**E-Commerce Product Classification and Analysis**

By Jaisurya | Data Science Project | June 2025

**Objective:**

The objective of this project is to scrape real-time product data from Amazon, clean and preprocess it for analysis, and perform exploratory data analysis to uncover key trends. The project applies unsupervised clustering to identify natural product groupings and builds supervised models to classify products. These models are tuned for optimal performance, evaluated using standard metrics, and the best model is saved for future use.

**Problem Definition:**

Manual product classification in e-commerce platforms is slow, inconsistent, and error-prone, especially with large inventories. The goal of this project is to build an automated classification system that predicts product categories using attributes like price, brand, rating, discount, and reviews.

**Data Collection:**

* Source: The dataset was created by scraping product data from Amazon India.
* Method: Web scraping was done using the requests library and HTML parsing techniques (without using BeautifulSoup in the final code).
* Features Collected:
* **Product Name:** Name or title of the product listed.
* **Price:** Listed price of the product at the time of scraping.
* **Discount:** Percentage or amount of discount applied.
* **Brand:** The brand or manufacturer of the product.
* **Category (target):** Product category label used as the target for classification.
* **Rating:** Customer rating of the product (if available).
* **Number of Reviews:** Count of customer reviews for each product.
* **Storage:** Collected data was stored in a **Pandas DataFrame**.

**Data Cleaning:**

* Handled Missing Values: Identified and addressed missing data in key columns like *Price*, *Rating*, and *Number of Reviews*.
* Corrected Data Types: Converted columns (e.g., *Price* to float, *Rating* to numeric) for accurate analysis.
* Removed Duplicates: Eliminated duplicate product entries to ensure data integrity.
* Normalized Data: Standardized text data (e.g., *Brand*, *Category*) for consistency.
* Cleaned Price & Discount Fields: Removed symbols (₹, %) and converted to numeric values.
* Outlier Detection**:** Checked for outliers in numerical columns (such as price, rating, and discount) using visualization and statistical methods.

**Database Storage:**

* Database Storage: The cleaned data was stored in a MySQL database for future use.

**Exploratory Data Analysis:**

* Distribution Analysis: Analyzed the distribution of numerical features like price, discount, rating, and number of reviews.
* Category Trends: Explored the most common product categories and their pricing patterns.
* Brand Insights: Identified top brands and compared their average prices and ratings.
* Correlation Study: Examined relationships between features (e.g., price vs. rating, discount vs. price).
* Visualization: Created bar charts, histograms, and scatter plots to uncover patterns and trends.

**Insights from EDA:**

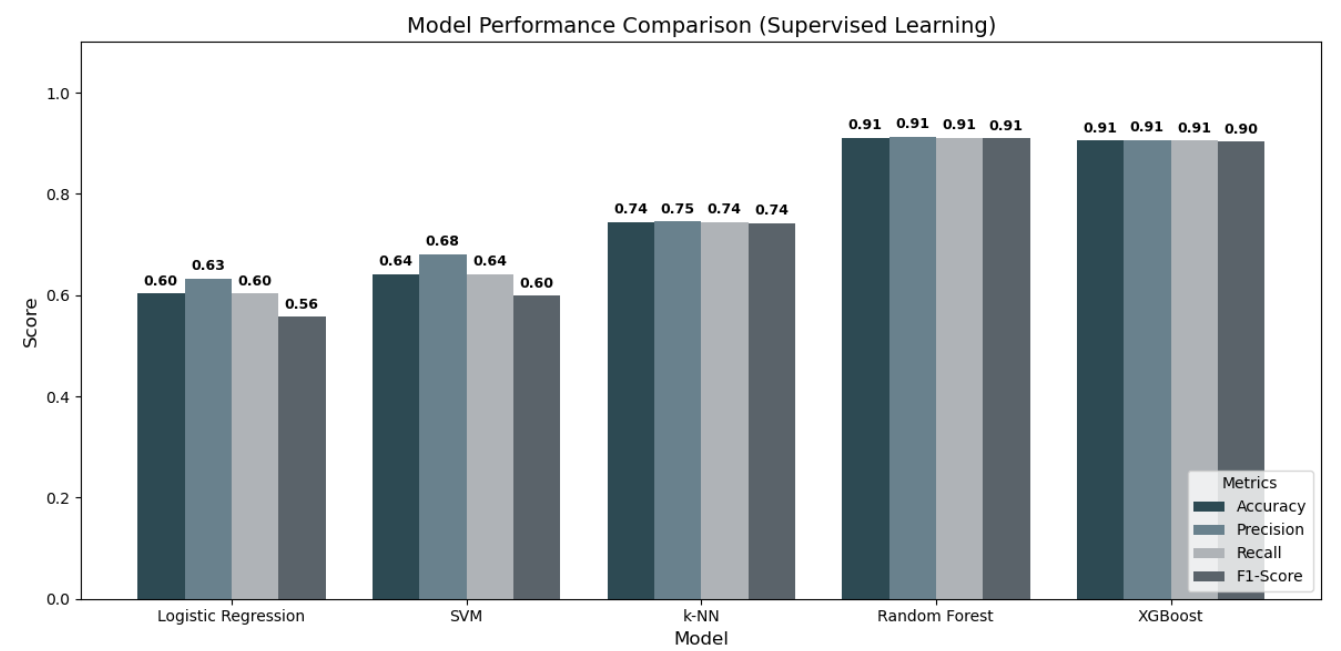
* Most product reviews fall between 3.7 and 4.2, with an average rating of 4, indicating overall positive customer feedback.
* Mobile phones and headphones received consistently good reviews, averaging around 4 stars.
* Laptops and mobiles were generally priced higher compared to other product categories.
* Smartwatches and headphones offered higher discounts, reflecting competitive pricing strategies in these segments.

**Unsupervised Learning:**

* Algorithm used: KMeans clustering with n\_clusters = 2 to group products.
* Dimensionality Reduction: Applied PCA to reduce feature space for 2D visualization of clusters.
* Clusters identified: Grouped products into clusters representing similar characteristics (e.g., premium vs budget products).
* Insight: Helped uncover natural product groupings based on features like price, rating, and discount without relying on labels.

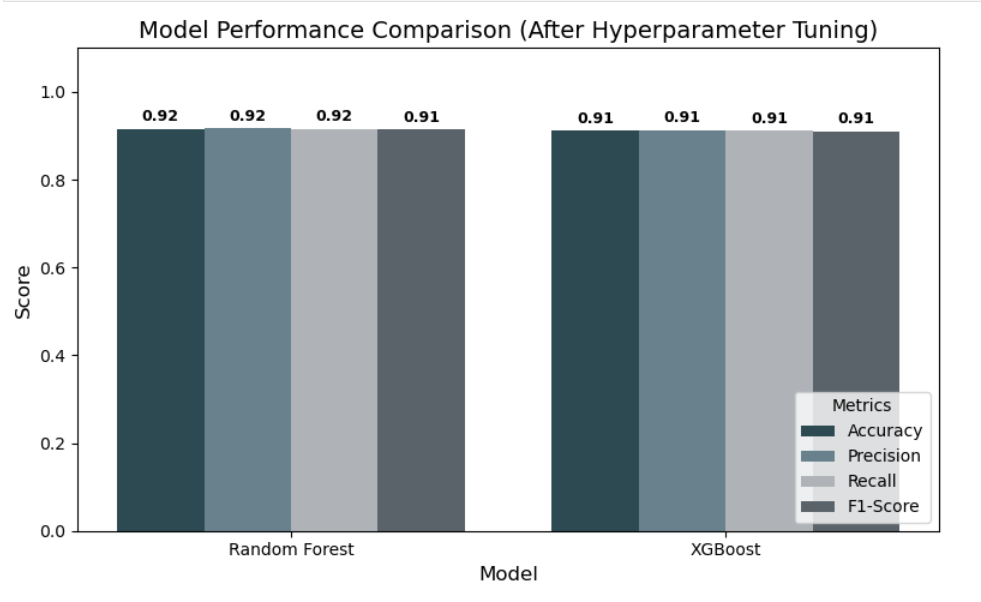
**Supervised Learning:**

* Built multiple models to classify products into their respective categories.
* Models used: Logistic Regression, SVM, k-NN, Random Forest, and XGBoost.
* Initial accuracies achieved:
* Logistic Regression: 60%
* SVM: 64%
* k-NN: 74%
* Random Forest: 91%
* XGBoost: 91%
* Evaluation metrics included accuracy, precision, recall, and F1-score to measure performance.



**Hyperparameter Tuning:**

* Used RandomizedSearchCV for hyperparameter tuning with cross-validation.
* Applied StratifiedKFold to ensure balanced class splits during validation.
* Tuned models: Random Forest and XGBoost.
* Parameters tuned included n\_estimators, max\_depth, min\_samples\_split, min\_samples\_leaf, learning\_rate, subsample, colsample\_bytree, gamma, reg\_alpha, and reg\_lambda.
* Result: Achieved improved and validated high-performing models.



**Model Saving:**

* The best Random Forest model was saved using pickle for future deployment.

**Future Scope:**

* Deploy the model as a web app using tools like Streamlit or Flask for real-time product classification.
* Extend the scraping pipeline to cover other e-commerce platforms like Flipkart or Snapdeal for richer datasets.
* Integrate CNN**s** or other deep learning models for image-based product classification.
* Explore multi-modal learning by combining numeric, text, and image features for enhanced accuracy.

**Conclusion:**

* Successfully built a machine learning pipeline for product classification from data scraping to model saving.
* Automated the product classification process to reduce manual effort and improve consistency.
* Performed exploratory data analysis to uncover trends in pricing, discounts, ratings, and reviews.
* Achieved high accuracy using tuned models like Random Forest and XGBoost.
* Prepared the model for future deployment in real-world e-commerce applications.
* Demonstrated the use of both unsupervised and supervised learning for comprehensive analysis.
* Established a scalable framework that can be extended to include more features or platforms.