Reports

### **Data Cleaning Report**

In this analysis, I began with cleaning and preprocessing the dataset to ensure the data was ready for model training and analysis.

The first step involved handling missing values. I addressed this by filling missing categorical variables with the most frequent value from the respective column. This approach helps maintain the integrity of the dataset while minimizing any potential bias from excluding data with missing entries.

In addition, I created a new score column using a VADER (Valence Aware Dictionary and sEntiment Reasoner) sentiment analysis tool to provide an objective measure of sentiment for each entry. This score was particularly useful for understanding customer feedback or sentiment-driven features that could potentially impact predictions.

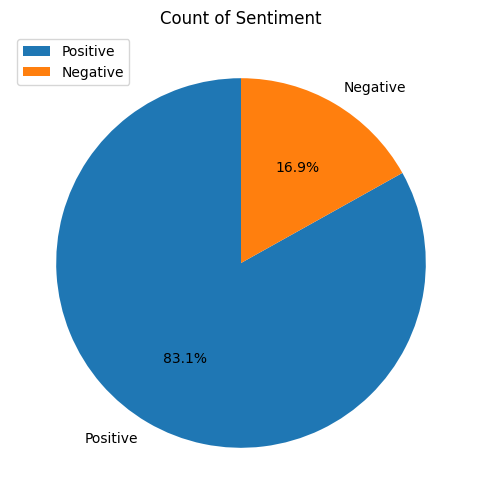
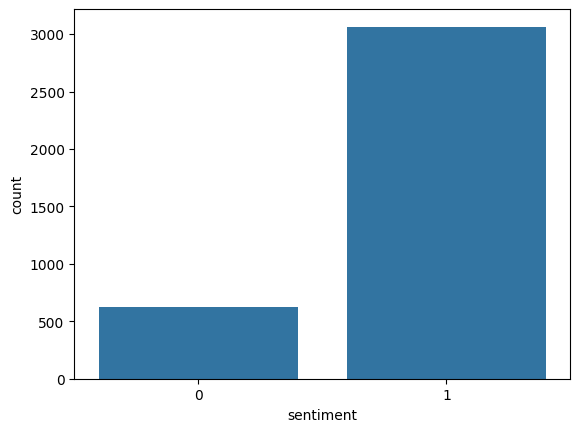
I also ensured that numerical features were scaled appropriately. Standardization or normalization was applied as needed to ensure all features contributed equally to model training, especially for models sensitive to feature scaling.

Finally, the dataset was split into training and testing sets. The training set was used for model building, while the test set was kept aside for evaluating model performance. These steps ensured the data was clean, properly structured, and ready for analysis, allowing for more accurate and efficient model training.

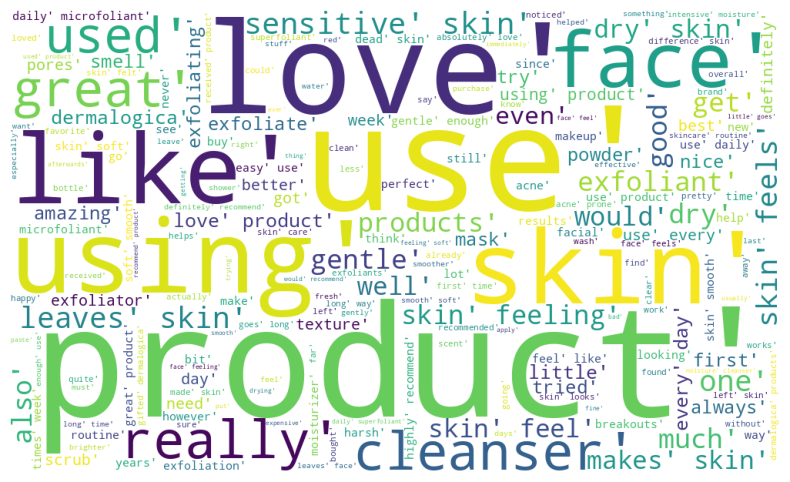
This preprocessing was essential in improving the quality and relevance of the dataset, thereby optimizing the machine learning models' performance

**Some visualization:**

The 0 and 1 represent positive sentiments and negative with 1 being the positive sentiments

****

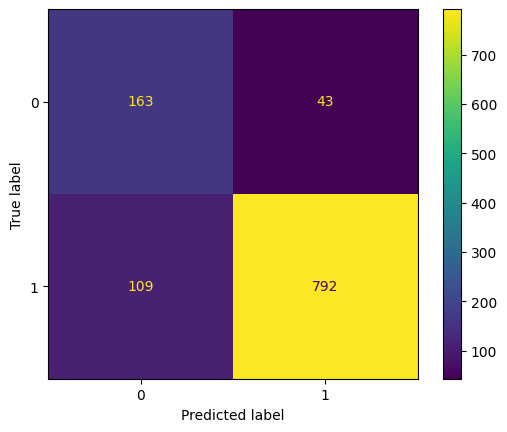
**Word cloud for the most used Word in the Reviews**



**Model Selection**

The performance of a weighted logistic regression model:

* **Class 0 (Minority Class):** Precision is 0.60, recall is 0.79, and the F1-score is 0.68, indicating the model struggles with precision but captures most true positives for this class.
* **Class 1 (Majority Class):** Precision is 0.95, recall is 0.88, and the F1-score is 0.91, showing strong performance for the majority class.
* **Overall Accuracy:** 86%, reflecting good general performance.
* **Confusion Matrix:** The model misclassified 43 instances of class 0 as class 1 and 109 instances of class 1 as class 0, indicating better recognition of the majority class.

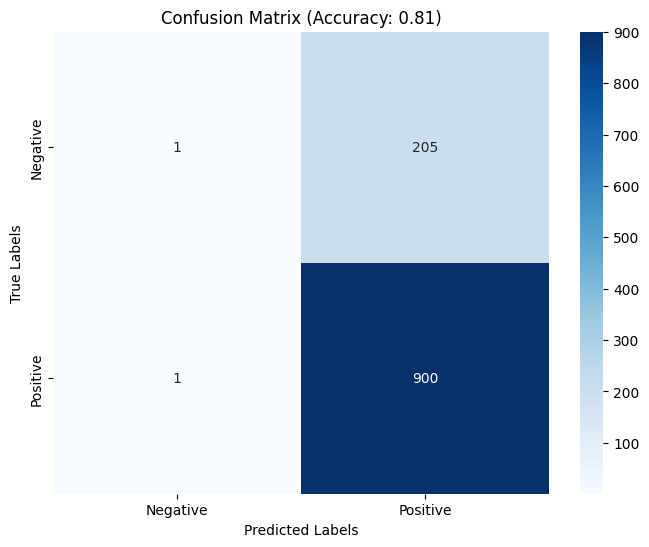


The results suggest the model is reasonably effective, with room for improvement in handling the minority class, possibly by further tuning the class weights or exploring other techniques like oversampling or ensemble methods.

**Naive Bayes:**

The Naive Bayes model shows an overall accuracy of 81%, driven by high performance on the majority class but poor handling of the minority class:

* Class 0 (Minority): Precision is 0.50, but recall is almost 0, resulting in an F1-score of 0.01. The model barely identifies any true positives for this class.
* Class 1 (Majority): Precision and recall are very high (0.81 and 1.00), with an F1-score of 0.90, showing the model predominantly predicts this class correctly.
* Confusion Matrix: Only 1 instance of class 0 was correctly classified, while 205 were misclassified as class 1.

****

The confusion matrix is showing 1 true negative 1 false negative, 205 false positives and also 900 true positives.

This suggests a strong bias toward the majority class, likely due to class imbalance. Improving the model may require balancing techniques or alternative algorithms better suited for imbalanced data and even though I used Smote to balance the data it doesn't seem like it worked well.

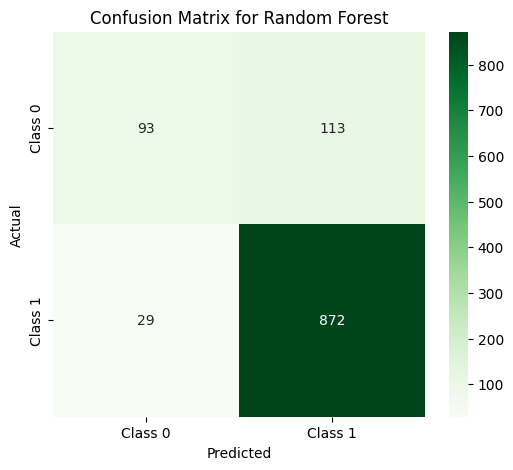
**Random forest**

The Random Forest model achieves an accuracy of 87%, and the performance across both classes is as follows:

* **Class 0 (Minority Class):** Precision is 0.76, recall is 0.45, and the F1-score is 0.57. The model struggles to identify most of the class 0 instances, as reflected in the low recall (0.45), meaning it misses many true positives for class 0.
* **Class 1 (Majority Class):** Precision is 0.89, recall is 0.97, and the F1-score is 0.92, showing strong performance in predicting class 1, with both high precision and recall.
* **Macro average:** The macro average F1-score of 0.75 indicates that the model's performance is unbalanced, with better overall performance on the majority class.
* **Weighted average:** The weighted average F1-score of 0.86 suggests that class 1's high recall and precision dominate the overall score.

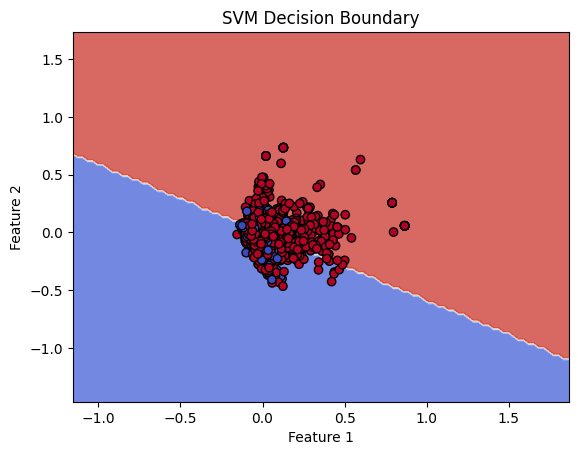
**Confusion Matrix Analysis:**

* The confusion matrix [[93, 113], [29, 872]] shows that 93 instances of class 0 were correctly predicted, while 113 were misclassified as class 1. On the other hand, 29 class 1 instances were misclassified as class 0, and 872 were correctly predicted as class 1.



**Conclusion:** The Random Forest model performs well on the majority class but struggles with the minority class. To improve performance for class 0, techniques like class weighting, oversampling, or undersampling could be applied to better capture the minority class.

**SVM Decision boundary:**

****

The plot shows a clear division between the two classes, with the decision boundary (the white line) separating them. This indicates that the SVM model is effectively distinguishing between the two classes based on the features provided.

Class Distribution:

* Red points are on the top (under the red region), indicating they belong to class 0.
* Blue points are on the bottom(under the blue region), indicating they belong to class 1.

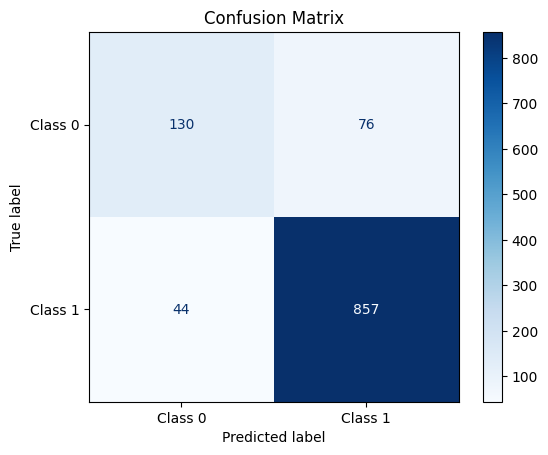
Decision Boundary Characteristics:

* The boundary is linear, using a linear kernel to separate the classes.
* The decision boundary is relatively straight, and it does not appear to have many instances close to it, indicating that the SVM has likely learned to classify the points well without overfitting (at least for the visible data).

**SVM Model**

he SVM model shows strong overall performance with an accuracy of 89%:

* **Class 0 (Minority Class):** Precision is 0.75, recall is 0.63, and the F1-score is 0.68, indicating moderate performance in identifying class 0, with some room for improvement in recall.
* **Class 1 (Majority Class):** Precision is 0.92, recall is 0.95, and the F1-score is 0.93, demonstrating excellent performance in correctly predicting class 1.
* **Confusion Matrix:** The model correctly identifies 130 instances of class 0 and misclassifies 76 instances of class 0 as class 1, while it misclassifies 45 instances of class 1 as class 0.



The model is effective overall, particularly for class 1, but could benefit from further adjustments to better capture the minority class. Techniques like class balancing or hyperparameter tuning might improve performance for class 0.

### **Conclusion:**

* **Best Model Overall**: **SVM (Support Vector Machine)**
  + The SVM model has the highest accuracy (0.89) and performs well for both classes, especially class 1. While recall for class 0 is not perfect, the overall performance makes SVM the best choice.
* **Second Best Model**: **Random Forest**
  + The random forest model also performs well, with high precision and recall for class 1. However, its recall for class 0 is lower than that of SVM, and it slightly lags behind in overall performance.
* **Logistic Regression**: The logistic regression model performs fairly well but lags behind in both precision and recall for class 0.
* **Naive Bayes**: This model has the worst performance, especially for class 0, making it unsuitable for this classification task.

### **Final Recommendation:**

I know I mentioned in my presentation t\hata the best model was random forest but after modifying and experimenting I realized SVM is the best one for both classes. When the priority is balancing performance across both classes, **SVM** would be the best choice, given its higher accuracy and relatively good performance for class 0 as well. If class 0 needs further improvement, experiment with resampling or fine-tuning the SVM model would be the best way.