Project Report

Sentiment Analysis on Yelp Reviews — Impact of Data Preprocessing on Model Performance

1. Project Statement

The goal of this project is to explore how effective data preprocessing enhances sentiment classification performance. Using a subset of the Yelp Reviews dataset, we analyze customer feedback and train machine learning models to predict sentiment (positive, neutral, or negative). We demonstrate the influence of text cleaning, transformation, and feature engineering on both data quality and model accuracy.

2. Data Collection

Dataset: Yelp Reviews Dataset (CSV version) —
https://www.kaggle.com/datasets/flyersteve/yelp-csv
Size: ~7 million reviews, 10 columns
Selected fields: text, stars, date
Sample used for analysis: ~50,000 reviews

The dataset contains real-world restaurant reviews from Yelp users, including the text of each review, a numerical star rating (1–5), and associated metadata.

3. Data Visualization (Preliminary EDA)

We performed initial exploratory analysis to understand the dataset's structure and key patterns:
- Distribution of ratings (stars) → shows class imbalance (mostly positive reviews).
- Review length before vs after cleaning → confirms text normalization reduces noise.
- Boxplots & histograms → reveal longer reviews tend to be more positive.
- Time trends (stars per year) → track variations in satisfaction over time.
- Word clouds → highlight most frequent words in positive vs. negative reviews.

Goal: Identify key relationships and ensure the data is representative before modeling.

4. Data Cleaning

Steps applied:
- Removed missing or empty texts.
- Dropped duplicate reviews.
- Removed very short reviews (less than 3 words).
- Normalized text (lowercasing, removing punctuation and symbols, stripping whitespace).

Effect: Reduced noise and improved text uniformity, ensuring meaningful features for vectorization.

5. Data Transformation

Applied TF-IDF Vectorization to transform raw text into numerical representations:
- Max 5000 features
- Removed English stop words
- Used train/test split (80/20)

Effect: Text transformed into a sparse matrix representing the importance of words for sentiment classification.

6. Feature Engineering

Added numeric features to enrich the dataset:
- review_len — number of words in each review
- year — extracted from the review date

These features capture temporal trends and linguistic richness, which help improve model interpretability and accuracy.

7. Preprocessing Pipeline

Constructed a unified ColumnTransformer + Pipeline for clean, reusable preprocessing:
- Text: TF-IDF vectorization
- Numeric: Standard scaling

Effect: Guarantees reproducibility and prevents data leakage.

8. Model Training & Evaluation

Two models were trained and evaluated:

Naive Bayes (text only): Accuracy = 81.7%
Logistic Regression (text + numeric): Accuracy = 85.8%

Evaluation metrics included accuracy, precision, recall, F1-score, and confusion matrices.

Insight: Adding numeric features and normalization improved classification performance by approximately 4%. The preprocessing pipeline also boosted model stability and interpretability.

9. Impact of Preprocessing

Before preprocessing:
- Inconsistent casing and punctuation.
- Redundant duplicates and short, meaningless reviews.
- High noise $\rightarrow$ poor model generalization.

After preprocessing:
- Cleaned, normalized, structured dataset.
- Higher accuracy (+4%).

- Clearer feature patterns and more robust predictions.

10. Conclusion

This project demonstrated that effective data preprocessing — including cleaning, transformation, and feature engineering — significantly improves model performance. The comparison between Naive Bayes and Logistic Regression confirms that text-only models are efficient baselines, but preprocessing and multi-feature integration yield measurable performance gains. The final model achieved 85.8% accuracy, showing how preprocessing transforms noisy text into meaningful, predictive insights.