PDAN8411 REPORT DOCUMENT

Makabongwe Lwethu Sibisi

ST10145439

Table of Contents

Introduction

This report outlines a project aimed at analysing hospital reviews to gain insights into patient feedback. Using a dataset of approximately 900 hospital reviews, the project sought to identify broad areas of patient concern through Latent Dirichlet Allocation (LDA) topic modelling. Furthermore, it analysed customer sentiment using both a supervised Logistic Regression model and a rule-based approach with VADER SentimentIntensityAnalyzer. This report details the data preparation, model development, and evaluation steps taken, culminating in a comparison of the two sentiment analysis methods to understand their effectiveness in capturing patient satisfaction.

Dataset Overview

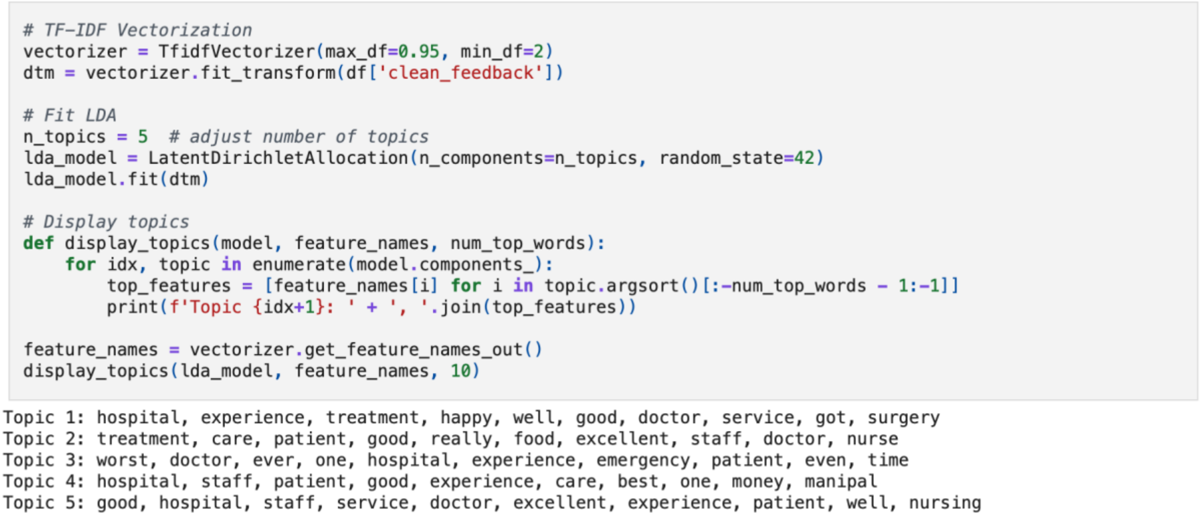
The dataset comprised 996 records of hospital reviews, containing three key columns: the "Feedback" column capturing unstructured patient comments, "Sentiment Label" indicating pre-classified sentiment (0=negative, 1=positive), and "Ratings" representing 1–5 star scores. Initial quality assessment revealed no missing values in these critical fields. An entirely null "Unnamed: 3" column was discarded during preprocessing, and 19 duplicate entries were identified and eliminated, resulting in 977 unique records for analysis. The dataset's combination of textual feedback and categorical sentiment labels proved well-suited for both topic modelling and sentiment analysis objectives.

Exploratory Data Analysis (EDA)

The Exploratory Data Analysis (EDA) phase aimed to understand the inherent characteristics of the hospital review dataset, identify patterns, and uncover insights crucial for subsequent modelling.

Initial inspection of the dataset revealed a predominant positive sentiment, with reviews heavily skewed towards higher star ratings (e.g., 5 and 4 stars). This distribution highlighted a potential class imbalance, particularly for the negative sentiment, which would need careful consideration during model training.

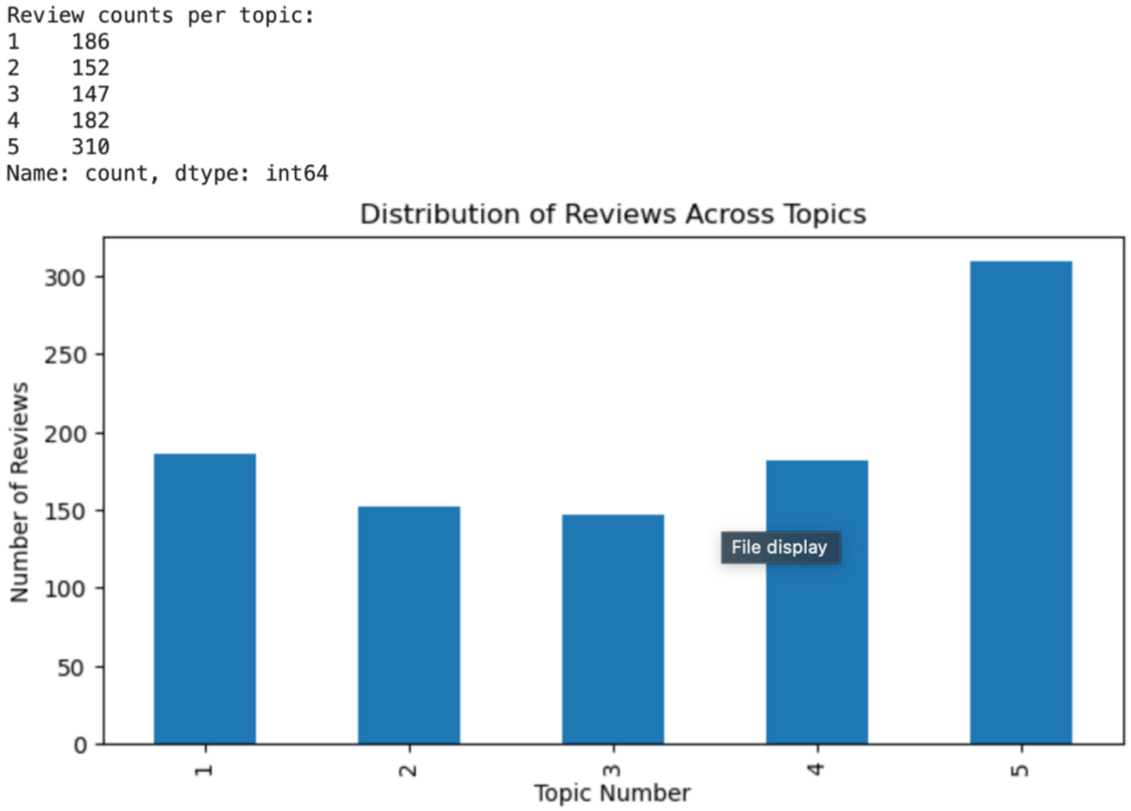
To understand the core themes within the patient feedback, Latent Dirichlet Allocation (LDA) was applied. The model identified five distinct topics, each characterized by recurring keywords. For instance, Topic 1 revolved around "hospital experience" (good, treatment, doctor), while Topic 3 focused on "negative experiences" (worst, emergency, time). The distribution of reviews across these topics was somewhat varied, with Topic 5 (good, hospital, staff, service) being the most prevalent (Ph.D., 2024).



```python
# TF-IDF Vectorization
vectorizer = TfidfVectorizer(max_df=0.95, min_df=2)
dtm = vectorizer.fit_transform(df['clean_feedback'])

# Fit LDA
n_topics = 5  # adjust number of topics
lda_model = LatentDirichletAllocation(n_components=n_topics, random_state=42)
lda_model.fit(dtm)

# Display topics
def display_topics(model, feature_names, num_top_words):
    for idx, topic in enumerate(model.components_):
        top_features = [feature_names[i] for i in topic.argsort()[:-num_top_words - 1:-1]]
        print(f'Topic {idx+1}: ' + ', '.join(top_features))

feature_names = vectorizer.get_feature_names_out()
display_topics(lda_model, feature_names, 10)
```

```
Topic 1: hospital, experience, treatment, happy, well, good, doctor, service, got, surgery
Topic 2: treatment, care, patient, good, really, food, excellent, staff, doctor, nurse
Topic 3: worst, doctor, ever, one, hospital, experience, emergency, patient, even, time
Topic 4: hospital, staff, patient, good, experience, care, best, one, money, manipal
Topic 5: good, hospital, staff, service, doctor, excellent, experience, patient, well, nursing
```

Overall Sentiment Distribution

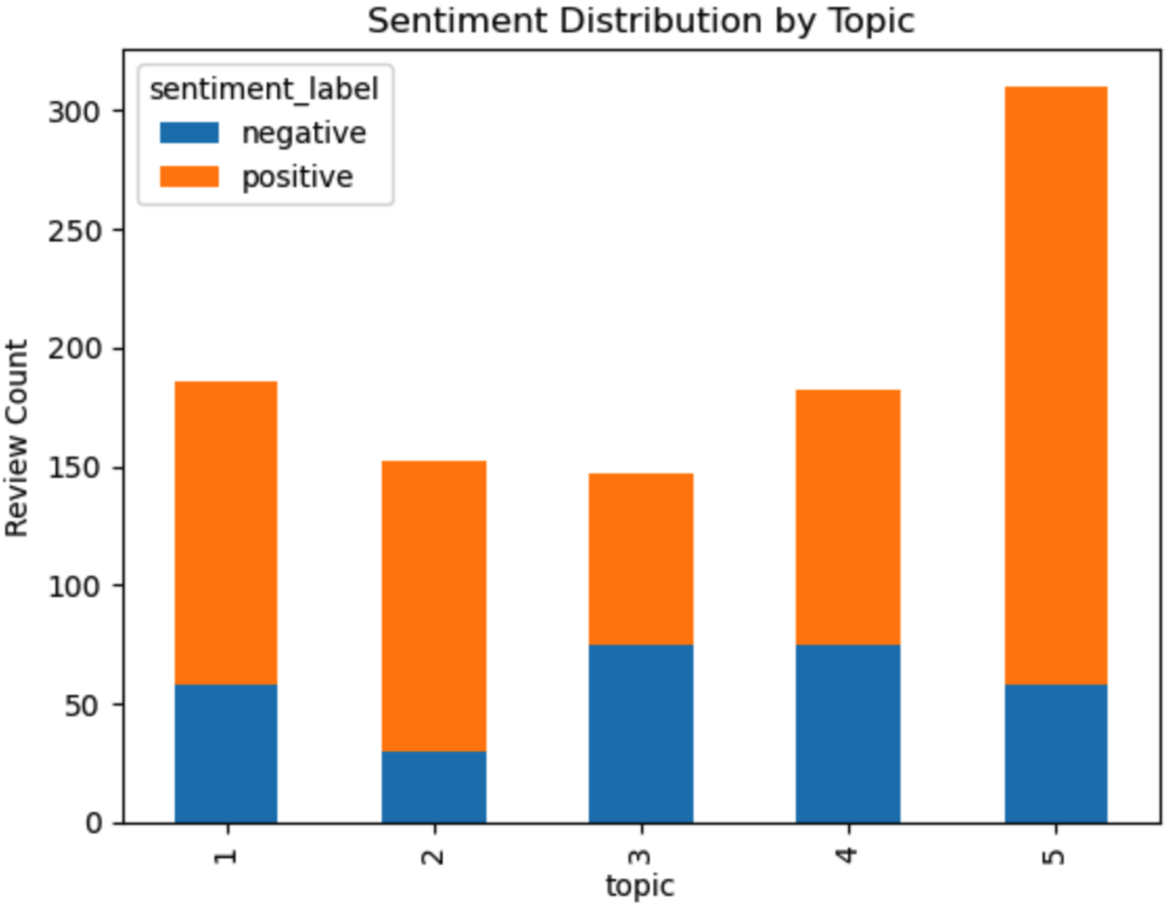


```
Review counts per topic:
1    186
2    152
3    147
4    182
5    310
Name: count, dtype: int64
```

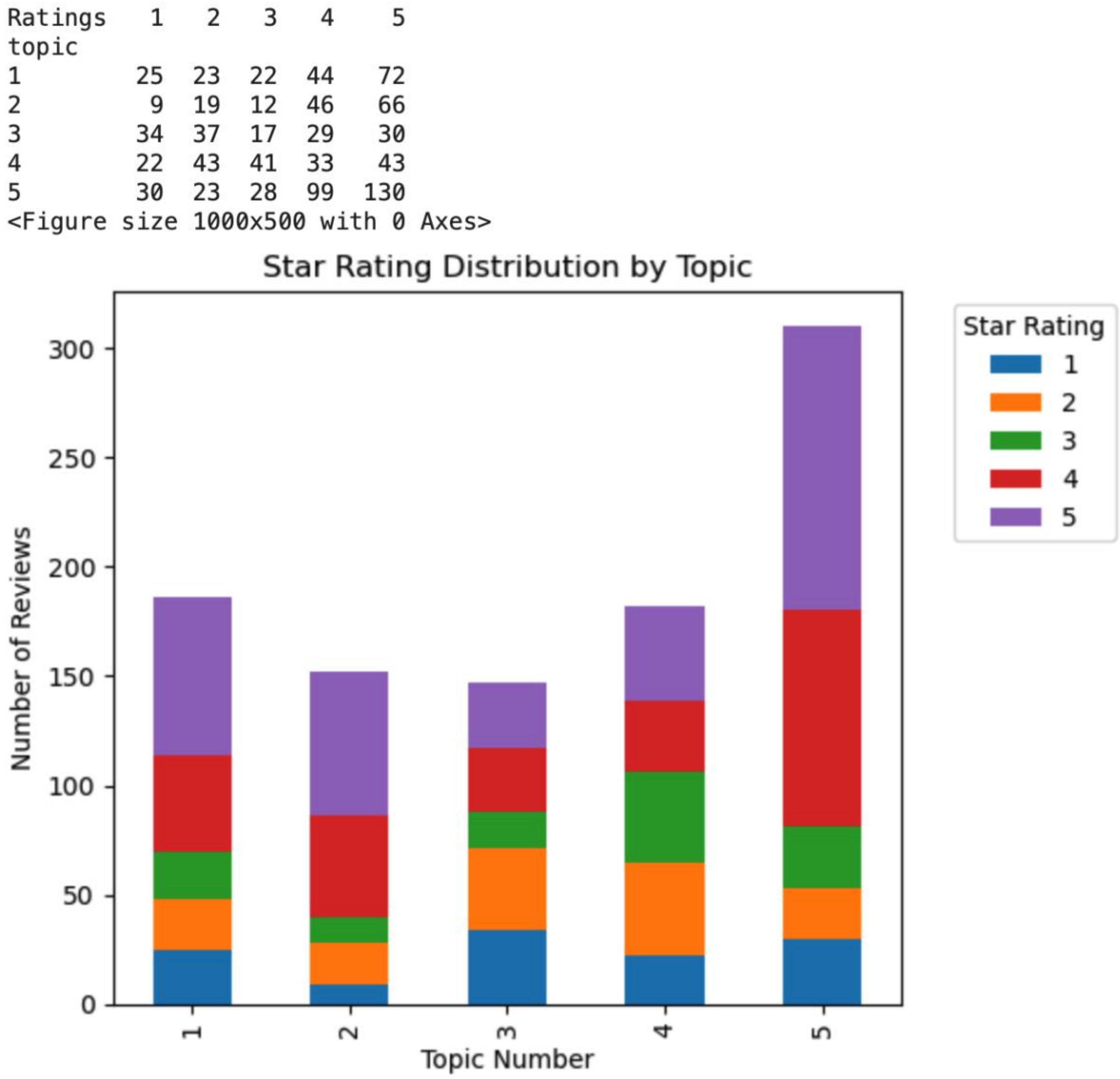
Further analysis delved into the relationship between these identified topics and sentiment/star ratings. Visualizations showed how certain topics were more strongly associated with positive or negative sentiment and specific rating ranges. For example, topics related to "worst experiences" naturally correlated with lower ratings and negative sentiment labels.

Sentiment Distribution by Topic

```
sentiment_label
positive    681
negative    296
Name: count, dtype: int64
sentiment_label  negative  positive
topic
1                      58       128
2                      30       122
3                      75        72
4                      75       107
5                      58       252
```



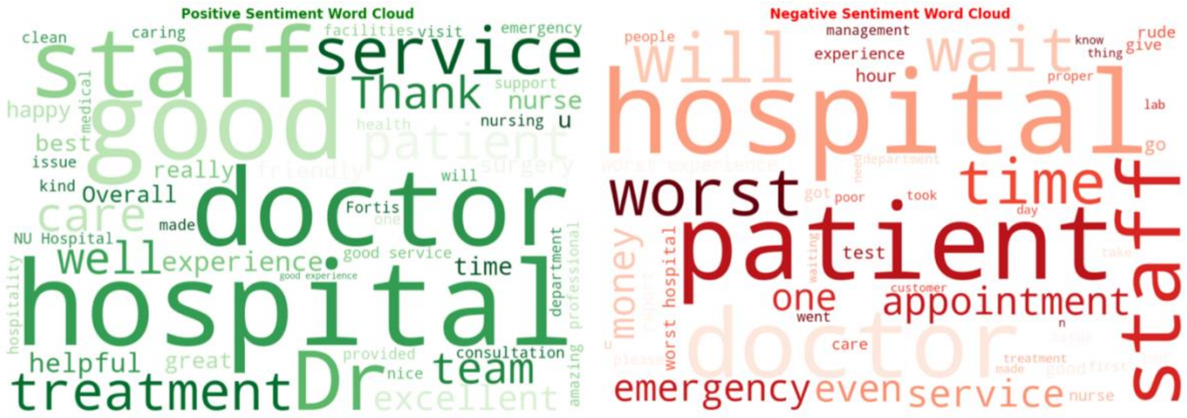
Star Rating Distribution by Topic



```
Ratings   1   2   3   4    5
topic
1        25  23  22  44   72
2         9  19  12  46   66
3        34  37  17  29   30
4        22  43  41  33   43
5        30  23  28  99  130
<Figure size 1000x500 with 0 Axes>
```

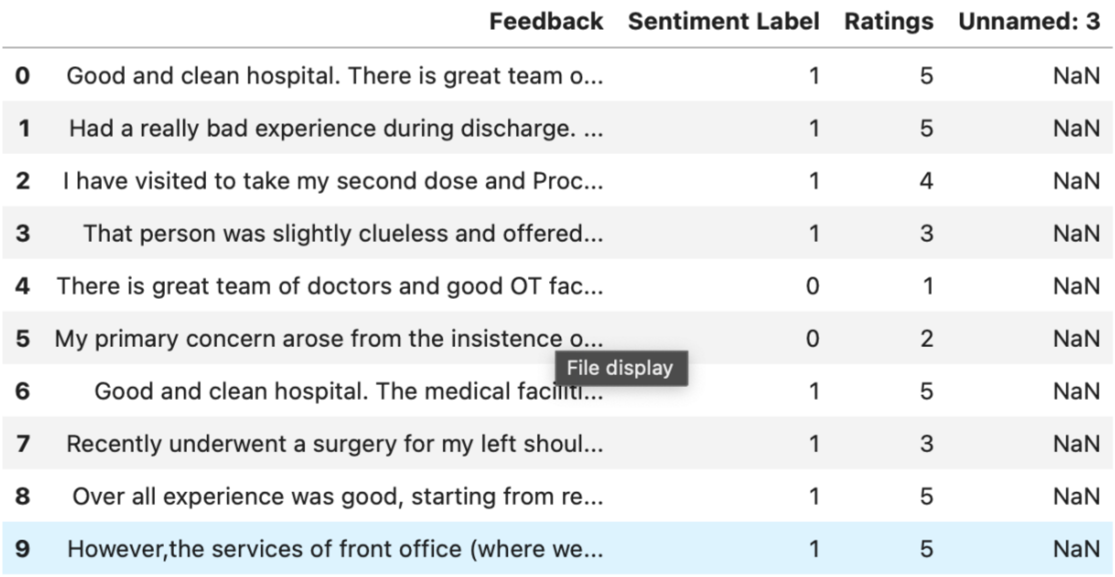
Finally, word clouds were generated for both positive and negative reviews to visually highlight the most frequently used terms. The word cloud for positive sentiment often featured words like "good," "doctor," "staff," and "service," indicating appreciation for these aspects. Conversely, the negative sentiment word cloud frequently contained terms such as "waiting," "time," "staff," and "appointment," pointing to common pain points for patients. This visual exploration provided a quick understanding of the vocabulary used by patients expressing different levels of satisfaction.

Word Cloud for Positive Reviews and Negative Reviews

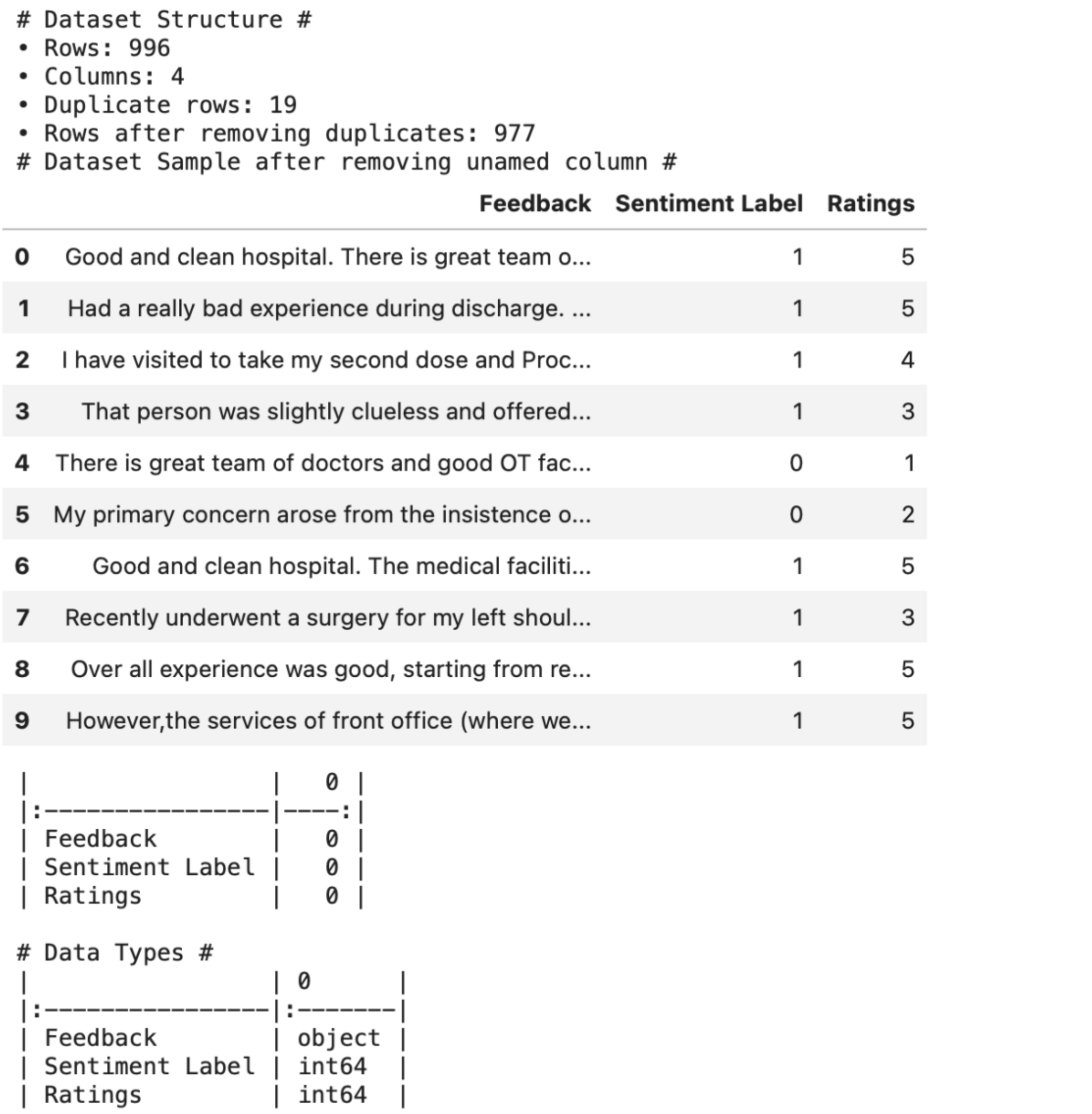


The EDA phase confirmed the dataset's suitability for both topic modelling and sentiment analysis, while also surfacing the class imbalance challenge that would require strategic handling in the modelling phase to ensure robust and accurate results.

Data Cleaning and Preprocessing



	Feedback	Sentiment Label	Ratings	Unnamed: 3
0	Good and clean hospital. There is great team o...	1	5	NaN
1	Had a really bad experience during discharge. ...	1	5	NaN
2	I have visited to take my second dose and Proc...	1	4	NaN
3	That person was slightly clueless and offered...	1	3	NaN
4	There is great team of doctors and good OT fac...	0	1	NaN
5	My primary concern arose from the insistence o...	0	2	NaN
6	Good and clean hospital. The medical faciliti...	1	5	NaN
7	Recently underwent a surgery for my left shoul...	1	3	NaN
8	Over all experience was good, starting from re...	1	5	NaN
9	However,the services of front office (where we...	1	5	NaN



```
# Dataset Structure #
• Rows: 996
• Columns: 4
• Duplicate rows: 19
• Rows after removing duplicates: 977
# Dataset Sample after removing unamed column #
```

	Feedback	Sentiment Label	Ratings
0	Good and clean hospital. There is great team o...	1	5
1	Had a really bad experience during discharge. ...	1	5
2	I have visited to take my second dose and Proc...	1	4
3	That person was slightly clueless and offered...	1	3
4	There is great team of doctors and good OT fac...	0	1
5	My primary concern arose from the insistence o...	0	2
6	Good and clean hospital. The medical faciliti...	1	5
7	Recently underwent a surgery for my left shoul...	1	3
8	Over all experience was good, starting from re...	1	5
9	However,the services of front office (where we...	1	5

```
|                 |   0 |
|:----------------|----:|
| Feedback        |   0 |
| Sentiment Label |   0 |
| Ratings         |   0 |

# Data Types #
|                 | 0      |
|:----------------|:-------|
| Feedback        | object |
| Sentiment Label | int64  |
| Ratings         | int64  |
```

Effective data cleaning and preprocessing are fundamental steps to ensure the quality and utility of textual data for analysis. For this project, the raw hospital review data underwent a series of transformations to prepare it for both LDA topic modelling and sentiment analysis.

Initially, the dataset contained 996 entries. A quality check revealed an entirely null column, "Unnamed: 3", which was removed as it provided no valuable information. Additionally, 19 duplicate records were identified and subsequently eliminated, reducing the dataset to 977 unique patient reviews. This deduplication step was crucial to prevent bias and ensure that each review contributed independently to the analysis.

The textual "Feedback" column then underwent a comprehensive preprocessing pipeline:

1. **Lowercasing:** All text was converted to lowercase to ensure consistency and prevent the model from treating identical words with different casing (e.g., "Good" and "good") as distinct terms.

2. **Punctuation and Number Removal:** Regular expressions were used to remove punctuation marks and numbers. This step focuses the analysis on the words themselves, as punctuation often carries little semantic weight for topic modelling and sentiment analysis in this context.

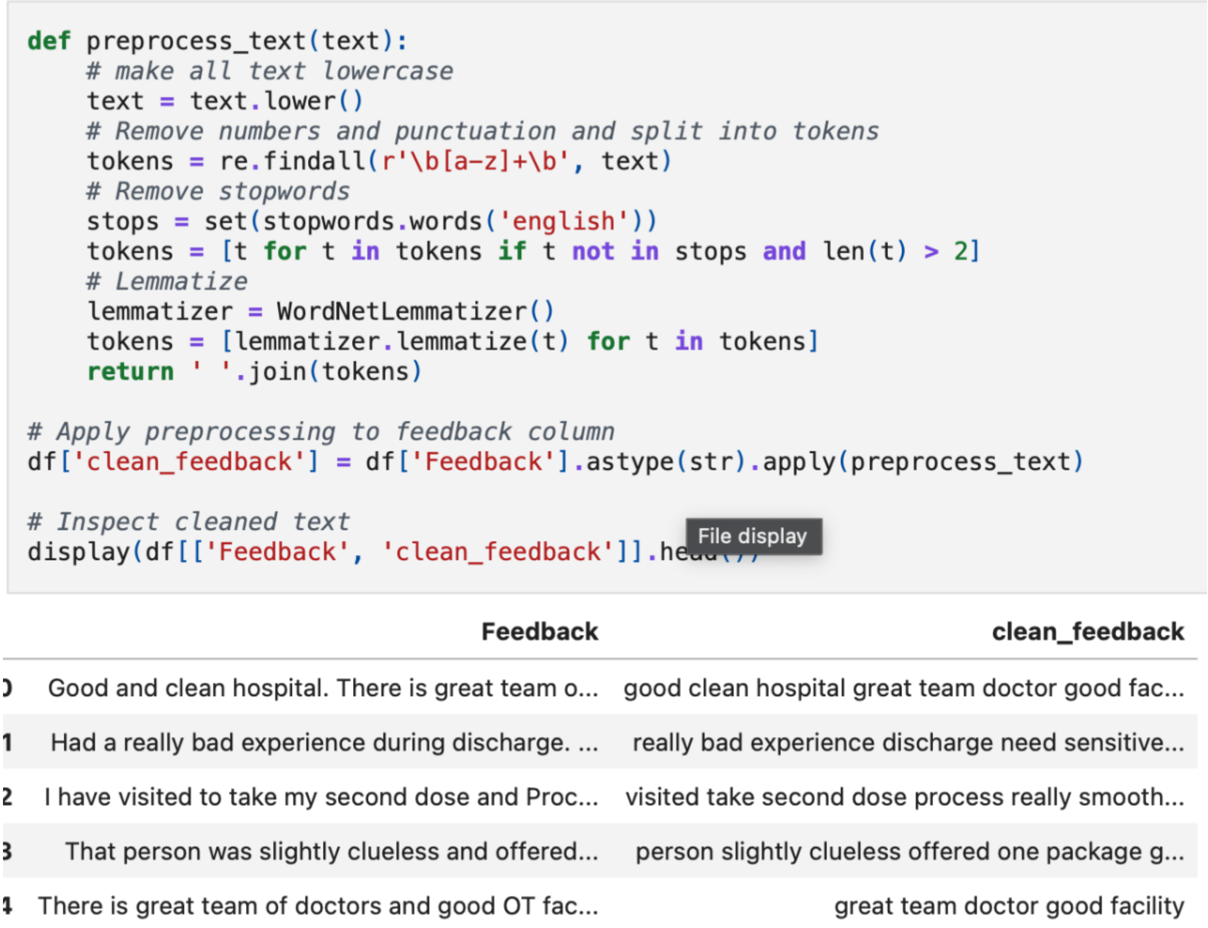
3. **Tokenization:** The cleaned text was split into individual words or "tokens," forming the basic units for analysis.

4. **Stop Word Removal:** Common English stop words (e.g., "the," "is," "and") were removed. These words are high-frequency but typically carry little semantic meaning and can obscure more important terms in topic models and sentiment analysis.

5. **Lemmatization:** Words were reduced to their base or root form (e.g., "running," "ran," "runs" all become "run"). This step normalizes vocabulary, reducing sparsity and improving the accuracy of both topic extraction and sentiment classification by grouping inflected forms of a word.

This rigorous preprocessing pipeline resulted in a clean_feedback column, ready for vectorization and model training. The transformation ensured that the textual data was consistent, relevant, and optimized for the analytical techniques applied in the subsequent stages of the project (Dylan, 2024).

Cleaned Feedback Sample



```python
def preprocess_text(text):
    # make all text lowercase
    text = text.lower()
    # Remove numbers and punctuation and split into tokens
    tokens = re.findall(r'\b[a-z]+\b', text)
    # Remove stopwords
    stops = set(stopwords.words('english'))
    tokens = [t for t in tokens if t not in stops and len(t) > 2]
    # Lemmatize
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(t) for t in tokens]
    return ' '.join(tokens)

# Apply preprocessing to feedback column
df['clean_feedback'] = df['Feedback'].astype(str).apply(preprocess_text)

# Inspect cleaned text
display(df[['Feedback', 'clean_feedback']].head())
```

	Feedback	clean_feedback
0	Good and clean hospital. There is great team o...	good clean hospital great team doctor good fac...
1	Had a really bad experience during discharge. ...	really bad experience discharge need sensitive...
2	I have visited to take my second dose and Proc...	visited take second dose process really smooth...
3	That person was slightly clueless and offered...	person slightly clueless offered one package g...
4	There is great team of doctors and good OT fac...	great team doctor good facility

Feature Selection

Feature engineering primarily involved transforming raw text into a numerical representation that machine learning models could understand. In this project, TF-IDF (Term Frequency-Inverse Document Frequency) vectorization was the chosen method for converting the preprocessed hospital reviews into a suitable feature set.

TF-IDF assigned a weight to each word, reflecting its importance in a specific document relative to the entire collection of documents (corpus). Words that appeared frequently in a particular review but rarely across the overall dataset received higher TF-IDF scores, indicating their unique significance. Conversely, common words that appeared in many reviews (like "the" or "is") were down-weighted, preventing them from dominating the feature space.

Specific parameters **were applied** during TF-IDF vectorization:

- **max_df=0.95**: This parameter **filtered out** terms that **appeared** in more than 95% of the documents. Such terms **were** often too common to be discriminative and **could be considered** noise (e.g., "hospital" itself, which might **have appeared** in almost every review).

- **min_df=2 (for LDA) / min_df=5 (for Logistic Regression)**: This parameter **excluded** terms that **appeared** in fewer than 2 (for LDA) or 5 (for Logistic Regression) documents. This **helped** remove rare words, potential typos, or highly specific mentions that **were** unlikely to contribute meaningfully to general topics or sentiment.

- **lowercase=True and stop_words='english'**: While significant preprocessing steps handled lowercasing and stop word removal beforehand, these parameters within the TfidfVectorizer reinforced the cleanliness of the features by ensuring consistency in tokenization.

The output of the TF-IDF vectorizer was a Document-Term Matrix (DTM), where rows represented individual reviews and columns represented the TF-IDF scores for each unique word. This DTM served as the input feature set for both the Latent Dirichlet

Allocation (LDA) model for topic extraction and the Logistic Regression model for supervised sentiment classification.

An important note was that VADER SentimentIntensityAnalyzer, being a lexicon- and rule-based approach, did not rely on these extracted TF-IDF features. Instead, it directly processed the raw text to determine sentiment scores based on its predefined lexicon and grammatical rules (Science, 2024).

Model Training

The project involved applying a rule-based sentiment analyser, VADER, and training two distinct models: a Latent Dirichlet Allocation (LDA) model for topic extraction and a Logistic Regression model for supervised sentiment classification.

For VADER Sentiment Analysis, no formal training was required. As a lexicon- and rule-based tool, VADER was initialized as an SentimentIntensityAnalyzer object. It then directly processed the preprocessed Feedback text, computing polarity scores (negative, neutral, positive, and compound) for each review based on its built-in dictionary and grammatical rules. These compound scores were subsequently mapped to categorical sentiment labels ("positive" or "negative") for comparison with the ground truth and the Logistic Regression model's outputs.

For LDA Topic Modelling, the TF-IDF vectorized clean_feedback (Document-Term Matrix) was used as input. An LDA model with 5 topics (n_components=5) was initialized and trained on this dataset. The choice of 5 topics was determined through an iterative process of evaluation, balancing interpretability and topic distinctiveness. The random_state=42 parameter was set to ensure reproducibility of the topic assignments. Once trained, the model identified prominent topics by grouping co-occurring words, allowing for a thematic understanding of the hospital reviews.

For Logistic Regression Sentiment Classification, the preprocessed text features (TF-IDF vectorized clean_feedback) and the Sentiment Label column (our target variable) were split into training and testing sets. A stratified 80-20 train-test split was employed, ensuring that the proportion of positive and negative sentiment labels was maintained in both the training and testing sets, which was crucial given the observed class imbalance. The TF-IDF vectorizer was fitted on the training data (X_train) and then used to transform both the training and testing data (X_train_tfidf, X_test_tfidf). A Logistic Regression classifier, with max_iter=1000 to ensure convergence, was then trained on the X_train_tfidf and y_train data. This training process involved the model learning the relationship between the word features and the sentiment labels, enabling it to predict sentiment for unseen reviews. The model's performance was then evaluated on the X_test_tfidf and y_test sets.

Model Evaluation

The evaluation phase focused on assessing the performance of both the supervised Logistic Regression model and the rule-based VADER sentiment analyser, with a particular emphasis on their accuracy in classifying sentiment and their ability to capture both positive and negative feedback.

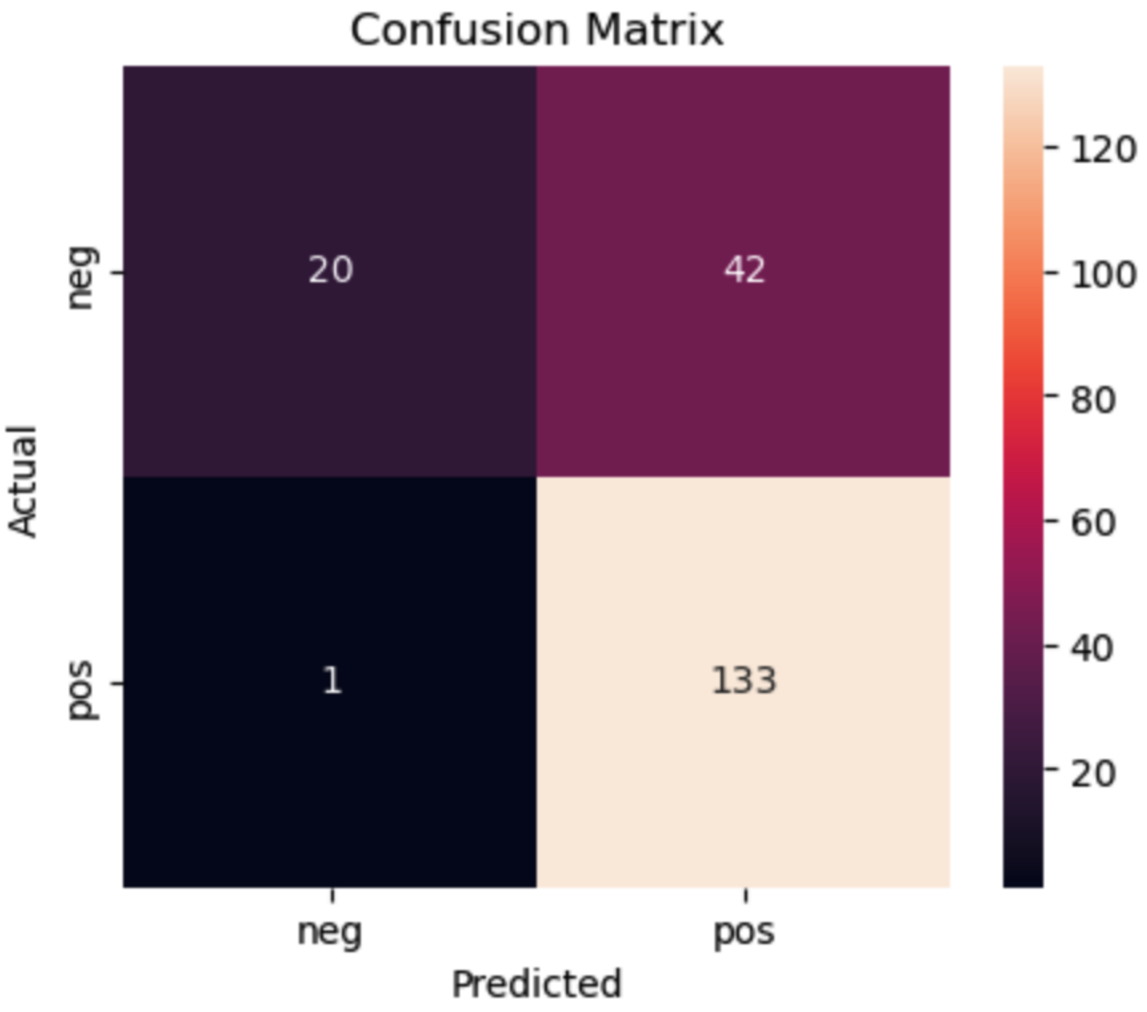
Logistic Regression Model Evaluation

The Logistic Regression model for sentiment classification was tested on a set of 196 unseen patient reviews. It achieved an overall accuracy of approximately 82.1%, correctly identifying the sentiment in 161 of the reviews. This strong performance suggests that the model is generally reliable in distinguishing between positive and negative sentiments, accurately predicting the correct label in more than four out of five cases.

A deeper look into the precision and recall metrics reveals a nuanced picture. The model demonstrated exceptional performance in detecting positive reviews, with a precision of 82% and a remarkably high recall of 97%. This means it was not only good at predicting when a review was positive but also very unlikely to miss truly positive sentiments. However, the model struggled with negative sentiment detection, with a recall of just 42.0% indicating that more than half of the actual negative reviews were misclassified as positive. Despite this, its precision for negative predictions was quite high at 85.0%, meaning that when the model did predict a negative review, it was often correct.

The confusion matrix clearly highlights a strong performance on predicting positive sentiment, with only 1 false negative, indicating the model almost never misses a truly positive review. However, it struggles with correctly identifying negative reviews, leading to 42 false positives, where negative feedback was mislabelled as positive. This trend confirms a significant bias toward predicting positive sentiment, which may stem from class imbalance in the dataset or limitations in feature separation. This outcome sets a meaningful baseline for comparison against VADER, a rule-based sentiment analyser, which may offer different strengths in identifying negative tones more explicitly..

Confusion Matrix - Logistic Regression



The key takeaway from the Logistic Regression evaluation was its strong ability to identify positive feedback but a considerable struggle to accurately detect dissatisfied customers. This imbalance in performance across classes suggested the need for techniques to improve negative recall, such as class weighting, oversampling of negative examples, or further feature engineering.

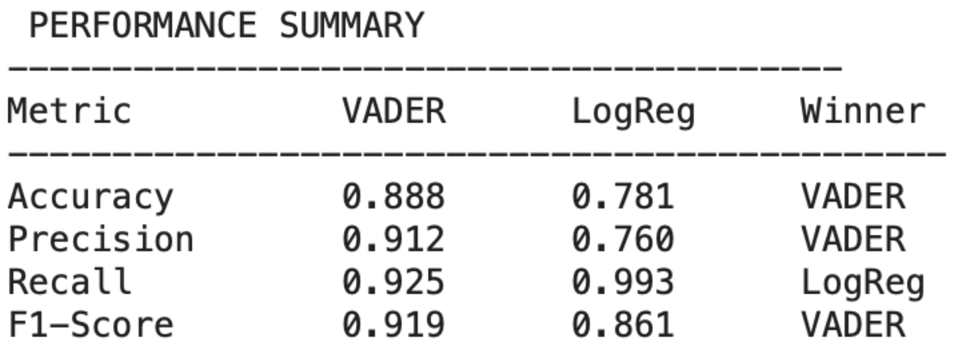
Comparison with VADER Sentiment Analyzer

To provide a robust evaluation of sentiment classification approaches, the VADER SentimentIntensityAnalyzer was applied to the same test dataset and directly compared against the Logistic Regression model using consistent evaluation metrics. The results revealed a clear advantage for VADER, which achieved a higher overall accuracy of 88.8%, outperforming Logistic Regression's 78.1%. This improvement in accuracy reflects VADER's better generalization across both positive and negative sentiments.

In terms of precision, recall, and F1-score, VADER consistently outperformed the supervised model. VADER achieved a precision of 91.2%, higher than Logistic Regression's 76.0%, indicating fewer false positives. Although Logistic Regression had an exceptional recall for the positive class (99.3%), it came at the cost of extremely poor recall for the negative class which resulted in a highly imbalanced model. In contrast, VADER offered a more balanced performance across both classes, with an overall recall of 92.5%, and crucially, a stronger ability to identify negative reviews.

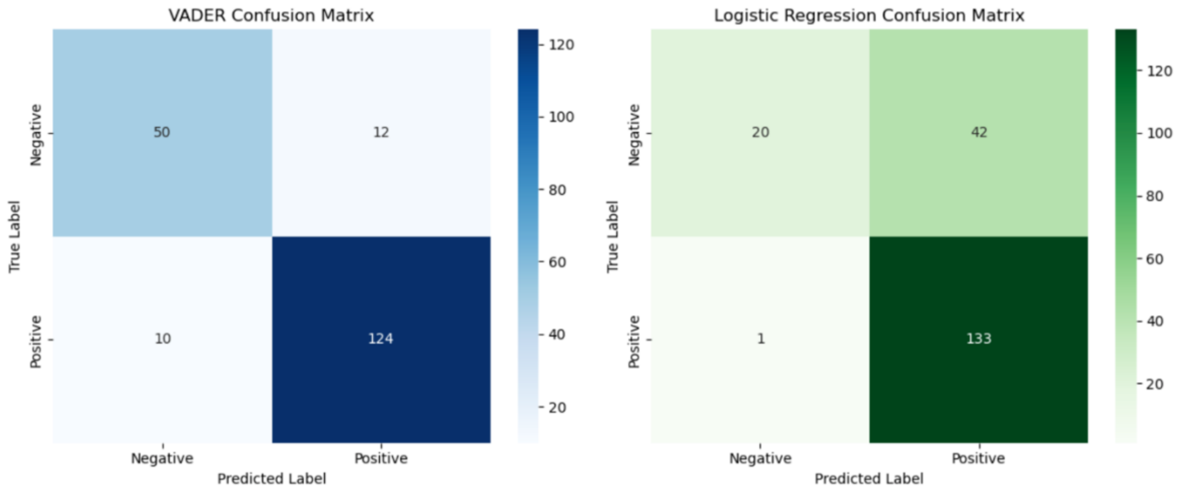
The F1-score further highlights this disparity in performance. VADER achieved an overall F1-score of 91.9%, compared to Logistic Regression's 86.1%. Most notably, VADER's F1-score for the negative class was 0.82, vastly outperforming the Logistic model's score of 0.48, indicating that VADER was much more reliable at correctly identifying negative sentiments. This comparison demonstrates that while Logistic Regression is highly sensitive to positive sentiment, VADER offers a more interpretable, balanced, and effective alternative for sentiment analysis in real-world feedback data.

Performance Summary - VADER vs. Logistic Regression



```
 PERFORMANCE SUMMARY
----------------------------------------
Metric          VADER      LogReg     Winner
---------------------------------------------
Accuracy        0.888      0.781      VADER
Precision       0.912      0.760      VADER
Recall          0.925      0.993      LogReg
F1-Score        0.919      0.861      VADER
```

Confusion Matrix – VADER and Updated Logistic Regression



Conclusion

This project analysed hospital reviews using text analytics to uncover common themes and assess patient sentiment. Topic modelling through Latent Dirichlet Allocation (LDA) revealed clear patterns in patient feedback, identifying key areas of satisfaction and concern. For sentiment classification, a comparative analysis showed that the VADER SentimentIntensityAnalyzer outperformed the supervised Logistic Regression model. While Logistic Regression achieved high recall for positive reviews, it struggled with identifying negative sentiment, leading to imbalanced results. In contrast, VADER demonstrated stronger precision, balanced recall across both sentiment classes, and a higher overall F1-score, making it a more effective and interpretable choice for real-world feedback analysis.

A critical takeaway from this study is the opportunity to improve efficiency in future sentiment analysis efforts. Since lower-rated reviews (3 stars or below) often correlate strongly with negative sentiment and areas of dissatisfaction, future models can focus on this subset to reduce computational overhead. This targeted approach enables faster, more focused analysis of high-priority feedback, allowing healthcare providers to take timely, informed action on issues impacting patient experience. By combining robust, balanced sentiment analysis with strategic focus, organizations can enhance service quality while maintaining analytical efficiency.

Bibliography

Ph.D., J. M., 2024. *What is Latent Dirichlet allocation?*. [Online]
Available at: https://www.ibm.com/think/topics/latent-dirichlet-allocation
[Accessed 27 June 2025].

Dylan, D. w., 2024. *Find the Most COMMON Words with Python WordCloud*. [Online]
Available at: https://www.youtube.com/watch?v=jB1XMrv_dCA
[Accessed 24 June 2025].

Science, R. &. M. D., 2024. *Hands-On Machine Learning: Logistic Regression with Python and Scikit-Learn*. [Online]
Available at: https://youtu.be/aL21Y-u0SRs?si=_AOJbJE3nXgkGYKp
[Accessed 24 June 2025].