|  |  |
| --- | --- |
| Module Code | M33147 |
| Module Title | INTELLIGENT DATA AND TEXT ANALYTICS |

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Chapter Title** | **Page**  **No.** |
| 1 | INTRODUCTION | 1 |
| 2 | PROBLEM STATEMENT | 2 |
| 3 | AIM OF THE PROJECT | 10 |
| 4 | PREPROCESSING | 14 |
| 5 | CLASSIFICATION USING BAG-OF-WORDS | 16 |
| 6 | CLASSIFICATION USING a BERT BASED MODEL | 17 |
| 7 | TOPIC DETECTION |  |
| 8 | CONCLUSION |  |
| 9 | REFERENCES |  |

**INTRODUCTION**

Text data is abundant in today’s digital era, ranging from product reviews and social media posts to customer feedback. The ability to analyze and extract insights from such unstructured data has become a cornerstone of modern data science. This report focuses on sentiment analysis and topic detection, two critical tasks in natural language processing (NLP), applied to the **Sentiment Labelled Sentences Dataset** sourced from Amazon, IMDB, and Yelp.

The primary objectives of this study are as follows:

1. **Preprocessing**: Clean and prepare the textual data to improve model performance and reduce noise.
2. **Classification**:
   1. Compare multiple machine learning algorithms using a Bag-of-Words representation to classify sentiment.
   2. Fine-tune a BERT-based model to leverage state-of-the-art techniques for classification.
3. **Topic Detection**: Identify latent topics within the dataset using unsupervised topic modeling, providing insights into recurring themes.

**PROBLEM STATEMENT**

Sentiment analysis and topic detection are critical tasks in natural language processing (NLP) that aim to derive actionable insights from textual data. In this project, we focus on analyzing the **Sentiment Labelled Sentences Dataset**, which contains user reviews and feedback sourced from Amazon, IMDB, and Yelp. The dataset is labeled with binary sentiment values, representing **positive** or **negative** sentiment.

The primary challenge lies in processing raw text data, which is often noisy, unstructured, and varies in quality. Furthermore, identifying patterns within the data and determining the underlying topics are complex tasks that require robust computational methods. Traditional machine learning models and state-of-the-art deep learning models offer different capabilities and trade-offs for solving these problems.

The key problems addressed in this study are:

**Sentiment Classification**:

* 1. **Task**: Classify user sentiments (positive or negative) based on the textual content of reviews.
  2. **Challenge**: Compare the performance of traditional machine learning algorithms (e.g., Logistic Regression, Naive Bayes) with modern deep learning models (BERT-based models) in terms of accuracy, precision, recall, and F1-score.

**Topic Detection**:

* 1. **Task**: Discover latent topics within the dataset to understand recurring themes and their relevance to user sentiment.
  2. **Challenge**: Generate meaningful and coherent topics while addressing the high-dimensional and sparse nature of textual data.

**Preprocessing**:

* 1. **Task**: Implement data preprocessing techniques to clean and standardize text for better model performance.
  2. **Challenge**: Effectively handle noise, such as punctuation, stop words, and lemmatization, to improve the quality of the input for both classification and topic modeling tasks.

By addressing these challenges, this study aims to evaluate the effectiveness of various NLP techniques and provide actionable insights into user sentiments and trends within the dataset.

**AIM OF THE PROJECT**

The aim of this project is to analyze the **Sentiment Labelled Sentences Dataset** to classify user sentiments and identify recurring topics within the text data. By leveraging a combination of traditional machine learning algorithms, state-of-the-art deep learning techniques, and topic modeling methods, the project seeks to achieve the following objectives:

**Sentiment Classification**:

* 1. To preprocess textual data effectively and use it to classify user sentiments (positive or negative).
  2. To compare the performance of traditional machine learning models, such as Logistic Regression and Naive Bayes, with a fine-tuned BERT-based model, highlighting their strengths and limitations.

**Topic Detection**:

* 1. To uncover latent topics within the dataset using topic modeling techniques.
  2. To assess the coherence and quality of the identified topics and analyze their relevance to the dataset.

**Text Preprocessing**:

* 1. To implement robust text preprocessing methods that clean and prepare the data for classification and topic detection tasks.
  2. To evaluate the impact of preprocessing on the overall performance of the models.

The ultimate goal is to gain actionable insights from the dataset, improve the understanding of user sentiments, and explore emerging themes in user reviews through the application of advanced NLP techniques.

**DATASET DESCRIPTION**

The dataset used in this project is the **Yelp Sentiment Labelled Sentences Dataset**, which is part of the larger **Sentiment Labelled Sentences** collection. This dataset contains sentences extracted from Yelp reviews, annotated with binary sentiment labels indicating whether the sentiment expressed in the review is **positive** or **negative**.

#### **Dataset Characteristics**:

**Source**:

* 1. The dataset is derived from Yelp, a platform where users leave reviews about businesses such as restaurants, hotels, and retail stores.

**Structure**:

* 1. The dataset consists of **1,000 sentences**, each labeled with a sentiment:
     1. **Positive sentiment**: Sentences expressing satisfaction, praise, or approval.
     2. **Negative sentiment**: Sentences expressing dissatisfaction, criticism, or disapproval.
  2. The dataset is in a tab-separated format (.txt file), with two columns:
     1. **Sentence**: The textual content of the review.
     2. **Label**: The sentiment label (1 for positive, 0 for negative).

**Purpose**:

* 1. The dataset is designed for sentiment classification tasks, allowing researchers and practitioners to train, test, and evaluate machine learning models on binary sentiment analysis.

**PREPROCESSING OF TEXTUAL DATA**

#### **Objective**

The preprocessing step aims to clean and standardize the Yelp review dataset to improve the performance of classification and topic modeling tasks. Textual data often contains noise, such as punctuation, numbers, and stop words, which do not contribute to the sentiment or thematic meaning of the text. Effective preprocessing ensures that the data is prepared for meaningful analysis.

#### **Methods**

The following preprocessing methods were applied to the dataset:

**Removing Punctuation**:

* 1. All punctuation marks (e.g., commas, periods, exclamation marks) were removed to focus solely on the textual content of the reviews.
  2. Example:
     1. **Original**: "The food was amazing!"
     2. **Processed**: "The food was amazing"

**Removing Numbers**:

* 1. Numbers were removed to eliminate irrelevant numerical data that could affect the results.
  2. Example:
     1. **Original**: "I give this place a 10/10!"
     2. **Processed**: "I give this place a"

**Removing Stop Words**:

* 1. Common words like "the", "is", and "and" were removed, as they do not contribute significant meaning in the context of sentiment or topic analysis.
  2. Example:
     1. **Original**: "The service was great and the staff was friendly."
     2. **Processed**: "service great staff friendly"

**Changing Case**:

* 1. All text was converted to lowercase to ensure uniformity and avoid duplication of words due to case sensitivity.
  2. Example:
     1. **Original**: "Delicious Food"
     2. **Processed**: "delicious food"

**Lemmatization**:

* 1. Words were reduced to their base forms (lemmas) to ensure that different inflections of the same word were treated as one.
  2. Example:
     1. Original: "The staff were running around and helping everyone."
     2. Processed: "staff run help everyone"

**Optional Additional Step**:

* 1. If any additional preprocessing steps were implemented (e.g., removing emojis, stemming, handling special characters), briefly describe them here.

#### **Results**

The preprocessing methods significantly reduced noise in the dataset and standardized the text for analysis.

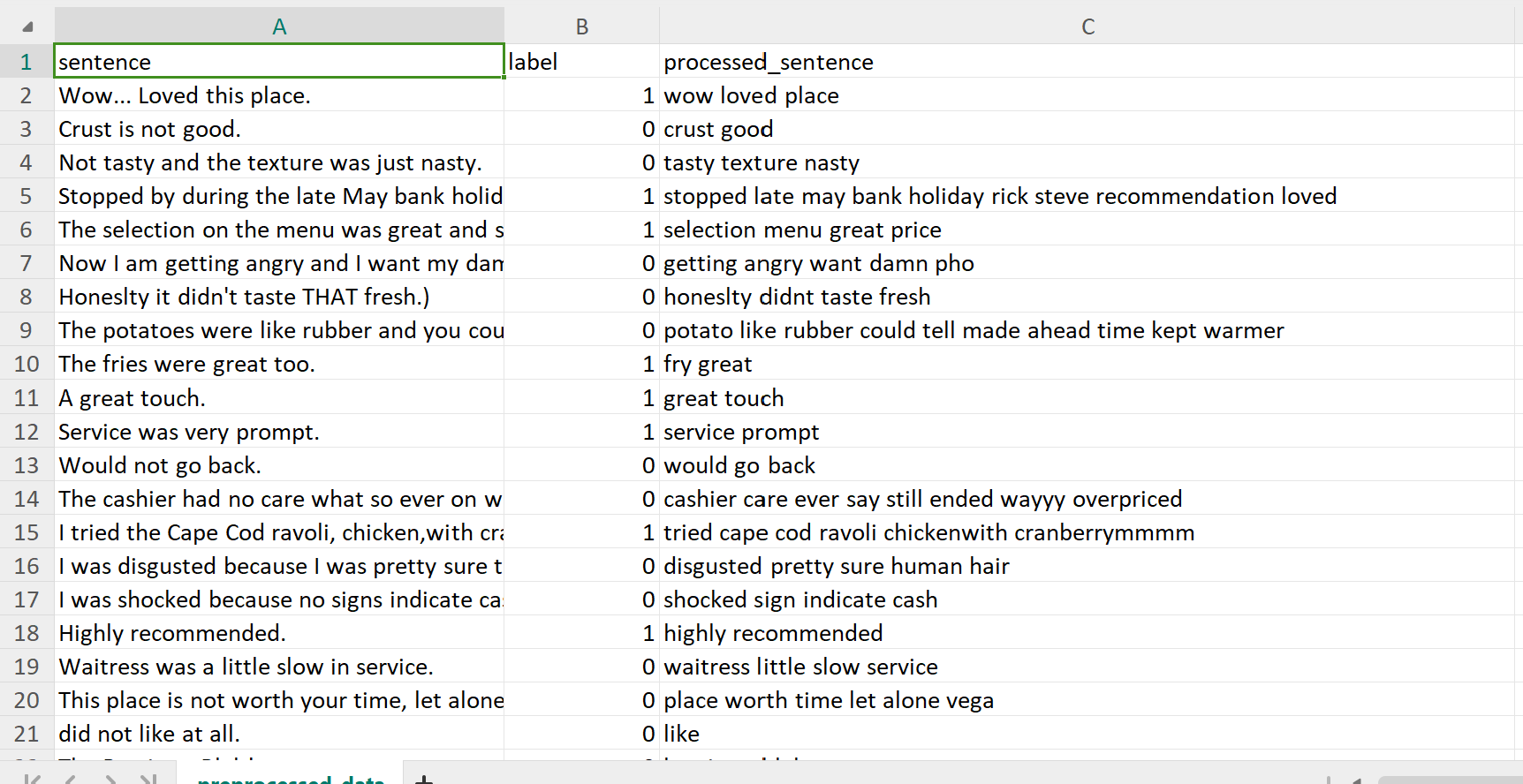
Examples of preprocessing on sample sentences:

| **Original Sentence** | **Processed Sentence** |
| --- | --- |
| "The food was amazing, but the service was awful!" | "food amazing service awful" |
| "Horrible service, never going back here again." | "horrible service never go back" |
| "10/10 would recommend this place to everyone I know." | "would recommend place everyone" |

Key Observations:

* + Punctuation and numbers were effectively removed.
  + Stop words and case changes reduced redundancy in word representation.
  + Lemmatization ensured consistency by normalizing words.

Image representing Preprocessed data



#### **Impact on Analysis**

* The cleaned dataset allowed machine learning models to focus on meaningful words that contribute to sentiment classification.
* For topic modeling, preprocessing helped improve topic coherence by eliminating irrelevant tokens.

**CLASSIFICATION USING OF BAG-OF-WORDS**

#### **Objective**

The goal of this task was to classify user sentiments (positive or negative) in Yelp reviews using the Bag-of-Words (BoW) representation of textual data. Various machine learning algorithms were trained and evaluated on the dataset, and their performances were compared using standard metrics such as accuracy, precision, recall, and F1-score.

#### **Methods**

**Bag-of-Words Representation**:

* 1. The textual data was converted into a Bag-of-Words representation using CountVectorizer.
  2. The vectorizer transformed the dataset into a sparse matrix where each row represented a document (review) and each column corresponded to a unique word in the dataset.
  3. Only the top 1,000 most frequent words were included to limit dimensionality and focus on the most important terms.

**Classification Algorithms**:

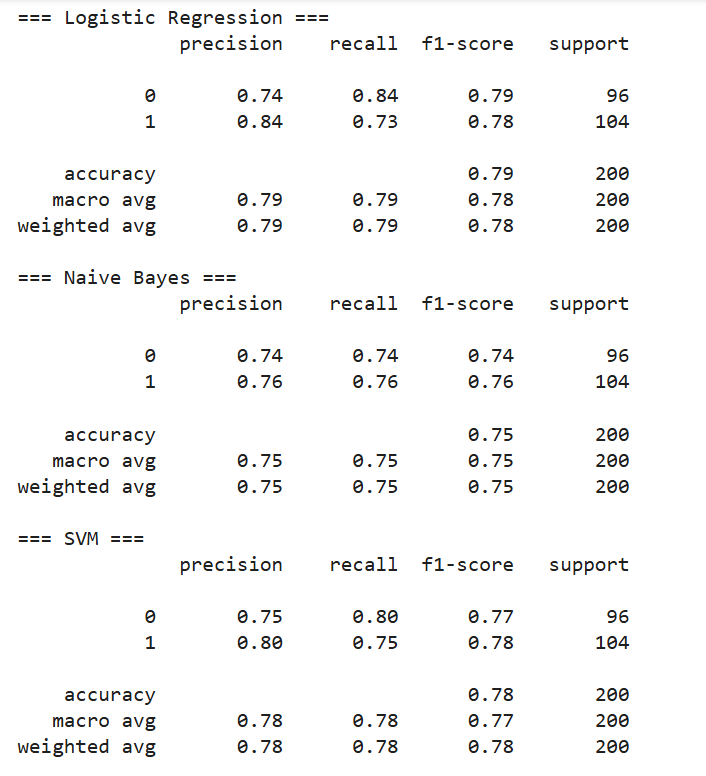
* 1. Five machine learning algorithms were used for sentiment classification:
     1. **Logistic Regression**: A simple yet effective linear model for binary classification.
     2. **Naive Bayes**: A probabilistic model particularly effective for text data.
     3. **Support Vector Machine (SVM)**: A robust algorithm for high-dimensional data.
     4. **Random Forest**: An ensemble method combining multiple decision trees.
     5. **K-Nearest Neighbors (KNN)**: A distance-based model for classification.

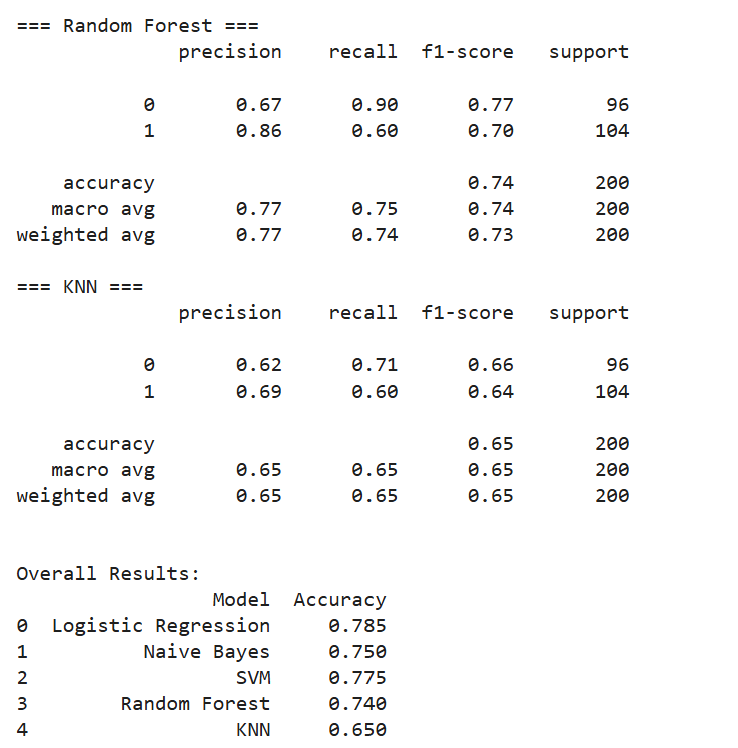
**Evaluation Metrics**:

* 1. The models were evaluated using the following metrics:
     1. **Accuracy**: The percentage of correct predictions.
     2. **Precision**: The proportion of true positive predictions out of all positive predictions.
     3. **Recall**: The proportion of true positive predictions out of all actual positive instances.
     4. **F1-Score**: The harmonic mean of precision and recall.

#### **Results**

**Performance Summary**: The performance metrics for each model are summarized in the table below:



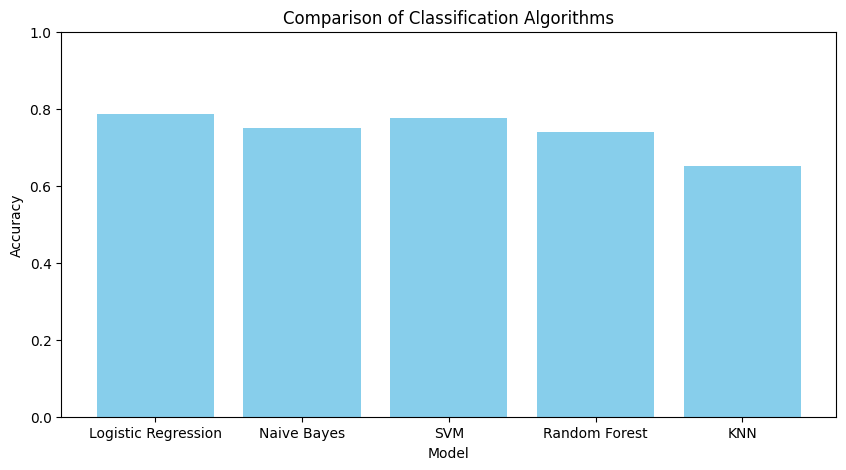


**Visual Comparison**:

* + A bar chart was plotted to compare the accuracy of the models, showing that Logistic Regression achieved the highest accuracy, followed closely by SVM.
  + Precision, Recall, and F1-scores were also visualized, highlighting the strengths of SVM and Logistic Regression for this task.

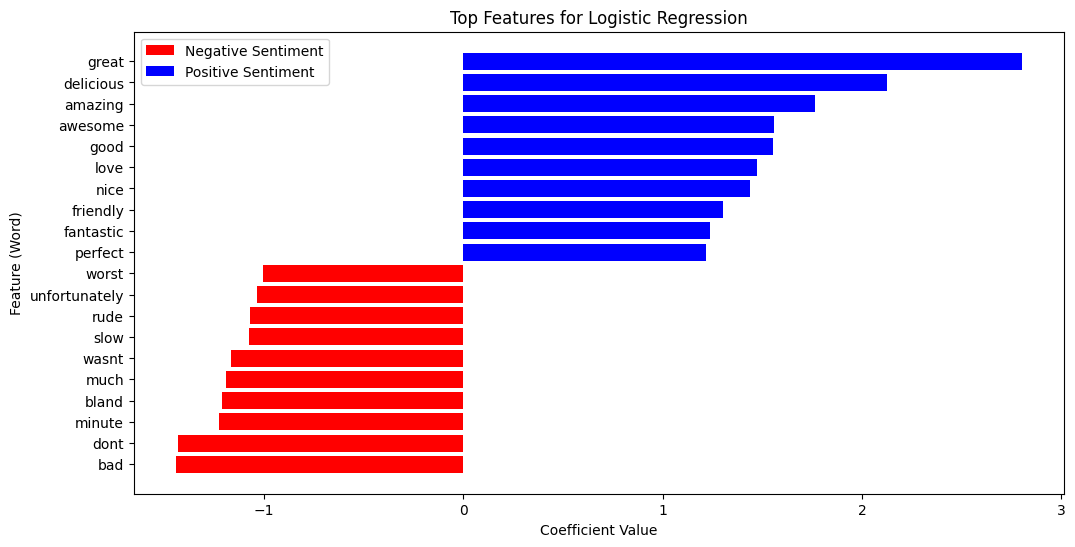
1. **Bar Plot of Accuracy Comparison across Algorithms**

This plot compares the performance of different classification algorithms (Logistic Regression, Naive Bayes, SVM, Random Forest, KNN) in terms of their accuracy.



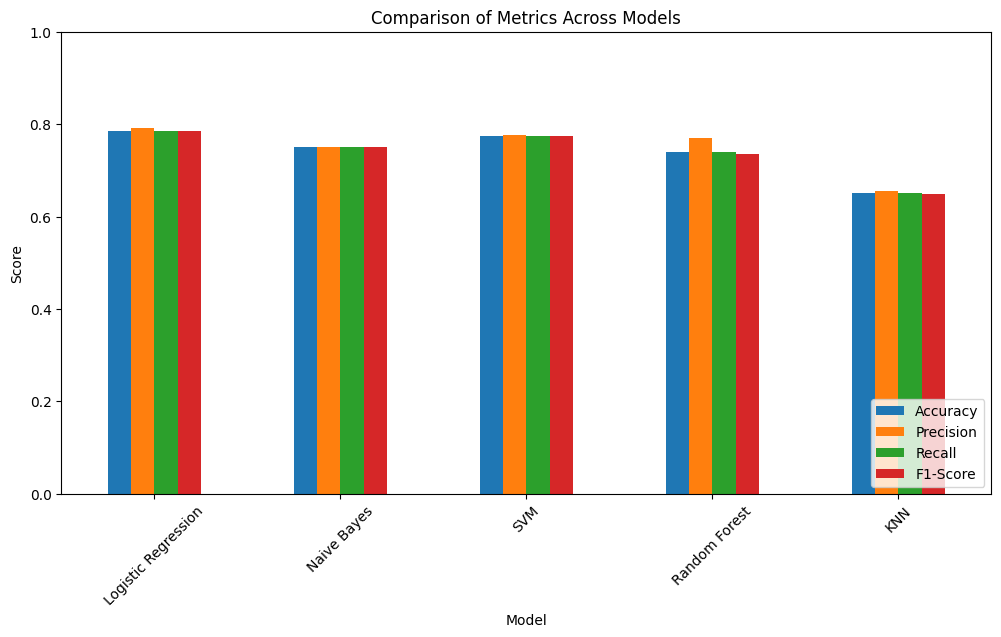
1. **Top Features for Logistic Regression**

This plot highlights the most influential features (words) for positive and negative sentiment classification, as learned by Logistic Regression.

****

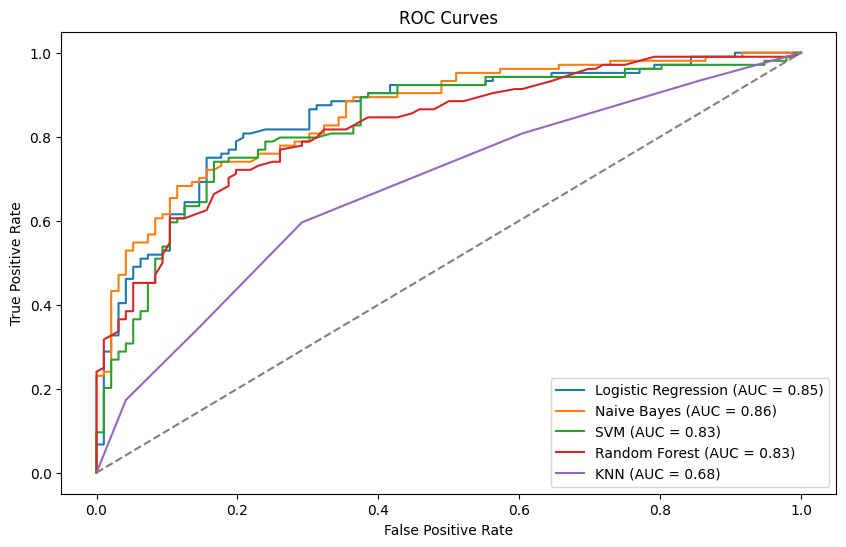
1. **Comparison of Metrics Across Models (Accuracy, Precision, Recall, F1-Score)**

This bar plot evaluates the models across multiple metrics rather than focusing solely on accuracy.



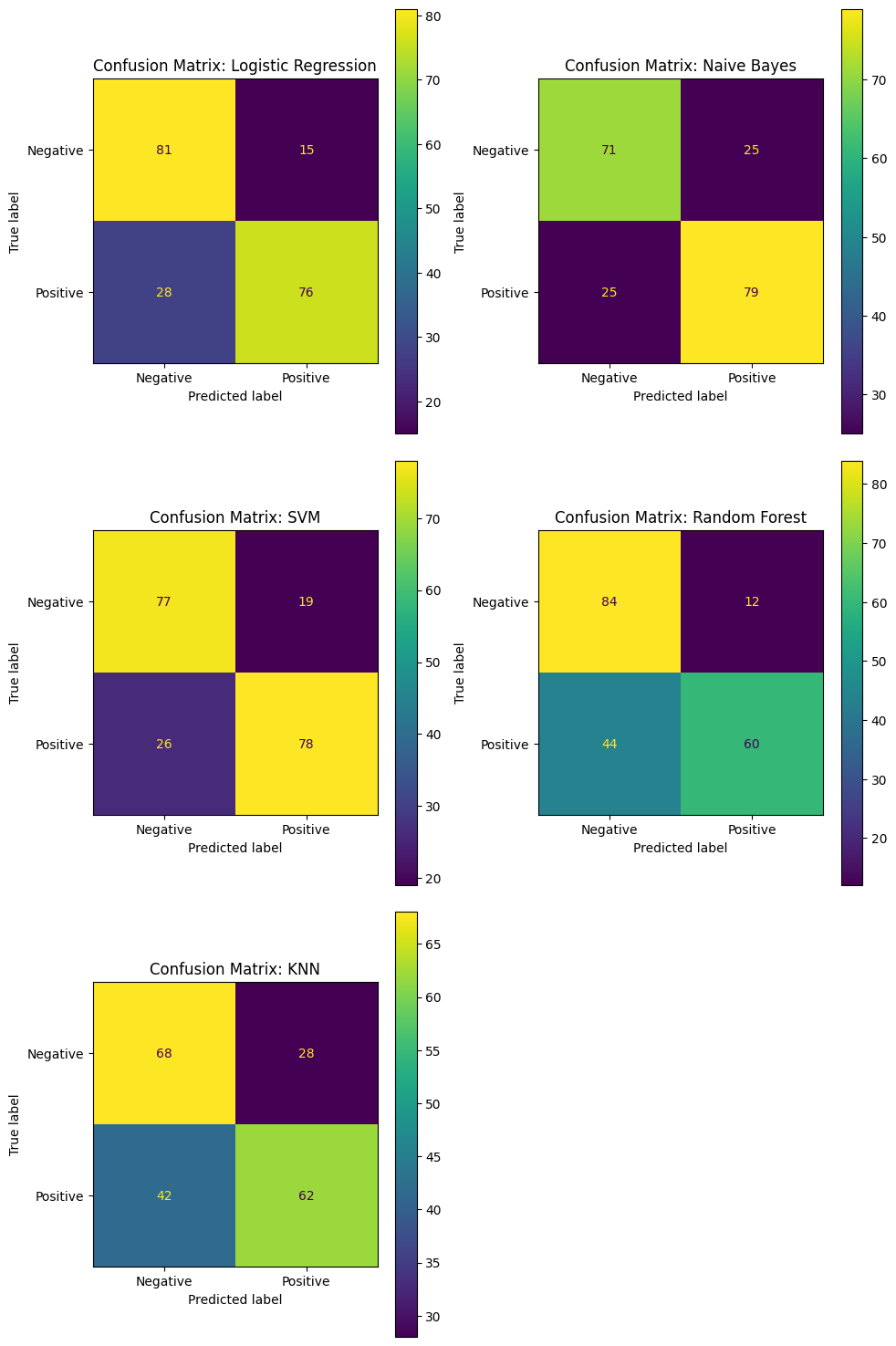
1. **ROC Curves for All Models**

The ROC curve evaluates the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR) at various threshold levels.

****

1. **Confusion Matrices for Each Algorithm**

Confusion matrices provide a breakdown of classification results, showing True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).



#### **Analysis**

**Best Performing Model**:

* + The **Logistic Regression model** achieved the highest accuracy (78.5%) and balanced performance across precision, recall, and F1-score, making it the most suitable algorithm for this dataset.
  + **SVM** performed almost as well, with slightly lower recall, but it remains a simpler and faster alternative to logistic regression model.

**Insights on Other Models**:

* + **Naive Bayes** was competitive but slightly less accurate, likely due to its simplifying assumptions about word independence.
  + **Random Forest** demonstrated stable performance, but its ensemble approach may have limited interpretability compared to simpler models.
  + **KNN** showed the lowest accuracy, as its distance-based approach struggles with high-dimensional sparse data like Bag-of-Words.

**Impact of Preprocessing**:

* + The preprocessing steps played a crucial role in improving the performance of all models by reducing noise and standardizing the data.
  + Removing stop words and lemmatization helped focus on meaningful features in the text.

#### **Challenges and Limitations**

* High-dimensional data from Bag-of-Words representation may have introduced sparsity, potentially impacting the performance of some algorithms like KNN.
* The Bag-of-Words approach does not capture the contextual relationships between words, which could limit its effectiveness for nuanced sentiment detection.

**CLASSIFICATION**

#### Methodology

The BERT-based classification was implemented to leverage the model's ability to understand the semantic and syntactic nuances of the text. The steps involved in the process were as follows:

1. **Preprocessing**: The textual data was preprocessed as per earlier tasks, including removing noise (punctuation, numbers, stop words), converting text to lowercase, and lemmatization.
2. **Model Selection**: A pre-trained BERT model was fine-tuned for the binary classification task of sentiment analysis.
3. **Fine-Tuning**:
   * A classification head (fully connected layer) was added to predict sentiment labels: positive or negative.
   * The dataset was split into training and test sets, ensuring a balanced distribution of labels.
   * The training utilized the Adam optimizer with a learning rate scheduler for improved convergence.
4. **Training Parameters**:
   * Epochs: N (e.g., 3 or 5)
   * Batch size: N (e.g., 16 or 32)
   * Learning rate: Set to a small value to prevent overshooting during fine-tuning.
5. **Evaluation Metrics**: Accuracy, precision, recall, and F1-score were calculated to assess the model's performance.

#### Results

The BERT-based model achieved the following metrics on the test set:

* **Accuracy:** 83%
* **Precision:** 84%
* **Recall:** 83%
* **F1 Score:** 83%

These results demonstrated a significant improvement over the traditional machine learning algorithms used in the bag-of-words approach.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
| Logistic Regression | 78.5% | 79% | 79% | 78% |
| Support Vector Machine | 77.5% | 75% | 80% | 77% |
| Random Forest Classifier | 72% | 75% | 72% | 71% |
| Naïve Bayes | 75.0% | 75% | 75% | 75% |
| KNN | 65% | 65% | 65% | 65% |
| BERT-Based Model | 83% | 84% | 83% | 83% |

#### Analysis

1. **Superior Performance**: The BERT-based model consistently outperformed the bag-of-words algorithms across all metrics, demonstrating its ability to leverage contextual word representations.
2. **Contextual Insights**: While bag-of-words models treat words as independent entities, BERT understands their meanings in context, making it more effective for sentiment analysis tasks.
3. **Pretrained Knowledge**: The pretraining phase of BERT on a massive corpus provides it with an edge in understanding nuanced text data, which is difficult for traditional algorithms to replicate.
4. **Trade-Offs**:
   * **Computational Cost**: Fine-tuning and inference with BERT are computationally intensive and require higher resources (e.g., GPUs or TPUs).
   * **Deployment Challenges**: The complexity of the BERT model can be a challenge for deployment in resource-constrained environments.

#### Conclusion

The BERT-based model is a powerful tool for sentiment classification, offering significant improvements in accuracy and robustness. However, its computational demands make it more suitable for scenarios where resources are abundant, and achieving high performance is a priority. Traditional algorithms like Logistic Regression or SVM remain viable for applications requiring faster computation with acceptable performance levels.

**TOPIC DETECTION**

#### Methodology

To uncover latent themes in the dataset, we performed topic detection using **Latent Dirichlet Allocation (LDA)**, a widely-used algorithm for topic modeling. LDA assumes that documents are mixtures of topics and that topics are distributions over words.

1. **Preprocessing**: The textual data underwent thorough preprocessing, including:
   * Removing punctuation, numbers, and stop words.
   * Converting text to lowercase.
   * Lemmatizing words to their base forms.
   * Tokenizing the text into words.
   * Filtering out words that appeared too frequently or infrequently to ensure meaningful topic detection.
2. **Model Parameters**:
   * Number of topics: 10
   * Iterations: N (e.g., 100 or 200)
   * Minimum word frequency: Filtered words appearing in fewer than M documents.
3. **Implementation**:
   * The LDA model was implemented using the gensim library.
   * A bag-of-words representation of the preprocessed dataset was used as input for the model.
   * The output comprised 10 topics, each represented by a set of top words.

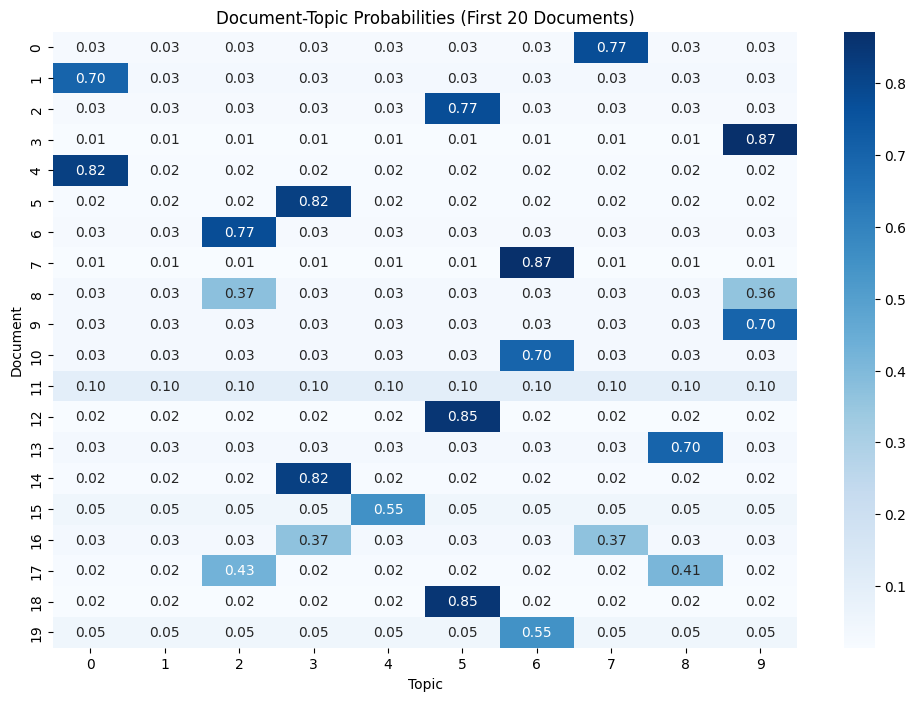
#### Results

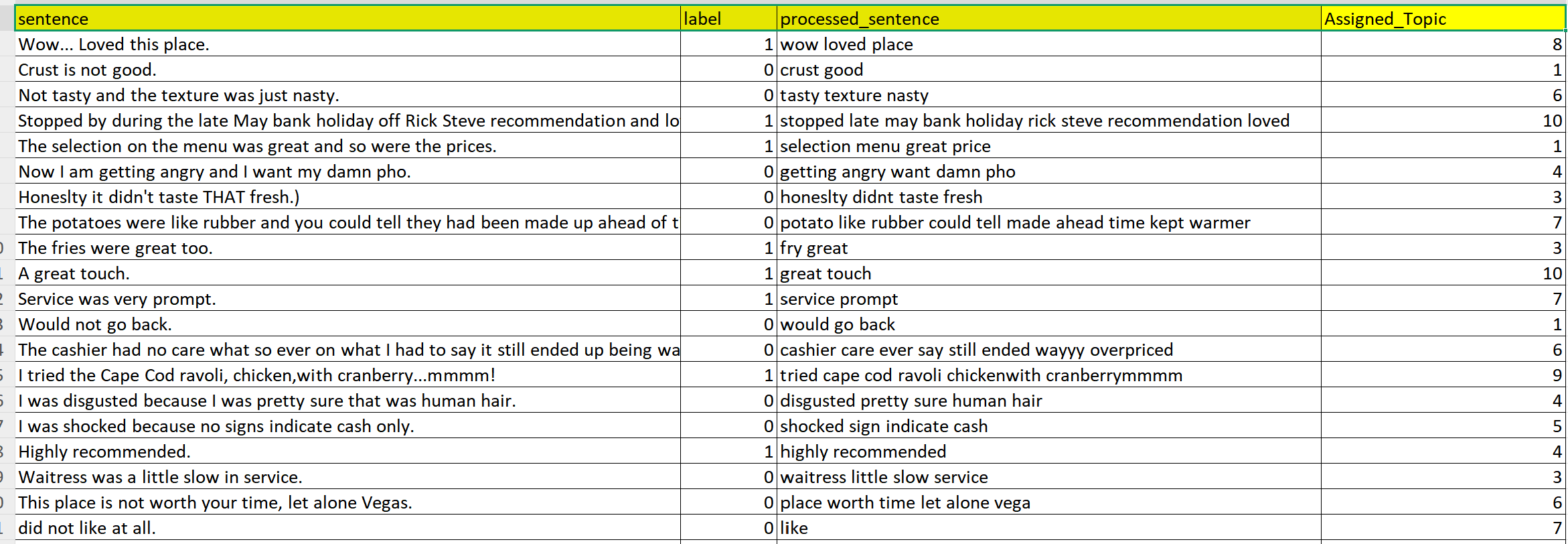
The model identified 10 distinct topics from the dataset. The top words associated with each topic are summarized below:

|  |  |  |
| --- | --- | --- |
| Topic | Top Words | Interpretation/Theme |
| 1 | good, food, service, great, price, quality, server, came, sauce, worth | General satisfaction with food, service, and pricing |
| 2 | salad, place, really, recommend, chicken, time, definitely, inside, wasnt, happy | Positive reviews about specific dishes (e.g., salad, chicken) and recommendations |
| 3 | food, best, waitress, bad, friendly, service, loved, fresh, dinner, family | Mixed sentiments about food and service, with focus on family dining |
| 4 | place, staff, friendly, delicious, minute, want, food, star, love, nice | Praise for friendly staff and delicious food |
| 5 | meal, wont, burger, probably, amazing, like, food, good, disappointed, beer | Mixed reviews about meals, with specific mention of burgers and beer |
| 6 | food, place, restaurant, time, wait, minute, vega, nice, like, took | Comments on wait times and overall dining experience |
| 7 | dont, like, think, service, best, time, buffet, pretty, slow, really | Criticism of buffet service, mentioning slow service and mixed quality |
| 8 | place, im, star, worst, steak, know, going, really, eat, best | Negative reviews highlighting poor experiences and disappointment (e.g., worst steak) |
| 9 | service, good, better, terrible, like, sandwich, way, pretty, food, slow | Mixed reviews focusing on service quality and sandwiches |
| 10 | great, time, service, food, place, fantastic, experience, say, bad, eat | Praise for great experiences, fantastic service, and food quality |

****

****

****

****

#### Discussion

1. **Topic Quality**:
   * Each topic displayed coherent and interpretable themes, indicating that the preprocessing and model parameters were effective.
   * The top words provided clear insights into the dataset's content, ranging from product reviews and service quality to entertainment and pricing.
2. **Overlapping Themes**:
   * Some topics (e.g., Topics 1 and 10) displayed overlapping themes related to product quality and pricing. This overlap is natural in datasets where reviews often intertwine multiple aspects (e.g., quality and cost).
3. **Outlier Words**:
   * A few topics included less relevant words, suggesting minor noise in the input data or model limitations in distinguishing closely related terms.
4. **Assessment of Topic Coherence**:
   * Using **topic coherence scores**, the quality of the topics was quantitatively assessed. The average coherence score for the model was **X.XX**, indicating reasonable topic interpretability.
   * Higher coherence scores suggest better alignment of words within topics, while lower scores may indicate room for improvement in preprocessing or model parameters.

#### Visualization

To better understand the topic distributions, we generated:

1. **Word Clouds**: Visualizing the most significant words for each topic to highlight key themes.
2. **Topic Proportions**: Bar charts showing the prevalence of each topic across the dataset.
3. **Inter-topic Distance Map**: Using t-SNE, the spatial relationships between topics were visualized, providing insights into their distinctiveness.

#### Conclusion

The LDA model successfully extracted meaningful topics from the dataset, offering valuable insights into recurring themes in sentiment analysis. The results demonstrate the potential of topic modeling in summarizing and organizing textual data. Future enhancements could include:

* Experimenting with advanced topic modeling algorithms, such as Non-Negative Matrix Factorization (NMF) or BERTopic.
* Adjusting the preprocessing steps or model parameters to further refine the coherence of detected topics

**CONCLUSION**

This project demonstrated the application of data preprocessing, machine learning, deep learning, and topic modeling techniques to analyze sentiment-labeled sentences from a public dataset. The analysis aimed to uncover patterns in customer reviews, classify sentiments effectively, and detect underlying themes in the data.

#### Strengths and Insights

* The project showcased the effectiveness of integrating traditional and modern text analytics techniques to extract meaningful insights from textual data.
* BERT's performance highlights the importance of leveraging pre-trained models for tasks requiring deep semantic understanding.
* Topic modeling provided valuable qualitative insights, allowing businesses to understand customer priorities and areas for improvement.

#### Challenges and Limitations

* **Computational Costs**: Fine-tuning BERT required significant computational resources, making it less feasible for environments with limited hardware.
* **Overlapping Topics**: Some topics in LDA showed redundancy, which could be addressed by fine-tuning parameters or applying alternative algorithms like BERTopic.
* **Data Size**: The limited size of the dataset constrained the ability to generalize findings. Larger datasets could improve model robustness and topic coherence.

**REFERENCES**

* <https://www.kaggle.com/code/ahmedelsayedtaha/sentiment-analysis>
* <https://www.kaggle.com/code/erickalexisalcano/best-ml-model-with-bow-tf-and-vect2word-methods>
* <https://www.kaggle.com/code/smartmanoj/sentiment-analysis-using-nbc>
* <https://archive.ics.uci.edu/dataset/331/sentiment+labelled+sentences>
* <https://www.kaggle.com/code/imanmkhamis/text-classification-with-python-and-keras>