AMAZON

ELECTRONICS

WEB SCRAPING–

CAPSTONE PROJECT

INTRODUCTION:

This capstone project represents a comprehensive, end-to-end Data Science workflow, focused on analyzing and extracting insights from the electronicsproduct domain using real-world e-commerce data. The primary objective is to explore customer preferences and product characteristics through data-driven techniques and predictive modeling, thereby supporting strategic decisions in e-commerce retail.

The project began with web scraping, where product-level data such as names, prices, ratings, number of reviews, and categories were extracted from Amazon’s electronics listings using Python libraries like BeautifulSoup and requests. Over 20,000 raw data points were initially collected for analysis.

The project employed both Unsupervised Learning techniques to discover natural product groupings and Supervised Learning algorithms to predict product rating classes. This capstone project showcases the real-world application of data science tools and methodologies, from data acquisition to actionable insights, demonstrating the capability to solve business problems using analytical thinking and technical implementation.

WEB SCRAPING

Web scraping was a foundational part of this capstone project, enabling the creation of a rich and diverse dataset for analysis and machine learning. The data was collected directly from Amazon’s Electronics category, focusing on extracting key product features such as the product name, price, customer rating, and number of reviews. To gather real-world product data for this project, web scraping was performed using Python’s requests and BeautifulSoup libraries which were used for sending HTTP requests and parsing HTML content, respectively. These tools allowed for efficient navigation through multiple pages and structured extraction of data elements from the webpage. A total of 20,000 electronics product listings were targeted and collected. The scraping logic was designed to iterate over multiple product listing pages, collecting data until the desired number of records was reached. Key considerations included:

* Simulating browser behavior using custom headers to avoid request blocking.
* Navigating pagination to access thousands of products.
* Handling missing values gracefully by assigning a default placeholder such as “Not Available”.
* Aligning all extracted attributes (names, prices, ratings, and reviews) correctly to maintain data consistency.

To reduce the risk of being blocked or flagged by Amazon’s servers, a delay was added between requests. This helped mimic human browsing behavior and ensured that scraping could continue over an extended period.

The scraping script was made robust with exception handling to bypass any network issues, incomplete loads, or unexpected changes in page structure. If an error occurred while loading a page, it would be logged, and the script would proceed to the next available page. The scraping process involved automating page navigation and extracting the required fields, including:

* Product Name
* Price
* Rating
* Number of Reviews

To ensure smooth data retrieval and avoid getting blocked, appropriate headers with a user-agent string were set. A while loop was used to paginate through the results until the desired number of records was collected. Below is a breakdown of the key components of the scraping logic:

while len(Names) < MAX\_ITEMS:

url=f'https://www.amazon.com/s?k=electronics&page=

{page\_num}'

response=requests.get(url,headers=HEADERS1) soup=BeautifulSoup(response.content,"html.parser")

names = soup.find\_all('a', class\_='a-link-normal s-line-clamp-2 s-link-style a-text-normal')

prices=soup.find\_all('span',class\_='a-offscreen')

ratings = soup.find\_all('i', class\_='a-icon a-icon-star-small a-star-small-4-5')

review\_count = soup.find\_all('span', class\_='a-size-base s-underline-text')

After the scraping and initial filtering processes, a total of 15,303 clean and unique product entries were successfully collected. This dataset laid the groundwork for all subsequent steps, including preprocessing, unsupervised clustering, classification, and visualization.

DATA CLEANING

To prepare the scraped dataset for effective analysis and modeling, we have to ensured data consistency, relevance, and accuracy. The raw data contained several missing entries in product price, rating and number of reviews initially filled with "Not Available", and later filled with median appropriately for analysis. To enable quantitative analysis in price column dollar symbols were removed and converted to float. In Number of Reviews column commas were stripped out and values were converted to integers. In rating the databconverted from strings (like "4.5 out of 5") to float for model training and grouping.

A large number of product entries were repeated across multiple pages. Nearly 1697 out of 20,000 entries were duplicates. This ensured a high-quality, unique dataset of electronic products. The Product Name field often contained lengthy descriptions, brand names, and product colors. To retain only the most relevant keywords we extracted the words up to the second comma, which generally contain the core product description. Additionally, we extracted the word after the last comma, as this typically represents the color of the product. This approach helped reduce noise and standardize product names for clustering, classification, and visualization tasks. The outliers were detected and removed. Hence the data left with 15303 rows which ensures uniqueness and reliability.

DATABASE STORAGE

Once the data was thoroughly cleaned and preprocessed, it became essential to store it in a structured format for further analysis and modeling. To achieve this, SQLAlchemy, a Python-based SQL toolkit was used.

SQLAlchemy played a vital role in bridging the gap between Python and SQL databases. It provided a convenient and efficient way to move the cleaned data from a Pandas DataFrame into a relational database system. In this project, a lightweight and easy-to-use database like SQLite was utilized during development for simplicity and speed.

The cleaned dataset which included key attributes like product name, price, rating, number of reviews, and classification labels was saved as a table within the database. This made it possible to perform fast, reliable SQL queries and ensured the data was persistent and organized.

Using SQLAlchemy allowed seamless integration of data storage with Python's data processing capabilities, laying the groundwork for a scalable architecture that can later be extended to cloud-based or production-level databases such as MySQL or PostgreSQL.

Overall, this step helped structure the data for easy access, supported efficient querying, and prepared the foundation for downstream analytics and machine learning tasks.

UNSUPERVISED LEARNING

The unsupervised learning was applied as an exploratory step to segment electronic products based on shared characteristics. Since the dataset originally lacked explicit target labels, clustering helped identify underlying groupings among the products. To segment the product data into meaningful groups, we applied the K-Means clustering algorithm. This technique helps uncover patterns in data by assigning each product to one of several clusters based on its features. Before clustering, we selected key numerical features such as: Product Price, Rating, Number of Reviews. These features were scaled to ensure that no variable dominated due to its range, which is essential for distance-based algorithms like K-Means. To determine the optimal number of clusters (K), we used the Elbow Method. We trained K-Means for different values of K (from 1 to 14). We calculated the WCSS (Within-Cluster Sum of Squares) for each. By plotting WCSS vs K, we observed an 'elbow' shape in the curve. The point where the curve starts to flatten indicates the optimal K. In this case, the elbow was observed at K = 4, suggesting 4 distinct clusters within our dataset. We then applied K-Means using 4 clusters. Each product was assigned a Cluster label (from 0 to 3). These cluster labels were added to the dataset for further analysis and visualization. By examining the cluster assignments and corresponding product data. We found that products were grouped by characteristics such as price range and popularity. For example, one cluster included affordable items with moderate reviews, while another had premium, high-rated gadgets. This clustering provided meaningful product groupings and was later used as a target for supervised learning classification, allowing us to train models that predict a product's cluster based on its attributes. As conclude that cluster 2 is the highest among other from that “Etekcity smart scale for body weight” is highly in demand and also it has maximum reviews and indication of popularity.

SUPERVISED LEARNING

In the supervised learning phase of our project, we aimed to build classification models capable of predicting product cluster labels, which were previously derived using unsupervised K-Means clustering. To train these models, we selected three significant features from the dataset Product Price, Rating, and Number of Reviews. These features were considered as they play a key role in determining product popularity and customer perception. The target variable for classification was the Cluster label, which segments products into distinct groups based on shared attributes.

We started by splitting the dataset into training and testing subsets about 80% and 20%. Before feeding the data into the models, we applied feature scaling using StandardScaler to normalize the range of the input variables, which helps improve the performance of distance-based algorithms and accelerates convergence in optimization-based models. Following this, we implemented several popular classification algorithms: Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (k-NN), Random Forest, and XGBoost. Each model was trained on the scaled training data and evaluated on the test data using Accuracy and F1 Score as the primary performance metrics.

Among all the models, Logistic Regression yielded the best results with an accuracy and F1 score of approximately 99.93%, showcasing its ability to effectively capture the underlying patterns in the data. To further improve this model, we applied GridSearchCV to perform hyperparameter tuning. This configuration achieved a best cross-validated accuracy of 99.88%.

The results demonstrate the robustness of our classification pipeline and its ability to predict product clusters with high precision, making it a reliable solution for categorizing electronics based on consumer-driven features.

KEY INSIGHTS

1. The most in-demand product is *"Etekcity Smart Scale for Body Weight and Fat Percentage"*, which stands out with 135,122reviews and 4.5 out of 5 rating.
2. Cluster 2 contains products with the highest number of reviews, indicating it houses the most popular and well-received products in the dataset.
3. Products in Cluster 3 maintain consistently good ratings while being cost-effective, appealing to price-sensitive yet quality-conscious customers.
4. A positive correlation exists between product ratings and number of reviews, suggesting that satisfied customers are more likely to engage and leave feedback.
5. The clustering approach successfully grouped products based on customer engagement patterns, helping identify distinct market segments.
6. Products with ratings above 4.5 tend to dominate in review count, indicating trust and credibility are closely tied to rating scores.
7. Shoppers are likely prioritizing both affordability and reliability, as seen by the highly rated, moderately priced products being frequently reviewed.
8. Using features like price, rating, and review count proved effective for both clustering and classification, showcasing a strong foundation for product recommendation systems.

CONCLUSION

This project explored the application of both unsupervised and supervised machine learning techniques to analyze a dataset of electronic products scraped from Amazon. Through a data cleaning process we ensured the dataset was well-prepared for insightful analysis. Special care was taken to extract meaningful components from product names, such as the key descriptive words and color attributes, which added more depth to the data.

Overall, the project demonstrates how machine learning can be a powerful tool in e-commerce analytics. By uncovering patterns in product performance, customer preference, and pricing strategies, businesses can make data-driven decisions to optimize marketing, inventory, and product recommendations. The success of both clustering and classification models also shows the strength of using structured product data and review metrics in predictive modeling. Future improvements may include incorporating natural language analysis on customer reviews or extending the dataset to include temporal trends.