**Final Project**

ST494: Statistical Learning

Dr. Xu (Sunny) Wang

April 4th, 2025.

Diya Mistry, 169033002

Janaki Patel, 169037500

Sufiya Rahemtulla, 169018559

**Executive Summary**

Our analysis of Sephora’s product dataset focused on two main goals: identifying brands that offer the best value to consumers and predicting product ratings based on key features. Through careful cleaning and preprocessing, we prepared a dataset of 8,494 beauty products with variables, including prices, ratings, and popularity indicators. Using K-means clustering, we grouped brands based on average price and rating into four meaningful clusters: luxury brands, affordable good-value brands, budget brands, and lower-value brands. We found that affordable and good-value brands dominate Sephora’s offerings, aligning with a middle-class customer base that prioritizes quality and affordability over luxury pricing. Although some overlap existed, the clustering provided valuable insight into brand positioning and customer preferences.

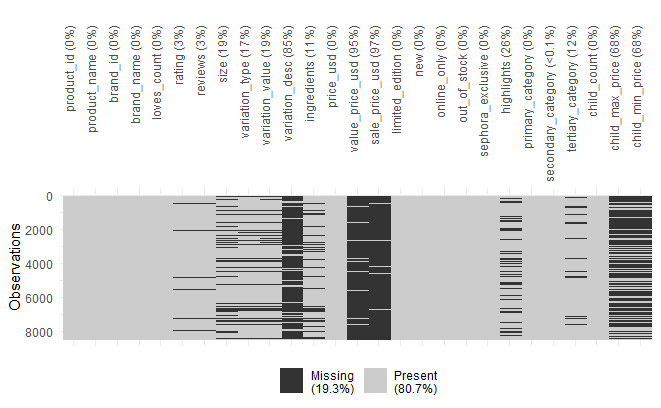
In the second part, we built a classification model to predict product ratings. Applying PCA helped simplify the data, and Random Forest models were trained both with and without class weights to handle the heavily imbalanced dataset. While the weighted model slightly improved predictions for rare ratings, it significantly reduced overall accuracy. Ultimately, the unweighted Random Forest achieved 67.8% accuracy, performing best for the dominant class (rating = 4) but struggling with minority classes. This reflects real-world patterns where most products receive average ratings, and highlights the challenges of predicting rare events like very high or very low ratings. These findings suggest that while basic product features like price and size offer some predictive power, more complex modeling or additional consumer behavior data may be needed for highly accurate rating prediction.

**About Our Dataset**

Our dataset provides detailed information on over 8,000 beauty and skincare products available through Sephora’s online store. It includes 27 attributes such as brand name, product name, price, and rating, as well as popularity indicators like loves\_count and number of reviews, which measure customer engagement. The goal of this project was to identify which brands offer the best value to Sephora customers and to explore whether it is possible to predict a product’s rating based on its features.

**Cleaning and Preprocessing**

To ensure accuracy of our classification and clustering analyses, we began our data preprocessing by addressing missing values and identifying irrelevant columns. We used “vis\_mis()” to detect and visualize missing data patterns.



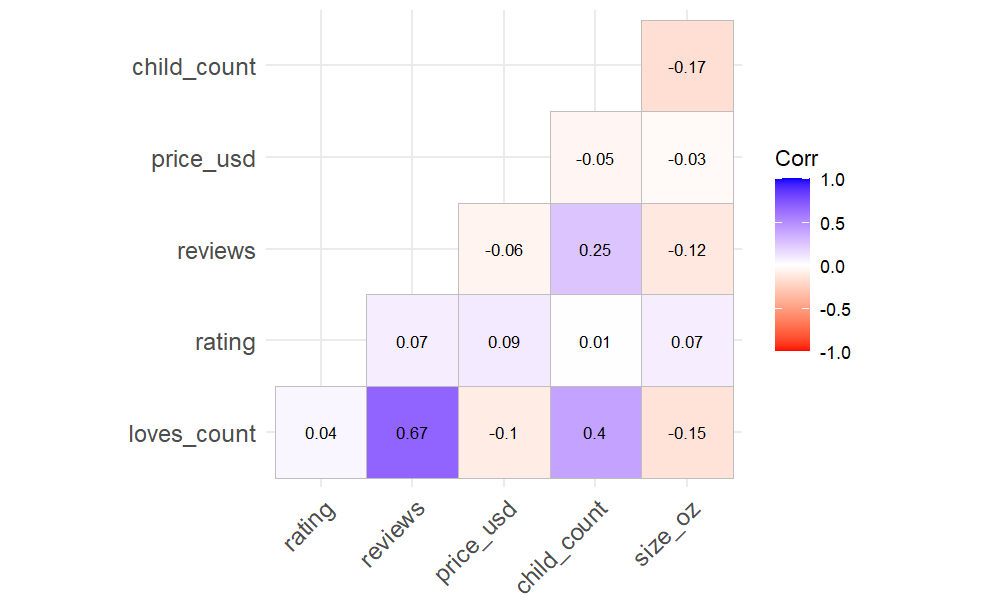
Columns including “value\_price\_usd”, “sale\_price\_usd”, “child\_max\_price”, and “child\_min\_price” had high percentages of missing data (>50%) and were also highly correlated with “price\_usd”, a target variable of interest; keeping all of these variables would have introduced distorted clustering. We did retain some variables with little amounts of missing values that could still provide insights. One challenge we ran in to with vis.mis() is that it was not detected many blank values as they were stored as empty strings (“”) rather than NA. We were able to address this by replacing empty strings with NA which helped us discover the many empty values in “variation\_desc”, “highlights”, etc. Overall, we removed the variables, “product\_id”, “ingredients”, “variation\_value”, and “variation\_desc”, “value\_price\_usd”, “sale\_price\_usd”, “child\_max\_price”, and “child\_min\_price”. These were all either irrelevant to our clustering/classifation goals, highly correlated with other variables, or had too many missing values, potentially skewing our models. We checked with missing values were associated with specific categories but did not discover anything significant.

Next, we applied one-hot encoding to the “primary\_category” and “secondary\_category” variables to convert these categorical features into a numerical format we could use in our models. Though it did expand the data, it allows the models to get a better understanding of each group member without any bias. After this, we rounded the “rating” variable to the nearest whole number between 1 and 5, as compared to their original decimal format, simplifying the feature to better reflect how users typically interpret product ratings and to align with classification models that could easily interpret and predict whole rating values. We then extracted numerical ounce values from the “size” field, while originally created inconstant text formats such as “1.7ox” or “8 x .02oz”. The new variable was created as “size\_oz”, and is easier for clustering and classification models to interpret, ensuring we’re comparing product volumes accurately and on the same weight scale. Further scaling for numerical values will be completed later in our use cases. All of these transformations help to reduce overall noise and address missing values to improve data consistency and allow our models to better capture meaningful patterns related to price segmentation and consumer preferences, ensuring a smooth and accurate downstream analysis.

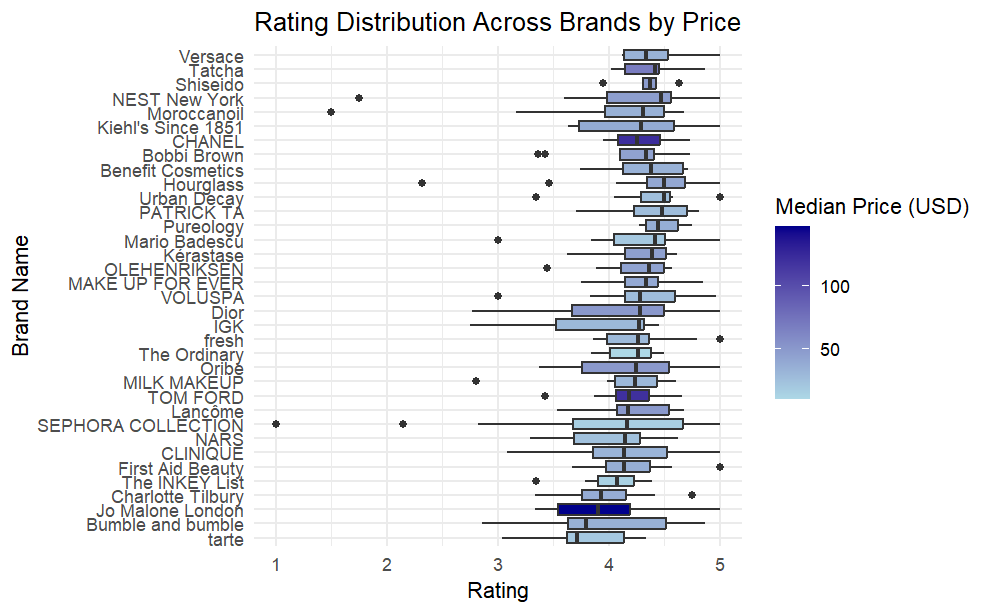
**Variable Exploration & Relationships**

**Cleaned Dataset Summary:** Our final cleaned dataset contains 8494 product listings across 71 variables, including key features like “price\_usd”, “reviews”, “loves\_count”, and includes parsed and converted variables such as “size\_oz” and “rating\_rounded”. Categorical features like “primary\_category” and “secondary\_category” are also included as dummy variables. We identified that some variables do still contain missing values (notably rating\_rounded, size\_oz, and reviews), which we will later address in our use cases. Overall, the dataset is now well-structured, with numerical and categorical features ready for clustering and modelling.

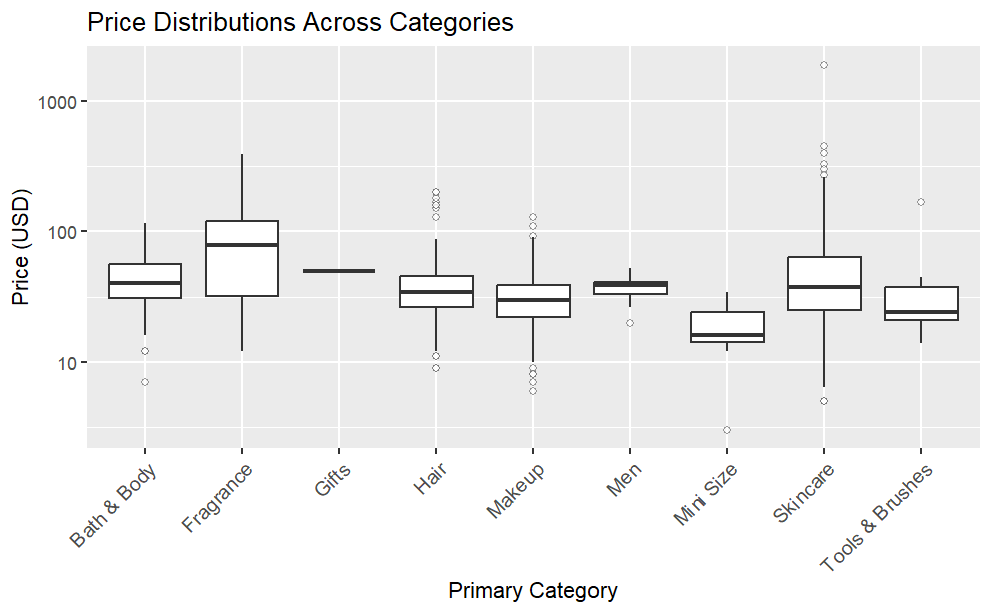
**Analyzing Correlations:** To further explore relationships between numeric variables in the dataset, we computed a correlation matrix. The results revealed that most variables are weakly correlated, indicating minimal linear relationships. The strongest correlation was between “loves\_count” and “reviews”, with r=0.67, which does make intuitive sense, as products with more reviews often have higher visibility and popularity, leading to more user engagement and loves. We also found a positive relationship between child\_count and loves\_count or r=0.4. This could indicate that products with more ‘children’ (various sizes, shades, etc.) are highly loves. From my experience, this is very reasonable, especially for face products such as foundation and concealer where a large shade range is important for a good collection. Most interestingly, “price\_usd” and “size\_oz” showed a very weak correlation (-0.03), suggesting that a higher price doesn’t always reflect larger product sizes, possibly due to the spectrum of budget vs luxury product pricing strategies. These findings are valuable for clustering and classification, as low correlations suggest that each variable could contribute uniquely to differentiating product categories. For instance, with clustering, the correlation analysis shows that brand\_id, reviews, and price\_usd are useful for clustering products based on popularity, value, and brand. Strong correlations between loves\_count and reviews suggest they can indicate popularity, while price\_usd helps differentiate product categories. Weak correlations with size suggest price alone isn't a strong predictor, so clustering will rely on combining these features. For classification of product ratings, the weak positive correlations between rating and reviews (0.07), rating and price\_usd (0.09), and rating and size\_oz (0.07) suggest that while there are slight relationships, these features alone may not be strong predictors of the rating. However, reviews and price\_usd could still provide some insight into rating trends, especially in differentiating between higher and lower ratings. The low correlations indicate that more complex models, incorporating other factors like loves\_count, might be needed for accurate classification.



**Rating Distribution and Price Across Brands:** Next, we explored a visualization “Rating Distribution and Price Across Brands” to examine the relationship between product ratings, brand identity, and price, using a sample of 1000 products to ensure the plot remained computationally efficient and interpretable. We focused on the 30 brands with the highest product counts to ensure we had meaningful comparisons. By plotting rating distributions across these brands and coloring them by median price, we aimed to understand whether highest-priced products tended to receive higher ratings. Interestingly, while premium brands like CHANEL and Tom Ford do show high average ratings, the overall trend suggests that price is not a consistent predictor of higher consumer satisfaction. For example, we see that Sephora Collection is at a much lower price range but still receives high ratings. This suggests that consumers may rate products based on perceived value rather than on price alone. These insights are important for our future clustering results, as they indicate that customers don’t always associate cost with quality, and for classification tasks predicting ratings, emphasizing that using price alone may not be a strong enough feature and we should incorporate combinations of variables for a more accurate prediction.



**Prices Across Categories:** This boxplot analysis was performed to explore how product prices vary across different primary categories, using a log scale to account for skewness and wide price ranges. The results show that skincare products have the high price variability and the highest overall price range, while Mini Size items are consistently lower in price, contradicting our earlier finding of a weak correlation between size and price. This suggests that category type may be a stronger predictor of price than size alone. From a classification perspective, especially in predicting price tiers or even ratings, category based price trends provide valuable context and can improve model performance by introducing meaningful variance.



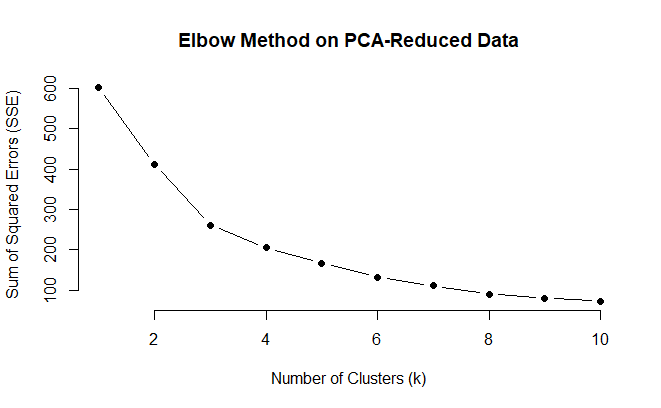
**Use Case 1: Determine what brand may be more worth purchasing**

**Objective:** To determine which brands provided the best value for their products in this case, we used the features brands, price, and rating. We grouped brands according to average pricing and average rating. We used k-means clustering to find 4 unique clusters that we may classify as luxury brands, good value brands, budget brands, and lower value brands.

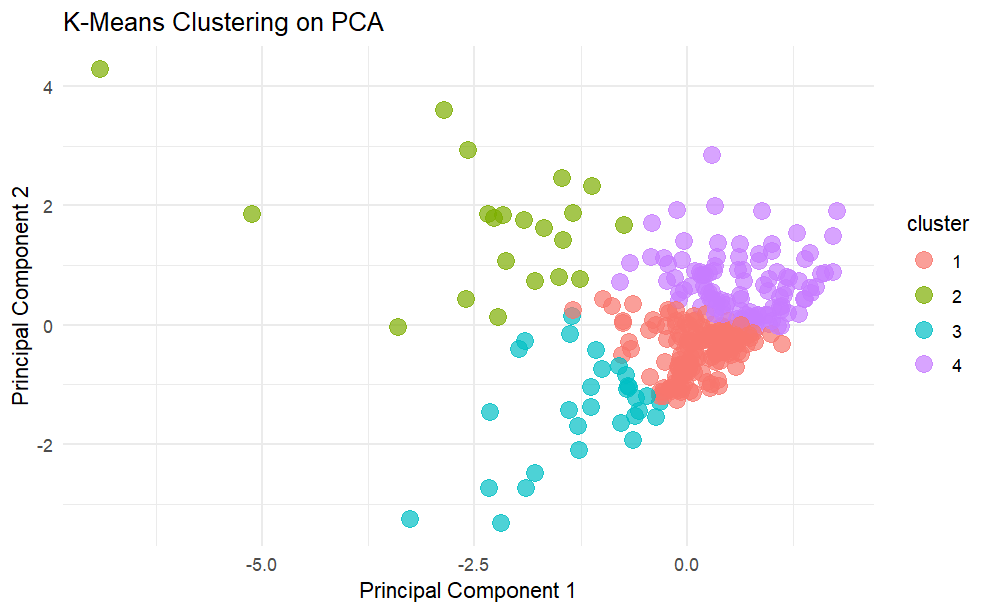
**Methodology:** For this use case we used k-means clustering, we believed this was a good choice for its interpretability to determine the different groups of brands as each brand is assigned to its nearest centroid. Since we are also only looking at three features (brand, price, and rating) it is small and easy for k-means to work with. In addition, we assessed cluster performance using the silhouette score to evaluate how effective the class assignments were done since there is no testing being done.

**Cleaning and Preparation:** We dropped rows with missing ratings. Since calculating average ratings required valid data, products with missing values could have skewed the results and would not accurately represent the brands. Because price and rating are on very different scales, we normalized both features before clustering. Without scaling, one variable could have dominated the clustering process. To simplify our analysis since brands have too many products, we grouped data by brand and calculated the average price and average rating per brand. Finally, before clustering we applied PCA to decorrelate the features and rotate the data to a new coordinate system making the clusters more spherical and easier to separate.

**Clustering Process:** To determine our optimal number of clusters we utilized the elbow method. Below on the graph we can see the sum of squared errors within clusters declined sharply up to k=4 at which it slowed down, suggesting that 4 was the best choice. To further confirm this was the best method we evaluated the silhouette score for k=3 and k=5 as well, confirming that k=4 is the best. Another method to choose k we tried was plotting the silhouette scores to determine what value would give the highest overall silhouette score. This method, however, did not work since we did not have very clearly separate data as brands tend to overlap in price and rating. Although the silhouette score did not help determine K it helped us evaluate the performance of the model. The average silhouette score was 0.68 meaning the clusters were reasonably well separated. While not perfect the separation was acceptable.



**Interpretation:** Each Cluster was characterized by its average price and average rating. Cluster 1(Affordable Brands): Have an average price of $38 and an average rating of 4.2, some of the brands in this cluster include 54 Thrones and ALTERNA Haircare, this cluster can be seen as affordable and good quality. Cluster 2(Luxury Brands): Have an average price of $197 and an average rating of 4.1, this includes brands such as Acqua di Parma appealing to customers who are willing to spend more on high-end products. Noticing that the ratings is almost the same as those of affordable brands is a good representation of our previous discovery that luxury does not indicate higher customer satisfaction. Cluster 3(lower-value brands): Has an average price of $51 and an average rating of 3.6. This cluster is more mixed containing brands like 19-69 which have higher end prices but lower ratings. Finally, Cluster 4(Good-Value Brands) highlights higher-end (not luxury) brands, with an average price of $54 and an average rating of 4.3, including brands such as ABBOTT. These brands offer slightly higher prices but deliver excellent customer satisfaction, shown by the highest average rating among the clusters.

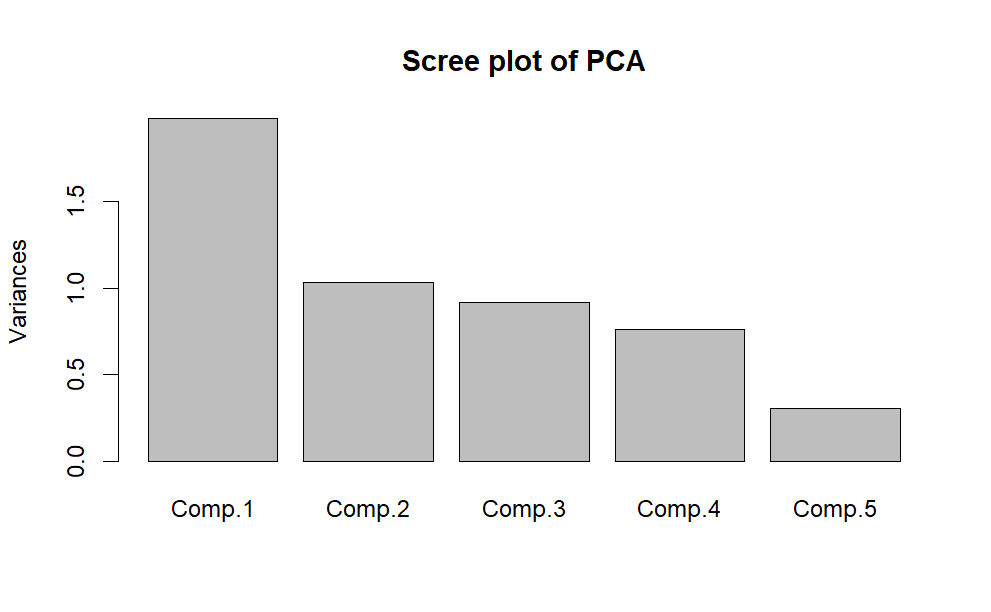


**Observations:** Cluster 2 contains only 21 brands, highlighting that luxury is rarer. In contrast, the more affordable brands in Clusters 1 and 4 had 150 and 101 brands, respectively. This increase in midrange brands reflects Sephora’s target audience, which appears to be more middle-class customers who favour affordable quality. Both Clusters 1 and 4 prioritize high ratings and moderate pricing, aligning with this customer preference. Cluster 3, however, demonstrates some limitations of the clustering process. Some brands, like 19-69, might fit better elsewhere if grouped by price alone, but since clustering considers both price and rating, some overlap and ambiguity naturally occur. Additionally, we chose to keep the outliers in the analysis, as they still represent brands, possibly extremely luxury brands or those with a lower singular rating. Another limitation is that we did not account for the number of products each brand offers. A brand could sell both high-end products (like an expensive perfume) and lower-cost items (like a makeup brush), affecting its overall average. While it was difficult to fully capture these details, the clustering exercise still provides valuable insight into brands as a whole.

**Use Case 2 – Predicting Ratings for Products**

**Objective:** The goal of this use case was to build a multiclass classification model to predict a product’s rating (ranging from 1 to 5 stars) based on product characteristics. Ratings are a key consumer signal and being able to forecast them from features such as price, size, category, and popularity can allow for better product insights, recommendation systems, and inventory decisions.

**Applying PCA:** Based on our earlier findings of a weak-to-moderate correlations, particularly between “loves\_count”, “reviews”, and “price\_usd”, we applied PCA to reduce dimensionality and address any multicollinearity among continuous variables. We started by selecting “price\_usd”, “size\_oz”, “reviews”, “loves\_count” and “child\_count” based on their numerical nature, relevance to consumer behaviour, and product characteristics. We also used binary features such as “limited\_edition”, “new” and “sephora-exclusive”. This PCA result was weak, with the first two principal components explaining only ~39% of the variance. After refining the PCA to use only continuous variables (excluding dummy categories), which intuitively would work better with PCA, we were left with “price\_usd”, “size\_oz”, “reviews”, “child\_count”, and “loves\_count”. With this, the first two principal components explained nearly 60% of the total variance in the data. Furthermore, these two new dimensions capture most of the meaningful variance across products, specifically along axes reflecting product popularity (e.g. reviews and loves\_count) and likely value as well (price and size). By compressing correlated information into uncorrelated components, this will help preserve the most informative aspects of consumer engagement and pricing behavior relevant for prediction of ratings without overfitting.

****

**Logistic Regression Model:** We applied Multinomial Logistic Regression as our initial classification model to predict product ratings (1 to 5), as it's a commonly used and interpretable method for multiclass classification tasks. It allowed us to evaluate how well our selected features including principal components derived from continuous variables (PC1, PC2), limited edition status, and primary product categories could predict consumer ratings. The model achieved an overall accuracy of 66.2%, but this is misleadingly high due to strong class imbalance. Class 4 dominates the dataset, and the model essentially defaults to predicting that class. It correctly classified almost all class 4 instances (sensitivity = 99.8%) but completely failed to detect minority classes (1–3), which represent lower-rated products that are critical for understanding dissatisfaction. This indicates that while logistic regression was methodologically appropriate, its assumptions and structure limit its usefulness in imbalanced real-world datasets like ours. These results emphasized the need for a more robust model (like Random Forest), which can better capture complex interactions between features and handle class imbalance without collapsing into majority-class predictions.

**Random Forest Model:** We first decided to try class weights in our Random Forest model because our dataset was highly imbalanced. Logistic regression already showed poor sensitivity on these minority classes, so we hoped class weights would help the model pay more attention to underrepresented groups. With class weights applied, the model did show improved sensitivity for class 3 (from near zero to 0.58) and for class 5 (0.64), meaning it was better at identifying these classes compared to the unweighted version. However, this came at a major cost: overall accuracy dropped to 33.78%, and the model severely struggled with the dominant class 4 (sensitivity = 0.19, down from 0.98 before). Precision for minority classes remained low—just 3% for class 2 and 13% for class 3—indicating many false positives. Ultimately, while class weights helped address imbalance slightly, they hurt overall performance, especially on the most frequent class, so we opted not to use them in the final model.

In the next case, we chose to not use class weights in the Random Forest model, primarily to see how well the model performed with the natural distribution of the classes, as class 4 was dominant in the dataset, and given the rather poor performance of using the weights. Without class weights, the model focuses more on correctly classifying the majority class, which in this case was class 4, and may not perform as well on minority classes. This approach helps increase overall accuracy but can lead to the underprediction of minority classes. The results without class weights show higher overall accuracy (67.8%) compared to when class weights were used, largely due to the model's strong performance on class 4 (sensitivity = 0.9724). However, the model still struggles with minority classes, particularly class 1 (sensitivity is 0.0000), and class 2, (sensitivity also at 0.0000). Class 3's sensitivity is slightly better at 0.012, but still very low. The model does well in class 4 with high specificity (0.9949) but has low specificity for class 4 (0.1290) as it misclassifies a lot of instances into other classes. Although the overall accuracy is decent, the balanced accuracy is much lower, highlighting that the model’s predictions are heavily influenced by the dominant class, resulting in poor generalization for the other classes.

Overall, we chose to stick with this model without class weights because class 4 represents the majority of products in the dataset, making it the most frequent class. Given the imbalance, it makes sense for a classification model to often predict class 4, as it maximizes the model's overall accuracy. While this results in poor performance for minority classes, the model still provides useful insights for predicting the most common class, which is likely the focus in many practical applications where the majority class is of primary interest.

**Other Considerations:** To improve our random forest model for this classification task, we could use cross-validation to ensure the model is generalizing well and not just overfitting to the training data, especially since the data is imbalanced. The class imbalance, where most products are in class 4, is a big issue, and we could address this by tuning hyperparameters like tree depth or the number of trees, or by using techniques like class weights or oversampling to give minority classes more importance in the training process. Since the model has trouble predicting minority classes, such as class 1, trying methods like the Synthetic Minority Over-sampling Technique could help by generating synthetic data for underrepresented classes. Additionally, feature engineering could help, especially by creating interaction terms between price, reviews, and category, which might help capture patterns that the random forest isn’t picking up. There is also potential to explore more complex models such as XGBoost, as it might perform better with the imbalanced data and handle complex relationships in the features more effectively.

**Insights & Interpretation:** For this classification task, we observed that predicting product ratings from the Sephora dataset is tricky, especially since most products fall under class 4, representing average ratings. This class imbalance made it difficult for the model to accurately predict the minority classes, like class 1 (very poor ratings) or class 5 (very good ratings). It’s similar to how most reviews you see online might skew toward the middle range, where people are generally satisfied but not overly excited. We also found that factors like price, size, and category were strongly associated with ratings, which makes sense given that higher-priced products or popular categories like skincare or makeup might get better reviews. However, the model often leaned toward predicting class 4 because it was the most common, showing that the imbalance in the data had a significant impact on its predictions. This insight is important for brands like Sephora, where accurately predicting both high and low ratings is just as crucial for managing customer expectations and improving product offerings.