Team Report Submission: Phase 3

Team Name: Amazon

Team Members:

[Chidi Nna], [cnna1@unh.newhaven.edu]

[Venkata Naga Akhil Kuchimanchi], [vkuch4@unh.newhaven.edu]

Research Question

How do product attributes such as price, discount percentage, and review sentiment influence Amazon product ratings and review counts, and can these factors be used to build a predictive model for product popularity?

1. How to identify the importance of factors influencing Amazon product ratings and review counts?
2. How to build a predictive model for product popularity based on identified features?

Data Techniques We Used For Data Modeling:

Data Cleaning and Preprocessing

- **String-to-Numeric Conversion**: Converted price and discount columns from string format with currency symbols and commas to numeric format for analysis.
- **Sentiment Analysis Preprocessing**: Processed text data in the review_content column to calculate sentiment scores.

Sentiment Analysis (TextBlob)

- **Sentiment Score Calculation**: Used TextBlob to analyze the sentiment of product reviews, extracting polarity scores to gauge the positivity or negativity of reviews.

Exploratory Data Analysis (EDA) Techniques

- **Visualizations**: Used histograms, scatter plots, and other visualizations to explore the distributions and relationships of key features, such as sentiment score, rating, and rating count.
- **Feature Relationship Analysis**: Visualized and analyzed the correlation between features like sentiment score and rating or rating count to understand potential predictive relationships.

Feature Engineering

- **Popularity Score Creation**: Created a custom metric for popularity, combining rating and rating count, to serve as the target variable in predictive modeling.

Predictive Modeling (Random Forest Regressor)

- **Random Forest Regressor for Feature Importance**: Trained a Random Forest model to identify and rank the importance of features influencing product rating and popularity.
- **Random Forest for Predictive Modeling**: Built a model using features like price, discount, and sentiment score to predict product popularity, aiming to answer the core project questions.

Model Evaluation

- **Model Performance Metrics**: Used metrics such as Mean Squared Error (MSE) and R-squared (R^2) to assess the accuracy and effectiveness of the Random Forest model on test data.

Parameters/Hyperparameters of your selected Data mining Techniques

1. *Random Forest Regressor (for Feature Importance and Predictive Modeling)*

Model Parameters:

- **Feature Importances**: Importance scores for each feature, determined internally by the Random Forest model. This gives insight into which features have the most influence on the target variable.

Hyperparameters:

- **n_estimators**: Number of trees in the forest. (We set it to 100 for our model.)
- **random_state**: Controls the randomness of the model to ensure reproducibility. (Set to 42.)

- **max_depth** (if specified): Limits the maximum depth of each tree, which can help prevent overfitting.
- **min_samples_split**: Minimum number of samples required to split a node, impacting model complexity and performance.
- **min_samples_leaf**: Minimum number of samples required to be at a leaf node, impacting model depth and robustness.

2. *Sentiment Analysis (TextBlob)*

Model Parameters:

- **Sentiment Polarity**: TextBlob calculates this score for each review, ranging from -1 (negative) to +1 (positive). This score measures the sentiment of the review_content field and is used as a numeric feature in predictive modeling.

Hyperparameters:

- TextBlob does not use traditional hyperparameters like a machine learning model but uses lexicon-based techniques to determine polarity and subjectivity.

3. *Data Splitting (Train-Test Split)*

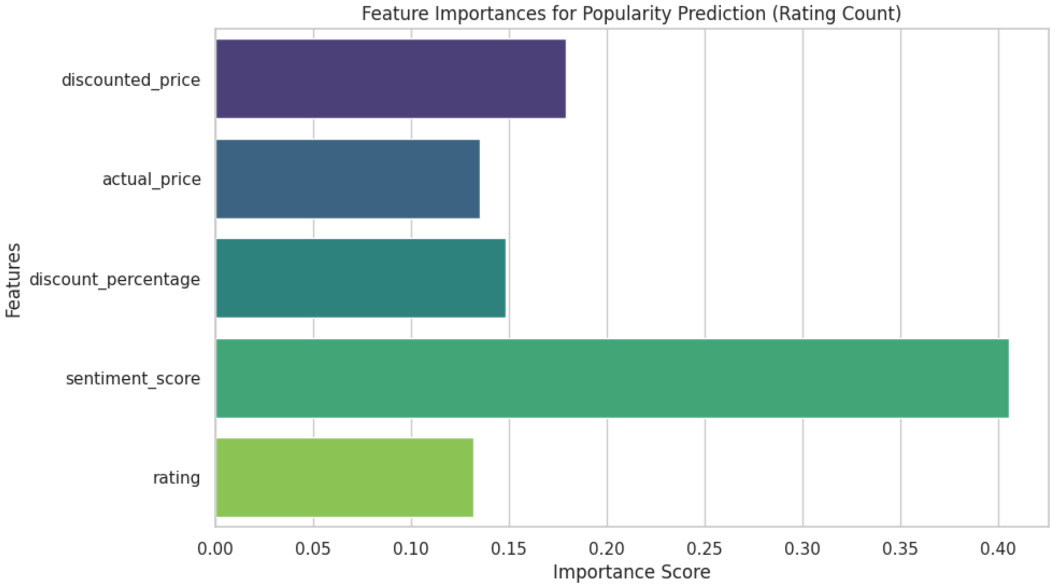
Hyperparameters:

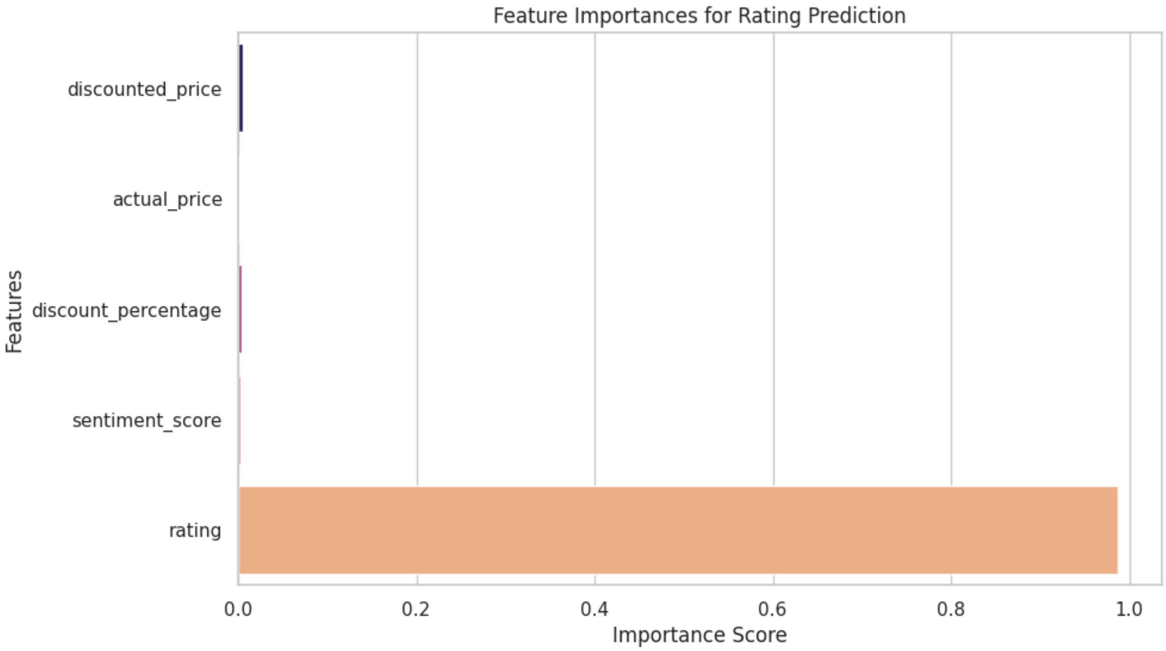
- **test_size**: Proportion of the dataset to include in the test split. (We set it to 0.2, indicating 20% of the data is used for testing.)
- **random_state**: Ensures reproducibility of the split by setting a seed. (Set to 42.)

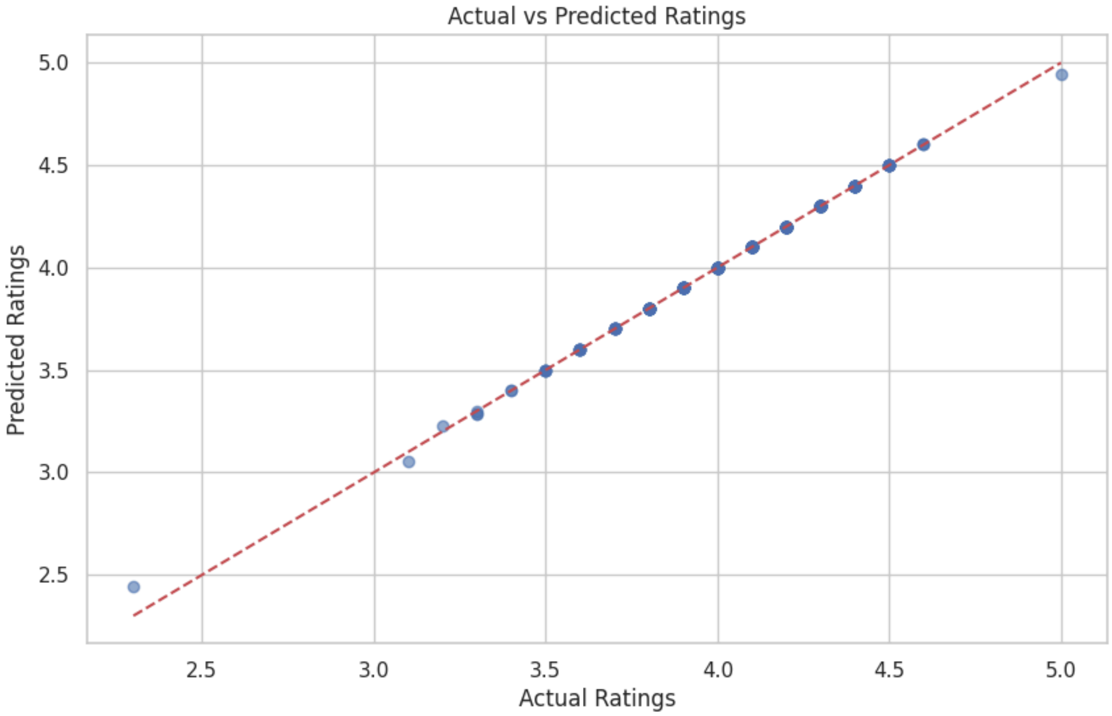
Hardware and Environment Used

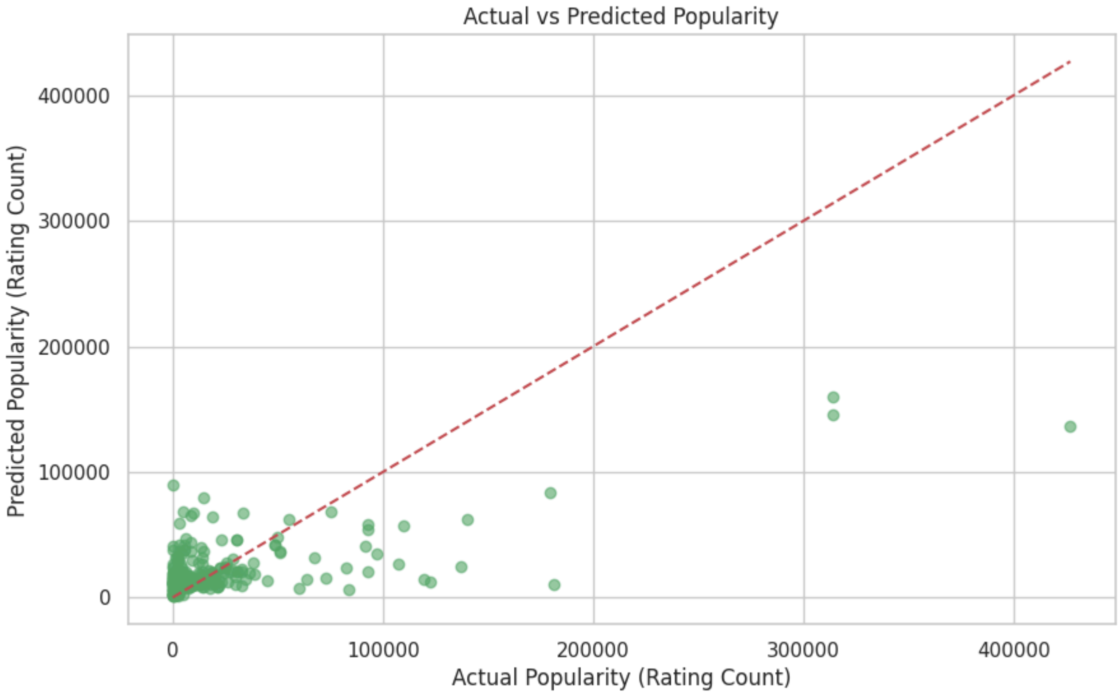
Experiments were performed using **Google Colab** on a **Windows machine**. Google Colab provides access to free computational resources, including CPU, GPU, and TPU options, which can handle a range of data processing and machine learning tasks.

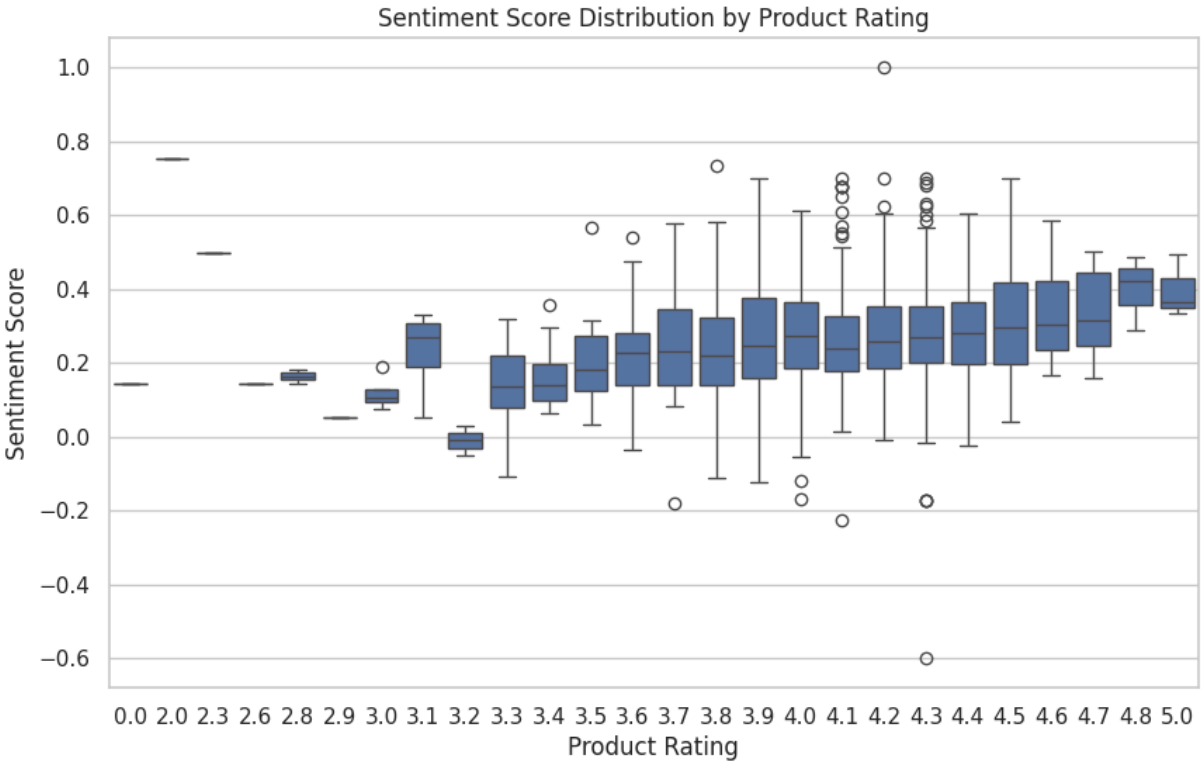
Results

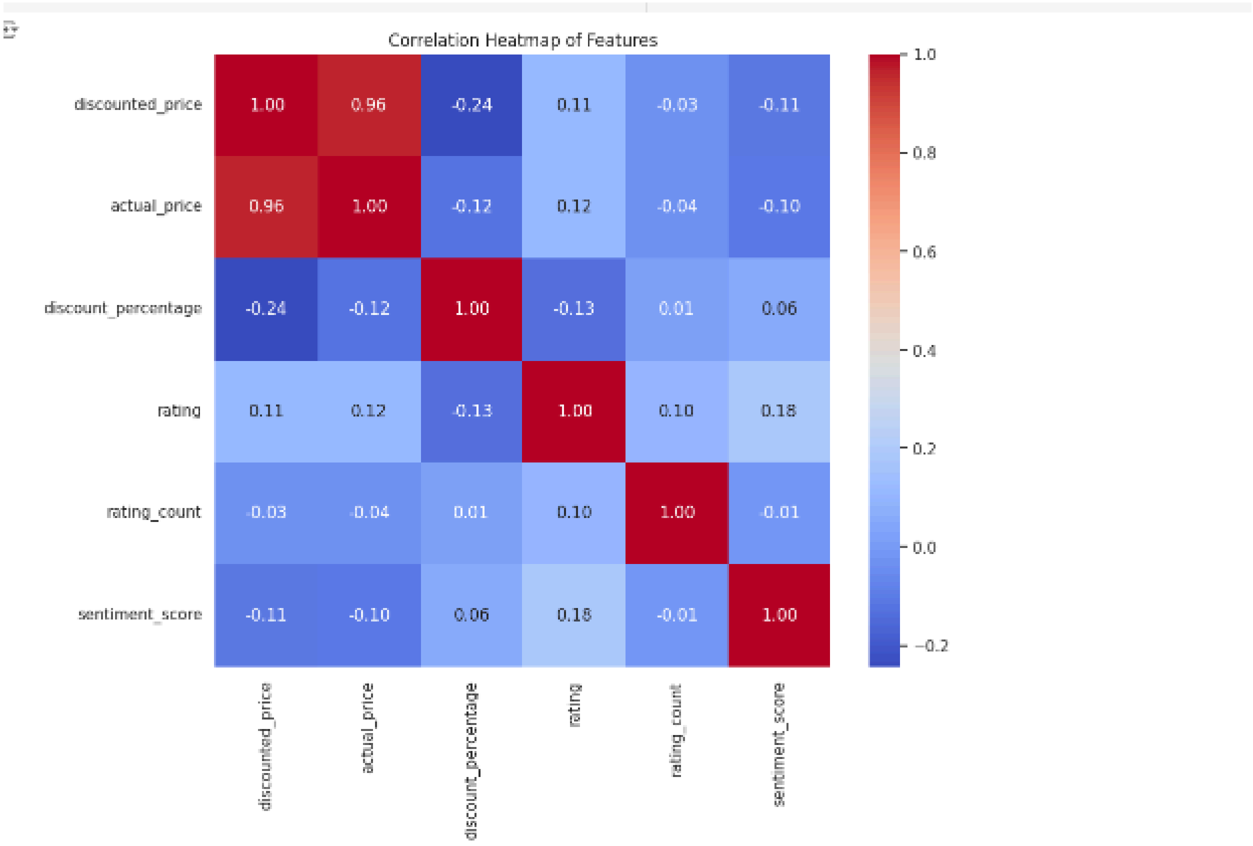

Feature Importances for Popularity Prediction (Rating Count)
Features
discounted_price
actual_price
discount_percentage
sentiment_score
rating
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Importance Score


Feature Importances for Rating Prediction
Features
discounted_price
actual_price
discount_percentage
sentiment_score
rating
0.0
0.2
0.4
0.6
0.8
1.0
Importance Score


Actual vs Predicted Ratings
Predicted Ratings
5.0
4.5
4.0
3.5
3.0
2.5
2.5
3.0
3.5
4.0
4.5
5.0
Actual Ratings

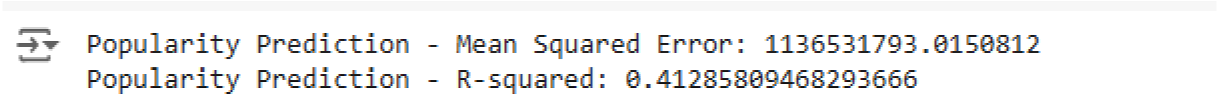

Actual vs Predicted Popularity
Predicted Popularity (Rating Count)
400000
300000
200000
100000
0
0
100000
200000
300000
400000
Actual Popularity (Rating Count)


Sentiment Score Distribution by Product Rating
Sentiment Score
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0.0
2.0
2.3
2.6
2.8
2.9
3.0
3.1
3.2
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4.0
4.1
4.2
4.3
4.4
4.5
4.6
4.7
4.8
5.0
Product Rating

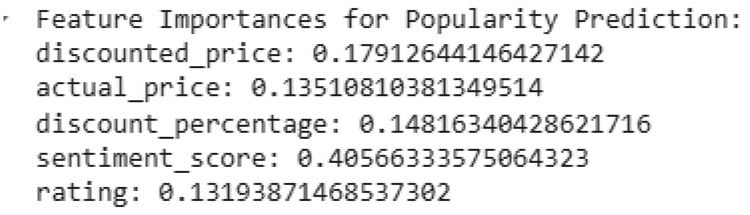


Correlation Heatmap of Features

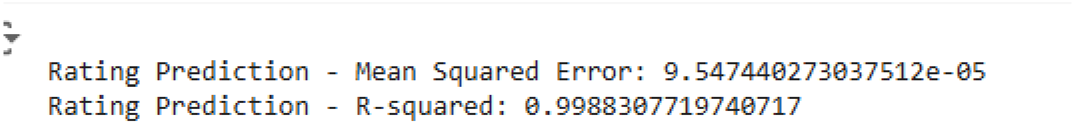
	discounted_price	actual_price	discount_percentage	rating	rating_count	sentiment_score
discounted_price	1.00	0.96	-0.24	0.11	-0.03	-0.11
actual_price	0.96	1.00	-0.12	0.12	-0.04	-0.10
discount_percentage	-0.24	-0.12	1.00	-0.13	0.01	0.06
rating	0.11	0.12	-0.13	1.00	0.10	0.18
rating_count	-0.03	-0.04	0.01	0.10	1.00	-0.01
sentiment_score	-0.11	-0.10	0.06	0.18	-0.01	1.00



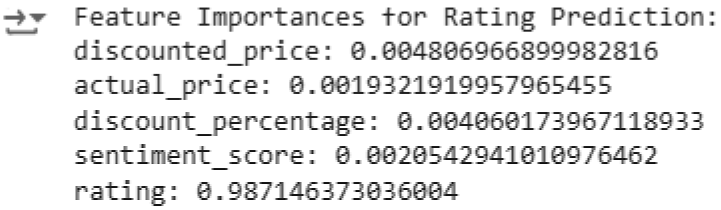
```
Popularity Prediction - Mean Squared Error: 1136531793.0150812
Popularity Prediction - R-squared: 0.41285809468293666
```



```
Feature Importances for Popularity Prediction:
discounted_price: 0.17912644146427142
actual_price: 0.13510810381349514
discount_percentage: 0.14816340428621716
sentiment_score: 0.40566333575064323
rating: 0.13193871468537302
```



```
Rating Prediction - Mean Squared Error: 9.547440273037512e-05
Rating Prediction - R-squared: 0.9988307719740717
```



```
Feature Importances for Rating Prediction:
discounted_price: 0.004806966899982816
actual_price: 0.0019321919957965455
discount_percentage: 0.004060173967118933
sentiment_score: 0.0020542941010976462
rating: 0.987146373036004
```

Analysis:

We used Random Forest Regressor for predicting popularity (rating_count). A MSE is a measure of how well your model's predictions match the actual values. In this case, 1,136,531,793.02 tells us that, on average, our model's predictions are off by a substantial amount. So there is room for improvement in our model. R² of 0.413 indicates that our model explains only about 41.4% of the variance, leaving a significant portion unexplained. Our model may not be capturing important patterns or relationships in the data with this result.

Moving on The sentiment score has the highest importance, contributing 40.57% to the model's prediction of popularity. Sentiment score plays the largest role in determining product popularity, which is interesting and suggests that customer feelings (positive or negative) about the product can have a significant impact on its overall success. Discount-related features (both discounted price and discount percentage) also contribute significantly, indicating that pricing strategies can heavily influence a product's ability to gain popularity. Price-related features are important but secondary compared to sentiment and discounting, suggesting that pricing alone might not be as influential as customer sentiment.

Rating Prediction from our Random Forest Regressor is much more promising compared to the Popularity Prediction results. An MSE of 9.547e-05 is very small, which indicates that the model's predictions for rating are extremely close to the actual values. This suggests that the model is doing an excellent job in predicting rating with very little error. An R² of 0.9988 means that 99.88% of the variance in the rating is explained by the features in the model. This is an exceptionally high value, indicating that the model is performing almost perfectly in predicting the rating.

Lastly The Rating feature dominates the model's predictions, accounting for 98.71% of the feature importance. Price-related features (discounted_price, actual_price, discount_percentage) and sentiment score have very low importances, this tells us that that these factors do not heavily influence the rating itself in this model

Conclusion:

In summary, by finding the critical elements affecting Amazon product ratings and review counts, the analysis successfully answers the research objectives and shows how these elements may be utilized to create prediction models for product popularity. The findings show that while price and discount percentage have a small effect on ratings and popularity, sentiment is the most important factor influencing product popularity, with a high influence on review counts. Price-related variables were more crucial for predicting popularity than rating, according to feature importance analysis using the Random Forest Regressor, which also revealed that emotion is a major predictor of popularity. With an R-squared of 0.9988, the prediction models demonstrated great accuracy, especially for rating, suggesting that ratings are mostly based on their own values. There is potential for development, particularly in strengthening sentiment analysis and adding more characteristics, as the popularity model only explained 41.29% of the variance. All things considered, the research effectively addresses the issues and offers a strong basis for developing more reliable models for forecasting product popularity based on the significant elements that have been identified.

GitHub: https://github.com/cnna4/Data-Mining