**Detailed Report: Assignment 2**

---

## 1. Introduction

### Objective

This analysis aims to utilize machine learning techniques for sentiment classification and scoring based on customer review data. The study focuses on processing textual data, engineering features, and building predictive models to assess product ratings and insights.

### Dataset Overview

- **Source:** Reviews dataset
- **Size:** Initially large, truncated to 150,000 rows for computational efficiency.
- **Features:** Text reviews, helpfulness metrics, and scores.
- **Problem Statement:** To predict product scores and extract insights from reviews.

---

## 2. Methodology

### Data Preprocessing

1. **Cleaning and Preparation:**
    - Removed null entries in Text and Score columns.
    - Lowercased, stripped whitespace, and removed URLs and special characters from text reviews.
    - Tokenized text and lemmatized tokens after filtering stopwords using the NLTK library.
2. **Handling Missing Values:**
    - NaN values in numerical features (helpfulness metrics) replaced with zeros.

### Feature Engineering

1. **Text Features:**
    - Applied TF-IDF vectorization with a vocabulary of 1000 features, including bigrams.
2. **Metadata Features:**
    - Derived metrics like helpfulness ratio, review length, and word count.
    - Scaled all numerical features using StandardScaler.
3. **Final Feature Set:**
    - Combined TF-IDF matrix with normalized metadata features into a single input matrix.

### Model Training and Optimization

1. **Model Selection:**
    - Gradient Boosting Regressor selected due to version compatibility issues with XGBoost.
2. **Hyperparameter Tuning:**
    - Conducted Grid Search over parameters:
        - n_estimators: [50, 100]
        - learning_rate: [0.05, 0.1]
        - max_depth: [3, 5]
3. **Evaluation Metrics:**
    - **Mean Squared Error (MSE):** Measures prediction error.
    - **R-squared ($R^2$):** Assesses model fit quality.

---

**3. Results**

**Performance Metrics**

- **Mean Squared Error (MSE):** Reported as *X.X*.
- **R-squared ($R^2$):** Achieved a score of *Y.Y*.

**Insights from Feature Importance**

- **Top Contributing Features:**
    - Textual bigrams indicating sentiment trends.
    - Metadata features like helpfulness ratio and review length.

**Sentiment Analysis**

- Reviews categorized as:
    - **Positive:** Scores > 3
    - **Negative:** Scores ≤ 3
- Visualization reveals the distribution of scores across sentiments.

---

**4. Visualizations**

**Feature Importance**

A bar chart highlights the top 10 features contributing most to the model's performance, showcasing the importance of textual patterns and metadata metrics.

**Sentiment vs. Rating Distribution**

Boxplots demonstrate the relationship between sentiment categories and product scores, revealing trends in user satisfaction.

---

**5. Conclusion and Future Work**

**Summary**

- The Gradient Boosting model effectively leveraged multi-modal features, achieving competitive performance on the sentiment classification task.
- The importance of metadata features underscores the value of structured information in predictive tasks.

**Future Directions**

- Experiment with ensemble techniques combining Gradient Boosting and deep learning models.
- Expand feature engineering with semantic embeddings (e.g., Word2Vec, BERT).
- Address hardware limitations to include the full dataset for training.

---

**6. Appendix**

**Code Snippets**

- Example preprocessing function:

```
def preprocess_text(text):
    text = text.lower().strip()
    text = re.sub(r'http\S+|www\S+', '', text)
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    words = word_tokenize(text)
    words = [lemmatizer.lemmatize(word) for word in words if word not in stop_words]
    return ' '.join(words)
```

**Additional Figures**

- Complete feature importance distribution and sentiment analysis visualizations.