TV TELEVISION PRICE PREDICTION AND CLUSTERING USING WEB SCRAPING AND MACHINE LEARNING

Submitted by

SRI HARIHARAN CR

Data Analytics & Data Science

KGiSL Micro College, Coimbatore – 641035.

1.Data Collection via Web Scraping

To begin the project, data was collected directly from Flipkart, one of India's largest e-commerce platforms, using web scraping techniques. The target dataset consisted of laptop product listings, which included attributes such as:

Key Data points

Product name
Price
Rating
Review
Os
Support apps
Sound system

Here BeautifulSoup were used to extract data from Flipkart's laptop section. Selenium handled the dynamic content loading and page navigation, while BeautifulSoup parse the HTML structure to consistently and accurately fetch relevant product details across multiple pages. The final scraped data was stored in a CSV file format for further analysis.

2.Data Preprocessing

After collecting the raw data through web scraping, preprocessing was carried out to ensure the quality and consistency of the dataset. This process involved appropriately handling missing or null values, standardizing data formats—such as converting prices and review counts into numeric values—and removing duplicate entries. These steps were essential to prepare a clean and reliable dataset for further analysis and modeling.

3.Exploratory Data Analysis (EDA)

After preprocessing, EDA was performed to understand the structure, distribution, and relationships within the data. Methods like .info() and .describe() provided insights into data types, missing values, and statistical summaries, while visual tools such as boxplots helped identify and manage outliers. Univariate, bivariate, and multivariate analyses were conducted using histograms, scatter plots, heatmaps, and other visualizations to uncover meaningful patterns and trends in variables like price, ratings, and discounts

Here various visualization techniques such as countplot, histograms, bar charts, and scatter plots were utilized to understand the distribution and relationships within the dataset. The countplot showed that most Tv television have ratings between 4.0 and 4.5, indicating overall customer satisfaction. Histogram analysis revealed that discounts typically range from 10% to 40%, reflecting common pricing strategies. A bar chart comparing top-rated laptop titles demonstrated that high customer ratings are not always dependent on large discounts. Finally, scatter plots showed a positive trend between rating and review count, and revealed that both low and high-priced tv television can maintain strong ratings across various discount levels

4.Data Storage

After cleaning, you will set up a relational database to store your refined data. Using SQLAlchemy or similar libraries, you will push the cleaned dataset into the database for future access and analysis

5.Unsupervised Machine Learning

Here focused on identifying potential clusters within the dataset, which could help in understanding product categories or customer preferences based on various attributes. The following steps were taken to perform unsupervised learning
.

Label Encoding: We used Label Encoding to convert categorical variables, such as the product titles, into numeric labels. This was necessary for the clustering algorithm to work effectively, as it requires numerical data.

Data Normalization: To ensure that the features were on a similar scale and did not bias the clustering algorithm, we applied Standardization using the StandardScaler. This step transformed the data so that all features had a mean of 0 and a standard deviation of 1, making the data more suitable for clustering.

K-Means Clustering: We then applied the KMeans Clustering algorithm to find patterns and group the data into clusters. After experimenting with different values of k, the optimal number of clusters was found to be 2, which was considered the most meaningful and interpretable outcome.

Model Evaluation: To evaluate the clustering quality, we calculated the Silhouette Score, which measures how similar the objects within a cluster are compared to objects in other clusters. A higher Silhouette Score indicates well-defined clusters. The Silhouette Score confirmed that 2 clusters were the best choice for the dataset.

To visualize the results of the KMeans clustering, a scatter plot was created, where each data point was color-coded based on its assigned cluster. This provided an easy-to-understand graphical representation of how the data points were grouped into two distinct clusters. The scatter plot not only confirmed the clustering output but also allowed us to see how well the model divided the data based on the features.In the analysis, two distinct clusters were identified using KMeans clustering.

Cluster 0 represents more affordable laptops, which are generally priced lower than average, often with lower ratings, fewer reviews, and higher discounts. These laptops are likely budget-friendly options that may appeal to price-conscious consumers looking for cost-effective choices.

Cluster 1 represents premium laptops, which tend to have higher ratings, more reviews, and higher prices. These laptops are generally associated with premium features and better performance, likely appealing to consumers willing to invest more in advanced technology. The higher discount percentages in Cluster 1 suggest that these premium laptops may still be available at significant markdowns to attract customers in competitive markets

6.Supervised Machine Learning

In this phase, several machine learning models were evaluated to predict the clusters obtained from the unsupervised learning phase. The models tested included Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and XGBoost. These models were trained on the scaled data, and Their performance was evaluated based on accuracy, precision, recall, and F1-score. The results of each model were as follows:

XGBoost achieved an accuracy of 96.7%, with a high precision and recall for both clusters, indicating the model's ability to effectively classify both affordable and TV Television .

SVM and KNN both had an accuracy of 81.00%, with SVM showing a slightly better performance in classifying cluster 1.

Random Forest yielded the highest performance with an accuracy of 98.29%. This The model showed the best balance between precision and recall for both clusters.

Logistic Regression also reached an accuracy of 78.29%, with excellent classification results across all metrics.

The Random Forest model was selected as the best-performing model due to its balanced precision and recall, providing high accuracy with minimal bias between the two clusters. The classification reports for each model demonstrated their strengths in predicting the clusters, particularly in classifying premium laptops and affordable laptops. These results highlight the potential of using machine learning for understanding customer preferences and segmenting products effectively.

7.Random Forest Classifier - Hyperparameter Tuning

To improve the performance of the Random Forest Classifier, hyperparameter tuning was performed using GridSearchCV. The goal was to find the optimal set of hyperparameters that would yield the highest model performance on the given dataset. The key hyperparameters tuned for the Random Forest model were:

max_leaf_nodes: The maximum number of leaf nodes in the decision trees.
min_samples_leaf: The minimum number of samples required to be at a leaf node.
n_estimators: The number of trees in the forest.

After conducting the search, the best hyperparameter combination was found to be:
max_leaf_nodes = 45
min_samples_leaf = 2
n_estimators = 50

The model was evaluated based on its performance on the test set, and the following results were achieved:

Accuracy: 98.29%
Precision: 98% (Class 1) / 100% (Class 0)
Recall: 100% (Class 1) / 94% (Class 0)
F1-Score: 98.1% (Macro average)

The hyperparameter tuning significantly improved the performance of the model, making it the best-performing classifier in terms of both accuracy and other classification metrics.

8.Model Hyperparameter Tuning and Saving the Final Model

After tuning the Random Forest model using GridSearchCV, the best-performing model was selected based on the hyperparameters: {'max_leaf_nodes': 45, 'min_samples_leaf': 2, 'n_estimators': 50}.

To preserve the final model for future use, it was saved into a .pkl file, allowing easy access and deployment without retraining. This ensures that the model can be loaded and reused for making predictions or further evaluations. The accuracy, precision, and recall obtained during training remain consistent when the model is reloaded from the saved file.