Sentiment Analysis of Sri Lankan Hotel Reviews: A Multi-Stage Study

Part A: Summary of Findings and Recommendation

Task 1: Data Collection and Initial Processing

Overview

- Collected and prepared a comprehensive dataset of Sri Lankan hotel reviews from TripAdvisor to support tourism revival efforts.
- Aimed to create a high-quality corpus reflecting genuine tourist sentiment, suitable for automated analysis.

Data Collection

- Curated 200 hotels covering diverse locations and categories (urban, beach resorts, boutique).
- Prioritized hotels with high traveler engagement using Traveler Rank.
- Used APIFY's Tripadvisor Reviews Actor scraper to extract 16,500+ raw English reviews.
- Initial sentiment distribution by rating:
    - Negative (1–2 stars): ~2,266
    - Neutral (3 stars): ~2,266
    - Positive (4–5 stars): ~12,000
- The dataset reflected real-world trends but was imbalanced, with positive reviews dominating.

Data Cleaning

A robust multi-step cleaning pipeline was applied:

- **Structural Filtering:** Removed duplicates and incomplete entries; standardized review schemas (hotel name, city, rating, date, review text).

- **Textual Cleaning:** Lowercased all text, tokenized and lemmatized using spaCy; removed stopwords, punctuation, and numeric tokens.
- **Language Verification:** Applied three layered filters (alphabetic checks, English vocabulary matching with NLTK, and language detection using langdetect) to ensure reviews were English.
- **Quality Control:** Excluded reviews with fewer than 20 meaningful tokens.

Post-cleaning, 11,317 reviews across all hotels remained, providing a high-quality dataset.

Exploratory Data Analysis

- **Sentiment Distribution:** Positive reviews were dominant but later balanced via downsampling.
- **Review Length:** Negative reviews were generally longer and more detailed, reflecting more expressive dissatisfaction.
- **Frequent Themes:** Words like *hotel*, *room*, *staff*, *food*, *service*, *pool*, and *experience* were most frequent, highlighting key satisfaction drivers.
- **Vocabulary Compression:** The cleaning pipeline effectively reduced vocabulary size by removing noise while preserving meaningful content.

Limitations

- Sampling bias favored higher-ranked hotels, underrepresenting budget and rural accommodations.
- Sentiment skew toward positive feedback was addressed but remains an inherent limitation.
- Language filtering may not be perfect; some non-English content might persist.

Conclusion

The outcome of Task 1 is a well-curated, cleaned, and quality-controlled dataset that serves as a solid foundation for sentiment labeling and predictive modeling.

Task 2: Establishing Ground Truth Sentiment Labels

Objective

To generate reliable sentiment labels, we used an ensemble of three classifiers combining lexicon and transformer-based approaches, and consolidated their outputs via majority voting.

Classifiers Used

- **VADER:** A lexicon and rule-based sentiment analyzer optimized for social media text.
- **BERT:** The nlptown/bert-base-multilingual-uncased-sentiment model predicting star ratings.
- **RoBERTa:** The cardiffnlp/twitter-roberta-base-sentiment-latest model, a transformer-based classifier trained on Twitter data.

Methodology

- **Threshold Optimization:** For VADER and BERT, thresholds for positive and negative sentiment were tuned via compound score sweeps and star-rating averages respectively.
- **Prediction Aggregation:** Each review was split into 512-token chunks if needed; predictions were averaged or aggregated across chunks for final sentiment determination.
- **Majority Voting:**
    - If at least two classifiers agreed, that label was assigned.
    - If all disagreed, the review was labeled neutral.

Performance

- VADER aligned with rating-based sentiment for 8,885 reviews; it struggled with informal or sarcastic language.
- BERT matched 9,614 reviews and handled neutral sentiments better but occasionally misclassified mixed sentiments.
- RoBERTa performed best without tuning, showing high contextual sensitivity and manual inspection indicated it aligned closest to human judgment.

Final Dataset

- Majority voting produced labels for all 11,317 reviews.
- To mitigate class imbalance, the dataset was downsampled to:
    - Negative: 1,907
    - Neutral: 1,022
    - Positive: 2,071

Analysis

- RoBERTa had the highest agreement with the majority vote.
- VADER was the most divergent, especially on neutral labels.
- Some rating-based labels were inconsistent with text sentiment, illustrating the importance of text-based classification.

Summary

Task 2 produced a robust, semantically meaningful sentiment-labeled dataset through ensemble classification and balancing, laying the groundwork for model training.

Task 3: Feature Extraction

Objective

To transform text into numerical features suitable for classification, four vectorization methods were explored, capturing both sparse lexical and dense semantic representations.

Methods

Method	Type	Feature Shape	Description & Strengths
Bag-of-Words	Sparse	(5000, 22,504)	Word frequency counts, simple and interpretable baseline
TF-IDF	Sparse	(5000, 22,504)	Weighted word frequencies emphasizing rare terms
GloVe (avg)	Dense	(5000, 100)	Pre-trained word embeddings averaged per review
Doc2Vec	Dense	(5000, 100)	Learned document embeddings capturing context

Insights

- Sparse representations excel in interpretability and fast computation but ignore word order and context.
- Dense embeddings incorporate semantic and contextual nuances, important for complex sentiment cues.
- Doc2Vec was trained specifically on our corpus, potentially capturing domain-specific semantics.

Conclusion

Using diverse feature types ensures a comprehensive representation of the text for downstream classification tasks.

Task 4: Text Classification with Classical Models

Objective

Evaluate classical machine learning models' performance on sentiment classification using extracted features.

Models

- **Random Forest:** Captures complex non-linear feature interactions, robust to noise.
- **Logistic Regression:** Reliable linear baseline with regularization.
- **Linear Support Vector Classifier (SVC):** Strong margin-based learner suited for high-dimensional sparse data.

Experimental Setup

- Balanced dataset split into training/testing sets.
- Bayesian optimization used for hyperparameter tuning (trees, depth, penalties, regularization strength).
- Evaluated primarily on balanced accuracy, F1 macro, and precision-recall metrics.

Results

- Linear SVC performed well on features, balancing accuracy and computational efficiency.
- Random Forest showed strength but was resource-intensive and less suited for dense vectors.
- Logistic Regression performed well with appropriate solvers and regularization.

Evaluation

- Confusion matrices revealed the main misclassifications occurred between neutral and adjacent classes.
- Learning curves showed good generalization with minimal overfitting.
- ROC and precision-recall curves indicated strong class discrimination, especially for positive and negative labels.

Interpretation

Linear classifiers particularly excelled with sparse features due to their regularization and margin maximization. Random Forest's ability to model non-linearities may benefit from dense embeddings.

Limitations and Future Work

- Dense embeddings (GloVe, Doc2Vec) and deep learning comparisons are needed for a full performance landscape.
- Larger and more diverse datasets could improve model robustness.

Task 5: Sentiment Classification Using Pre-Trained Contextual Embeddings

Objective

- Extend the sentiment classification pipeline by using **pre-trained contextual embeddings**.
- Evaluate their impact on classical machine learning models.
- Deep learning model comparison was planned but not executed due to resource constraints; recommended for future work.

Embeddings

- Contextual embeddings (e.g., **RoBERTa**) produce dynamic word vectors influenced by surrounding context.
- Capture nuances such as sarcasm, negation, and word sense better than static embeddings (e.g., GloVe, Doc2Vec).
- Extracted token-level embeddings averaged to create fixed-length review vectors.

Models

- Retrained classical classifiers using contextual embeddings as inputs:
    - **Random Forest**
    - **Logistic Regression**
    - **Linear Support Vector Classifier (SVC)**
- The deep learning model (e.g., fine-tuned transformer or neural network) was not implemented but proposed for future exploration.

Experimental Setup

- Maintained the same training/testing splits and balanced dataset as previous tasks.
- Used Bayesian optimization for hyperparameter tuning, optimizing for **balanced accuracy**.

Results

- All classical models showed improved performance with contextual embeddings compared to BoW, TF-IDF, or static embeddings.
- **Logistic Regression and Linear SVC** benefited most due to better handling of dense, semantic-rich features.
- **Random Forest** showed moderate gains but was less efficient with high-dimensional dense data.

Interpretation & Conclusion

- Contextual embeddings substantially enhance sentiment classification effectiveness with classical models.
- Though not tested here, deep learning models are expected to outperform classical classifiers when using contextual embeddings.
- Future work should include implementing and comparing deep learning models for further improvements.

Part B: Reflective Report on Learning Process

Project Reflection and Key Learnings

This project has been a valuable learning experience, enhancing my skills in applied Natural Language Processing (NLP) and machine learning across the entire workflow—from data collection to classification using contextual embeddings. The challenges faced provided deep insights and strengthened my practical abilities.

Importance of Data Quality

- Initially underestimated the impact of noisy, unstructured, and skewed data on model performance and interpretability.
- Developed a multi-stage cleaning pipeline including:
    - Language filtering
    - Vocabulary validation
    - Semantic cleaning
- Learned to balance noise removal with preserving meaningful content despite informal text issues (spelling errors, emojis, multilingual content).

Robust Labeling via Ensemble Methods

- Gained practical knowledge of ensemble techniques by combining lexicon-based (VADER) and transformer-based (BERT, RoBERTa) classifiers.
- Majority voting helped mitigate biases of individual models.
- Discovered discrepancies between star ratings and textual sentiment, emphasizing the importance of semantic-based labeling.

Sparse vs. Dense Representations

- Sparse methods like TF-IDF combined with linear classifiers showed strong baseline performance.
- Contextual embeddings from transformer models significantly outperformed static embeddings (GloVe, Doc2Vec), capturing subtle linguistic nuances.

Role of Generative AI

- Used ChatGPT as an interactive learning companion rather than just a code generator.
- Tasks supported by ChatGPT included:
    - Clarifying complex library functions (e.g., skopt, spaCy, Hugging Face transformers)
    - Debugging token limit and prediction issues
    - Validating interpretation of evaluation metrics (balanced accuracy, F1 macro, precision-recall)
    - Designing ensemble labeling strategies, feature engineering, and class balancing
    - Accelerated report writing and documentation, offering phrasing suggestions and helping me stay within word limits.
- This collaboration improved efficiency and confidence while keeping me actively involved in critical thinking and decision-making.

Final Reflection

- Strengthened ability to design, implement, and evaluate end-to-end NLP pipelines.
- Developed better intuition about feature-model interactions and hyperparameter tuning.
- Learned to balance model interpretability with predictive power.
- Cultivated a habit of critical reflection on methodological choices, essential for data science professionals.