Hotel reviews

Gehad Mamdouh : CO-Occurrence feature extraction

Gehad Ahmed : TF-IDF feature extraction

Aalaa shehata: n-grams feature extraction

Israa Alaa: CountVectorizer feature

Supervisor: Dr. Wesam Ahmed

Group Number: 2

Submission date: May 12th, 2025

**Chapter 1**

**Introduction:**

In this project, our team of four members collaborated to build a sentiment analysis pipeline for text data. The workflow began with thorough data preprocessing, including cleaning, stopword removal, and lemmatization, to ensure high-quality input for the modeling phase.

**Each team member** was assigned a different feature extraction technique to apply to the preprocessed dataset:

N-grams

CountVectorizer

TF-IDF

Co-occurrence Matrix

Preprocessing:

The dataset we worked with contained multilingual text data, so the first step in our preprocessing pipeline was translating all text into English to ensure consistency across the entire dataset.

After translation, we applied a series of text-cleaning operations to prepare the data for feature extraction. The key preprocessing steps included:

**Converting emojis to descriptive words**, to preserve sentiment-related expressions often conveyed through emojis.

**Lowercasing all text**, to reduce redundancy caused by case differences.

**Removing punctuation** and extra whitespace, which do not carry semantic meaning.

**Eliminating unnecessary numerical values** that were irrelevant to the sentiment.

**Removing stop words,** such as "the", "is", "and", which do not contribute significantly to sentiment detection.

**Tokenizing the text,** i.e., breaking sentences into individual words or tokens.

**Lemmatizing tokens** to reduce words to their base or dictionary form (e.g., “running” → “run”).

**To gain a better understanding of the data**, we also generated word clouds to visually represent the most frequent terms in the dataset after cleaning.

This comprehensive preprocessing ensured that the input to our models was clean, standardized, and semantically meaningful.

Chapter 2

Code Implementation:

https://colab.research.google.com/drive/1fc6liQkipSdazvwIsZCkAUkghbgFRTz-?usp=sharing

<https://colab.research.google.com/drive/1dyGtZby8gORE-ofKefDMPKFB8mo8hd-a?usp=sharing>

<https://colab.research.google.com/drive/1NxV0WT9slG8EUM2jtog-3VXFIeVlt6g-?usp=sharing>

Figure 2.1: Word cloud



**1-N-grams method (Aalaa Shehata)**

|  |  |  |
| --- | --- | --- |
| Model | Training Accuracy | Testing Accuracy |
| SVM | 90% | 85% |
| Logistic regression | 88% | 85% |
| Random forest | 78% | 76% |

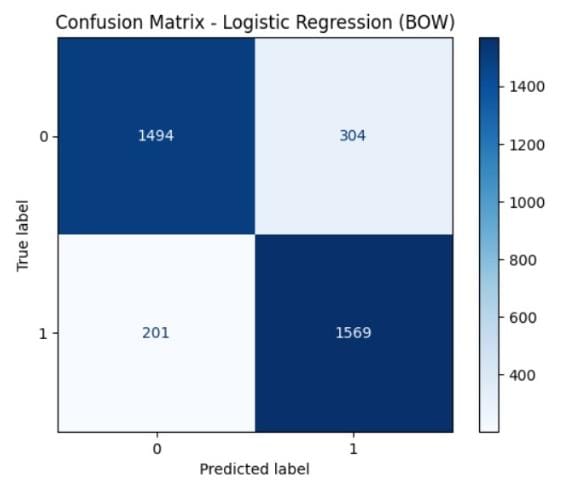
**2 -Tf-idf method (Gehad Ahmed)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Training Accuracy | Testing Accuracy | Precision | Recall |
| SVM | 87% | 85% | 85% | 86% |
| Naïve Bayes | 86% | 84% | 72% | 79% |
| Random forest | 76% | 75% | 85% | 82% |
| Dense | 92% | 87% | 86% | 88% |
| **Logistic**  **Regression** | 90% | 87% | 86% | 88% |
| **XGBoost** | 82% | 79% | 78% | 82% |

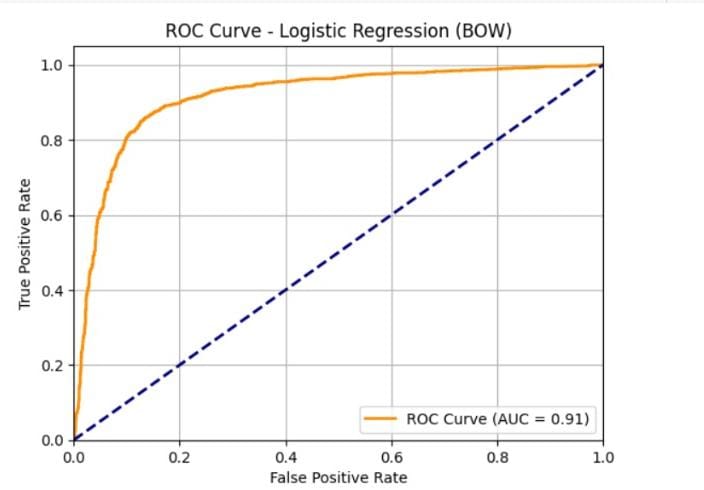
**3 - co-occurrence(Gehad mamdouh)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MODEL | Training accuracy | Test accuracy | Precisin | Recall | F1\_score | AUC |
| Logistic regression | 88.84% | 85.85% | 84% | 89% | 86% | 91% |
| Gradient boosting | 89.86% | 85.09% | 84% | 87% | 85% | 93% |
| Naive bayes | 90.72% | 85.43% | 84% | 87% | 86% | 91% |
| Dense | 89.76% | 86.32% | 89.37% | 82.20% | 85.64% | 94.39% |

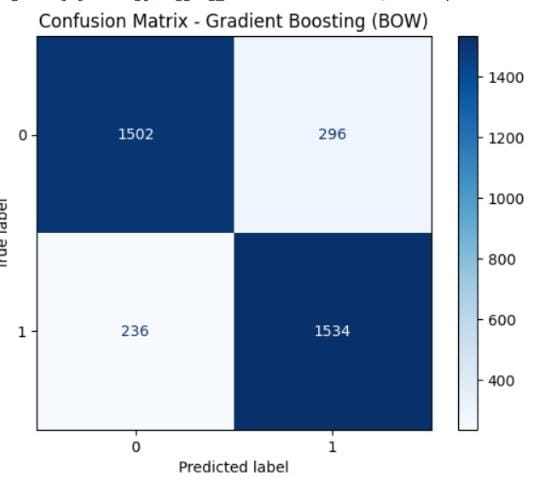
**-Confusion matrix of logisitic regression**

****

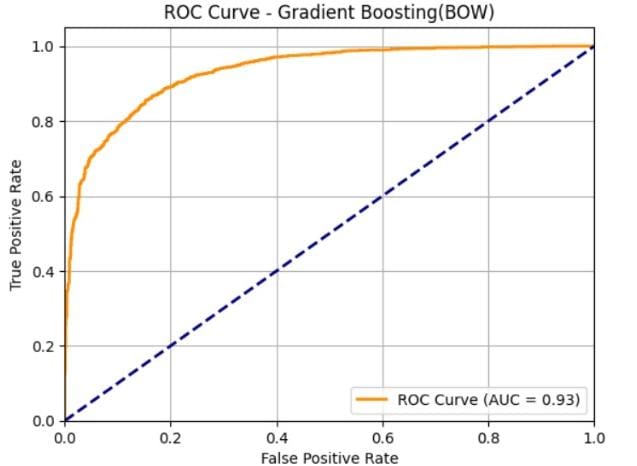
**-ROC of logistic regression**

****

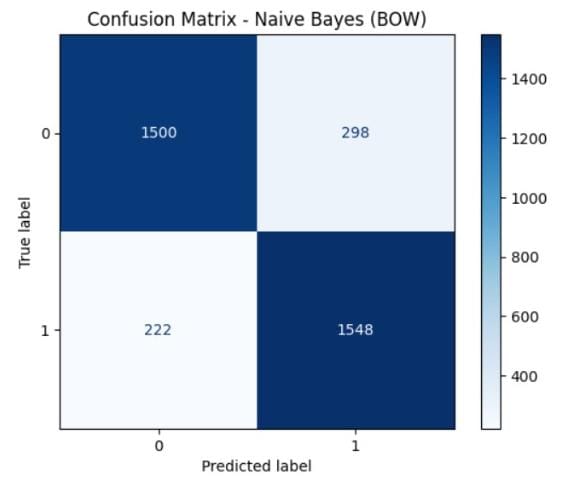
**-confusion matrix of gradient boosting**



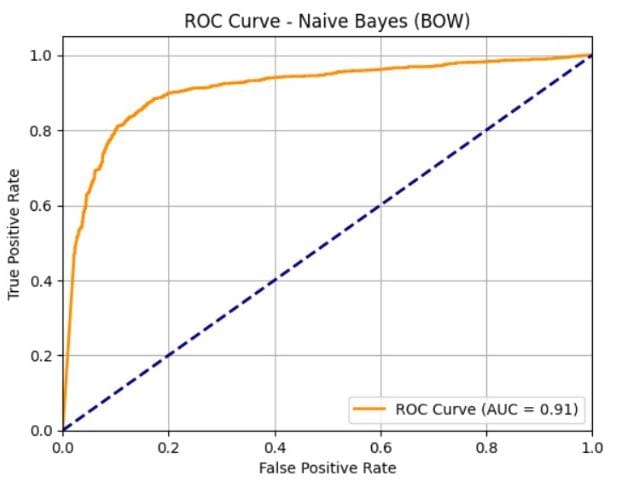
**-ROC of gradient boosting**



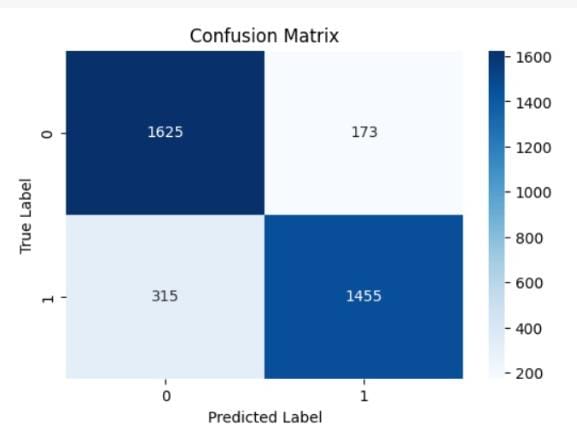
**-confusion matrix of Naive bayes**



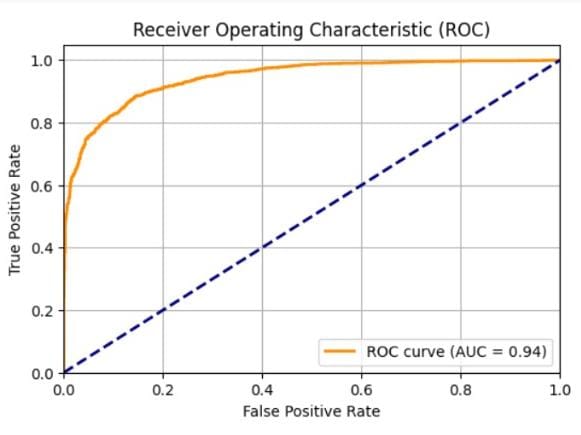
**-ROC of naive bayes**



**-confusion matrix of Dense**

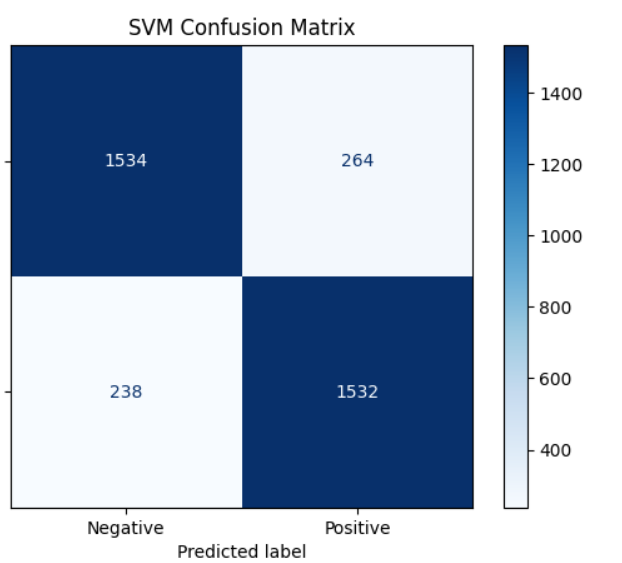


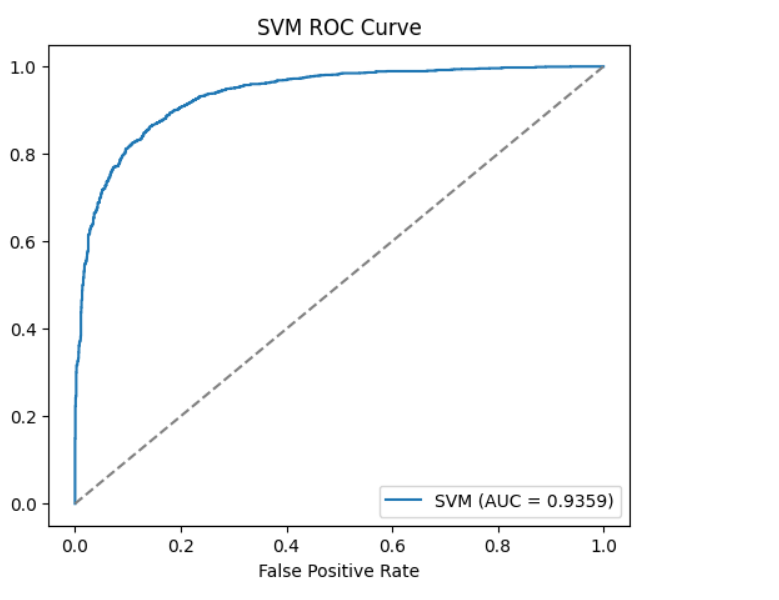
**-ROC of Dense**



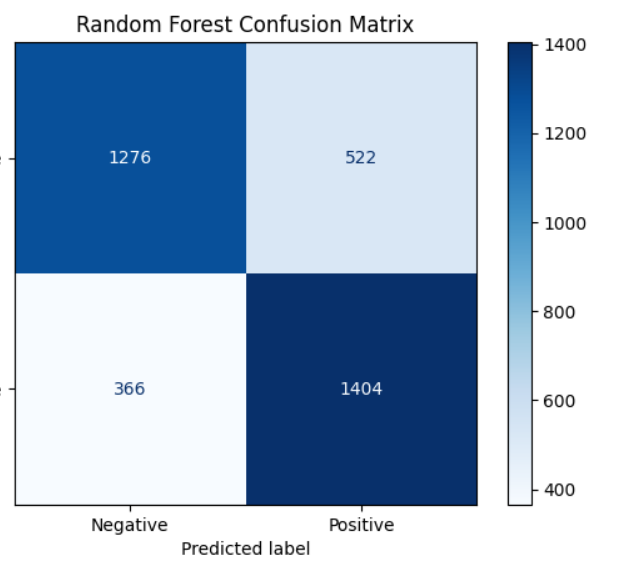
**2-Gehad Ahmed**

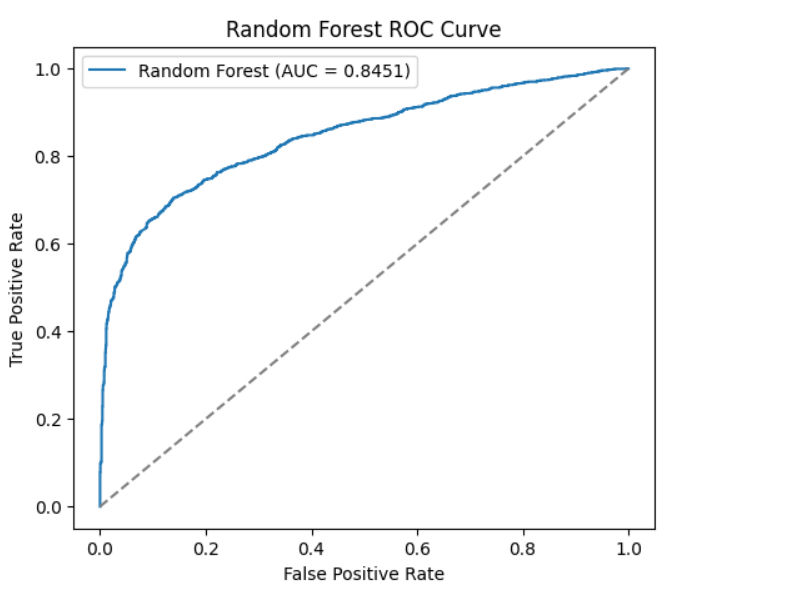
**1-svm**

****

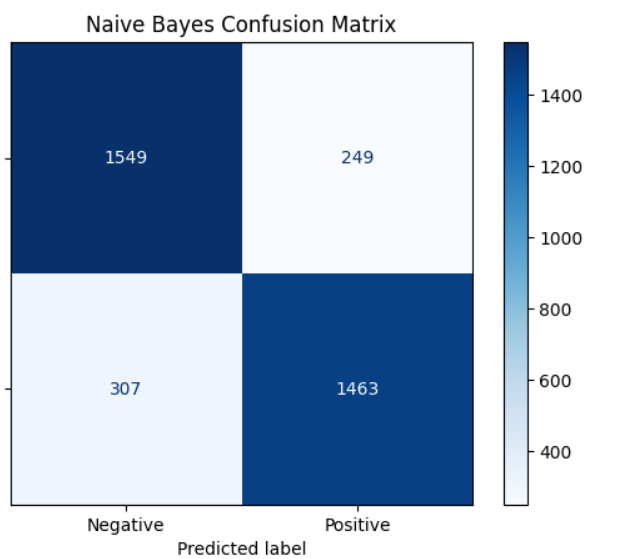
****

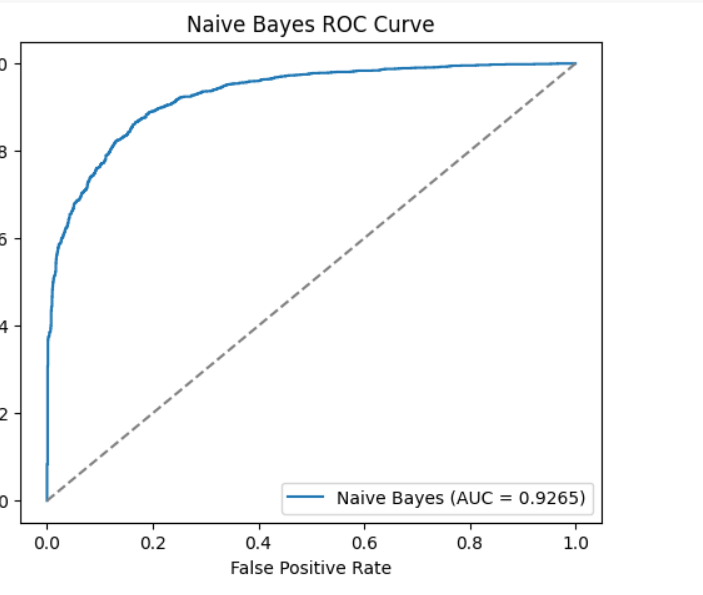
2-**Random Forest**



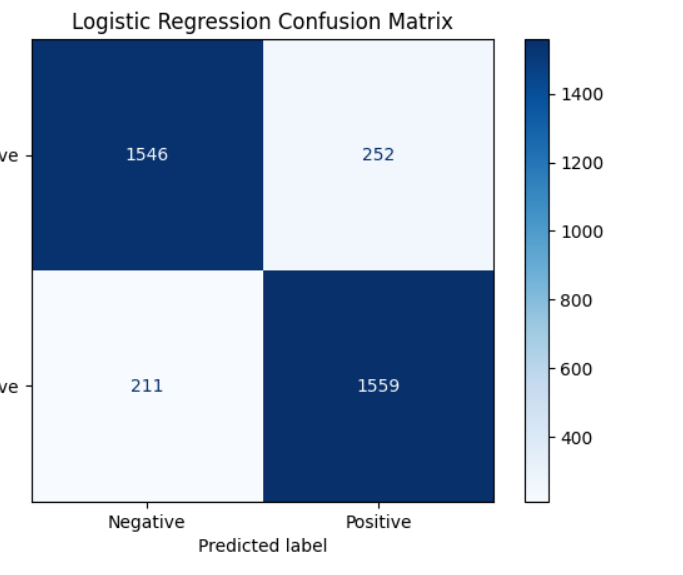


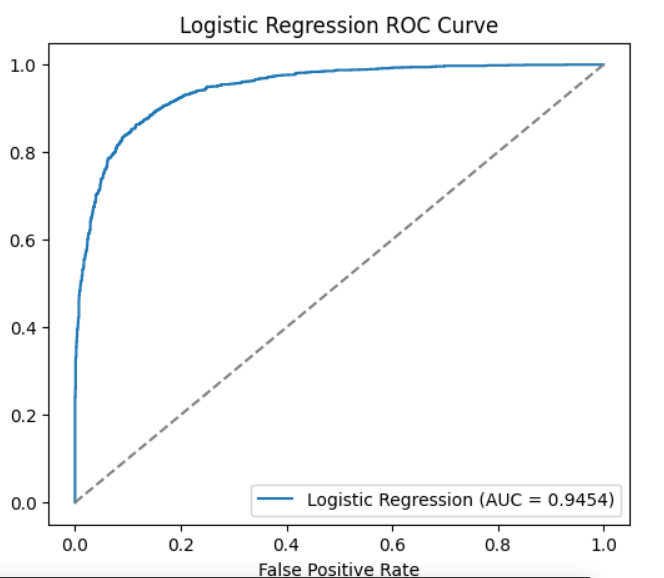
3-**Naive Bayes**





4**-Logistic Regression**





5- **XGBoost**

