**Title: Sentiment Analysis Classification Report Documentation**

**1. Introduction**

This document provides a comprehensive analysis of sentiment classification performance for various subtheme sentiment categories. The sentiment analysis model was trained and evaluated using a dataset annotated with subtheme sentiment categories. The classification reports presented herein offer insights into the precision, recall, and F1-score metrics for each sentiment category, along with overall accuracy.

**2. Objective**

The objective of this sentiment analysis project is to analyze subtheme sentiment categories within a dataset and develop a predictive model to classify sentiment for each category. By training separate classifiers for each subtheme sentiment category, we aim to capture nuanced sentiment variations and provide targeted insights for specific topics or themes.

**3. Methodology**

**Data Preprocessing:** We begin by cleaning the text data, which involves removing punctuation, lower-casing, removing stop words, emojis, and rare words. These preprocessing steps help to standardize the text data and remove noise.

**Feature Engineering**: Next, we vectorize the preprocessed text data using the CountVectorizer method. This transforms the text data into a numerical representation that can be used as input for machine learning models.

**Model Training:** We train logistic regression classifiers for each subtheme sentiment category using the vectorized text data. By training separate classifiers for each category, we aim to capture the unique sentiment characteristics of different topics or themes.

**Model Evaluation:** Finally, we evaluate the performance of the trained classifiers using metrics such as accuracy, F1-score, and classification reports. This allows us to assess the effectiveness of the sentiment analysis model in accurately predicting sentiment for each subtheme category.

**4. Implementation**

To implement the sentiment analysis pipeline, we provide code snippets and a step-by-step guide detailing each stage of the process. The code is written in Python using popular libraries such as pandas, scikit-learn, and NLTK for data manipulation, machine learning, and natural language processing tasks, respectively. Detailed explanations and comments are included within the code to enhance understanding.

**5. Classification Reports**

**Positive Sentiment:**

1. **Value for Money:**
   * Precision: 0.95
   * Recall: 0.95
   * F1-score: 0.95
   * Support: 937
   * Accuracy: 0.953
2. **Garage Service:**
   * Precision: 0.94
   * Recall: 0.77
   * F1-score: 0.85
   * Support: 367
   * Accuracy: 0.842
3. **Ease of Booking:**
   * Precision: 0.97
   * Recall: 0.79
   * F1-score: 0.87
   * Support: 238
   * Accuracy: 0.911
4. **Location:**
   * Precision: 0.98
   * Recall: 0.86
   * F1-score: 0.91
   * Support: 212
   * Accuracy: 0.955
5. **Length of Fitting:**
   * Precision: 0.98
   * Recall: 0.75
   * F1-score: 0.85
   * Support: 121
   * Accuracy: 0.920
6. **Delivery Punctuality:**
   * Precision: 0.99
   * Recall: 0.80
   * F1-score: 0.88
   * Support: 91
   * Accuracy: 0.959
7. **Tyre Quality:**
   * Precision: 0.99
   * Recall: 0.70
   * F1-score: 0.82
   * Support: 90
   * Accuracy: 0.927
8. **Advisor/Agent Service:**
   * Precision: 0.99
   * Recall: 0.51
   * F1-score: 0.67
   * Support: 49
   * Accuracy: 0.965
9. **Mobile Fitter:**
   * Precision: 0.99
   * Recall: 0.76
   * F1-score: 0.86
   * Support: 45
   * Accuracy: 0.983
10. **Advisor/Agent Service:**
    * Precision: 0.99
    * Recall: 0.49
    * F1-score: 0.65
    * Support: 35
    * Accuracy: 0.963

**Negative Sentiment:**

1. **Garage Service:**
   * Precision: 0.98
   * Recall: 0.96
   * F1-score: 0.97
   * Support: 1947
   * Accuracy: 0.951
2. **Change of Date:**
   * Precision: 0.99
   * Recall: 0.98
   * F1-score: 0.99
   * Support: 1964
   * Accuracy: 0.975
3. **Wait Time:**
   * Precision: 0.99
   * Recall: 0.94
   * F1-score: 0.96
   * Support: 1969
   * Accuracy: 0.932
4. **Delivery Punctuality:**
   * Precision: 0.99
   * Recall: 0.98
   * F1-score: 0.99
   * Support: 1978
   * Accuracy: 0.975
5. **Ease of Booking:**
   * Precision: 0.99
   * Recall: 0.98
   * F1-score: 0.98
   * Support: 1976
   * Accuracy: 0.963
6. **Mobile Fitter:**
   * Precision: 0.99
   * Recall: 0.99
   * F1-score: 0.99
   * Support: 1978
   * Accuracy: 0.964
7. **Advisor/Agent Service:**
   * Precision: 0.99
   * Recall: 0.99
   * F1-score: 0.99
   * Support: 2004
   * Accuracy: 0.986
8. **Booking Confusion:**
   * Precision: 0.99
   * Recall: 0.98
   * F1-score: 0.99
   * Support: 2004
   * Accuracy: 0.979
9. **Discounts:**
   * Precision: 1.00
   * Recall: 0.99
   * F1-score: 0.99
   * Support: 2000
   * Accuracy: 0.988
10. **Length of Fitting:**
    * Precision: 1.00
    * Recall: 0.99
    * F1-score: 0.99
    * Support: 2003
    * Accuracy: 0.984

**6. Discussion**

* The classification reports reveal variations in classification performance across different subtheme sentiment categories.
* Categories such as "Value for Money" and "Location" exhibit high precision and recall, indicating robust sentiment classification.
* Categories like "Garage Service" and "Length of Fitting" show lower precision and recall, suggesting potential challenges in accurately classifying sentiment for these themes.
* Overall, the sentiment analysis model demonstrates promising performance with an average accuracy of 0.95 across all subtheme categories.

**7. Conclusion**

In conclusion, the sentiment analysis classification reports provide valuable insights into the performance of the sentiment analysis model for various subtheme sentiment categories. The model showcases strong classification capabilities for certain themes, while highlighting areas for improvement in others. Moving forward, further refinement of the model and exploration of advanced techniques could enhance sentiment classification accuracy and reliability.

**8. Recommendations**

Based on the findings from the classification reports, the following recommendations are proposed:

* Conduct targeted data augmentation or sampling strategies to address class imbalances for categories with lower precision and recall.
* Explore ensemble learning techniques or deep learning architectures to improve sentiment classification performance, especially for challenging subtheme categories.
* Incorporate domain-specific features or context-aware sentiment analysis methods to enhance the model's ability to capture nuanced sentiment variations within specific themes.