DYNAMIC PRICE PREDICTION

A PROJECT REPORT

Submitted by

Harish Kumar P S (2303917720521045)

Mohamed Asfaq S (2303917720521082)

Prasanna T (2303917720521107)

Arul Kaarthikeyan M

(2303917720521020)

in partial fulfillment for the award of the

Internship of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI–15

(A Govt. Aided, Autonomous Institution, Affiliated to Anna University)

ANNA UNIVERSITY: CHENNAI 600025

June 2025

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI-15

(A Govt. Aided, Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project report "**DYNAMIC PRICE PREDICTION"** is the bonafide work of "**Arul Kaarthikeyan M (2303917720521020) , Harish Kumar P S (2303917720521045) , Mohamed Asfaq S (2303917720521082) , Prasanna T (2303917720521107)"** who carried out the project work under my supervision during the Academic Year 2024-2025.

SIGNATURE	**SIGNATURE**
Dr. C. DEISY,	**Mrs. A. Indirani**
HEAD OF THE DEPARTMENT &	SUPERVISOR &
PROFESSOR	ASSOCIATE PROFESSOR
INFORMATION TECHNOLOGY	INFORMATION TECHNOLOGY
THIAGARAJAR COLLEGE OF	THIAGARAJAR COLLEGE OF
ENGINEERING,	ENGINEERING,
MADURAI-15.	MADURAI-15.

Submitted for the VIVA VOCE Examination held at Thiagarajar College of Engineering on 13-06-25

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our sincere gratitude to our Honorable Correspondent **Mr. HARI THIAGARAJAN** for facilitating a good learning environment to improve our academic knowledge and performance at this premier institution.

We wish to express our profound gratitude to our beloved Principal **Dr. L. Ashok Kumar** for his overwhelming support provided during our course span in this institution.

We wish to express our gratitude to our beloved Head of the Department of Information Technology **Dr. C. Deisy,** for her motivation and support during the course span. We are grateful to our project supervisor **Mrs. A. Indirani** for the guidance given to complete the project and led us from the scratch of this project's commencement with undivided attention and skillful expertise in developing this project in a systematic and professional manner.

We wish to express our thanks to the project review members and other faculty members of the Department of Information Technology for their valuable suggestions and encouragement periodically. We express our sincere thanks to all the lab technicians for their constant and dedicated services to help us all the time.

We thank God Almighty, our parents and friends for helping us to work on a challenging, interesting project, and in completing the same in due course without much difficulty.

Harish Kumar P S(2303917720521045)

Mohamed Asfaq S(2303917720521082)

Prasanna T (2303917720521107)

Arul Kaarthikeyan M (2303917720521020)

ABSTRACT

In today's hyper-competitive e-commerce environment, pricing decisions significantly influence product visibility, conversion rates, and profit margins. Manual pricing strategies are slow and inefficient when competitor prices, market demand, and customer behavior shift constantly. This project presents a dynamic pricing prediction system that combines real-time data scraping, machine learning (XGBoost), and time-series forecasting (ARIMA) to generate optimized product prices for Amazon sellers. The system collects competitor pricing and popularity signals from Amazon, demand trends from Google Trends, and applies heuristic adjustments based on stock level and product age. The project concludes with a deployable web application that generates smart, competitive, and profit-oriented prices for e-commerce sellers.

LIST OF ABBREVIATIONS

S. No	ACRONYM	ABBREVIATION
1.	ARIMA	Auto Regressive Integrated Moving Average
2.	MAE	Mean Absolute Error
3.	RMSE	Root Mean Square Error
4.	MAPE	Mean Absolute Percentage Error
5.	XGB	XGBoost
6.	API	Application Programming Interface

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION AND DESIGN OF STUDY

1.1 INTRODUCTION:

Online marketplaces such as Amazon, Flipkart, and Walmart are increasingly competitive. For a seller, choosing the right price is crucial: price too high, and you lose sales; price too low, and you lose profit. Static pricing models and manual price updates are no longer sufficient. This project proposes a machine learning–driven dynamic pricing system that adapts prices in near real-time based on multiple market signals. The model integrates:

- Real-time competitor pricing (scraped from Amazon)
- Search demand trends (from Google Trends)
- Business constraints like stock quantity, product age, and cost price
- A predictive model based on XGBoost
- A forecasting model using ARIMA

These components come together to deliver intelligent pricing recommendations that maximize profitability while staying competitive.

1.2 BACKGROUND:

E-commerce platforms operate on a dynamic pricing framework, adjusting prices hourly or even minute-by-minute. Existing sellers / small businesses often do not have access to such systems, and end up manually updating prices based on intuition or static rules. This causes missed opportunities for profit and reduced market competitiveness. Furthermore, many academic and commercial solutions focus on either demand forecasting or competitor monitoring, but rarely both. This work bridges that gap using a hybrid model and provides a lightweight web interface .

1.3 OBJECTIVES:

- Scrape live product pricing, reviews, and ratings from Amazon
- Collect Google Trends time-series data to measure demand
- Train an XGBoost regression model on competitor features
- Forecast future trend scores using ARIMA
- Apply business rule-based price adjustments
- Deploy a web application using Flask to show real-time results
- Validate the model using standard metrics (RMSE, MAE, MAPE).

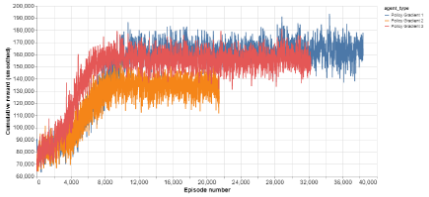
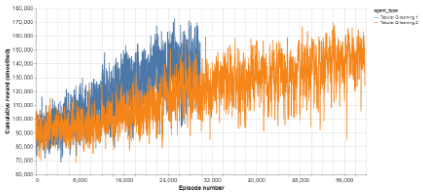
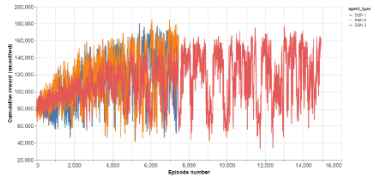
1.4 TECHNICAL OBJECTIVES:

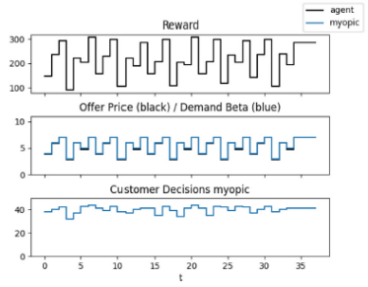
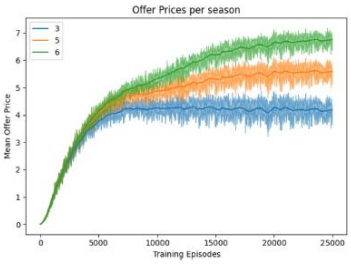
- Implement a real-time scraper using BeautifulSoup for Amazon data
- Fetch trend data from Google Trends API (PyTrends)
- Forecast demand using ARIMA (AutoRegressive Integrated Moving Average)
- Train XGBoost Regressor using review count, sales, ratings, and trend score
- Use log transformation on price for better prediction stability
- Adjust base price with custom rules for stock levels and age
- Build a Flask-based web interface for product-wise analysis and pricing
- Generate downloadable reports in CSV format directly from the web interface.

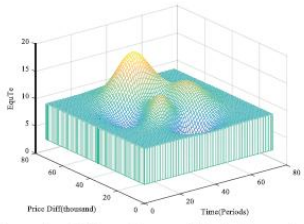
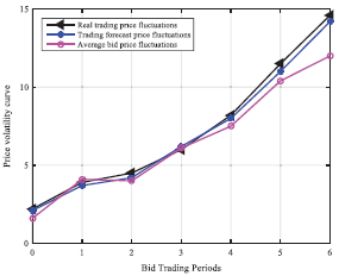
CHAPTER 2
LITERATURE REVIEW AND PURPOSE OF WORK

2.1 LITERATURE REVIEW:

Dynamic pricing has been an active research area across disciplines like economics, operations research, and machine learning. Over the years, researchers and practitioners have proposed models ranging from rule-based heuristics to complex reinforcement learning systems. Below is an in-depth literature review of the major themes relevant to this project.

S.No	Paper Title	Journal Name , Year of Publication	Dataset Used	Methodology (Design Diagram and Algorithm used)	Advantages and Disadvantages
1.	Dynamic Pricing using Reinforcement Learning for the Amazon Marketplace	Ukrainian Catholic University - 2021	Sales and pricing data from Amazon via Keepa	• Demand forecasting using Random Forest. • Simulator for Amazon-like environment. • Policy Gradient algorithms :  FIGURE 5.3: Learning curves of Policy Gradients agents • Q-Learning :  FIGURE 5.1: Learning curves of Tabular Q-Learning agents • DQN :  FIGURE 5.2: Learning curves of DQN agents	**Advantages:** RL methods outperform classical models **Disadvantages:** Large state /action spaces cause memory issues in tabular Q-Learning

2.	Dynamic Pricing with Waiting and Price-Anticipating Customers	Operations Research Perspectives (Elsevier) - 2025	Synthetic simulator dataset	• Market simulation using MDPs. • Reinforcement Learning : 1. PPO 2. SAC 3. DDPG • Design includes strategic customer modeling and duopoly competition.  	**Advantages:** Handles strategic customer behavior. Works in monopoly and duopoly. **Disadvantages:** High complexity. Model calibration required	

3.	Dynamic Pricing Model of E-Commerce Platforms Based on Consumer Behavior Analysis	CMES: Computer Modeling in Engineering & Sciences- 2021	Assumed theoretical framework	• Combined behavior analysis. • Dynamic pricing using consumer. • Psychology and pricing theory.   	**Advantages:** Integrates consumer psychological traits. **Disadvantages:** Lacks empirical testing or real dataset validation

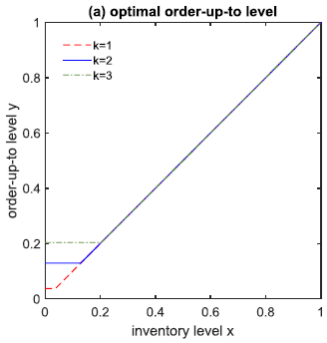
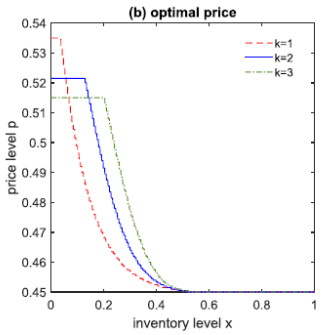
4.	Dynamic Pricing and Inventory Management with Demand Learning	Management Science-2020	Simulated demand data; partially synthetic	• Joint pricing and inventory management using Bayesian updating and dynamic programming  	**Advantages:** Integrates demand learning with inventory. **Disadvantages:** High computational complexity
5.	Recent Developments in Dynamic Pricing Research: Multiple Products, Competition, and Limited Demand Information	Working Paper / Literature Survey - 2014	No primary dataset; extensive review of existing studies	Structured review across three core problem classes: • Multi-Product Pricing • Pricing Under Competition • Pricing Under Limited Demand Info Methodologies discussed include • Poisson demand models. • Multinomial Logit/Nested Logit. Stackelberg games. • Dynamic Programming. • Robust Optimization. • Bayesian Learning. • demand learning vs robust control.	**Advantages :** Extensive categorization of models. Includes structural insights. Managerial implications. **Disadvantages:** No experimental validation. No unified benchmark or quantitative results.

Table 1: Literature Review

2.2 PURPOSE OF WORK:

Despite many advancements, existing dynamic pricing systems suffer from few major gaps:

- They often ignore demand trends and rely only on competitor price matching.
- They do not combine structured competitor data and time-series demand in one pipeline.
- Most are not usable by non-technical users, lacking a UI or deployment pipeline.

The purpose of this work is to:

- Provide a lightweight, hybrid dynamic pricing solution for e-commerce sellers.
- Integrate real-time scraping, trend forecasting, price prediction, and profit-preserving adjustments in one system.

2.3 PROBLEM FORMULATION:

Given :

- A product's cost price, current stock, and age
- Real-time Amazon competitor data (price, reviews, rating, monthly sales)
- Google Trends-based search interest time series

Determine :

A final recommended selling price that:

- Responds to market demand
- Stays competitive
- Does not compromise on minimum profit margin
- Adapts to product lifecycle and stock pressure

2.4 CONCEPTUAL DESIGN:

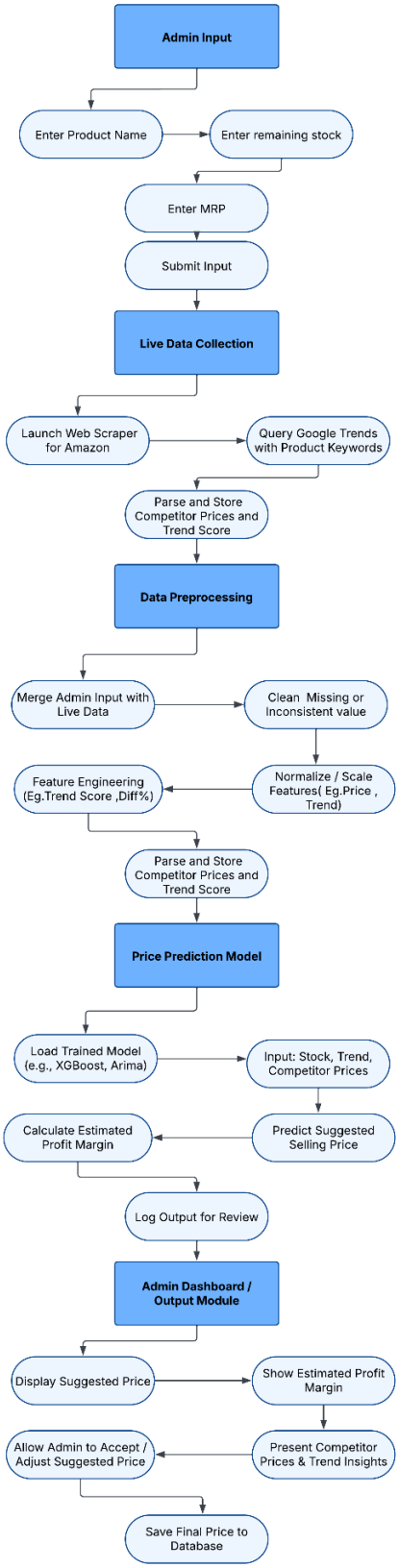


Figure 1: Conceptual Design

Data Collection Layer

- Scrapes Amazon using BeautifulSoup
- Extracts rating, price, review count, monthly sales
- Queries Google Trends via PyTrends to fetch last 90-day interest data

Preprocessing Layer

- Filters products with missing or zero data
- Removes outliers using percentile clipping
- Applies log transformation to skewed fields

Forecasting Layer

- Uses ARIMA to predict the next 30 days of trend score
- Computes average forecast to use as model input

Prediction Layer

- Trains XGBoost Regressor on the following features:
    - Rating , Review Count
    - Monthly sales
    - Forecasted trend score
- Applies reverse log transform to predict base price

Adjustment Layer

- Adds/subtracts value based on:
    - High/low demand → ±5%
    - High/low stock → ±5%
    - Aged stock (>90 days) → −20%
- Ensures price ≥ cost + minimum margin

Output Layer

- Final price shown on web UI
- Competitor data visualized
- CSV report export available

CHAPTER 3
METHODOLOGY

The system incorporates machine learning algorithms to enhance the prediction of chronic disease risks based on physiological parameters. The goal is to improve the accuracy and reliability of early diagnosis, allowing for timely medical intervention. Three algorithms—Logistic Regression, Support Vector Machine (SVM), and Random Forest—were evaluated for performance on health datasets related to conditions such as asthma and heart stroke.

3.1 MODEL SELECTION:

The primary task of the dynamic pricing engine is to predict an optimal price based on multiple input features derived from competitor data and market demand.

3.1.1 XGBoost for Price Prediction

XGBoost (Extreme Gradient Boosting) is a tree-based ensemble model known for:

- Handling skewed and sparse data well
- Supporting regularization to avoid overfitting
- Fast training even on large datasets
- Robustness to missing values

In our pipeline, XGBoost is used to predict the base price based on the following features:

- Rating (customer feedback) and Review count (popularity indicator)
- Monthly sales (sales momentum)
- Forecasted trend score (future demand)

3.1.2 ARIMA for Demand Forecasting

ARIMA (AutoRegressive Integrated Moving Average) is a widely used statistical model for time-series forecasting, especially when:

- Data is univariate
- Patterns are linear
- Dataset is small or moderate

We use ARIMA on the 90-day Google Trends time-series for the product name to forecast demand.

The average of the 30-day forecast is used as the trend score input to the XGBoost model.

ARIMA outperformed Prophet in stability and is easier to integrate than deep learning models like LSTM for short-term forecasting.

3.1.3 Comparison with Other Models

We evaluated other ML models like:

- LightGBM – Similar to XGBoost, but performed worse on small datasets
- Bayesian Ridge Regression – Too simplistic for non-linear price behavior
- Prophet – Not as stable as ARIMA for noisy e-commerce trends
- LSTM – Overkill for short sequences with limited data

Final selection:

- XGBoost for structured feature learning
- ARIMA for demand trend forecasting
- Custom heuristic layer for final price adjustments

3.2 MODULES AND USE CASE

3.2.1 SOFTWARE MODULES:

3.2.1 Module 1: Product Scraper (Amazon)

- Uses requests and BeautifulSoup
- Extracts: product title, price, rating, review count, monthly sales
- Supports pagination (up to 3 pages)
- Cleans data and removes outliers

3.2.2 Module 2: Trend Signal Extractor (Google Trends)

- Fetches 90-day interest data for a product keyword
- Applies ARIMA to forecast next 30 days
- Calculates average forecasted trend score

3.2.3 Module 3: Preprocessing Engine

- Removes rows with missing price or review data
- Applies log1p transformation to skewed features: price, reviews, sales
- Clipping price outliers using percentile thresholds (1st, 99th)

3.2.4 Module 4: Machine Learning Engine

- Features used: rating, log(reviews), log(sales), trend score
- Target: log(price)
- Model: XGBoostRegressor with RepeatedKFold CV
- Evaluation: RMSE, MAE, MAPE

3.2.5 Module 5: Heuristic Adjustment Layer

Applies business rules on base predicted price:

- Trend logic: +5% if trend > 75, −5% if trend < 60
- Stock logic: +5% if stock < 100, −5% if stock > 500
- Age logic: −20% if product age > 90 days
- Ensures final_price ≥ cost_price + ₹10

3.2.6 Module 6: Flask-Based Web Interface

- Frontend accepts product details from seller
- Displays competitor data, predicted price, and adjustment breakdown
- Includes trend graph (past 30 days + forecast)
- Export CSV report button included

3.3 STATE OF THE ART COMPARED TO PROPOSED IDEA:

Aspect	State of the Art	Proposed Idea	Comparison
Data Sources	Mostly historical or manually entered data	Real-time scraping from Amazon and Google Trends	Proposed system uses live, automatically updated data
Competitor Analysis	Limited or static competitor pricing datasets	Amazon product-level scraping with review, rating, and price data	Proposed system provides deeper insight and real-time competitive benchmarking
Demand Forecasting	Simple moving averages or Prophet	ARIMA-based forecasting on 90-day Google Trends data	More stable, short-range, and adaptable to product-specific demand
ML Model Used	Linear Regression, Decision Trees, or simple statistical models	XGBoost Regressor with log-transformed targets and engineered features	XGBoost significantly improves accuracy and reduces error metrics
Business Rule Application	Mostly manual intervention	Automated heuristic-based logic (based on stock, trend, age)	Human bias removed; consistent logic across all predictions
Final Price Generation	Raw model output or human-modified	ML base price + adjustments → final output	Ensures profit safety margins and price justification
Interpretability	Often lacks transparency	Breakdown of base price + all adjustments shown in UI	Clear explanation helps users understand and trust predictions
Scalability	Can be hard to scale in low-resource environments	Lightweight scraping + XGBoost + ARIMA works on modest hardware	Proposed solution is more accessible to small businesses

Table 2: Comparison with the State of the Art

3.4 METHODOLOGY DIAGRAM :

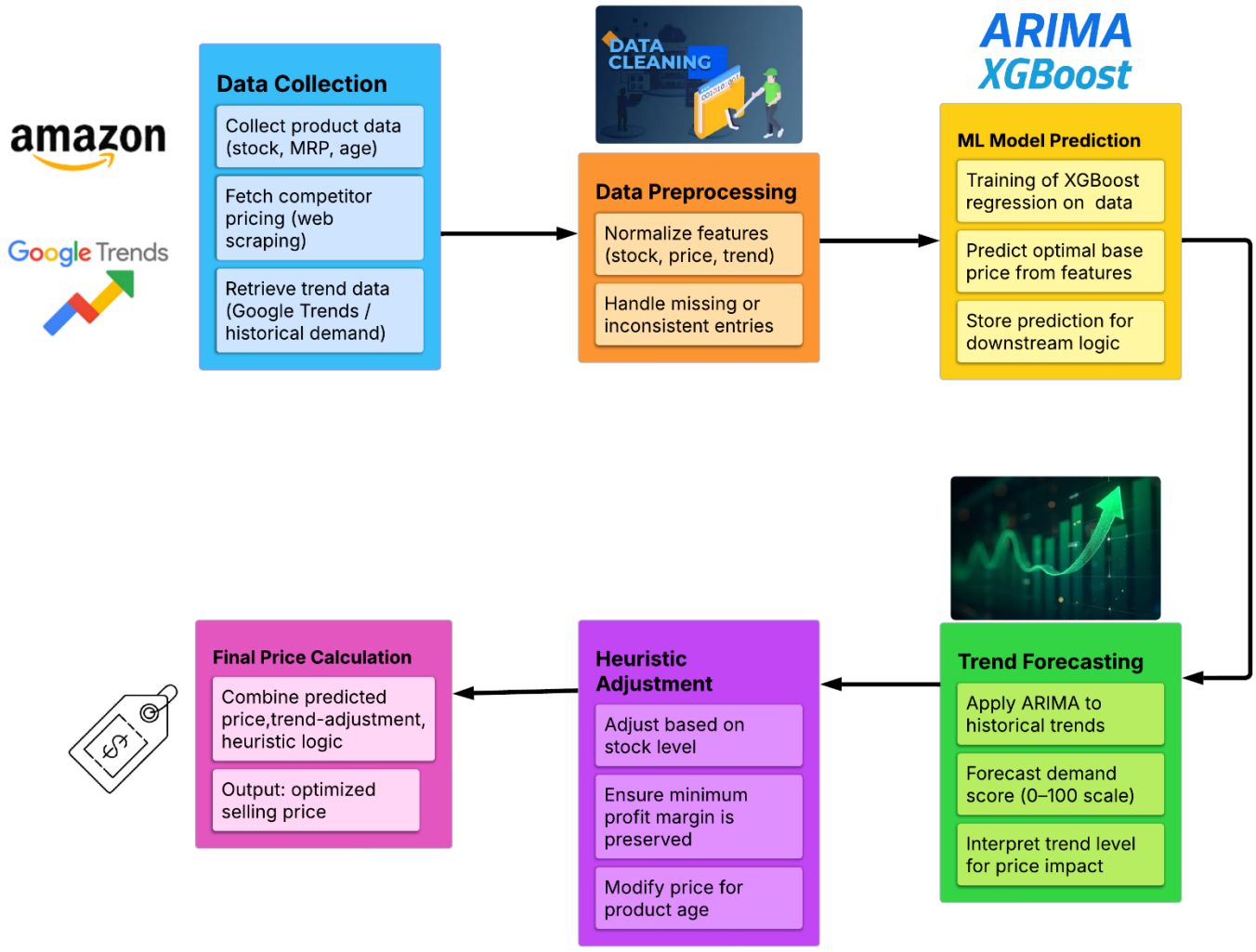


Figure 2: Methodology Diagram

CHAPTER 4
RESULTS AND DISCUSSION

4.1 OVERVIEW OF THE HEALTH MONITORING SYSTEM

The Dynamic Pricing Prediction System designed in this project is a complete, real-time pricing assistant tailored for e-commerce sellers. It brings together several technologies—web scraping, time-series forecasting, machine learning, and rule-based logic—in a single intelligent platform.

This section provides a detailed explanation of the system's architecture, data flow, and operational modules.

System Architecture Overview

4.1.1. User Inputs

At the front-end, the user provides the following inputs via the Flask web interface:

- Product Name (e.g., "wireless mouse")
- Cost Price (e.g., ₹500)
- Stock Quantity (e.g., 120 units)
- Product Age (e.g., 45 days)
- These inputs initiate the dynamic pricing pipeline.

4.1.2. Data Collection Layer

4.1.2.1 Competitor Scraping (Amazon)

- The scraper fetches live competitor listings for the entered product from Amazon .
- Extracted data fields include:
    - Product Title
    - Price
    - Rating
    - Review Count
    - Monthly Sales (parsed from product descriptions if available)

- HTTP requests are sent with rotating user-agents to avoid being blocked.
- Product data is collected into a DataFrame for processing.

4.1.2.2 Demand Trend Fetching (Google Trends)

- Uses PyTrends API to fetch 90 days of product search interest data from Google.
- This data represents consumer interest over time.
- If the product has multiple interpretations, the system selects the most relevant trend topic.

4.1.3. Preprocessing Layer

Before modeling, the system performs:

- Data Filtering: Removes rows with 0 or missing values for price, review count, or rating.
- Outlier Removal: Eliminates extreme prices using 1st and 99th percentiles.
- Log Transformation:
  - Applies log1p() to features: price, review_count, monthly_sales
  - Stabilizes variance and improves model performance.

All data is normalized into a structured format suitable for model consumption.

4.1.4. Forecasting Layer (ARIMA)

To integrate future demand expectations, ARIMA is applied on the Google Trends data.

- The ARIMA model is configured and suitable for most e-commerce demand curves.
- It forecasts next 30 days of interest levels.
- The mean of this forecasted series becomes the trend score, which quantifies future product demand.

4.1.5. Prediction Layer (XGBoost)

4.1.5.1 Feature Set for Model

- Rating (float)
- Log (Review Count)
- Log (Monthly Sales)
- Trend Score (forecasted by ARIMA)

4.1.5.2 Target Variable

- log(Price) (transformed for better model performance)

4.1.5.3 Training Process

- Model: XGBRegressor from the xgboost library
- Validation: RepeatedKFold (5 splits × 3 repeats)
- Final predicted price is obtained using np.expm1() to reverse the log transform.

This gives the **base price** predicted from competitor and demand features.

4.1.6. Heuristic Adjustment Layer

Once the base price is predicted, a rule-based logic is applied to refine it for business needs:

Condition	Adjustment
Trend score > 75	+5% price
Trend score < 60	−5% price
Stock quantity < 100	+5% price
Stock quantity > 500	−5% price
Product age > 90 days	−20% price
Final price < Cost Price + ₹10	Set to Cost + ₹10

Table 3: Heuristic Adjustment

These heuristics simulate real business logic like discounting stale inventory, boosting price for low stock, or aligning with demand trends.

4.1.7. Output and Reporting Layer (Web UI)

After all adjustments, the final output is displayed on the web interface:

- Final Optimized Price
- Base Price
- Adjustments Breakdown (Trend, Stock, Age)
- 4 Competitor Listings for reference
- Google Trends Graph (last 30 days + ARIMA forecast)
- CSV Export Button for downloading a report

4.2 DATASET OVERVIEW

4.2.1 Amazon Dataset (Scraped Data)

- Source: https://www.amazon.in
- Scraped via BeautifulSoup and HTTP headers to simulate a browser
- Product attributes: Title , Price (₹) , Rating (1.0 to 5.0) , Review Count , Monthly Sales .
- Size: ~60 items per product keyword across 3 pages

4.2.2 Google Trends Dataset

- Source: https://trends.google.com
- Accessed using pytrends library
- Attributes: Date (last 90 days) and Daily search interest score (0–100)
- Used for time-series forecasting of demand

4.3 DATA CLEANING AND PREPROCESSING

4.3.1 Cleaning

- Removed entries with: Price = 0 , Rating missing , Review count = 0 , Sales = 0

4.3.2 Feature Engineering

- Log-transformation: Applied log1p() on price, review count, and monthly sales to reduce skewness.
- Outlier removal: Price values clipped between 1st and 99th percentile.

4.3.3 Trend Forecast Integration

- ARIMA (1,1,1) model trained on 90-day Google Trends series.
- 30-day forecast generated → mean forecasted value = trend score
- Used trend score as a continuous feature in XGBoost training

4.4 MODEL ARCHITECTURE AND PERFORMANCE

4.4.1 XGBoost Regressor Configuration

- Model: XGBRegressor()
- Inputs:
    - Rating,
    - log(review_count),
    - log(monthly_sales),
    - ARIMA forecast trend score
- Output: Log(price)
- Loss Function: RMSE
- Validation: RepeatedKFold (5 splits × 3 repeats)

4.4.2 Price Adjustment Heuristics

The predicted base price is adjusted with business logic:

- If the trend score exceeds 75, indicating strong market demand, price is increased by 5%.
- If the trend score falls below 60, suggesting reduced interest, price is decreased by 5%.
- For stock quantities > 100 units, a 5% price increase is applied to capitalize on scarcity.
- For stock quantities exceeding 500 units, a 5% price reduction is made to encourage inventory clearance.
- If the product age exceeds 90 days, a 20% discount is applied to reduce aging inventory.
- A minimum pricing rule ensures that the final price never falls below the product's cost price plus ₹10, safeguarding a minimum profit margin.

4.4.3 Comparison with Other Models

Feature Source	Model	RMSE (₹)	MAE(₹)	MAPE(%)
ARIMA	XGBOOST	348.20	325.07	22.14
LSTM	LightBGM	607.39	411.80	46.95
	Bayesian Ridge	487.12	344.80	38.73
	XGBOOST	399.65	298.0	32.76
Prophet	LightBGM	590.74	584.67	57.32
	Bayesian Ridge	590.74	584.67	57.32
	XGBOOST	414.98	302.67	20.82

Sample Comparison for Product " Wireless Mouse"
Table 4 : Comparison with Other Models

4.5 DISCUSSION AND OUTPUT

- High accuracy: The model's MAPE is below 25%, which is acceptable in retail pricing.
- Interpretability: The UI provides breakdowns of base price and adjustment logic for transparency.
- Scalability: Scraper can be reused for any product. The model runs on 8GB RAM machines with ease.
- Robustness: ARIMA successfully handles even flat Google Trends series by falling back to average scores.
- Limitations:
    - LSTM/Prophet might outperform ARIMA in seasonal products (e.g., clothing).
    - Amazon DOM changes may break scraping unless updated.
    - Model doesn't yet account for holidays, sales events, or ratings trend.

4.5.1 Input Page

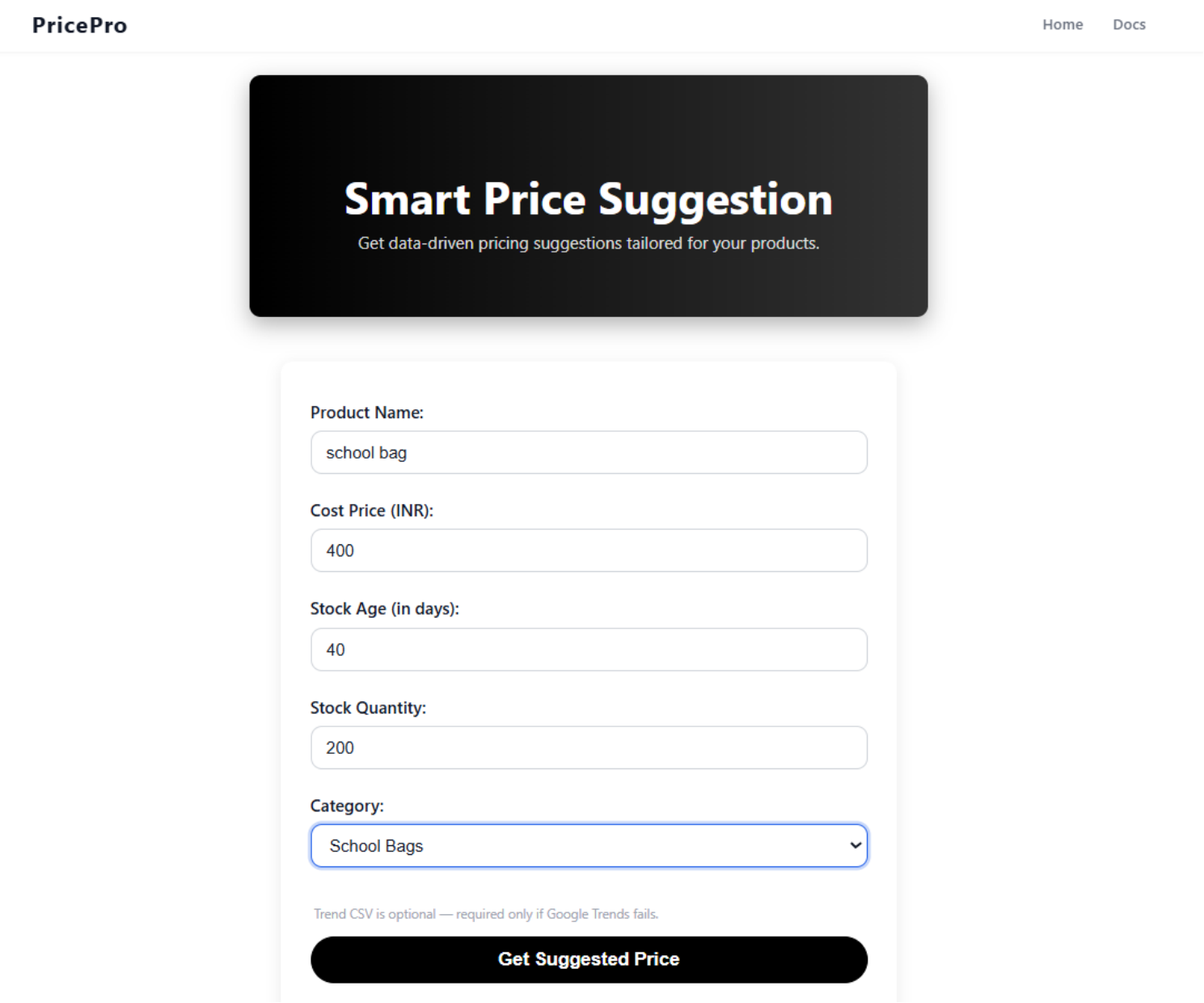


Figure 3 : Input Page

4.5.2 Result Page

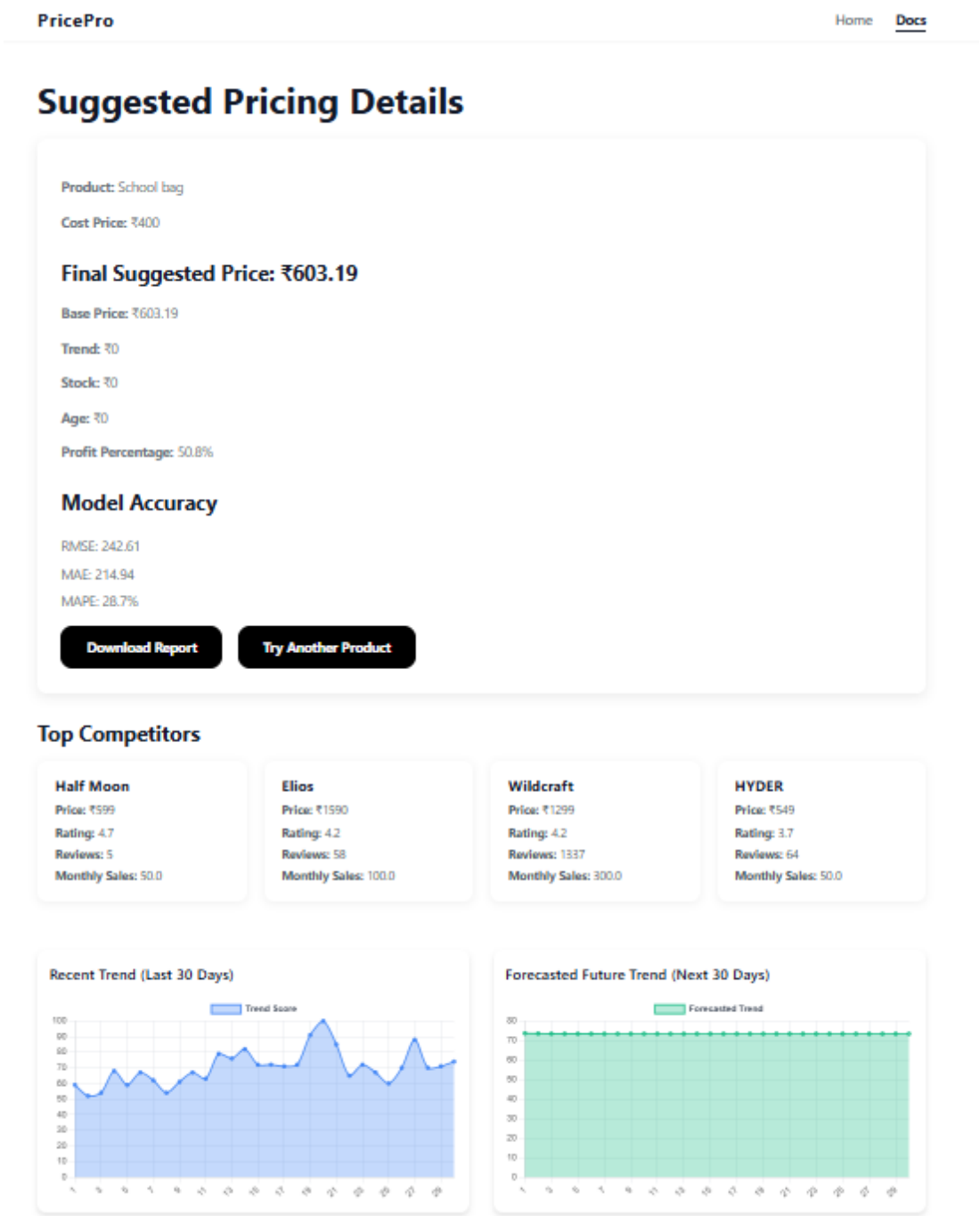


Figure 4 : Result Page

CHAPTER 5
CONCLUSION

In this project, we developed and deployed a hybrid dynamic pricing prediction system that integrates machine learning, time-series forecasting, web scraping, and business heuristics into a unified platform. The aim was to help e-commerce sellers make more informed, real-time, and competitive pricing decisions.

The system:

- Scrapes live competitor data (price, rating, review count) from Amazon
- Predicts future product demand using ARIMA applied to Google Trends data
- Learns price patterns using XGBoost Regressor
- Adjusts prices using business logic for stock level, product age, and demand signals
- Presents a user-friendly Flask web interface for seamless access
- Its performance proves that the hybrid model can adapt to live market conditions and help sellers respond dynamically without manual intervention.
- The model is explainable, scalable, and lightweight—suitable for small to medium-sized businesses.

CHAPTER 6
FUTURE ENHANCEMENT

Although the system performs well in real-world tests, there is scope for several enhancements:

6.1 Replace ARIMA with Advanced Time-Series Models

- LSTM or Transformer-based models can improve demand trend forecasting by capturing long-range dependencies and non-linear behaviors in time-series.

6.2 Seasonality and Festival Awareness

- Introduce seasonal multipliers for festival periods (e.g., Diwali, Prime Day).
- Use calendar-aware models to adjust prices based on upcoming events

6.3 Competitor Expansion

- Extend scraper support to Flipkart, Snapdeal, or niche platforms
- Aggregate competitor data from multiple sites for a more robust view

6.4 Dashboard and User Accounts

- Implement user login system, personalized product tracking
- Provide a dashboard with analytics, historical price trends, and alerts

6.5 Scheduled Batch Pricing

- Allow users to run the system on a cron job or schedule-based trigger for automatic daily price updates

6.6 Integration with E-Commerce APIs

- Auto-update prices to seller dashboards (Amazon Seller Central, Shopify, etc.) using public APIs

CHAPTER 7
REFERENCES:

1. Google Trends API. [Online].

Available:https://trends.google.com/

2. PyTrends API Documentation – General Mills. [Online].

Available: https://github.com/GeneralMills/pytrends

3. BeautifulSoup Documentation – Leonard Richardson. [Online].

Available:https://www.crummy.com/software/BeautifulSoup/bs4/doc/

4. Baabak, "Dynamic Pricing Using Machine Learning," *Medium*, [Online].

Available: https://medium.com/@baabak/dynamic-pricing-using-machine-learning-5e882282effe

5. Scraping Amazon Product Information Using BeautifulSoup, *GeeksforGeeks*, [Online].Available:vhttps://www.geeksforgeeks.org/scraping-amazon-product-information-using-beautiful-soup/

6. Dynamic Pricing Algorithm Overview, *AI Multiple Research*, [Online].

Available: https://research.aimultiple.com/dynamic-pricing-algorithm/