Dozen)

Oranges 1kg

Price: ₹110

Add to Cart (Oranges 1kg)

Grapes 500g

Price: ₹80

Add to Cart (Grapes 500g)

Pineapple 1pc

Price: ₹90

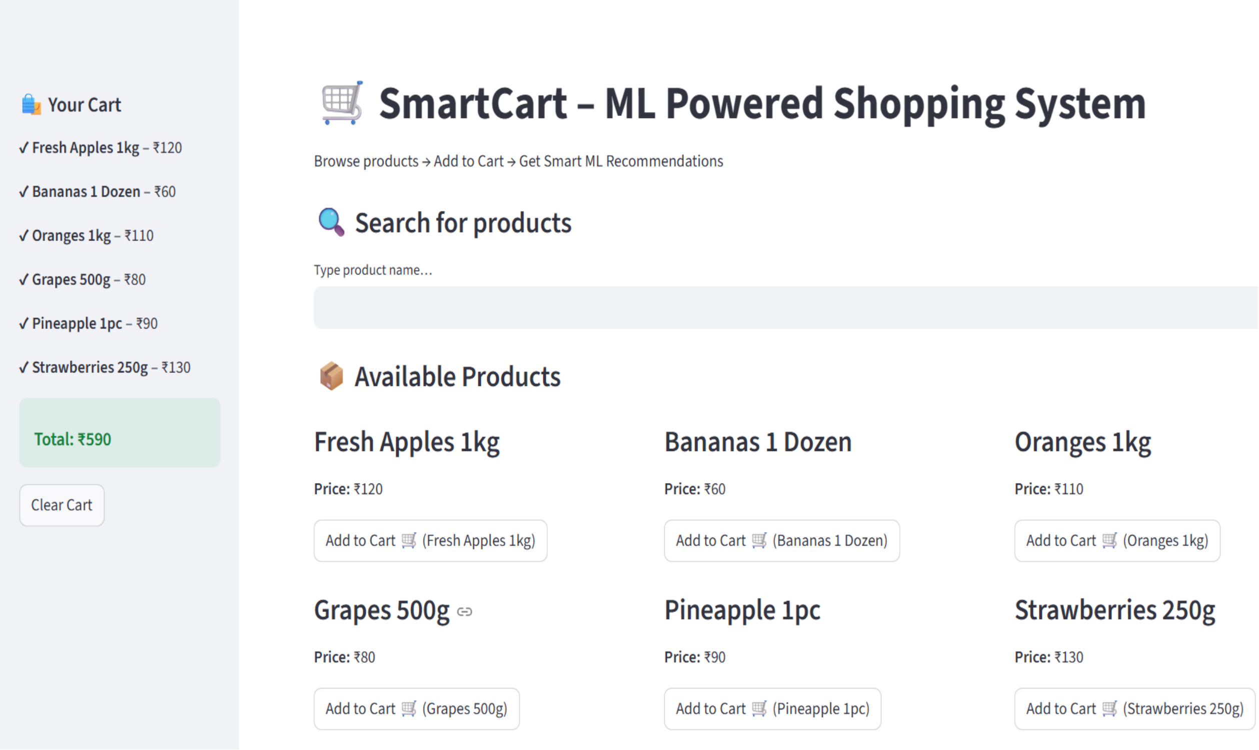
Add to Cart (Pineapple 1pc)

Strawberries 250g

Price: ₹130

Add to Cart (Strawberries 250g)

Cart sidebar



🛍️ Your Cart

✓ Fresh Apples 1kg – ₹120

✓ Bananas 1 Dozen – ₹60

✓ Oranges 1kg – ₹110

✓ Grapes 500g – ₹80

✓ Pineapple 1pc – ₹90

✓ Strawberries 250g – ₹130

Total: ₹590

🛒 SmartCart – ML Powered Shopping System

Browse products → Add to Cart → Get Smart ML Recommendations

🔍 Search for products

Type product name...

📦 Available Products

Fresh Apples 1kg

Price: ₹120

Add to Cart 🛒 (Fresh Apples 1kg)

Bananas 1 Dozen

Price: ₹60

Add to Cart 🛒 (Bananas 1 Dozen)

Oranges 1kg

Price: ₹110

Add to Cart 🛒 (Oranges 1kg)

Grapes 500g

Price: ₹80

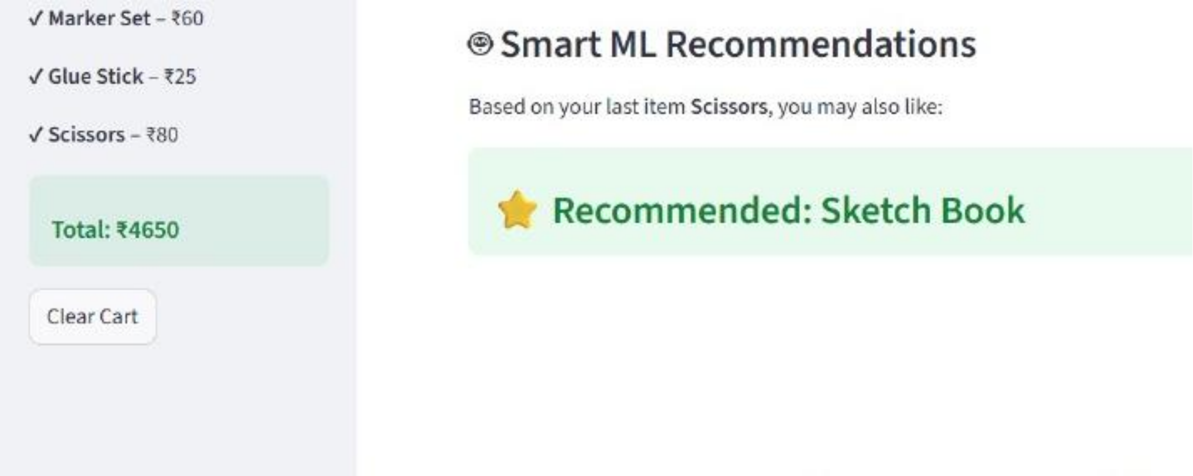
Pineapple 1pc

Price: ₹90

Strawberries 250g

Price: ₹130

Recommendation output



✓ Marker Set – ₹60

✓ Glue Stick – ₹25

✓ Scissors – ₹80

Total: ₹4650

Smart ML Recommendations

Based on your last item **Scissors**, you may also like:

⭐ Recommended: Sketch Book

9. Conclusion

SmartCart demonstrates how a simple Machine Learning model can enhance the user shopping experience by providing intelligent product recommendations. The project successfully integrates ML with a real-time web application and replicates key features of modern-day online shopping platforms.

The system can be expanded further with:

- Content-based filtering
- Collaborative filtering
- Neural network-based recommender models
- Full e-commerce integration

10. Future Scope

- Add product categories and metadata
- Implement image-based recommendations
- Use cosine similarity between product embeddings
- Expand model using deep learning
- Deploy application online using cloud services

11. References

- Scikit-learn Documentation
- Streamlit Official Documentation
- Python Pandas Documentation
- Various online ML tutorials and datasets