جامعة أم القرى
UMM AL-QURA UNIVERSITY

Collage of computing

Data Analysis 2

COURSE PRESENTER

(DR. Omaima A. Fallatah)

DURATION:

25 Oct 2024

Task 2

“Text Analysis:”

Student Name	ID
Reuf Bakr Al-sharif	443002428
Jana Hesham Al-Hless	443002347

TABLE OF CONTACT

INTRODUCTION

Text analysis is a core area in data science and AI, focusing on extracting insights from unstructured text data. With applications in customer feedback systems, recommendation engines, and automated decision-making, text analysis transforms raw text into structured data that can reveal patterns and trends. Key steps include preprocessing (like tokenization and lemmatization) and feature extraction, which enable machine learning models to classify or interpret text effectively.

In this project, we analyze a market basket dataset to explore customer purchasing behavior. Each transaction includes details like Bill Number, Item Name, Quantity, Date, Price, Customer ID, and Country. By using association rule mining and the Apriori algorithm, we identify items frequently bought together and uncover patterns in purchasing. This analysis aims to support business decisions on cross-selling, inventory planning, and targeted promotions. Through data cleaning, transformation, and rule mining, we gain valuable insights into product relationships and customer preferences.

DATASET

The data used in this analysis comes from the

Fake News Detection Dataset on Kaggle,

which provides labeled news articles for evaluating text-based classification models.

Overview of the Dataset:

The dataset contains transaction data from a retail store with the The dataset contains news articles categorized as "real" and "fake." Each article includes a set of attributes, such as the article's title, full text content, author's name, and publication date. These fields are used to analyze and determine if an article contains reliable or unreliable information, assisting in building a predictive model capable of detecting fake news based on the text content and associated details.

Dataset includes the following columns:

id: A unique identifier for each article.

title: The article's title.

author: The name of the article's author.

text: The full content of the article.

label: The classification of the article.

(0 for real articles, 1 for fake articles).

OBJECTIVES

In general, the primary objective of text analysis is to extract meaningful insights and patterns from textual data. This often involves classifying, clustering, or evaluating content to make informed decisions or predictions. For datasets involving news articles, the objective is typically centered around identifying reliable versus unreliable content, analyzing trends, or categorizing information based on certain criteria.

For this particular dataset of news articles categorized as ***"real"*** and ***"fake,"*** the objective is to develop a predictive model that can distinguish between trustworthy and untrustworthy news based on specific features like the title, full text, author, and publication date. By analyzing these attributes, the aim is to enhance the detection of fake news, providing valuable insights for content verification and helping reduce the spread of misinformation.

STEP 1 | DATA CLEANING AND PREPROCESSING

1.1 Remove Duplicates:
Duplicate rows were identified and removed to ensure each article in the dataset is unique, reducing redundancy and enhancing model accuracy.

1.2 Handle Missing Values:
Missing values were examined, especially in critical columns like "Title," "Text," and "Author." Articles with missing "Text" (which is essential for content analysis) were removed. For columns with non-critical missing values, records were retained as they wouldn't significantly affect the analysis.

1.3 Convert Data Types:
Key columns like "Date" were converted to a datetime format for easy manipulation, such as filtering articles by publication date if needed.

1.4 Text Standardization:
Text fields, including "Title" and "Text," were processed to remove leading and trailing spaces. This standardization ensures that slight variations in spacing don't affect the analysis or model training.

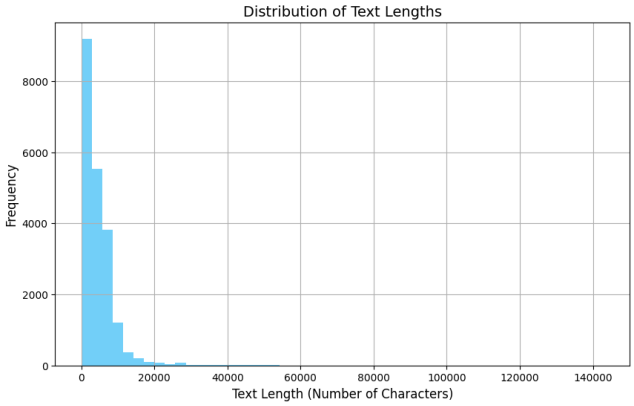
1.5 Text Preprocessing:
To prepare text for analysis, steps like lowercasing, removing punctuation, and eliminating stop words were performed. Lowercasing helps reduce case sensitivity issues, while removing punctuation and stop words minimizes irrelevant data, making it easier for the model to focus on meaningful words.

1.6 Feature Engineering:
Additional features were derived from the existing columns where applicable. For instance, the length of the "Text" and "Title" was calculated to see if length might correlate with article reliability. This helps create informative variables that the model can use for better predictions.

1.7 Outlier Detection and Handling:
Extreme values in quantitative columns (e.g., unusually long or short articles) were identified. While not removed entirely, outliers were flagged for further analysis to determine if they could impact model results.

STEP 2 | EXPLORATORY DATA ANALYSIS

2.1 Text Length Distribution:



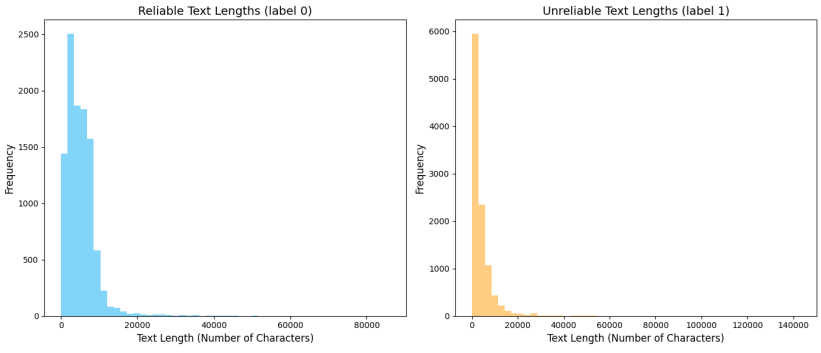
This plot illustrates the frequency distribution of the number of characters in the articles, showing that most articles have relatively short text lengths. The graph helps us identify outliers and understand the overall variability of the text data, which can be important for preprocessing steps like trimming overly long articles or filling in missing data for very short articles.

This distribution highlights that the majority of articles have fewer than 20,000 characters, with very few articles exceeding that length. This observation can guide feature engineering decisions or text length normalization, which may enhance the performance of machine learning models that rely on text input.

2.2 Distribution of Text Lengths by Category:

~~**Reliable Texts (label 0):**~~ Most reliable news articles have shorter texts, with a large concentration of articles containing fewer than 10,000 characters. Longer text lengths are rare, as indicated by the few outliers in the distribution.

~~**Unreliable Texts (label 1):**~~ Unreliable articles display a similar pattern, with most articles being short.However, the number of unreliable articles is higher compared to reliable ones in this shorter text range.

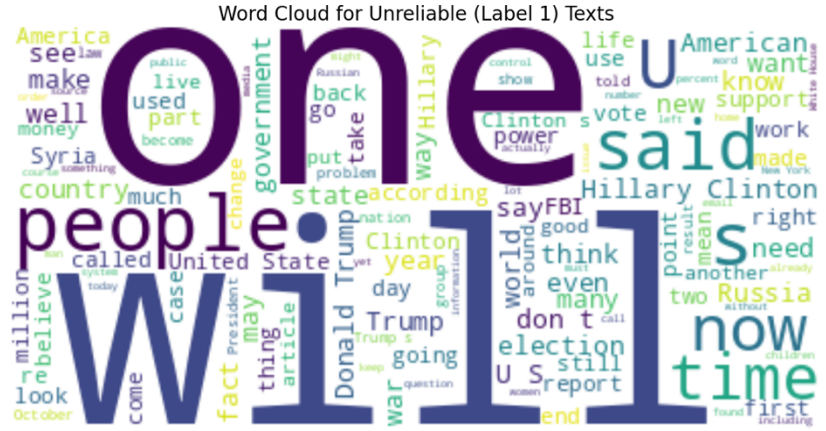


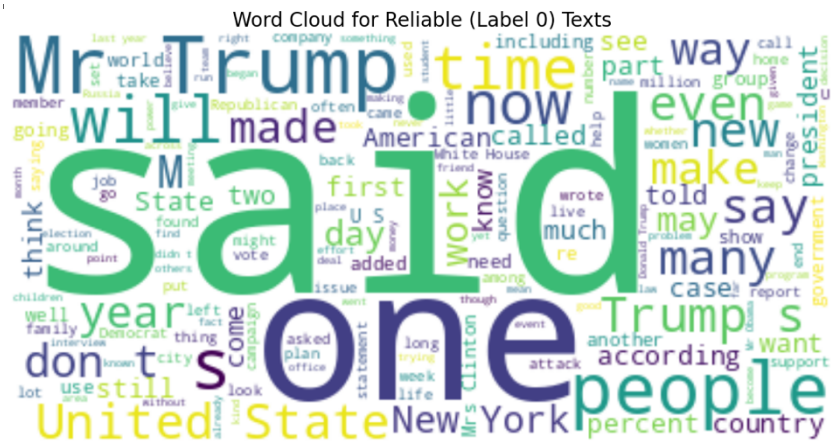
STEP 2 | (cont)..

2.2 Comparison of Common Words in Reliable and Unreliable Text:

Word Cloud for Reliable Texts (Label 0): The most common words in articles classified as reliable include words like "said," "Trump," "one," "people," and "will." This suggests that the content in these articles may frequently discuss political figures and events, particularly involving the former U.S. president, Donald Trump, and general statements or quotations.

Word Cloud for Unreliable Texts (Label 1): In unreliable articles, words like "one," "people," "will," and "said" appear prominently. There are also mentions of "Hillary Clinton" and other political figures or events. The word frequencies in unreliable articles suggest a similar theme but with possibly exaggerated or misleading information.





STEP 3 | Comparison of Sentiment Analysis:

In this analysis, we used two sentiment analysis techniques, ***TextBlob*** *and* ***VADER****,* to classify the news articles as **positive**, **negative**, or **neutral**. Both methods use different approaches, and we compared the results to understand their strengths and differences.

3.1 TextBlob vs VADER:

TextBlob, which uses a rule-based approach, provided a relatively higher count of positive classifications across the dataset, indicating that it might capture a broader scope of positive sentiments even in slightly neutral or less emotionally charged content. Its results leaned towards positive and neutral classifications.

	text	sentiment
0	house dem aide: we didn’t even see comey’s let...	positive
1	ever get the feeling your life circles the rou...	positive
2	why the truth might get you fired october 29, ...	positive
3	videos 15 civilians killed in single us airstr...	positive
4	print \nan iranian woman has been sentenced to...	negative

On the other hand, **VADER** (which is fine-tuned for social media and news sentiments) produced more balanced results with higher precision in distinguishing between positive, neutral, and negative sentiments. VADER's classification appeared stricter, identifying a significant number of articles as negative, especially those with strong negative connotations or words that carry negative emotional weight.

	text	sentiment
0	house dem aide: we didn’t even see comey’s let...	positive
1	ever get the feeling your life circles the rou...	negative
2	why the truth might get you fired october 29, ...	positive
3	videos 15 civilians killed in single us airstr...	negative
4	print \nan iranian woman has been sentenced to...	negative

STEP 3 | (cont)..

3.2 Converting Product Columns to Binary Values:

- Word Frequency Analysis Without Cleaning:

Using the Counter object, we calculated the frequency of words within the "no_sw" text column, which contained text without stopwords. The results showed high frequencies for words like "mr." and "said," but also included non-alphanumeric symbols like "—" and "”," which were appearing frequently.

Text Cleaning:

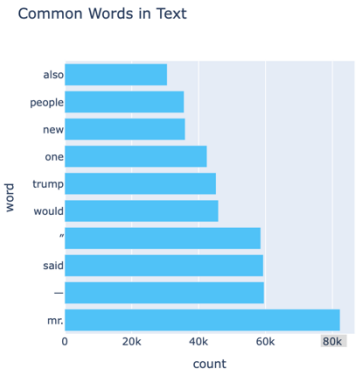
A function was applied to remove special characters from the text using the re library. This function eliminated punctuation and special symbols such as periods, commas, and dashes, aiming to improve the quality of text for analysis.

Word Frequency Analysis After Cleaning:

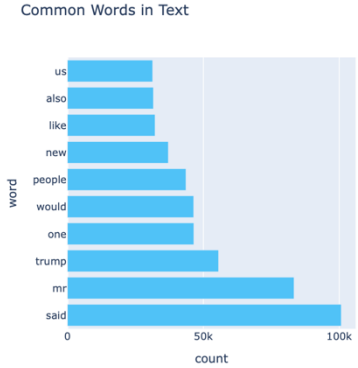
After cleaning the text, word frequencies were recalculated. Results showed a shift in the most frequent words, as special symbols were no longer inflating certain word counts, allowing for a clearer picture of the important words in the text.

Results:

~~Before Cleaning:~~ The analysis showed a high frequency of certain symbols like "—" and "”," which were unrelated to actual word content.



~~After Cleaning:~~ The focus shifted toward meaningful words such as "said," "Trump," and "people," providing a more accurate representation of the text content.



STEP 3 | (cont)..

3.3 Removing Frequent Words and Analyzing Removed Words in Text Data:

In this section, we aimed to enhance text data quality by removing frequently occurring, low-value words. This process helps in retaining only the more relevant terms that contribute meaningfully to analysis.

Steps:

1 - Identifying High-Frequency Words:

We calculated the frequency of each word in the dataset and identified the top 10 most common words, such as "like," "us," and "one," which added little analytical value.

2 - Removing Frequent Words:

Using a custom function, we removed these common words from each text entry to focus on more informative words. The processed text is stored in a new column, wo_stopfreq.

3 - Analyzing Removed Words:

We tracked the words removed from each entry to ensure only filler words were omitted, allowing a sharper focus on meaningful content.

Results and Insights:

This refinement removed repetitive terms, reducing noise and improving content clarity, which will aid in more accurate model training and analysis.

STEP 4 | Text Processing and High-Frequency Word Removal

In this text processing step, the primary objective was to remove high-frequency, non-informative words from the dataset. This technique improves text quality by reducing redundant terms, which could otherwise obscure the main topics and insights within the text.

4.1 Process Overview:

~~Identify Frequent Words:~~ Using a frequency analysis, the ten most common words across the dataset were identified.

~~Remove High-Frequency Words:~~ These frequent words, often referred to as "stop words" or "filler words," were removed from the text. The purpose of this step is to retain only the content-rich and informative terms in the dataset.

~~Text Refinement:~~ The resulting text is refined and becomes more readable, aiding subsequent analytical tasks such as sentiment analysis, topic modeling, and classification.

Benefits: Removing frequent words simplifies the data, allowing more meaningful patterns to emerge in subsequent analyses. This filtered version better highlights unique themes and sentiments present in the text, enhancing the effectiveness of model training.

STEP 4 | (cont)..

4.2 Lemmatization of Text Data:

this step, **Lemmatization** was applied to the text, a process that converts words to their base or root form. This helps reduce the number of distinct words with similar meanings, enhancing the quality of text data for analytical applications like sentiment analysis and classification.

Analysis Steps:

1 - Load WordNet: The WordNet library, containing English word dictionaries, was used to perform lemmatization.

2 - Initialize Lemmatizer: The WordNetLemmatizer tool was set up to convert words to their root forms.

3 - Apply Lemmatization: Lemmatization was applied to each word in the text stored in the 'wo_stopfreq' column, and results were stored in a new column, 'wo_stopfreq_lem'.

Benefit:

This process helps reduce redundancy and standardize terms, improving the effectiveness of further analysis.

highly likely to purchase multiple herb markers in one transaction.

	text	sentiment	no_sw	wo_stopfreq	wo_stopfreq_lem
0	house dem aide: we didn't even see comey's let...	positive	house dem aide: didn't even see comey's letter...	house dem aide: didn't even see comey's letter...	house dem aide: didn't even see comey's letter...
1	ever get the feeling your life circles the rou...	negative	ever get feeling life circles roundabout rathe...	ever get feeling life circles roundabout rathe...	ever get feeling life circle roundabout rather...
2	why the truth might get you fired october 29, ...	positive	truth might get fired october 29, 2016 tension...	truth might get fired october 29, 2016 tension...	truth might get fired october 29, 2016 tension...
3	videos 15 civilians killed in single us airstr...	negative	videos 15 civilians killed single us airstrike...	videos 15 civilians killed single airstrike id...	video 15 civilian killed single airstrike iden...
4	print \nan iranian woman has been sentenced to...	negative	print iranian woman sentenced six years prison...	print iranian woman sentenced six years prison...	print iranian woman sentenced six year prison ...

STEP 4 | (cont)..

4.3 Data Preparation for Sentiment Classificatione:

In this step, the dataset was cleaned and prepared for training a sentiment classification model. Columns like text, no_sw, and wo_stopfreq were dropped to streamline the data. Each review was assigned a binary label under the sentiment column, where "negative" sentiment was represented as 0 and "positive" as 1. This binary classification setup prepares the dataset for further training and testing processes in sentiment analysis.

4.4 Tokenization of Reviews:

This step involves breaking down each review into individual words, a process known as tokenization. By applying a function to split each review into a list of words, the text is transformed from sentences into discrete tokens, facilitating further processing and analysis. Tokenization is essential in text processing as it allows each word to be analyzed independently, making it possible to perform tasks like sentiment analysis or feature extraction on individual tokens.

he reviews have been tokenized, which means each review is represented as a list of words. Below are some examples of tokenized reviews:

Review 1: [house, dem, aide:, didn’t, even, see, comey’s...]

Review 2: [ever, get, feeling, life, circle, roundabout,...]

Review 3: [truth, might, get, fired, october, 29,, 2016,...]

Review 4: [video, 15, civilian, killed, single, airstrik...]

Review 5: [print, iranian, woman, sentenced, six, year,...]

STEP 5 | Naive Bayes Model Evaluation

5.1 Complement Naive Bayes Model Evaluation:

we conducted classification on the testing dataset. The **CNB** model achieved an accuracy of 77.65%.

```
ComplementNB model accuracy is 77.65%
--------------------------------------------------
Confusion Matrix:
      0     1
0  1645   583
1   554  2305
--------------------------------------------------
Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.74      0.74      2228
           1       0.80      0.81      0.80      2859

    accuracy                           0.78      5087
   macro avg       0.77      0.77      0.77      5087
weighted avg       0.78      0.78      0.78      5087
```

5.2 Multinomial Naive Bayes Model Evaluation:

The **MNB** model was also evaluated on the testing dataset, achieving an accuracy of **77.30%**.

```
MultinominalNB model accuracy is 77.75%
--------------------------------------------------
Confusion Matrix:
      0     1
0  1629   599
1   533  2326
--------------------------------------------------
Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.73      0.74      2228
           1       0.80      0.81      0.80      2859

    accuracy                           0.78      5087
   macro avg       0.77      0.77      0.77      5087
weighted avg       0.78      0.78      0.78      5087
```

5.3 Bernoulli Naive Bayes Model Evaluation:

The **BNB** model achieved an accuracy of **77.55%**

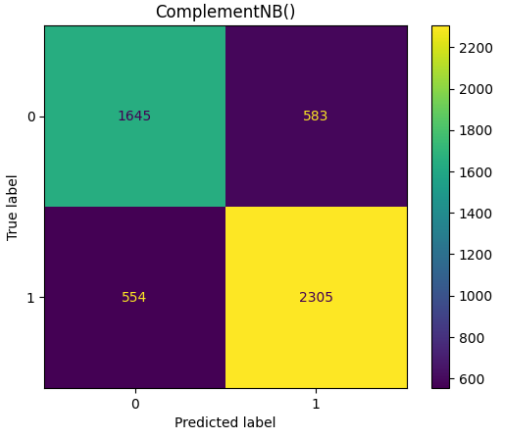
```
BernoulliNB model accuracy = 77.55%
--------------------------------------------------
Confusion Matrix:
      0     1
0  1610   618
1   524  2335
--------------------------------------------------
Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.72      0.74      2228
           1       0.79      0.82      0.80      2859

    accuracy                           0.78      5087
   macro avg       0.77      0.77      0.77      5087
weighted avg       0.77      0.78      0.77      5087
```

STEP 5 | (cont)..

5.4 Performance of Naive Bayes Models Using Confusion Matrices:

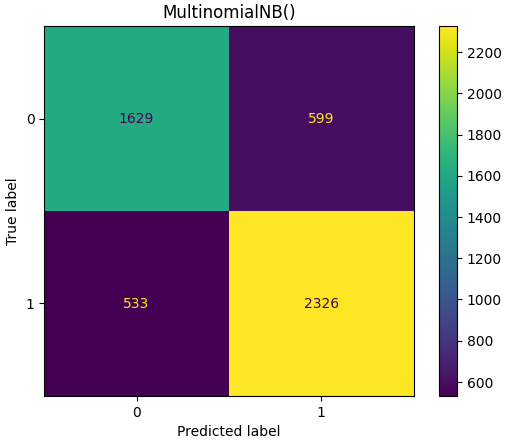
- ComplementNB:

- **TN**: 1645, **FP**: 583, **FN**: 554, **TP**: 2305.
- ComplementNB reduces false positives but has slightly more false negatives compared to MultinomialNB.



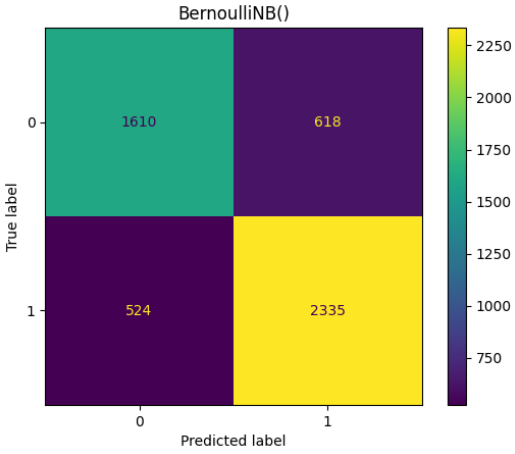
- MultinomialNB:

- **TN**: 1629, **FP**: 599, **FN**: 533, **TP**: 2326
- It shows a good balance by reducing both false positives and false negatives, making it the best performing overall.



- BernoulliNB:

- **TN**: 1610, **FP**: 618, **FN**: 524, **TP**: 2335
- This model has a high number of false positives, making it less reliable for the positive class.



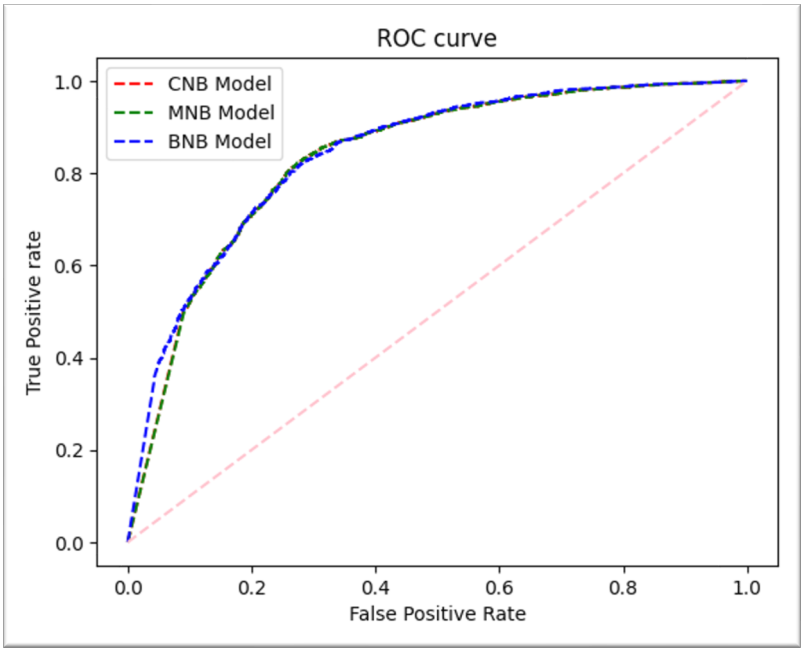
STEP 5 | (cont)..

5.5 ROC Curve Comparison:

The ROC curve (Receiver Operating Characteristic curve) shows the trade-off between the True Positive Rate (sensitivity) and the False Positive Rate for the three Naive Bayes models: Complement Naive Bayes (CNB), Multinomial Naive Bayes (MNB), and Bernoulli Naive Bayes (BNB).

- Complement Naive Bayes (CNB) (Red line): AUC = 0.83
- Multinomial Naive Bayes (MNB) (Green line): AUC = 0.83
- Bernoulli Naive Bayes (BNB) (Blue line): AUC = 0.84

The higher the AUC (Area Under the Curve), the better the model's ability to distinguish between classes. As seen, all three models perform similarly, with **Bernoulli Naive Bayes (BNB)** having a slightly higher AUC score (0.84), indicating a marginally better classification performance compared to CNB and MNB.



STEP 6 | Comparison of TF-IDF and Bag of Words (BoW) Accuracy

MultinomialNB:

This model achieved **78.69% accuracy** with **BoW**,a notable improvement over the **68.61% accuracy** it reached with **TF-IDF**. This suggests that BoW provided better text representation for MultinomialNB, making it more effective for classification on this dataset.

~~**Logistic Regression**:~~

With **TF-IDF n-grams**, Logistic Regression reached the highest accuracy among the models at **81.88%**. This result shows that TF-IDF, combined with n-grams, enhances Logistic Regression's ability to capture context within text, leading to more accurate predictions. BoW was not tested with this model, but it's likely that n-grams in TF-IDF contributed significantly to its success.

~~**BernoulliNB**:~~

The model achieved a **73.80% accuracy** using **TF-IDF**. Although not tested with BoW, BernoulliNB's performance with TF-IDF was solid, indicating it can perform well when representing text data as binary features through this method.

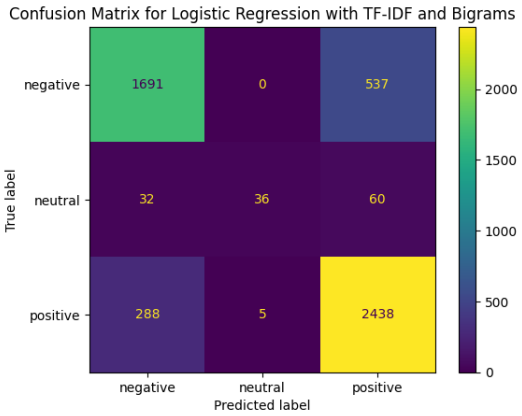
~~**ComplementNB**:~~

ComplementNB also performed well with **TF-IDF**, achieving **73.36% accuracy**. While it wasn't evaluated with BoW, TF-IDF's weighting scheme seemed to benefit ComplementNB, especially given its design to handle imbalanced datasets.

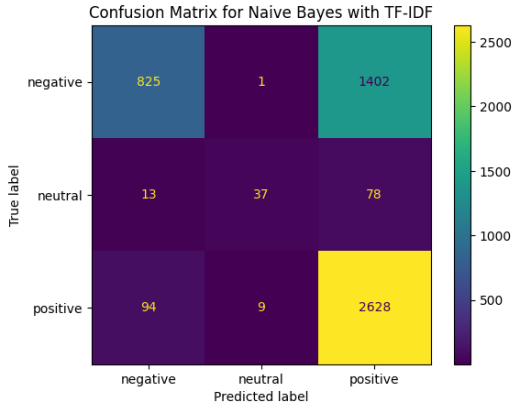
These findings suggest that **Bag of Words** is particularly effective for **MultinomialNB**, while **Logistic Regression** excels when paired with **TF-IDF and n-grams**. Each model benefits from specific text representation techniques, with Logistic Regression achieving the highest accuracy on this dataset using TF-IDF with n-grams.

STEP 6 | (cont)..

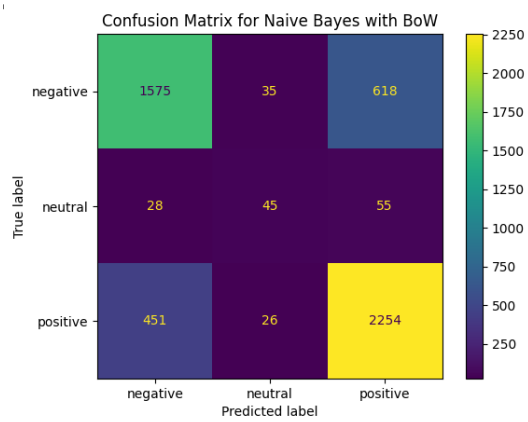
Logistic Regression with TF-IDF and Bigrams performed best overall, especially in classifying positive and negative sentiments, but struggled with neutral predictions.



Naive Bayes with TF-IDF had strong performance on positive classifications but performed poorly on negatives.



Naive Bayes with Bag of Words showed improved classification for negative instances but was less effective at predicting the positive class compared to its TF-IDF counterpart.



CONCLUSION

The process of building and evaluating text classification models began with data cleaning and preparation, where text data was preprocessed to remove noise and irrelevant information. Techniques such as tokenization, removing stop words, and applying **TF-IDF** and **Bag of Words (BoW)** were used to convert the text into a format suitable for machine learning models. After preprocessing, the data was split into training and testing sets to ensure fair evaluation.

Several models were tested, including **Logistic Regression with TF-IDF and Bigrams** and **Naive Bayes** models with both **TF-IDF** and **BoW** representations. The results indicated that **Logistic Regression** with **TF-IDF and Bigrams** provided the highest accuracy, particularly for positive and negative sentiments. However, the model struggled with neutral classifications. **Naive Bayes** performed well in classifying negative sentiments, especially when using **BoW**, but it had difficulty maintaining precision when classifying positive cases.

Confusion matrices were generated to visually assess the performance of each model, highlighting areas where models performed well and where they misclassified certain sentiment categories. Logistic Regression, while overall accurate, missed many neutral cases, while Naive Bayes showed strengths and weaknesses depending on the chosen text representation method.

The project concluded with a comparison of the models and their performance, with the findings emphasizing the importance of selecting the right text representation technique for each model. The results underscore that different models respond better to different approaches, such as TF-IDF or BoW, and that the use of n-grams can significantly improve a model's ability to capture context.