Evolving Arabic Sentiment Analysis:
Machine Learning-Driven Methodologies

The University of Jordan

King Abdullah II School for Information Technology

NATURAL LANGUAGE PROCESSING - 1905380

Done By:

Khaled Shrouf 0215106

Anas Moosa 0212567

Osama Khandaqgee 0213612

Ahmad Allhham 0215697

Contents

1.0 Introduction

Customer feedback in the form of text reviews is a rich source of insights but is often unstructured and complex. Analyzing this data manually is not feasible due to the volume, velocity, and variety of the data. Moreover, sentiments expressed in texts are influenced by cultural nuances and language-specific elements, especially in languages like Arabic, which poses additional processing challenges.

This project aims to automate the sentiment analysis of Arabic text reviews using advanced machine learning techniques. The goal is to categorize these reviews into positive, negative, or neutral sentiments, thereby enabling company owners and stakeholders to gauge customer satisfaction and identify areas for service improvement effectively.

The project will follow a structured approach, starting with finding the suitable dataset which fits to our aim. Next, we will extract relevant features using sophisticated techniques and apply machine learning models like Multinomial Naive Bayes (Arthur V, 2021), Random Forest (IBM, 2020), and Logistic Regression (Saini, 2024) to classify the sentiments of the reviews.

1.1 Need Analysis and Description

Customer feedback plays an important role in shaping the reputation and success of a company. Sentiment analysis is an excellent way to give insights into customer satisfaction and highlight areas that need improvement.

1.2 Project Constraints

- **Imbalanced Dataset**: The distribution of the sentiments (Positive, Negative, and Neutral) is uneven, which poses a challenge in making better models.

- **Language Specificity**: This project deals with text reviews in Arabic, which requires very specific preprocessing steps, such as handling right-to-left text, stemming, etc.
- **Data Quality**: The presence of null and duplicated values makes it important to clean the data by handling those values.
- **Evaluation Metrics**: Ensuring that the models are evaluated using meaningful metrics that align with the business's objectives.

1.3 System Environment

- **Operating System**: Windows
- **Programming Language**: Python 3
- **Libraries**:
    - Data Manipulation and Analysis: Pandas and Numpy. (Thai, 2019)
    - Visualization: Plotly, Seaborn, and Matplotlib (Gunay, 2023)
    - Text Processing: Nltk, Qalsadi, Demoji, and WordCloud (Otten, 2022)
    - Machine Learning: Scikit-Learn, and Imblearn (Wei, 2024), (Soni, 2020)

1.4 Project Software and Hardware Requirements

1.4.1 Software Requirements:

1. Python 3.x: The programming language used for the project.
2. Google Cloud – Jupyter Notebook: For code and output demonstration.

1.4.2 Hardware Requirements:

- Processor: A multi-core processor to handle the computational load of training machine learning models.
- RAM: At least 8 GB of RAM to efficiently manage data processing and model training tasks.

- Storage: At least 256 GB to accommodate the dataset and any intermediate files generated during processing (SSD preferred for faster read/write speeds).

2.0 Related Works

2.1 Toward Qualitative Evaluation of Embeddings for Arabic Sentiment Analysis

(Barhoumi, 2020)

This study was conducted to propose protocols for evaluating embeddings specifically for Arabic Sentiment Analysis (ASA), that is because the Arabic language is considered rich in terms of morphology.

The study uses different types of corpora, including polar and non-polar datasets. The researchers used the Large Arabic-Book Reviews (LABR) corpus, focusing on binary classification (positive/negative).

The researchers evaluated the embeddings using a neural architecture based on the Convolutional Neural Network (CNN), and the experiments provided excellent results with 91.9% accuracy.

2.2 Arabic Sentiment Analysis: A Systematic Literature Review (SLR)

(Mohsen A. &., 2020)

The paper aims to conduct a systematic review of existing literature on Arabic Sentiment Analysis (ASA) to support further research in the field, identify potential areas for future studies, and assist other researchers in finding relevant studies.

The data in this study consist of literature related to ASA, including research papers, articles, and other scholarly works.

The main algorithmic contribution of this paper is the development of a taxonomy for sentiment classification methods within the context of ASA. It categorizes different approaches used in sentiment analysis of Arabic text.

The findings of the review highlight limitations in existing approaches to ASA, particularly in preprocessing, feature generation, and sentiment classification methods. This suggests areas where future research can focus on improvement.

2.3 Arabic Sentiment Analysis of YouTube Comments (Musleh D. A., 2023)

The paper proposes an NLP-based model to classify Arabic YouTube comments as positive or negative, addressing the challenge of assessing video quality without watching it, because the dislike count feature was removed.

The researchers used a dataset consisting of 4212 labeled Arabic YouTube comments and used six classifiers: SVM, Naïve Bayes, Logistic Regression, KNN, Decision Tree, and Random Forest.

The researchers found that the Naïve Bayes algorithm provided the highest performance, whereas the Decision Tree algorithm provided the lowest performance.

2.4 Towards Arabic Multimodal Dataset for Sentiment Analysis (Haouhat A. B., 2023)

The paper focuses on advancing Arabic Deep Learning-based Multimodal Sentiment Analysis (MSA) by addressing the lack of standard datasets and validating the approach with state-of-the-art models.

The researchers created an Arabic Multimodal dataset using transformers, feature extraction tools, and word alignment techniques. The researchers designed a pipeline to create the dataset. Even though the size of the dataset was small, the experiments indicated that Arabic multimodal sentiment analysis is very promising.

2.5 summary of the related works

	Reference	Task Description	Techniques		Performance			
Work	---	---	preprocessing	F.E	Accuracy	F1	Recall	Precision
Arabic Sentiment Analysis: A Systematic Literature Review (SLR)	(Mohsen & Ali, 2020)	Reviews various techniques and datasets used in Arabic sentiment analysis from 191 papers (2013-2018).	Text cleaning, normalization, tokenization, stop words removal, stemming	Bag of Words, TF-IDF, Word Embeddings .	some models report up to 90% accuracy.	Not reported	Not reported	Not reported
Toward Qualitative Evaluation of Embeddings for Arabic Sentiment Analysis	(Barhoumi , et al., 2020)	conducted to propose protocols for evaluating embeddings specifically for Arabic Sentiment Analysis (ASA), that is because the Arabic language is considered rich in terms of morphology.	Not reported	Not reported	91.9%	Not reported	Positive: 97.76% Negative : 58.46%	Positive: 93.06% Negative : 82.01%
Arabic Sentiment Analysis of YouTube Comments: NLP-Based Machine Learning Approaches for Content Evaluation	(Musleh, et al., 2023)	Sentiment analysis of YouTube comments	Text normalization, tokenization	N-grams, TF-IDF	94.62%	94.62%	94.64%	94.64%
Towards Arabic Multimodal Dataset for Sentiment Analysis	(Haouhat, Bellaouar, Nehar, & Cherroun, 2023)	Multimodal sentiment analysis of Arabic videos	Speech, text extraction, segmentation	Linguistic, audio, visual features	Not reported	Not reported	Not reported	Not reported

3.0 Proposed Methodology

3.1 Pipeline of the Proposed Methodology

The pipeline consists of several steps: (show figure 1)

1. Data Acquisition: loading (Company_reviews.csv, 2021) dataset from Kaggle.
2. Data Preprocessing: Cleaning and normalizing the text data.
3. Feature Extraction: Transforming text into numerical features suitable for model training.
4. Feature Classification: Applying machine learning algorithms to classify the sentiments of the reviews into categories like positive, negative, and neutral.

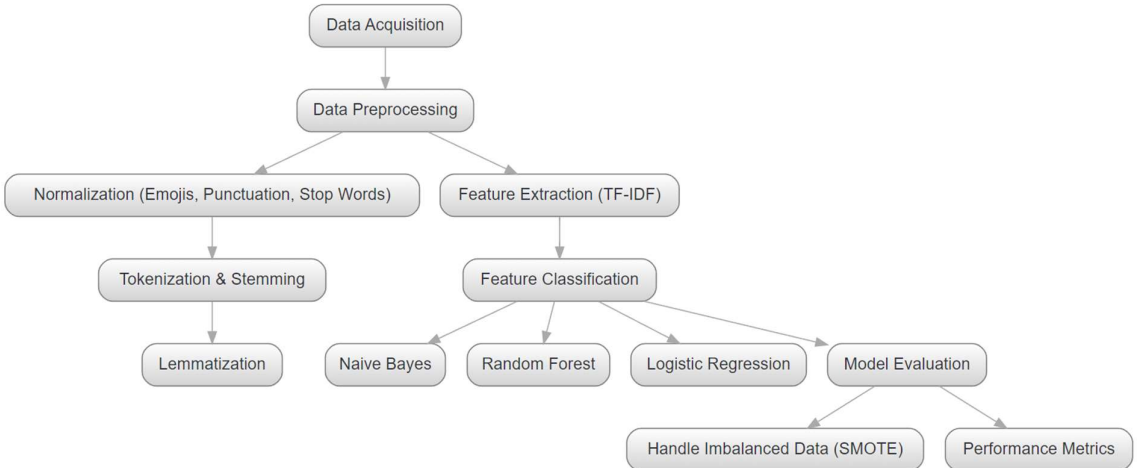

Figure 1 : Pipeline

3.2 Technical and Implementation Description

The project began with loading the dataset from a CSV file using Pandas. Preprocessing steps included handling missing values, removing duplicates, and normalizing text data by removing emojis, punctuation, and stop words. Stemming and lemmatization were applied to unify different forms of the same word. Features were extracted using TF-IDF vectorization (Jain, 2024), transforming the text data into numerical features suitable for machine learning models. The models were trained and evaluated on both the imbalanced and balanced datasets, with performance metrics and visualizations used to interpret the results.

3.3 Dataset Description

The dataset used for this project was sourced from: Kaggle (Company_reviews.csv, 2021)

It consists of reviews of various companies. It contains several key features: review_description, which holds the text of the review; rating, which represents the sentiment of the review categorized as 1 (positive), 0 (neutral), or -1 (negative); and company, which specifies the company being reviewed.

The dataset includes 40046 review samples and three features. The distribution of ratings in the dataset shows 23921 positive reviews, 1925 neutral reviews, and 14200 negative reviews. This distribution highlights the imbalance present in the dataset, which necessitated the application of techniques such as SMOTE (Maklin, 2022) to create a balanced dataset for training the machine learning models.

3.3.1 Sentiment Ratings

The pie chart in figure 2 shows the sentiment distribution within the dataset. The majority of the reviews, 59.7%, are classified as positive, demonstrating a generally favorable opinion towards the services or products offered by the companies. Negative reviews constitute 35.5% of the dataset, reflecting significant areas of dissatisfaction that could be pivotal for business improvement strategies. Neutral sentiments make up the smallest portion, at 4.8%, indicating that most reviewers tend to express a clear positive or negative stance rather than a neutral one.

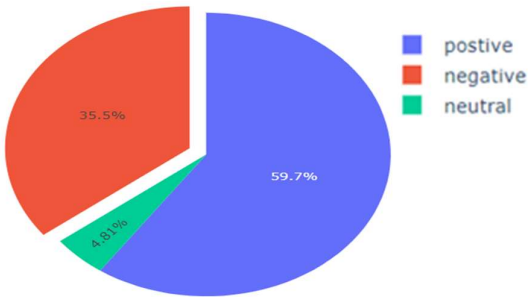

Figure2 : Pie chart for rating distribution

3.3.2 Company Distribution

The dataset consists of reviews from various companies, as illustrated in the donut chart shown in figure 3. The majority of the reviews, 80.1%, pertain to a single dominant company, indicating a significant concentration of feedback from this entity. This is followed by smaller segments representing other companies such as Swvl (11.7%) and Venus (5.22%), with the remaining companies like Raya, Telecom Egypt, Hilton, Domty, Nestle, Elsewedy, Capiter, TMG, and Ezz Steel contributing less than 1% each. This distribution highlights the varied nature of the data, encompassing a wide range of sectors and providing a comprehensive overview of different customer experiences and perceptions.

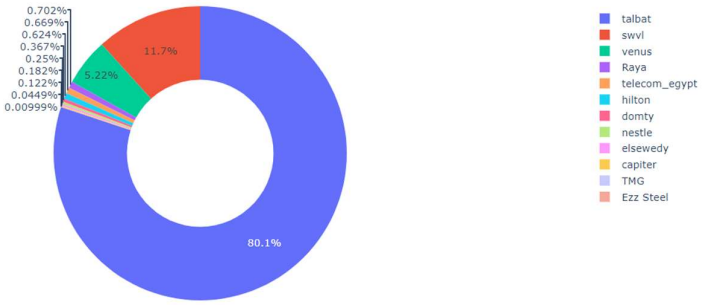

Figure3 :Donut chart for company distribution

3.3.3 Word Cloud Analysis

To better understand the most common words used in the Arabic text data, we generated a word cloud, as shown in Figure 4. A word cloud (Rai, 2020) is a visual representation of text data where the size of each word indicates its frequency or importance within the dataset. In this word cloud, larger words appear more frequently in the text corpus.

Figure4 : Word cloud

From the word cloud, we can observe that certain words such as "طبق", "طلب", "طعم" and "خدمة" are prominently featured.

This visualization provides a quick and intuitive understanding of the key topics within the dataset. It helps to identify the most relevant words for sentiment analysis, allowing us to focus on words that might carry strong positive or negative connotations. For example, word such as "حلو" could be indicative of positive sentiments, whereas words like "رفض" and "عطل" might reflect negative experiences.

3.4 Data Preprocessing

3.4.1 .Drop the null values

Removing null values ensures that the model doesn't process empty inputs, which could lead to errors during analysis or skew the model's understanding of the data.

3.4.2 .Remove duplicate entries

Duplications can introduce bias and affect the model's generalization capabilities. Removing duplicates ensures each input is unique, which helps in fair training and testing of the model.

3.4.3 .Remove emojis

Emojis in text data can be problematic because they are often not handled by typical text processing methods, which might misinterpret them or lead to unnecessary noise in the data.

'جائع جدا😋😋😋😋😋😋😋😋😋' after remove emojis 'جائع جدا'

3.4.4 .Remove punctuations

Punctuation marks can distort the model's understanding of words if not removed, as they might lead to different tokens being created for the same word (e.g., "مرحبا" vs. "!مرحبا").

3.4.5 .Remove stop words

Stop words are commonly used words (such as "ال", "لك","أصبح" etc.) that usually have little lexical content. In sentiment analysis, these can distract from more meaningful words in text.

'أصبح غالي جداً' after remove stop words 'غالي جداً'

3.4.6 .Tokenization (Gökdemir, 2023)

Breaking down text into individual words or tokens is crucial for most natural language processing tasks.

3.4.7 .Normalization (Stemming) (Nabil, 2023)

Stemming reduces words to their base or root form, which helps in reducing the complexity of the language model by mapping related words to the same stem.

3.4.8 .Lemmatization (Nabil, 2023)

Lemmatization is similar to stemming but brings context to the words. It links words with similar meanings to one word, enhancing the model's performance by aggregating common variants of a word.

These preprocessing steps are critical for ensuring that the dataset is clean, uniform, and optimized for the machine learning algorithms to perform effectively.

3.5 Features Extraction

Feature extraction was performed using TF-IDF vectorization (Jain, 2024), a powerful technique that transforms textual data into numerical features. TF-IDF (Term Frequency-Inverse Document Frequency) captures the importance of words in a document relative to the entire dataset. This method not only considers the frequency of a word in a single document but also how common or rare the word is across all documents.

By applying TF-IDF vectorization, the textual data was converted into a matrix of numerical features, where each row represented a review, and each column represented a unique word from the entire dataset. This transformation enabled the use of machine learning algorithms on the textual data, allowing the models to learn patterns and relationships within the text effectively.

3.6 Features Classification

In this section, we detail the classification strategies used to analyze sentiment based on the extracted and preprocessed features from our dataset. We employed various machine learning models to ensure robustness and accuracy in our predictions, focusing on Multinomial Naive Bayes, Random Forest, and Logistic Regression. Each of these models has distinct advantages in text classification scenarios.

3.6.1 Multinomial Naive Bayes: (Arthur V, 2021)

This model is particularly suited for classifications with discrete features, such as word counts in text classification. It uses the probabilities of each feature to make predictions, making it highly efficient for large datasets.

3.6.2 Random Forest Classifier: (IBM, 2020)

As an ensemble method, Random Forest combines multiple decision trees to improve the model's accuracy and stability. This method is less likely to overfit compared to a single decision tree and is very effective for handling high-dimensional data.

3.6.3 Logistic Regression: (Saini, 2024)

Logistic Regression is typically used for binary classification but can also handle multi-class problems using strategies like one-vs-rest. It is valued for its simplicity and efficiency, making it a popular choice in text classification.

3.6.4 Hyperparameter Tuning (Samancı, 2024)

To enhance the performance of our Logistic Regression model, we utilized GridSearchCV. This method systematically explores numerous combinations of parameters, performing cross-validation to determine the optimal settings that maximize model performance.

4.0 Experimental Results and Analysis

4.1 Performance Measures (Mudadla, 2023)

The performance of the models was evaluated using several key metrics:

Accuracy: The proportion of correctly predicted instances out of the total instances, providing an overall measure of model performance.

Precision: The proportion of correctly predicted positive observations to the total predicted positives, indicating the model's ability to avoid false positives.

Recall: The proportion of correctly predicted positive observations to all observations in the actual class, reflecting the model's ability to identify all relevant instances.

F1-Score: The weighted average of precision and recall, providing a balance between the two metrics and offering a single measure of model performance.

Confusion Matrix: A visual representation of the performance showing the correct and incorrect predictions for each class.

ROC Curve: The Receiver Operating Characteristic curve visualizes the performance of the classification model across all classification thresholds, highlighting the trade-off between true positive rate and false positive rate.

4.2 Experimental Results

The experimental results for the models before and after handling data imbalance are detailed as follows:

4.2.1 Before Handling Imbalanced Data:

- Naive Bayes: The model achieved an accuracy of [0.81], with detailed metrics showing precision, recall, and F1-score (Chavan, 2021) for each class shown in figure 5.

Accuracy: 0.81

	precision	recall	f1-score	support
Positives	0.79	0.76	0.78	2836
Neutral	0.20	0.00	0.00	404
Negatives	0.82	0.91	0.86	4561
accuracy			0.81	7801
macro avg	0.60	0.56	0.55	7801
weighted avg	0.78	0.81	0.79	7801

Figure 5 :Performance metrics for Naive Bayes

- Random Forest: The model achieved an accuracy of [0.81], with detailed metrics showing precision, recall, and F1-score for each class shown in figure 6 .

Accuracy of RandomForestClassifier: 0.81

	precision	recall	f1-score	support
Positives	0.78	0.79	0.79	2836
Neutral	0.22	0.01	0.03	404
Negatives	0.83	0.90	0.87	4561
accuracy			0.81	7801
macro avg	0.61	0.57	0.56	7801
weighted avg	0.78	0.81	0.79	7801

Figure 6 :Performance metrics for Random Forest

- Logistic Regression: The model achieved an accuracy of [0.82], with detailed metrics showing precision, recall, and F1-score for each class shown in figure 7 .

Accuracy: 0.82

	precision	recall	f1-score	support
Positives	0.82	0.77	0.79	2836
Neutral	0.15	0.00	0.01	404
Negatives	0.82	0.92	0.87	4561
accuracy			0.82	7801
macro avg	0.60	0.57	0.56	7801
weighted avg	0.79	0.82	0.80	7801

Figure 7 :Performance metrics for Logistic Regression

The classification report reveals that while the overall accuracy of the Multinomial Naive Bayes model is 82%, the performance metrics for the 'Neutral' class in the three models are significantly lower compared to the 'Positive' and 'Negative' classes. Specifically, the precision for the 'Neutral' class for example in the Multinomial Naive Bayes is 0.15, the recall is 0.00, and the F1-score is 0.01. This indicates that the model struggles to correctly identify and classify 'Neutral' reviews.

The low recall for the 'Neutral' class means that very few of the actual 'Neutral' reviews are being correctly identified by the model. Similarly, the low precision implies that among the reviews predicted as 'Neutral', a large proportion are incorrectly classified. Consequently, the

F1-score, which is the harmonic mean of precision and recall, is also very low for the 'Neutral' class, reflecting poor overall performance in identifying these reviews.

This performance disparity is primarily due to the imbalance in the dataset, where 'Neutral' reviews are significantly underrepresented compared to 'Positive' and 'Negative' reviews. Such imbalances can lead to biased models that favor the majority classes, resulting in poor performance on minority classes.

To address this issue, techniques such as oversampling, undersampling, and Synthetic Minority Over-sampling Technique (SMOTE) (Maklin, 2022) can be employed. SMOTE, for instance, generates synthetic samples for the minority class, thereby balancing the class distribution in the training data. By doing so, the model is exposed to a more balanced dataset during training, which helps improve its ability to correctly classify reviews from all classes, including the underrepresented 'Neutral' class. Implementing these techniques is crucial for enhancing the robustness and fairness of the model, ensuring that it performs well across all classes and provides more reliable sentiment analysis.

4.2.2 After Handling Imbalanced Data (using SMOTE):

- Naive Bayes: The application of SMOTE improved the model's accuracy to [0.74], with balanced precision, recall, and F1-score across all classes, as shown in figure 8 .

Naive Bayes Classification Report:

	precision	recall	f1-score	support
Positives	0.755628	0.799228	0.776816	4662.000000
Neutral	0.683743	0.728719	0.705515	4652.000000
Negatives	0.794718	0.695141	0.741602	4589.000000
accuracy	0.741279	0.741279	0.741279	0.741279
macro avg	0.744697	0.741029	0.741311	13903.000000
weighted avg	0.744478	0.741279	0.741335	13903.000000

Figure 8 :Performance metrics for Naive Bayes after SMOTE

- Random Forest: The application of SMOTE significantly enhanced the model's accuracy to [0.877], with improved precision, recall, and F1-score across all classes, as shown in figure 9 .

Random Forest Classification Report:

	precision	recall	f1-score	support
Positives	0.863314	0.861647	0.862480	4662.000000
Neutral	0.905837	0.957438	0.930923	4652.000000
Negatives	0.864297	0.816082	0.839498	4589.000000
accuracy	0.878659	0.878659	0.878659	0.878659
macro avg	0.877816	0.878389	0.877634	13903.000000
weighted avg	0.877867	0.878659	0.877795	13903.000000

Figure 9 :Performance metrics for Random Forest after SMOTE

- Logistic Regression: The application of SMOTE increased the model's accuracy to [0.756], with balanced precision, recall, and F1-score across all classes, as shown in figure 9.

Logistic Regression Classification Report:

	precision	recall	f1-score	support
Positives	0.826560	0.732947	0.776944	4662.000000
Neutral	0.658055	0.859630	0.745456	4652.000000
Negatives	0.838299	0.674439	0.747494	4589.000000
accuracy	0.756024	0.756024	0.756024	0.756024
macro avg	0.774305	0.755672	0.756632	13903.000000
weighted avg	0.774052	0.756024	0.756688	13903.000000

Figure 10 :Performance metrics for Logistic Regression after SMOTE

4.3 Visualization of Results:

- Confusion Matrices: (Malviya, 2023) The confusion matrices for each model provided insights into the distribution of correct and incorrect predictions, highlighting areas where the models performed well and where they struggled. Show figures (11-13)

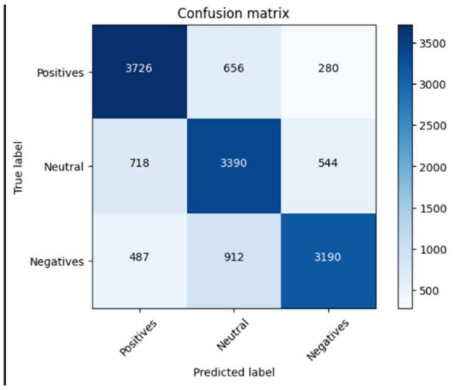

Figure 13 :Naive Bayes Confusion Matrix

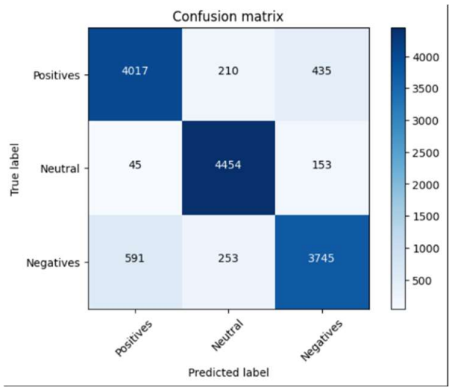

Figure 11 :Random Forest Confusion Matrix

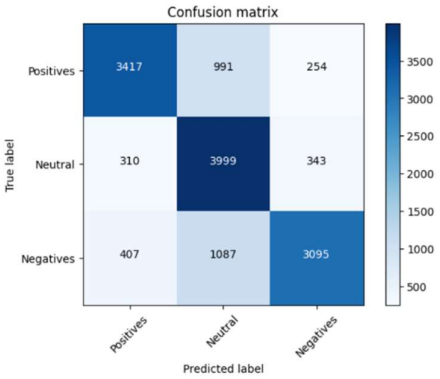

Figure 12 :Logistic Regression Confusion Matrix

Based on the confusion matrices, accuracy and report, we found that the Random Forest model outperformed the other models, achieving higher accuracy and better balance across the classes. The Random Forest model showed the highest number of correct predictions for each class, indicating its robustness (Dubrov, 2023) and effectiveness in handling the sentiment analysis task.

- Learning Curves: (Olamendy, 2024) The learning curves for the Random Forest model showed how training and validation accuracy evolved as the number of training samples increased, indicating the model's learning capacity and potential overfitting issues. show figure 14

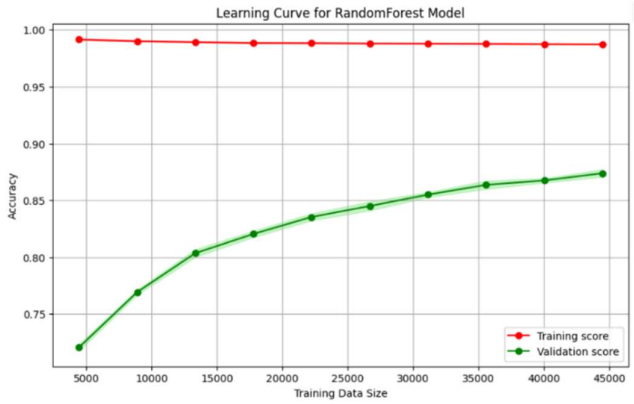

Figure 14 :Learning Curve for Random Forest Model

- ROC Curves: (Nahm, 2022) The ROC curves illustrated the trade-off between true positive rate and false positive rate for each class, providing a comprehensive view of model performance across different thresholds. Show figure 15

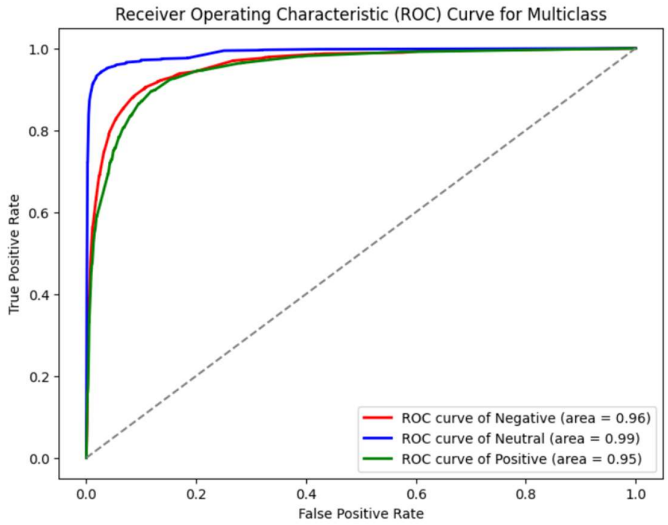

Figure 15 :Receiver Operating Characteristic (ROC) Curve for Multiclass

- Training and Validation Error vs. Number of Trees: plots the error rates on the y-axis against the number of decision trees in the ensemble on the x-axis. Training error refers to the model's error rate on the dataset used to build the model, while validation error measures the error on a separate, unseen dataset. Show figure 16

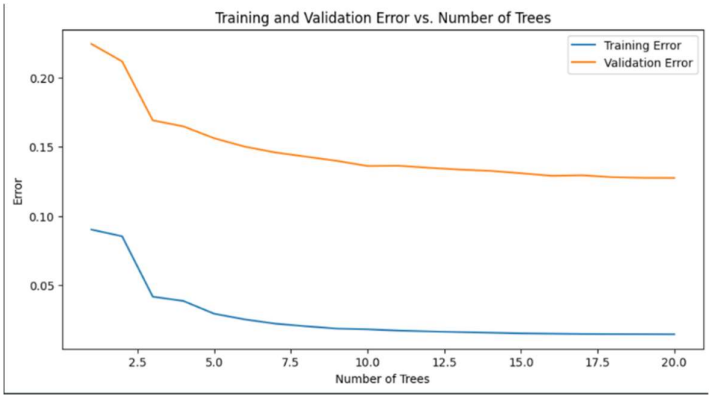

Figure 16 :Training and Validation Error vs. Number of Trees

Overall, the use of SMOTE significantly improved the model's ability to handle imbalanced classes, resulting in more balanced precision and recall across all classes. These results demonstrate the effectiveness of the proposed methodology in addressing sentiment analysis of company reviews and highlight areas for further improvement.

4.4 Model Testing with New Comments

To further evaluate the model's performance, we tested the built Random Forest model with new, unseen comments to determine their sentiment. Several comments were provided, and the model was able to classify them as positive, negative, or neutral. This real-world testing validated the model's capability to generalize and correctly predict sentiments of new reviews.

Example Comments and Predictions:

Comment: "وصلني الطعام متأخرا والاكل بارد" (The food arrived late and cold)

Predicted Label: Negative

Comment: "أفضل تطبيق" (Best application)

Predicted Label: Positive

Comment: "ليس سيء و ليس جيد" (Not bad, not good)

Predicted Label: Neutral

Comment: "مش عارف" (I don't know)

Predicted Label: Neutral

Comment: "تأخر الأكل" (The food was delayed)

Predicted Label: Negative

Comment: "تجربة ممتازة" (Excellent experience)

Predicted Label: Positive

These predictions demonstrate the model's effectiveness in understanding and classifying sentiments in Arabic text. The ability to accurately predict the sentiment of new comments indicates that the model is well-trained and can be used in practical applications to analyze user feedback and reviews.

5.0 Conclusions and Future Works

In this study, we conducted an extensive analysis of Arabic sentiment data using various machine learning models, achieving a final accuracy of 88%. The implementation of SMOTE (Synthetic Minority Over-sampling Technique) was critical in addressing data imbalance across the three sentiment classes. This allowed us to create a more robust and reliable sentiment analysis model. The preprocessing steps, including tokenization, stop words removal, stemming, and lemmatization, played a significant role in preparing the Arabic text data for analysis.

5.1 Strengths

- High Accuracy: The final model achieved an accuracy of 88%, demonstrating its effectiveness in sentiment classification.

- Data Imbalance Handling: The use of SMOTE successfully addressed the issue of data imbalance, ensuring that all sentiment classes were adequately represented in the training process.
- Comprehensive Preprocessing: The extensive preprocessing pipeline, including emoji removal, punctuation removal, tokenization, stop words removal, stemming, and lemmatization, significantly improved the quality of the input data.
- Visualization Tools: The use of visualizations such as word clouds, pie charts provided valuable insights into the dataset and the distribution of sentiments and companies.

5.2 Weaknesses

- Dialectal Variations: The model struggled with accurately classifying sentiments across different Arabic dialects, which have distinct vocabulary and syntax.
- Limited Lexical Resources: The lack of comprehensive Arabic sentiment lexicons posed a challenge in enhancing the model's performance.
- Complexity in Preprocessing: The extensive preprocessing required for Arabic text added complexity and increased the processing time.

5.3 Future Works

- Enhancing Dialect Handling: Future work should focus on improving the model's ability to handle various Arabic dialects. This could involve training dialect-specific models or incorporating dialectal variations into the preprocessing pipeline.
- Real-Time Sentiment Analysis: Developing a real-time sentiment analysis application that can process and classify sentiments in live text streams, such as social media feeds, could be a valuable extension of this work.

- Exploring Deep Learning Models: Investigating the use of deep learning models such as BERT (Jacob Devlin, 2018) or GPT (Anis Koubaa, 2024) for Arabic sentiment analysis could provide insights into further improving model performance and handling more complex linguistic structures.
- Deployment into an Environment: Deploying the model into an environment using frameworks like Flask (Steffi, 2022) or Streamlit (Aslan, 2024) would allow users to interact with the model easily. This would enable the sentiment analysis tool to be accessible for real-time usage and testing by end-users.

Table of Figures

List Of References

Anis Koubaa, A. A. (2024). *Arabian GPT*. Retrieved from https://arxiv.org/abs/2402.15313

Arthur V, R. (2021). *Multinomial Naïve Bayes' For Documents Classification and Natural Language Processing (NLP)*. Retrieved from https://towardsdatascience.com/multinomial-na%C3%AFve-bayes-for-documents-classification-and-natural-language-processing-nlp-e08cc848ce6

Aslan, F. (2024). *Streamlit*. Retrieved from https://medium.com/@furkan-aslan/streamlit-the-magic-of-data-storytelling-03d138a4b301

Barhoumi. (2020, may). *Qualitative Evaluation of Embeddings for Arabic Sentiment Analysis.* . Retrieved from aclanthology: https://aclanthology.org/2020.lrec-1.610.pdf

Barhoumi, A., Camelin, Nathalie, Aloulou, C., Esteve, Y., & Belguith, L. (2020, May). Toward Qualitative Evaluation of Embeddings for Arabic Sentiment Analysis. Retrieved from https://aclanthology.org/2020.lrec-1.610.pdf

Chavan, M. (2021). *Precision, Recall & F1-Score*. Retrieved from https://medium.com/@mahesh.chavan1997/what-is-precision-recall-f1-score-b65b1965804c

Company_reviews.csv. (2021). *kaggle*. Retrieved from https://www.kaggle.com/datasets/fahdseddik/arabic-company-reviews

Dubrov, V. (2023). *Understanding Machine Learning Robustness*. Retrieved from https://medium.com/@slavadubrov/understanding-machine-learning-robustness-why-it-matters-and-how-it-affects-your-models-5e2cb5838dab

Gökdemir, E. (2023). *What is Tokenization?* Retrieved from https://medium.com/sabancidx/what-is-tokenization-a60c77ea6424

Gunay, D. (2023). *Basics of Matplotlib & Seaborn*. Retrieved from https://medium.com/@denizgunay/basics-of-matplotlib-seaborn-74f6c1d5fc53

Haouhat, A. B. (2023). *Arabic Multimodal Dataset for Sentiment Analysis*. Retrieved from https://ar5iv.labs.arxiv.org/html/2306.06322

Haouhat, A., Bellaouar, S., Nehar, A., & Cherroun, H. (2023). Towards Arabic Multimodal Dataset for Sentiment Analysis. Retrieved from https://ar5iv.labs.arxiv.org/html/2306.06322

IBM. (2020). *What is random forest?* Retrieved from https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,Decision%20trees

Jacob Devlin, M.-W. C. (2018). *BERT Paper*. Retrieved from https://arxiv.org/abs/1810.04805

Jain, A. (2024). *TF-IDF in NLP (Term Frequency Inverse Document Frequency)*. Retrieved from https://medium.com/@abhishekjainindore24/tf-idf-in-nlp-term-frequency-inverse-document-frequency-e05b65932f1d#:~:text=TF%2DIDF%20is%20a%20numerical,its%20rarity%20across%20multiple%20documents.

Maklin, C. (2022). *Synthetic Minority Over-sampling TEchnique (SMOTE)*. Retrieved from https://medium.com/@corymaklin/synthetic-minority-over-sampling-technique-smote-7d419696b88c

Malviya, N. (2023). *Confusion Matrix*. Retrieved from https://medium.com/@nikitamalviya/confusion-matrix-870739a1ec31

Mohsen, A. &. (2020, january 2). *Arabic Sentiment Analysis: A Systematic Literature Review (SLR)*. Retrieved from https://www.hindawi.com/journals/acisc/2020/7403128/

Mohsen, A., & Ali, Y. (2020, January 29). Arabic Sentiment Analysis: A Systematic Literature Review (SLR). Retrieved from https://www.hindawi.com/journals/acisc/2020/7403128/

Mudadla, S. (2023). *NLP Model Metrics*. Retrieved from https://medium.com/@sujathamudadla1213/nlp-model-metrics-b3fa32373269

Musleh, D. A. (2023, july 3). *Arabic Sentiment Analysis of YouTube Comments: NLP-Based Machine Learning Approaches for Content Evaluation.* Retrieved from https://www.mdpi.com/2504-2289/7/3/127

Musleh, D., Alkhwaja, I., Alkhwaja, A., Alghamdi, M., Abahussain, H., Alfawaz, F., . . . Abdulqader, M. (2023, July 3). Arabic Sentiment Analysis of YouTube Comments: NLP-Based Machine Learning Approaches for Content Evaluation. Retrieved from https://www.mdpi.com/2504-2289/7/3/127

Nabil, M. (2023). *Text normalization (e.g., stemming, lemmatization, lowercasing)*. Retrieved from https://medium.com/@madali.nabil97/text-normalization-e-g-stemming-lemmatization-lowercasing-833558a3ca21

Nahm, F. S. (2022). *Receiver operating characteristic curve*. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8831439/#:~:text=The%20ROC%20curve%20is%20an,or%20absence%20of%20a%20disease.

Olamendy, J. C. (2024). *Understanding the Learning Curves in ML*. Retrieved from https://medium.com/@juanc.olamendy/understanding-the-learning-curves-in-ml-2ec442d91b8f#:~:text=Learning%20curves%20are%20not%20just,of%20the%20model's%20learning%20journey.

Otten, N. V. (2022). *Arabic NLP*. Retrieved from https://medium.com/@neri.vvo/arabic-nlp-how-to-overcome-challenges-in-preprocessing-ed56de0c43e2

Rai, S. (2020). *Word Cloud: A Text Visualization tool*. Retrieved from https://medium.com/analytics-vidhya/word-cloud-a-text-visualization-tool-fb7348fbf502

Saini, A. (2024). *Guide to Logistic Regression*. Retrieved from https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/

Samancı, B. (2024). *Hyperparameter Tuning*. Retrieved from https://medium.com/@betulsamancii/hyperparameter-tuning-9b7717b89e19#:~:text=Hyperparameter%20tuning%20is%20the%20process,in%20a%20support%20vector%20machine.

Soni, P. (2020). *Handling Imbalanced Datasets With imblearn Library*. Retrieved from https://medium.com/thecyphy/handling-imbalanced-datasets-with-imblearn-library-df5e58b968f4#:~:text=Imblearn%20library%20is%20specifically%20designed,the%20imbalance%20from%20the%20dataset.

Steffi. (2022). *What is Flask?* Retrieved from https://medium.com/data-science-ai-learning-journey/what-is-flask-cab5eb6e74f0#:~:text=Flask%20is%20a%20microframework%20for,Request%20Routing

Thai, C. (2019). *NumPy and Pandas*. Retrieved from https://medium.com/@christhai_6937/data-science-in-python-part-1-numpy-and-pandas-befcdddc0a05

w3schools. (n.d.). *Pandas Introduction*. Retrieved from https://www.w3schools.com/python/pandas/pandas_intro.asp

Wei, D. (2024). *Scikit-learn*. Retrieved from https://medium.com/@weidagang/essential-python-libraries-for-machine-learning-scikit-learn-the-ml-magician-6d099e6b4c87#:~:text=Scikit%2Dlearn%2C%20also%20known%20as,and%20building%20machine%20learning%20models.