UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Faculty of Engineering, Built Environment and Information Technology
Fakulteit Ingenieurswese, Bou-omgewing en Inligtingtegnologie / Lefapha la Boetšenere, Tikologo ya Kago le Theknolotši ya Tshedimoš

DEPARTMENT OF INFORMATICS

Assignment Cover Page

Student Number(s)								Surname	Initials
1	9	0	1	5	7	2	2	Braga	CLB
2	4	2	3	7	6	0	1	Welters	DJ
0	4	8	6	9	4	6	1	Wood	FS
2	1	4	7	5	6	5	3	Van der Merwe	ML
2	1	5	2	9	8	1	8	van Rooyen	C
2	1	5	6	2	8	3	2	Coetzee	A
2	1	4	4	8	5	6	7	Cress	CW
2	1	6	0	0	3	9	3	Adams	MS
2	1	4	4	3	2	5	5	Greyling	A

Module Code	**INF :**	7	9	1
Assignment Number	3			
Mark Allocation				
Date of Submission	18 October 2024			
Name of Lecturer	Dr. WA NKONGOLO MIKE NKONGOLO			

Table of Contents

1. Introduction *4*

2. Literature Review *5*

3. Data Collection and Processing *6*

Data collection 6

Data translation process 7

Data preprocessing 8

Sentiment score assignment 9

Data Cleaning Steps 9

4. Methodology *10*

Data Cleanup and Analysis 15

5. Results *15*

6. Discussion *42*

7. Conclusion *43*

8. Future Research *44*

9. References *45*

Table of figures

Figure 1: Dataset Load 10
Figure 2: Sentiment Analysis 10
Figure 3: Machine Learning 11
Figure 4: Sentence Dataset Load 12
Figure 5: Translate Sentences 13
Figure 6: Interactive Sentence and Language Selection 14
Figure 7: Feature Columns 15
Figure 8: Probability Prediction 16
Figure 9: ROC Curve for Logistic Regression 19
Figure 10: Confusion Matrix for Decision Tree 20
Figure 11: ROC Curve for Decision Tree 23
Figure 12: Confusion Matrix for Random Forest 24
Figure 13: ROC Curve for Random Forest 27
Figure 14: Confusion Matrix for SVM 28
Figure 15: ROC Curve for SVM 31
Figure 16: Count for Ciluba Words 32
Figure 17: Count for French words 33
Figure 18: Count of sentiment 34
Figure 19: Distribution of Sentiment Scores 35
Figure 20: Count of Word Nature 36
Figure 21: Language Translation Prompts 37
Figure 22: English Sentiment Values 37
Figure 23: Afrikaans Sentiment Values 38
Figure 24: Zulu Sentiment Values 39
Figure 25: Zulu Sentiment Values 40
Figure 26: Zulu Sentiment Values 41

1. Introduction

Artificial Intelligence (AI) has transformed the field of translation due to its ability to rapidly translate large sets of text, exceeding human capabilities (Bouguesmia, 2020). The power of AI can be beneficial in countries like South Africa, where there are 11 official languages. These technologies can enforce seamless and accurate language translations (Tanaka & Rossi, 2024), creating efficient and accessible communication across diverse linguistic groups. However, challenges arise when trying to accurately translate the meaning and emotional tone across different language translations (Van Atteveldt et al., 2021). AI machine learning is presented as a solution to this.

This study focuses on the expansion of a Lexicon for sentiment analysis and translation across a selection of South African languages alongside French and English. The goal is to explore the potential of AI-powered systems for accurate sentiment identification. A selection of machine-learning models are used to perform sentiment analysis that fully captures diverse cultural nuances that enhance language translations. This is important for enhancing translations and computing sentiment analysis across areas with multiple language groups, ultimately achieving an accurate representation of diverse linguistic groups. The study is composed as follows: the second section provides a detailed literature review conceptualising the key concepts and challenges related to sentiment analysis translations in multilingual environments. The third section presents the data collection and processing. The fourth section provides the methodology followed in this paper, followed by the results obtained and a discussion of the results presented in sections 5 and 6. Finally, this paper concludes with an outline of the key findings identified.

2. Literature Review

Social media monitoring and customer feedback analysis are just some applications in which sentiment analysis has been incorporated with natural language processing (Bing, 2012). By using a predefined list of words which are then associated with sentiment labels, a lexicon-based approach can be used due to its interpretability and simplicity (Taboada et al., 2011). By looking into the intensity and polarity of words, these methods can be used by sentiment lexicons for the classification of text. However, when numerous languages are involved, it can pose a particular challenge for the lexicon when translating between these different languages and cultures. In a country like South Africa for example which contains a total of 11 official languages, to construct sentiment lexicons, it requires an understanding of cultural differences as well as linguistic expertise (Chen & Skiena, 2014). Unfortunately, Africa has low resource languages which pose an extra challenge. However, by making use of machine learning models and integrating it with lexicon-based methods, these limitations can be addressed. This can enhance the sentiment classification by analysing the data and learning patterns (Chen & Skiena, 2014). For more flexible and dynamic classification models to be developed, hybrid approaches have been developed that are able to combine supervised learning with lexicons. This then allows for ambiguous and complex texts to be analysed (Cambria et al., 2013). However, there are different connotations with words depending on the culture and the context in which they are used. Thus, it is important to make sure that contextual shifts and linguistic variations are carefully considered. Therefore, it is important that multilingual lexicons exist as they play a crucial role in improving the sentiment analysis of systems.

3. Data Collection and Processing

For this assignment, there were two key datasets that were made use of to perform the sentiment analysis. This analysis included different languages that were high-resource and low-resource languages. This was done so that automated sentiment classification tools could be applied in multilingual environments, specifically with African languages. To fully understand the datasets used, an in-depth explanation is as follows:

Data collection

1. Tshikama dataset

This dataset includes translations between Ciluba and French languages and offers a bilingual lexicon. Ciluba is a Bantu language that is spoken in the DRC (Democratic Republic of Congo). Each word in the dataset is then classified with a sentiment label:

- Positive (Positif)
- negative (Negatif)
- Neutral (Neutre)

Once this has been done, they are then categorised into groups based on its grammatical nature (parts of speech). This sentiment analysis is useful underrepresented languages where linguistic resources are limited.

2. Lexicon Expanded with Sentiment VADER Dataset

This dataset includes the preprocessing that has been done in order to explain on the Tshukama dataset as it includes the translations into different languages, including:

- Afrikaans
- English
- Zulu
- Xhosa

Sentiment scores were allocated to each word that has been translated into the different languages by making use of tools such as VADER sentiment analysis that is commonly used for sentiment classification in text. The expansion into the different languages is important for this

analysis as it allows for a more comparative analysis across different linguistic contexts that include both high-resource languages and low- resource languages.

Data translation process

In order for our machine learning model to accurately output the correct results, the translation of words between the translation between the different languages had to be prepared and processed. in order to achieve this, the following had to be conducted:

1. Manual translations

When translating words from Ciluba and French into the different languages (English, Afrikaans, Zulu and Xhosa), each word was verified manually by making use of tools such as Google Translate. This was done to ensure that the sentiments were accurately captured. When translating the words manually, careful attention was placed on the context of each word as they carry different emotional tones across the different languages.

2. Machine translations and VADER

When looking at high-resource languages such as English, VADER sentiment analysis tools are able to automatically assign sentiment scores by analysing the word usage and tone to generate sentiment scores. This is done to ensure that the translation aligns with the meaning of the original text. The scores that are given are the following:

- Positive: The words that were translated maintain their corresponding positive sentiment scores.
- Negative: The words in the dataset were translated to ensure that the emotional tones were kept.
- Neutral: The words in the dataset remained neutral in the processing of translating.

Data preprocessing

Before the sentiment analysis was done, preprocessing of the dataset had to be performed so that the machine learning model was trained on clean and consistent data. In order to achieve this, the following steps were taken:

1. Normalisation

- To standardise data across the different languages, the formatting of the text had to remain consistent. This included making sure that all the words were in lower case and that there were no characteristics such as bullet points before or after the words. This is done so that there are no discrepancies caused by case sensitivity.

2. Handling missing values

- If any row was found with a missing value (this includes missing translations, sentiment scores or the grammatical nature of text), it was investigated and analysed until the corrected result was found. This step was done to ensure that the dataset was complete and reliable.

3. Whitespace and formatting

- White spaces were removed before and after the words were removed. This was a minor step but is crucial to ensure that there are no formatting issues that arise during the sentiment analysis. This is especially important when the translated words are matched against the predefined sentiment lexicons.
- Bullet points and numbering were also removed that had the potential to interfere with the analysis.

Sentiment score assignment

1. Sentiment scores by language

Each word in the dataset (including all the different languages) was given a corresponding sentiment score. The scores were calculated by making use of VADER for English and then were mapped to the rest of the languages (Zulu, Xhosa and Afrikaans). The following scale was used:

- Positive sentiment: words with a score greater than 0.
- Negative sentiment: words with a score less than 0.
- Neutral sentiment: words with the score of 0.

2. Translation consistency

Each sentiment score that was given to the translated word was carefully reviewed to ensure that the scores remained consistent between the different languages. For example, taking a positive sentiment score in Ciluba, the translated word in Afrikaans and Xhosa had to carry the same sentiment.

Data Cleaning Steps

1. Removal of non-alphabetic characters:

When running the machine learning model, some words were translated into symbols, specifically Arabic letters. This had to be removed and retranslated into the correct word to ensure that the dataset focuses solely on the textual content.

2. Data validation

A final review was conducted to ensure and validate that all the translations align correctly with the corresponding sentiment scores so that no data was duplicated or corrupted during the process.

4. Methodology

4.1 Loading and Translating the Dataset

The dataset was loaded, and feature columns were named correctly. Sentences were split into words and the words were translated using the lexicon. The translated words were then concatenated together to form the translated sentence.

```
# Load the dataset
file_path = r'C:\Users\AlexGreyling\OneDrive - Agile Bridge\Honours\Semester 2\INF 791\lexicon_expanded.xlsx'
df = pd.read_excel(file_path)

# Ensure columns are named correctly and include English and South African Languages
df.columns = ['ciluba', 'french', 'score', 'sentiment', 'nature', 'english', 'afrikaans', 'zulu', 'xhosa']

# Step 1: Translation functions
def translate_text_using_lexicon(text, lexicon):
    words = text.lower().split()
    translated_words = [lexicon.get(word, word) for word in words]
    return ' '.join(translated_words)

# Create translation lexicon from the dataset
translation_lexique = dict(zip(df['french'].str.lower(), df['ciluba']))

# Step 2: Sentiment Analysis Function
lexique = dict(zip(df['ciluba'].str.lower(), df['score']))
```

Figure 1: Dataset Load

4.2 Sentiment analysis

Sentiment analysis was performed on the translated sentence using the sentiment score for the translated language within the lexicon dataset. The analyse_sentiment function analyses the sentences by matching the words in text to scores in the lexicon. The lexique is a dictionary with words from the Ciluba column mapped to their sentiment scores variable from score. Each word score is summed up to calculate a positive or negative value resulting in a positive or negative sentiment for the sentence.

```
# Step 2: Sentiment Analysis Function
lexique = dict(zip(df['ciluba'].str.lower(), df['score']))

def analyse_sentiment(text):
    words = text.lower().split()
    word_scores = {word: lexique.get(word, 0) for word in words}
    score = sum(word_scores.values())
    if score > 0.05:
        sentiment = "Positive"
    elif score < -0.05:
        sentiment = "Negative"
    else:
        sentiment = "Neutral"
    return score, sentiment, word_scores
```

Figure 2: Sentiment Analysis

4.3 Machine Learning Pipeline

Pre-processing was done by means of dropping rows with undefined or missing values. The french column was set to the X value and the sentiment column to the Y value. The dataset was then split into a 30 test / 70 train ratio. Feature extraction was performed using the TfidfVectorizer method to limit extracted features to 5000. Model training an evaluation was executed using four machine learning models namely: Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine. These models are explained below.

```
# Binarize the output for multi-class ROC
y_bin = label_binarize(y, classes=['Negatif', 'Neutre', 'Positif'])
n_classes = y_bin.shape[1]

# Split into training and testing datasets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Feature extraction using TF-IDF
vectorizer = TfidfVectorizer(max_features=5000)  # Limit to top 5000 features
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

# Initialize machine learning models
models = {
    "Logistic Regression": LogisticRegression(max_iter=1000),
    "Decision Tree": DecisionTreeClassifier(),
    "Random Forest": RandomForestClassifier(),
    "Support Vector Machine": SVC(probability=True)
}

# Train and evaluate each model
for model_name, model in models.items():
    model.fit(X_train_tfidf, y_train)
    y_pred = model.predict(X_test_tfidf)
    y_proba = model.predict_proba(X_test_tfidf) if hasattr(model, "predict_proba") else None
```

Figure 3: Machine Learning

Logistic Regression

According to (Starbuck, 2023), logistic regression is a general linear model used for modelling relationships between non-continuous variables. It is according to (Craig. S, 2023), a suitable model to predict probability of observations between various classes of categorical data and supports multi-class datasets.

Decision Tree

Decision Trees are a form of classification trees used to discover individual variances within dataset features (Carrillo et al, 2023). (Yuanyang et al. 2021) states that decision trees are suitable visual representation for humans to understand feature importance within datasets. (Carrillo et al, 2023) further states that decision trees classify features in hierarchical form.

Random Forest

(Rahnasto et al. 2024) states that Random Forest models are able to display nonlinear relationships between a predefined input and output variable. (Rahnasto et al. 2024) defines random forests as aggregated models capable of producing final predictions based on multiple decision trees.

Support Vector Machine

Support Vector Machine is best suited for recognition of patterns, identifying objects, and image classification (Ramkumar et al, 2023). Values are separated and grouped based on a calculated weight bias which reduces costs of operation. (Hamed et al, 2023)

Training and evaluation

For each model, training was done on the trained data to make predictions on the training data. The classification report was then printed for each trained model and displayed a confusion matrix which showed true positive, true negative, false positive, and false negative values.

If the model allows for probability predictions, it will calculate the ROC AUC curve for each sentiment class. This plots them for visual performance values shown in figure 6. Once the model was trained, the sentences dataset was loaded, displaying the first five sentences as displayed in .

```
print(df_sentences.head())

Columns in the dataset: Index(['ciluba', 'french', 'score', 'sentiment', 'nature', 'english',
       'afrikaans', 'zulu', 'xhosa'],
      dtype='object')
                                English Sentences
0   On the top of the hill, the stars began to fade.
1  In the middle of the forest, the moonlight ill...
2  She was reading her favorite book when a myste...
3  As the music played softly, an unexpected visi...
4  He opened the door to find they exchanged stor...
```

Figure 4: Sentence Dataset Load

Translate Text Function

The translate_text function uses MyMemoryTranslator translator which initialises the translator with the specific source and target language. If the translator fails, it tries to translate another 2 times. After each failed attempt, it would wait 2 seconds before retrying again. If it fails a third time, the function returns an error message.

Sentiment Function

The analyse_sentiment function analyses the sentiment of a piece of text and returns a score and a label. It uses TextBlob which calculates the polarity of text, which ranges between -1 and 1. The polarity is scaled between -9 and 9 which produces the sentiment score. Scores above zero are positive, zero is neutral, and below zero negative in sentiment value.

Translate Sentences

The translate_sentences function translates a specific number of sentences, which includes the data frames and target language, to form sentiment analysis on both the original and translated text. The function loops over the first number of sentences and rows in the data frame. Each English sentence is translated into the target language using the translate_text function. The results from the translated text are then appended to a list. Once completed, a new data frame, translated_df, is created from the translated text as depicted in figure 5.

```python
# Function to translate a selected number of sentences into the chosen language and perform sentiment analysis
def translate_sentences(df, num_sentences, target_language):
    translated_data = []

    for idx, row in df.head(num_sentences).iterrows():
        english_sentence = row['English Sentences']
        translated_sentence = translate_text(english_sentence, source_language='en-GB', target_language=target_language)

        # Perform sentiment analysis on both the original and translated sentences
        original_sentiment_score, original_sentiment_label = analyze_sentiment(english_sentence)
        translated_sentiment_score, translated_sentiment_label = analyze_sentiment(translated_sentence)

        # Append data to the list
        translated_data.append({
            'Original English': english_sentence,
            'Original Sentiment': f"{original_sentiment_label} (Score: {original_sentiment_score})",
            'Translated': translated_sentence,
            'Translated Sentiment': f"{translated_sentiment_label} (Score: {translated_sentiment_score})"
        })

    # Create a DataFrame from the translated data
    translated_df = pd.DataFrame(translated_data)
    return translated_df
```

Figure 5: Translate Sentences

Interactive Sentence and Language Selection

The interactive_translation function prompts the user to choose a number of sentences to translate. The function then returns three languages to choose from namely: Afrikaans, Zulu, and Xhosa as the target translation language. Based on this input, the *translate_language* function is

called to handle the translation and sentiment analysis as described above. Once translated, the result is displayed to the user. This function is displayed in

```python
# Main function to interactively select number of sentences and language
def interactive_translation(df):
    # Prompt the user to choose the number of sentences (up to a maximum of 10)
    while True:
        try:
            num_sentences = int(input("How many sentences would you like to translate? (Max 10): "))
            if 1 <= num_sentences <= 10:
                break
            else:
                print("Please enter a number between 1 and 10.")
        except ValueError:
            print("Invalid input. Please enter a valid number.")

    # Prompt the user to choose a target language
    languages = {
        '1': ('af-ZA', 'Afrikaans'),
        '2': ('zu-ZA', 'Zulu'),
        '3': ('xh-ZA', 'Xhosa')
    }

    print("\nSelect the language you want to translate to:")
    print("1. Afrikaans")
    print("2. Zulu")
    print("3. Xhosa")

    while True:
        language_choice = input("Enter the number of the language (1, 2, or 3): ")
        if language_choice in languages:
            target_language_code, target_language_name = languages[language_choice]
            break
        else:
            print("Invalid choice. Please select 1, 2, or 3.")

    # Translate the selected number of sentences and perform sentiment analysis
    translated_df = translate_sentences(df, num_sentences, target_language_code)

    # Display the translated sentences and their sentiment scores
    print(f"\nTranslated {num_sentences} sentences into {target_language_name}:")
    print(translated_df)
```

Figure 6: Interactive Sentence and Language Selection

Data Cleanup and Analysis

Each word in the lexicon was translated and verified using google translate to ensure that translation between languages were as accurate as possible. This was essential to maintain consistent contexts between languages.

There were capitalisation inconsistencies throughout the lexicon, this was addressed by converting all text to lowercase by utilising the PROPER function to reformat the words. This ensures that any case sensitive issues were mitigated during analysis.

A preliminary translation was done on the dataset using a python library called deep-translator and during this translation process it added arabic words to the dataset, as well as numbers in words. Non-alphabetic characters such as Arabic letters were removed from the dataset as well as any numbers included in words.

Furthermore, bullet points and whitespace, before and after words, were removed to prevent any formatting issues that might occur.

5. Results

```
Columns in the dataset: Index(['ciluba', 'french', 'score', 'sentiment', 'nature', 'english',
       'afrikaans', 'zulu', 'xhosa'],
      dtype='object')
                                English Sentences
0   On the top of the hill, the stars began to fade.
1  In the middle of the forest, the moonlight ill...
2  She was reading her favorite book when a myste...
3  As the music played softly, an unexpected visi...
4  He opened the door to find they exchanged stor...
```

Figure 7: Feature Columns

The above figure shows the different columns that exist in the dataset.

- Ciluba: Represents words from the Ciluba language.
- French: Represents words translated into French.
- Score: Represents the sentiment score.
- Sentiment: Represents the sentiment (positive, negative or neutral)

- Nature: Represents the parts of speech (eg. verb, word)
- English: Represents words from the English language.
- Afrikaans: Represents words from the Afrikaans language.
- Zulu: Represents words from the Zulu language.
- Xhosa: Represents words from the Xhosa language.

```
Classification Report for Logistic Regression:
              precision    recall  f1-score   support

     Negatif       0.70      0.08      0.15        84
      Neutre       1.00      0.49      0.66        41
     Positif       0.89      1.00      0.94       775

    accuracy                           0.89       900
   macro avg       0.86      0.52      0.58       900
weighted avg       0.88      0.89      0.85       900
```

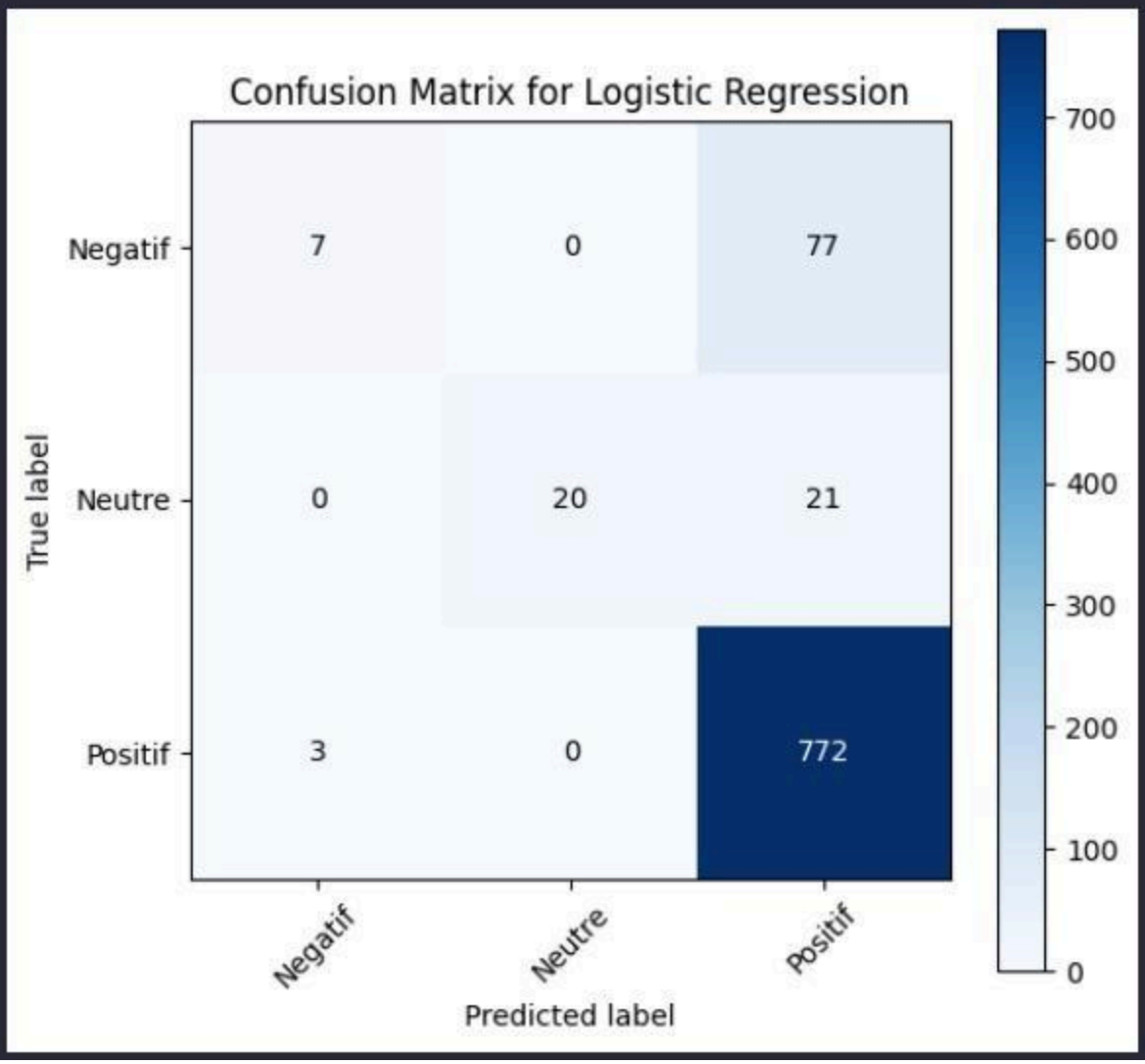

Figure 8: Probability Prediction

The above figure shows a classification report and a confusion matrix for the logistic model applied to the dataset

Classification Report

The performance of the logistic regression model is measured across the three sentiment categories:

- **Precision:** The percentage of correct positive predictions that have been given for each label.
    - **Negative:** 0.7, this means 70% of the predicted negative labels were correct
    - **Neutral:** 1, this means 100% of the predicted neutral labels were correct
    - **Positive:** 0.89, this means 89% of the predicted positive labels were correct
- **Recall:** The percentage of actual positive instances that have been identified correctly.
    - **Negative:** 0.08, very low recall, only 8% of actual negative instances were correctly identified
    - **Neutral:** 0.49, 49% recall, moderate accuracy in identifying neutral labels
    - **Positive:** 1.00, perfect recall, all actual positive labels were correctly identified
- **F1-Score:** The harmonic mean of precision and recall.
    - **Negative:** 0.15, low F1-score due to the very low recall
    - **Neutral:** 0.66, moderate performance
    - **Positive:** 0.94, strong performance
- **Support:** The number of actual instances of each class in the test data.
    - **Negative:** 84 samples.
    - **Neutral:** 41 samples.
    - **Positive:** 775 samples.
- **Overall Metrics:**
    - **Accuracy**: The overall accuracy of the model is 89% which means it correctly identified 89% of all instances.
    - **Macro Average**: The average precision, recall and F1-score which has been equally weighted for each call
        - **Precision**: 0.86
        - **Recall**: 0.52

        - **F1-Score**: 0.58
    - **Weighted Average**: The average weighted by the number of instances in each class
        - **Precision**: 0.88
        - **Recall**: 0.89
        - **F1-score**: 0.85

Confusion Matrix

The confusion matrix shows how the logistics regression performed in terms of predicting each class. The graph depicts the actual vs predicted labels.

- **Negative**: Only a few actual negative instances were correctly predicted as the diagonal for the class is a lighter colour, meaning most of the negative instances were misclassified.
- **Neutral**: Around half of the neutral instances were correctly predicted as can be seen by a darker shading around this area
- **Positive**: Majority of the positive instances were correctly predicted, as can be seen by the dark colour, which also related to the high precision and recall for this specific class.

The logistic regression model performs very well for the positive class with both precision and recall being high and as can be seen from the confusion matrix, majority of the predictions were concentrated in this class. The logistic regression model however struggled with the negative class with a very low recall of 0.08 and a very light colour on the matrix. The neutral predictions were very balanced as half of the predictions were correct, however some of these predictions misclassified as positive. It is also clear to see the imbalance in the dataset as there are 775 positive instances, 84 negative instances and 41 neutral instances.

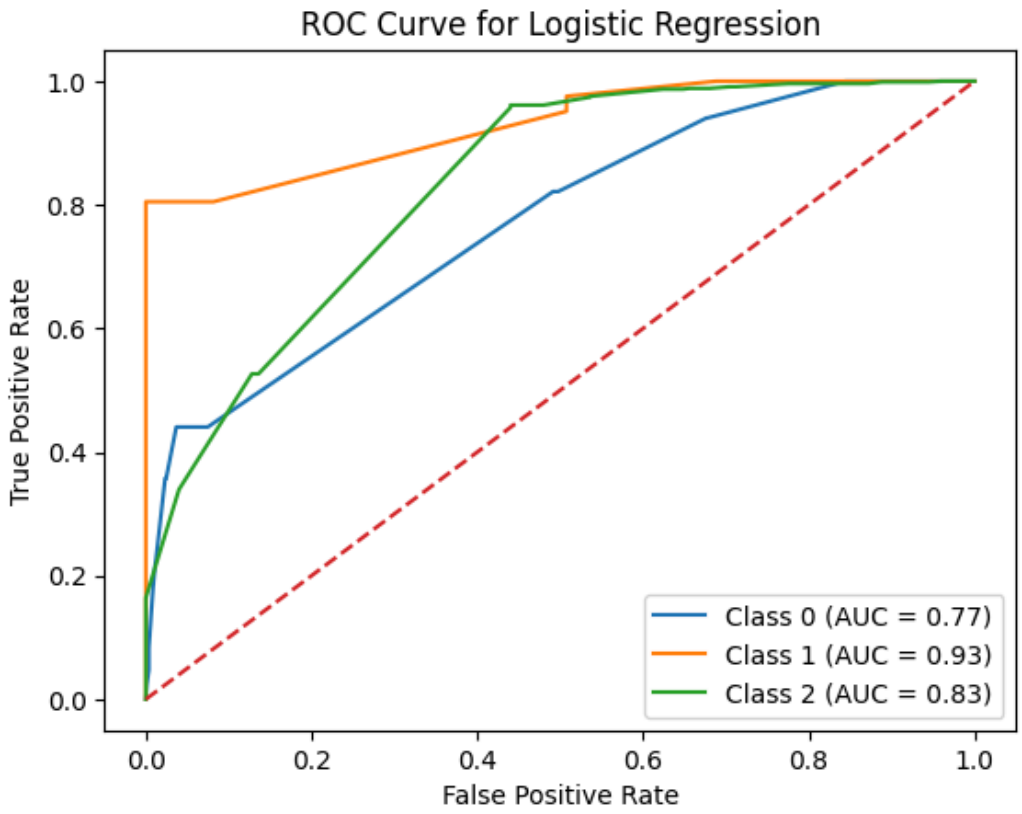

Figure 9: ROC Curve for Logistic Regression

The image above shows the logistic regression model ROC curve for the three sentiment classes. The ROC curve can be used to evaluate the performance of a classification model by plotting the True Positive Rate (TPR) but the False Positive Rate (FPR). The area under the curve (AUC) shows the overall ability of the model to distinguish between the three sentiment classes.

- Class 0 / Negative: The blue line represents Class 0 or the Negative class. This class has an AUC score of 0.77 which is the lowest performing class compared to the others. That means this class is less effective at correctly identifying negative instances.
- Class 1 / Neutral: The orange line represents Class 1 or the Neutral class. This class has an AUC score of 0.93 which is the highest performing class compared to the others. That means this class is more effective at correctly identifying true positives and is also effective at keeping the false positives low.
- Class 2/ Positive: The green line represents Class 2 or the Positive class. This class has an AUC score of 0.83 which is the middle performing class. That means this class has a

reliable balance and is effective at correctly identifying true positives and is also effective at keeping the false positives low.

Class 1 has the highest AUC at 0.93 indicating the model is very good at distinguishing neutral sentiment. Class 2 has the second highest AUC at 0.83 indicating it performs well at correctly classifying positive sentiment. Class 0 has the lowest AUC at 0.77 meaning the model struggled at correctly identifying negative sentiment which is consistent with the poor recall and precision for the negative class.

Classification Report for Decision Tree:

	precision	recall	f1-score	support
Negatif	0.56	0.44	0.49	84
Neutre	1.00	0.71	0.83	41
Positif	0.93	0.96	0.94	775
accuracy			0.90	900
macro avg	0.83	0.70	0.76	900
weighted avg	0.90	0.90	0.90	900

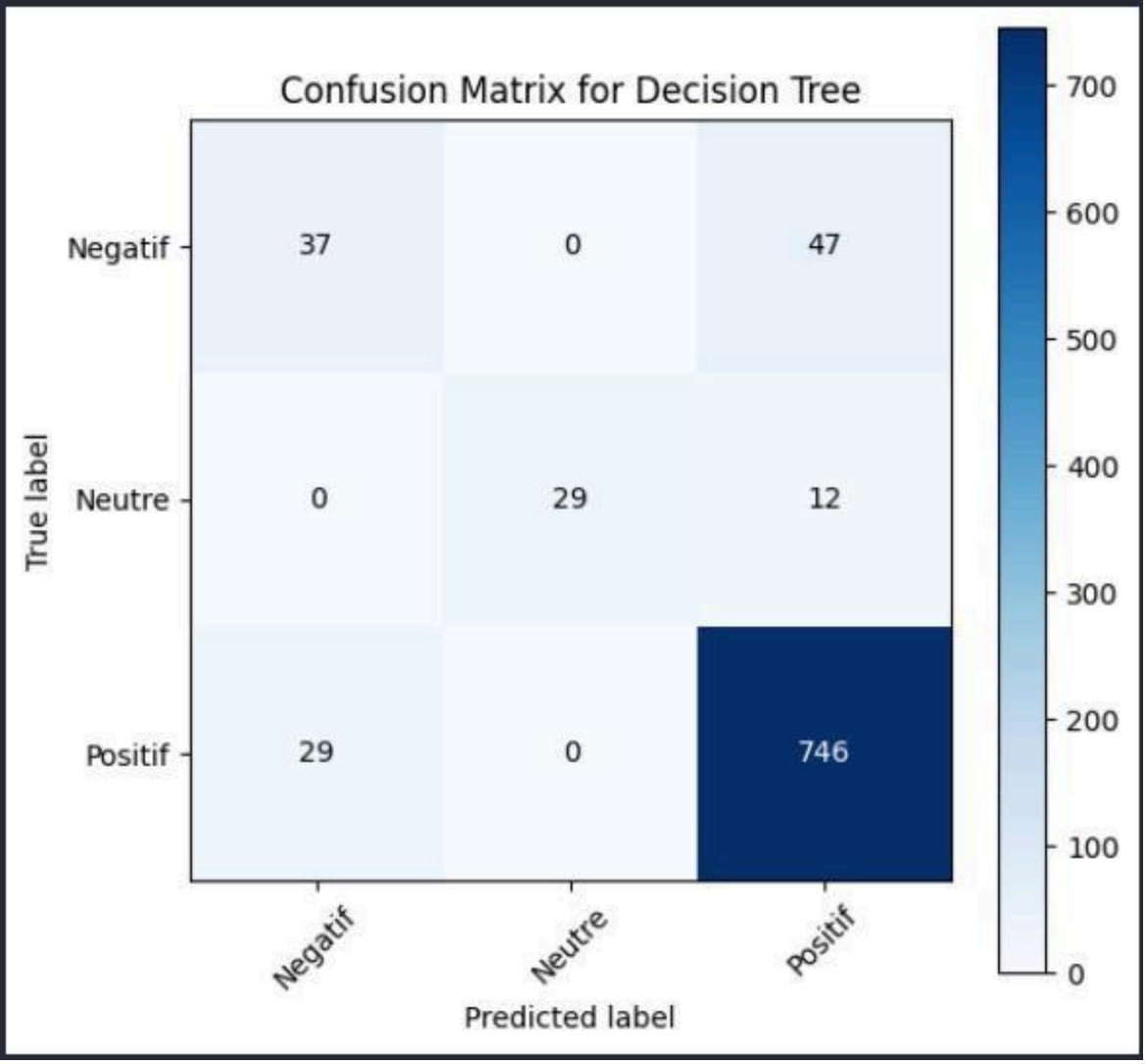

Figure 10: Confusion Matrix for Decision Tree

The above figure shows a classification report and a confusion matrix for the data tree model applied to the dataset.

Classification Report

The performance of the data tree model is measured across the three sentiment categories:

- **Precision:** The percentage of correct positive predictions that have been given for each label.
    - **Negative:** 0.56, this means 56% of the predicted negative labels were correct
    - **Neutral:** 1, this means 100% of the predicted neutral labels were correct
    - **Positive:** 0.93, this means 93% of the predicted positive labels were correct
- **Recall:** The percentage of actual positive instances that have been identified correctly.
    - **Negative:** 0.44, low recall, only 44% of actual negative instances were correctly identified
    - **Neutral:** 0.71, 71% recall, moderate accuracy in identifying neutral labels
    - **Positive:** 0.96, almost perfect recall,96% actual positive labels were correctly identified
- **F1-Score:** The harmonic mean of precision and recall.
    - **Negative:** 0.49, lower F1-score due to the low recall
    - **Neutral:** 0.83, strong performance
    - **Positive:** 0.94, strong performance, good balance between precision and recall
- **Support:** The number of actual instances of each class in the test data.
    - **Negative:** 84 samples.
    - **Neutral:** 41 samples.
    - **Positive:** 775 samples.
- **Overall Metrics:**
    - **Accuracy**: The overall accuracy of the model is 89% which means it correctly identified 89% of all instances.
    - **Macro Average**: The average precision, recall and F1-score which has been equally weighted for each call
        - **Precision**: 0.83
        - **Recall**: 0.70
        - **F1-Score**: 0.76

- **Weighted Average**: The average weighted by the number of instances in each class
  - **Precision**: 0.90
  - **Recall**: 0.90
  - **F1-score**: 0.90

Confusion Matrix

The confusion matrix shows how the data tree performed in terms of predicting each class. The graph depicts the actual vs predicted labels.

- **Negative**: A fair number of negative instances were correctly predicted as the diagonal for the class is a lighter colour, meaning some negative instances were misclassified.
- **Neutral**:A decent amount of neutral instances were correctly predicted as can be seen by a medium shading around this area
- **Positive**: Majority of the positive instances were correctly predicted, as can be seen by the dark colour, which also related to the high precision and recall for this specific class.

The data tree model performs very well for the positive class with both precision and recall being high and as can be seen from the confusion matrix, majority of the predictions were concentrated in this class. The data tree model also performs quite well in the neutral class, having a strong F-1 score and recall. However, the model struggles with the negative class with a lower recall of 0.44 and F1-score of 0.49. It is also clear to see the imbalance in the dataset as there are 775 positive instances, 84 negative instances and 41 neutral instances.

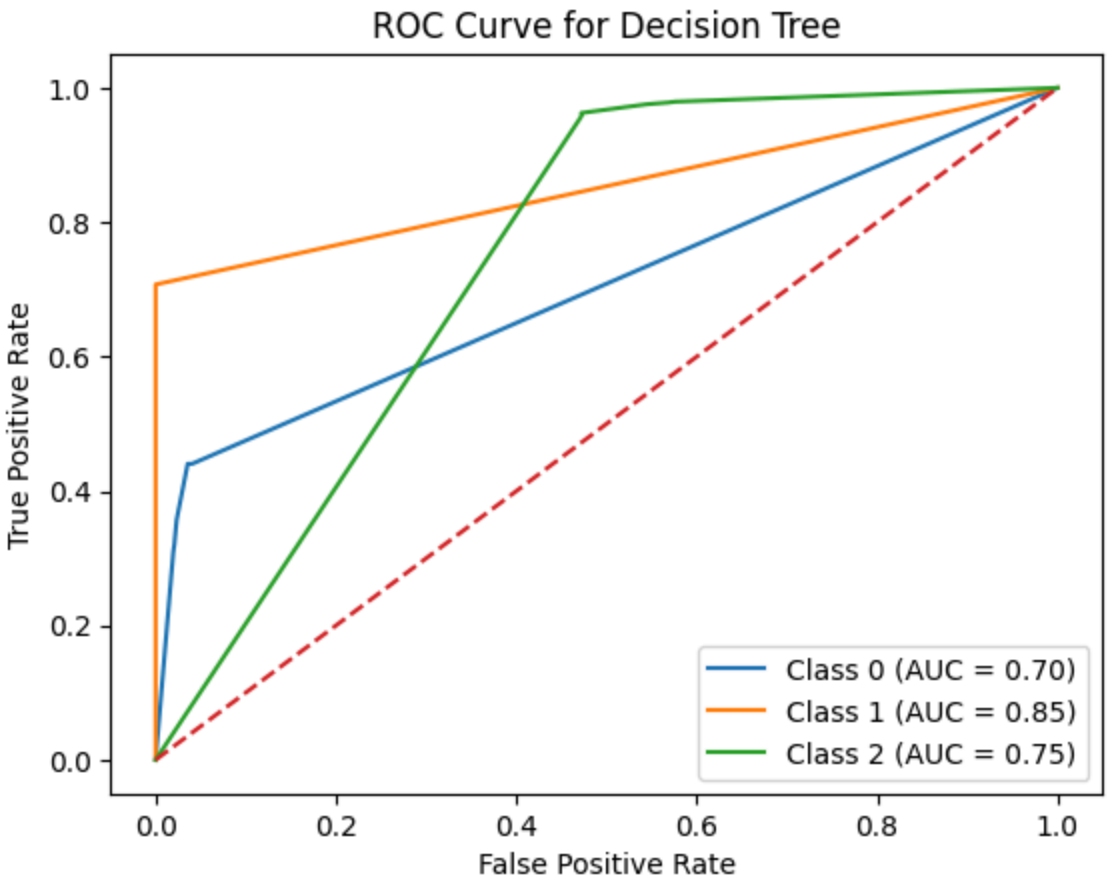

Figure 11: ROC Curve for Decision Tree

The image above shows the data tree model ROC curve for the three sentiment classes. The ROC curve can be used to evaluate the performance of a classification model by plotting the True Positive Rate (TPR) but the False Positive Rate (FPR). The area under the curve (AUC) shows the overall ability of the model to distinguish between the three sentiment classes.

- **Class 0 / Negative:** The blue line represents Class 0 or the Negative class. This class has an AUC score of 0.70 which is the lowest performing class compared to the others. That means this class is less effective at correctly identifying negative instances.
- **Class 1 / Neutral:** The orange line represents Class 1 or the Neutral class. This class has an AUC score of 0.85 which is the highest performing class compared to the others. That means this class is more effective at correctly identifying true positives and is also effective at keeping the false positives low.

- **Class 2/ Positive:** The green line represents Class 2 or the Positive class. This class has an AUC score of 0.75 which is the middle performing class. That means this class has a reliable balance and is effective at correctly identifying true positives and is also effective at keeping the false positives low.

Class 1 has the highest AUC at 0.85 indicating the model is very good at distinguishing neutral sentiment. Class 2 has the second highest AUC at 0.75 indicating it performs well at correctly classifying positive sentiment. Class 0 has the lowest AUC at 0.70 meaning the model struggled at correctly identifying negative sentiment which is consistent with the poor recall and precision for the negative class.

Classification Report for Random Forest:

	precision	recall	f1-score	support
Negatif	0.59	0.36	0.44	84
Neutre	1.00	0.68	0.81	41
Positif	0.92	0.97	0.94	775
accuracy			0.90	900
macro avg	0.84	0.67	0.73	900
weighted avg	0.89	0.90	0.89	900

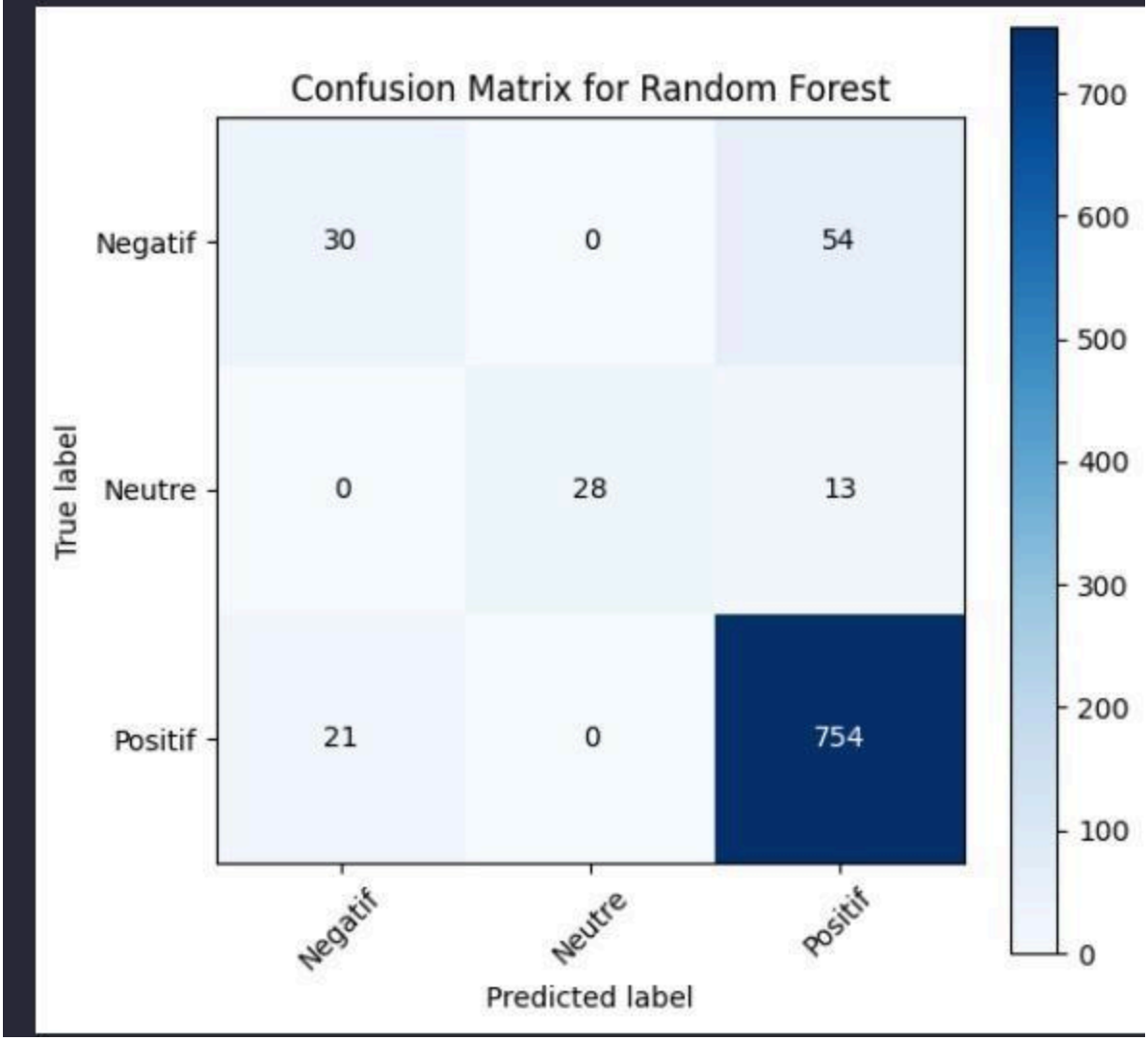

Figure 12: Confusion Matrix for Random Forest

The above figure shows a classification report and a confusion matrix for the random forest model applied to the dataset.

Classification Report

The performance of the random forest model is measured across the three sentiment categories:

- **Precision:** The percentage of correct positive predictions that have been given for each label.
    - **Negative:** 0.61, this means 61% of the predicted negative labels were correct
    - **Neutral:** 1, this means 100% of the predicted neutral labels were correct
    - **Positive:** 0.92, this means 92% of the predicted positive labels were correct
- **Recall:** The percentage of actual positive instances that have been identified correctly.
    - **Negative:** 0.36, low recall, only 36% of actual negative instances were correctly identified
    - **Neutral:** 0.68, 68% recall, moderate accuracy in identifying neutral labels
    - **Positive:** 0.98, almost perfect recall, 98% actual positive labels were correctly identified
- **F1-Score:** The harmonic mean of precision and recall.
    - **Negative:** 0.45, moderate F1-score due to the low recall
    - **Neutral:** 0.81, strong performance
    - **Positive:** 0.95, strong performance, good balance between precision and recall
- **Support:** The number of actual instances of each class in the test data.
    - **Negative:** 84 samples.
    - **Neutral:** 41 samples.
    - **Positive:** 775 samples.
- **Overall Metrics:**
    - **Accuracy**: The overall accuracy of the model is 90% which means it correctly identified 90% of all instances.
    - **Macro Average**: The average precision, recall and F1-score which has been equally weighted for each call
        - **Precision**: 0.84
        - **Recall**: 0.67
        - **F1-Score**: 0.74

- **Weighted Average**: The average weighted by the number of instances in each class
  - **Precision**: 0.89
  - **Recall**: 0.90
  - **F1-score**: 0.89

Confusion Matrix

The confusion matrix shows how the data tree performed in terms of predicting each class. The graph depicts the actual vs predicted labels.

- **Negative**: A fair number of negative instances were correctly predicted as the diagonal for the class is a lighter colour, meaning some negative instances were misclassified.
- **Neutral**:A decent amount of neutral instances were correctly predicted as can be seen by a medium shading around this area
- **Positive**: Majority of the positive instances were correctly predicted, as can be seen by the dark colour, which also related to the high precision and recall for this specific class.

The random forest model performs very well for the positive class with both precision and recall being high and as can be seen from the confusion matrix, majority of the predictions were concentrated in this class. The random forest model also performs quite well in the neutral class, having a strong F-1 score and recall. However, the model struggles with the negative class with a lower recall of 0.36 and F1-score of 0.45. It is also clear to see the imbalance in the dataset as there are 775 positive instances, 84 negative instances and 41 neutral instances.

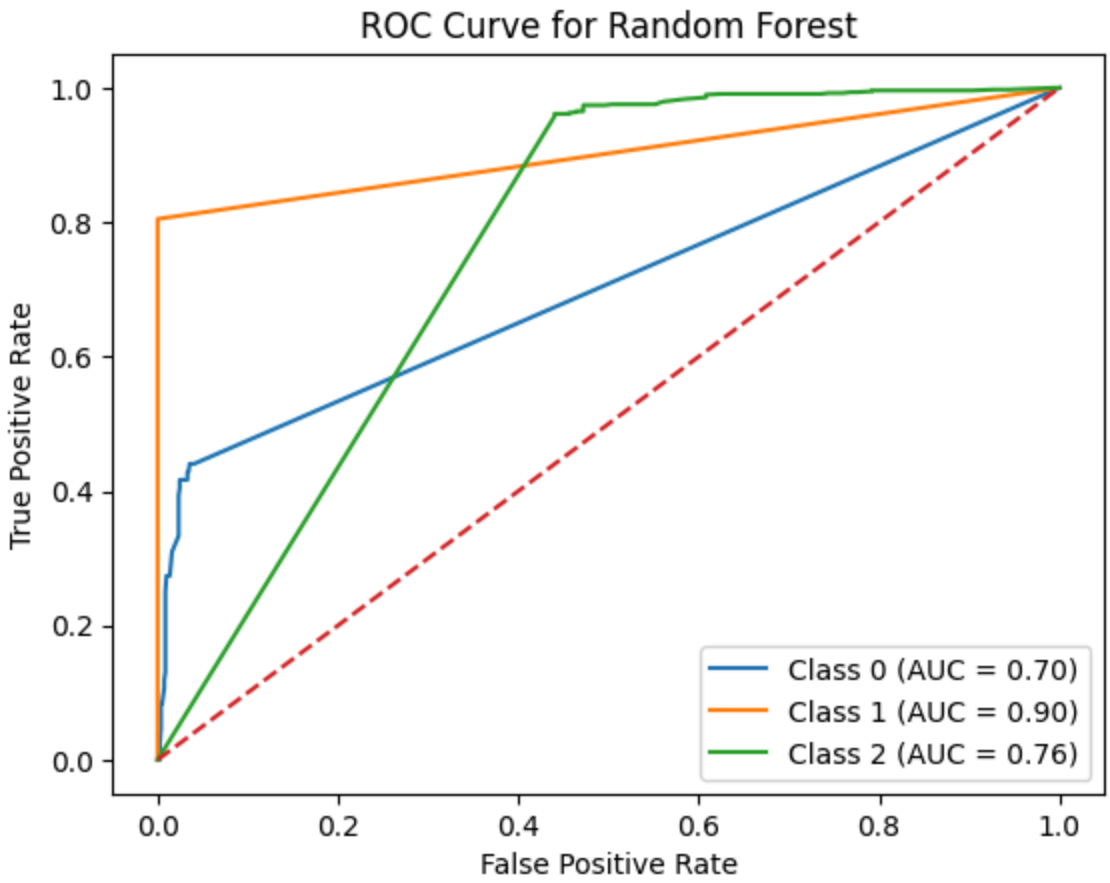

Figure 13: ROC Curve for Random Forest

The image above shows the data tree model ROC curve for the three sentiment classes. The ROC curve can be used to evaluate the performance of a classification model by plotting the True Positive Rate (TPR) but the False Positive Rate (FPR). The area under the curve (AUC) shows the overall ability of the model to distinguish between the three sentiment classes.

- **Class 0 / Negative:** The blue line represents Class 0 or the Negative class. This class has an AUC score of 0.70 which is the lowest performing class compared to the others. That means this class is less effective at correctly identifying negative instances.
- **Class 1 / Neutral:** The orange line represents Class 1 or the Neutral class. This class has an AUC score of 0.90 which is the highest performing class compared to the others. That means this class is more effective at correctly identifying true positives and is also effective at keeping the false positives low.

- **Class 2/ Positive:** The green line represents Class 2 or the Positive class. This class has an AUC score of 0.76 which is the middle performing class. That means this class has a reliable balance and is effective at correctly identifying true positives and is also effective at keeping the false positives low.

Class 1 has the highest AUC at 0.90 indicating the model is very good at distinguishing neutral sentiment. Class 2 has the second highest AUC at 0.76 indicating it performs well at correctly classifying positive sentiment. Class 0 has the lowest AUC at 0.70 meaning the model struggled at correctly identifying negative sentiment which is consistent with the poor recall and precision for the negative class.

Classification Report for Support Vector Machine:

	precision	recall	f1-score	support
Negatif	0.68	0.18	0.28	84
Neutre	1.00	0.68	0.81	41
Positif	0.90	0.99	0.95	775
accuracy			0.90	900
macro avg	0.86	0.62	0.68	900
weighted avg	0.89	0.90	0.88	900

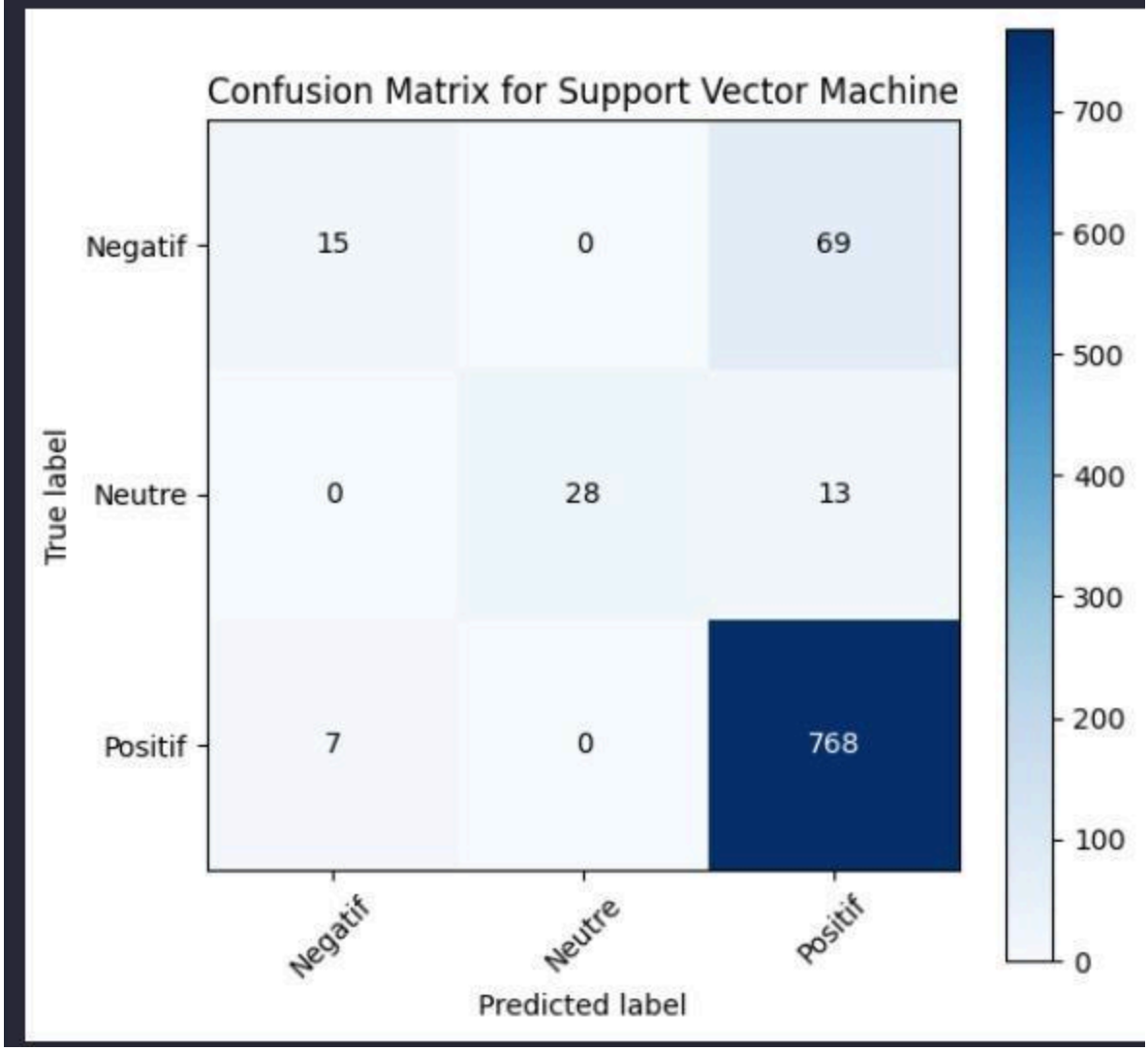

Figure 14: Confusion Matrix for SVM

The above figure shows a classification report and a confusion matrix for the support vector machine model applied to the dataset.

Classification Report

The performance of the support vector machine model is measured across the three sentiment categories:

- **Precision:** The percentage of correct positive predictions that have been given for each label.
    - **Negative:** 0.68, this means 68% of the predicted negative labels were correct
    - **Neutral:** 1, this means 100% of the predicted neutral labels were correct
    - **Positive:** 0.90, this means 90% of the predicted positive labels were correct
- **Recall:** The percentage of actual positive instances that have been identified correctly.
    - **Negative:** 0.18, very low recall, only 1/% of actual negative instances were correctly identified
    - **Neutral:** 0.68, 68% recall, moderate accuracy in identifying neutral labels
    - **Positive:** 0.99, almost perfect recall, 99% actual positive labels were correctly identified
- **F1-Score:** The harmonic mean of precision and recall.
    - **Negative:** 0.28, low F1-score due to the low recall
    - **Neutral:** 0.81, strong performance
    - **Positive:** 0.95, strong performance, good balance between precision and recall
- **Support:** The number of actual instances of each class in the test data.
    - **Negative:** 84 samples.
    - **Neutral:** 41 samples.
    - **Positive:** 775 samples.
- **Overall Metrics:**
    - **Accuracy**: The overall accuracy of the model is 90% which means it correctly identified 90% of all instances.
    - **Macro Average**: The average precision, recall and F1-score which has been equally weighted for each call
        - **Precision**: 0.86

        - **Recall**: 0.62
        - **F1-Score**: 0.68
    - **Weighted Average**: The average weighted by the number of instances in each class
        - **Precision**: 0.90
        - **Recall**: 0.90
        - **F1-score**: 0.88

Confusion Matrix

The confusion matrix shows how the data tree performed in terms of predicting each class. The graph depicts the actual vs predicted labels.

- **Negative**: A fair number of negative instances were correctly predicted as the diagonal for the class is a lighter colour, meaning some negative instances were misclassified.
- **Neutral**: A decent amount of neutral instances were correctly predicted as can be seen by a medium shading around this area
- **Positive**: Majority of the positive instances were correctly predicted, as can be seen by the dark colour, which also related to the high precision and recall for this specific class.

The support vector machine model performs very well for the positive class with both precision and recall being high and as can be seen from the confusion matrix, majority of the predictions were concentrated in this class. The support vector machine model also performs quite well in the neutral class, having a strong F-1 score and recall. However, the model struggles with the negative class with a lower recall of 0.18 and F1-score of 0.28. It is also clear to see the imbalance in the dataset as there are 775 positive instances, 84 negative instances and 41 neutral instances.

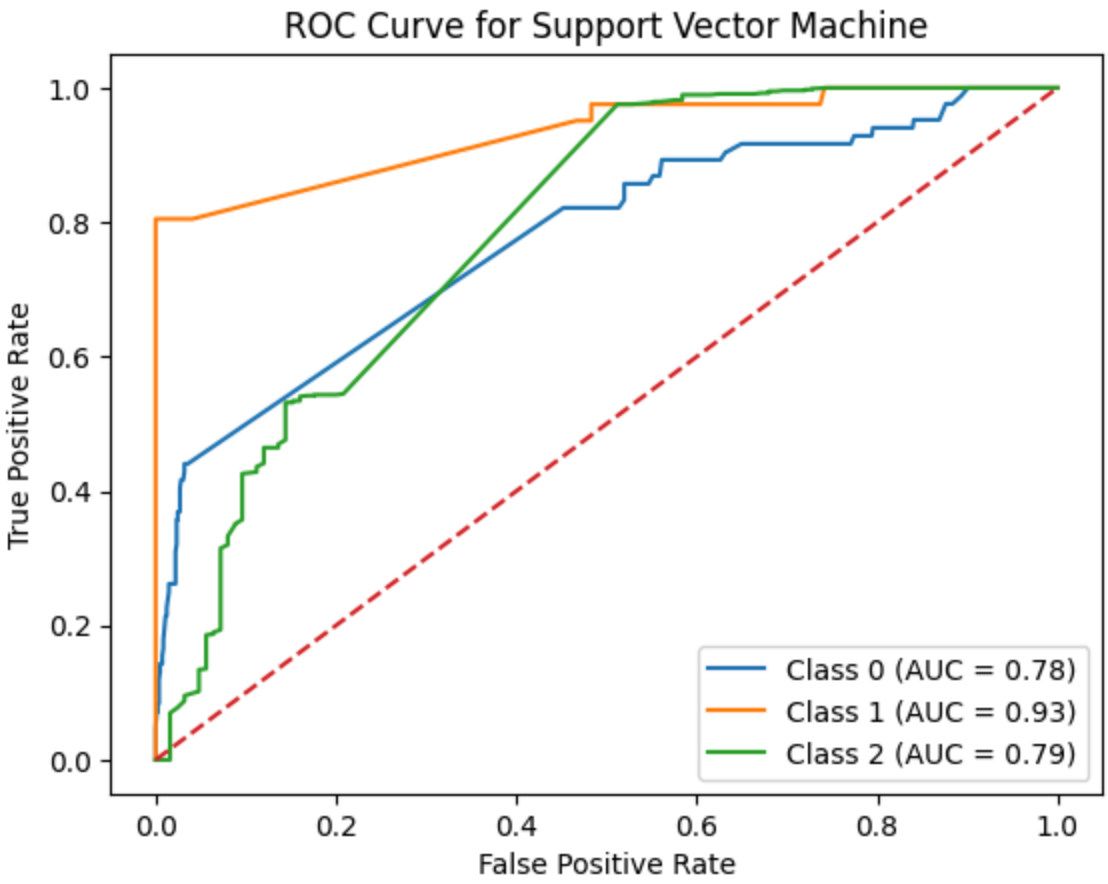

Figure 15: ROC Curve for SVM

The image above shows the support vector machine model ROC curve for the three sentiment classes. The ROC curve can be used to evaluate the performance of a classification model by plotting the True Positive Rate (TPR) but the False Positive Rate (FPR). The area under the curve (AUC) shows the overall ability of the model to distinguish between the three sentiment classes.

- **Class 0 / Negative:** The blue line represents Class 0 or the Negative class. This class has an AUC score of 0.78 which is the lowest performing class compared to the others. That means this class is less effective at correctly identifying negative instances.
- **Class 1 / Neutral:** The orange line represents Class 1 or the Neutral class. This class has an AUC score of 0.93 which is the highest performing class compared to the others. That means this class is more effective at correctly identifying true positives and is also effective at keeping the false positives low.

- **Class 2/ Positive:** The green line represents Class 2 or the Positive class. This class has an AUC score of 0.79 which is the middle performing class. That means this class has a reliable balance and is effective at correctly identifying true positives and is also effective at keeping the false positives low.

Class 1 has the highest AUC at 0.90 indicating the model is very good at distinguishing neutral sentiment. Class 2 has the second highest AUC at 0.93 indicating it performs well at correctly classifying positive sentiment. Class 0 has the lowest AUC at 0.79 meaning the model struggled slightly at correctly identifying negative sentiment which is consistent with the lower recall and precision for the negative class.

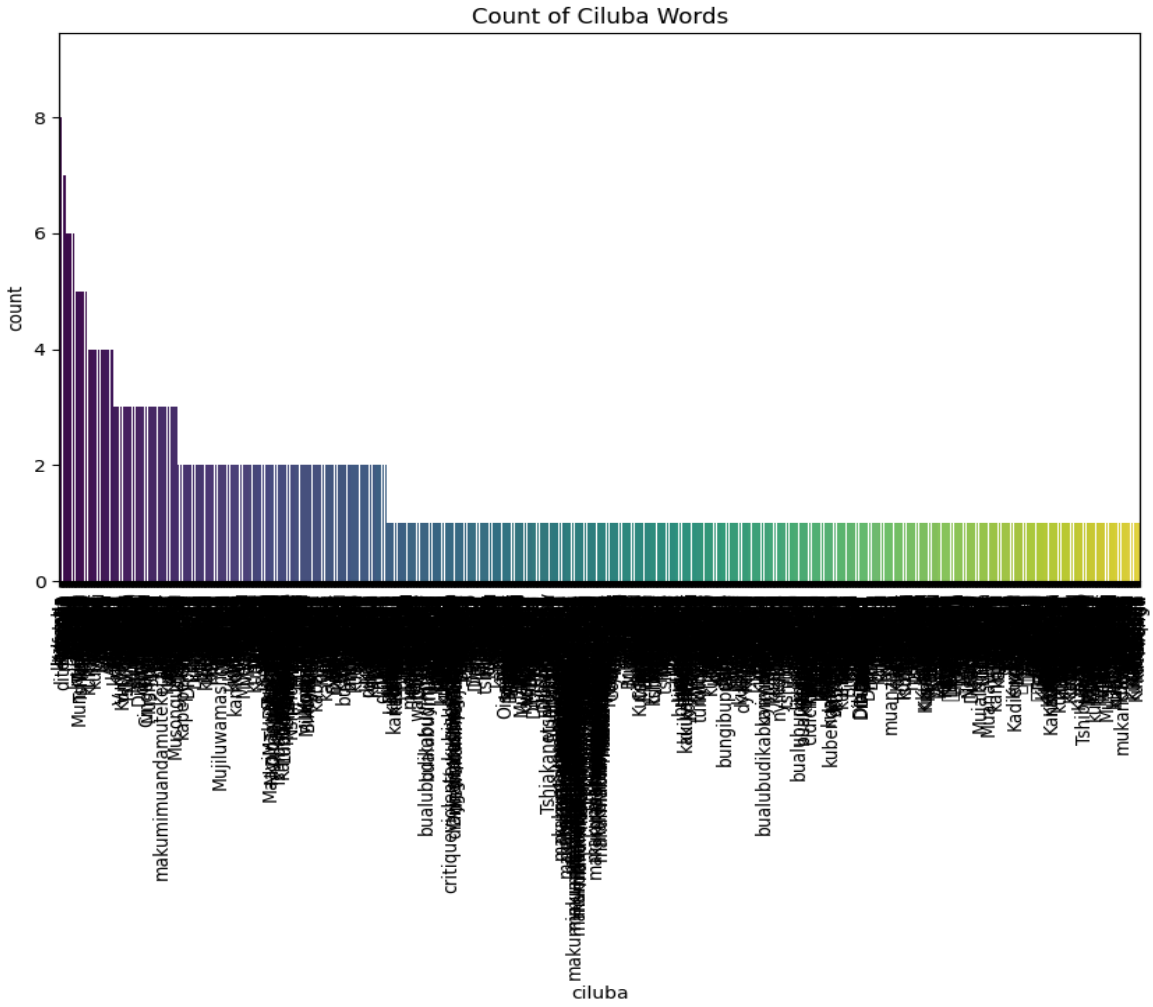

Figure 16: Count for Ciluba Words

The image above shows the amount of Ciluba words in the dataset and the frequency of these words in the dataset. From this graph we can see that some words appear frequently (up to 7 times), indicating repetition whereas most words only appear once.

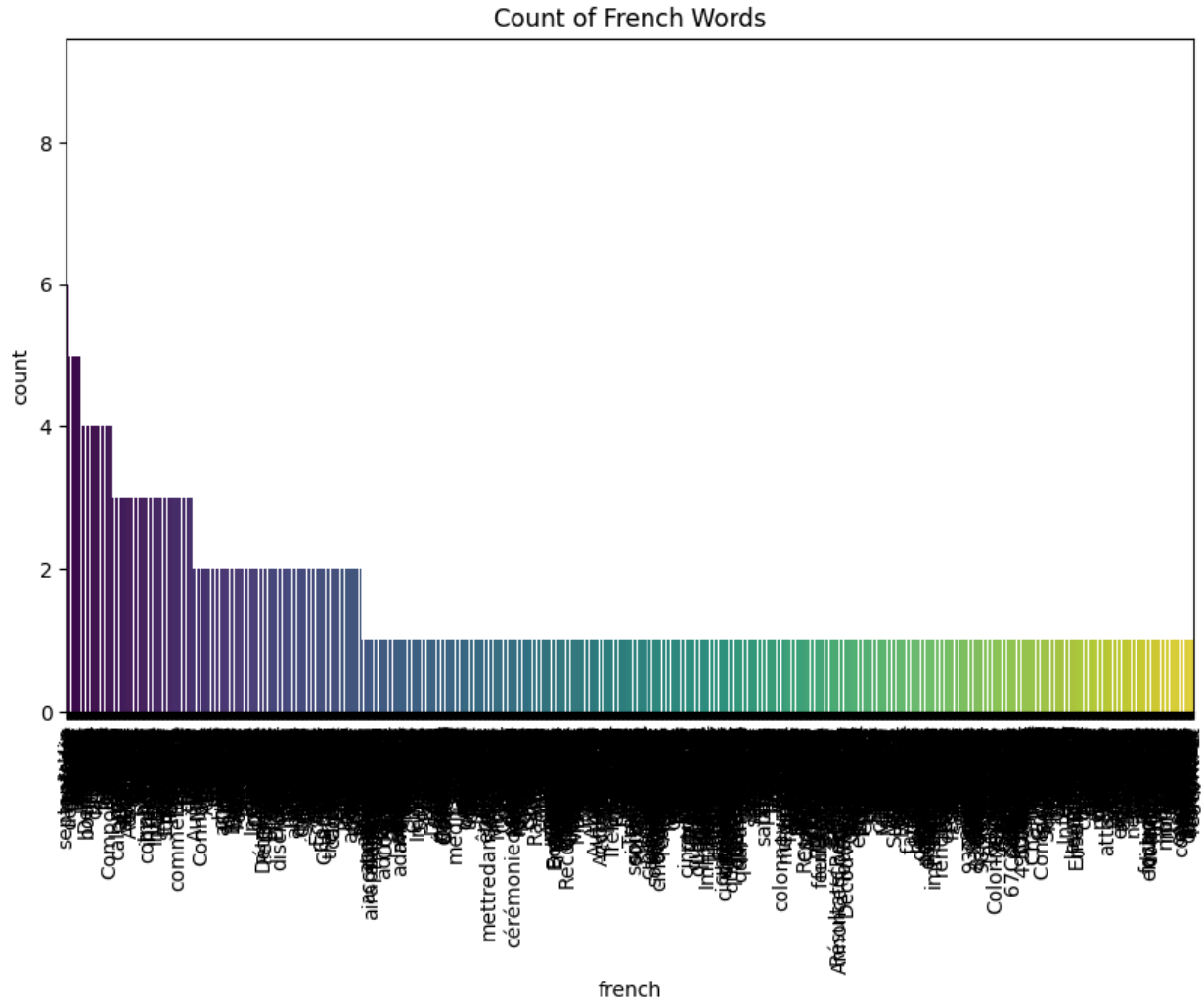

Figure 17: Count for French words

The image above shows the number of French words in the dataset and the frequency of these words in the dataset. From this graph we can see that some words appear frequently (up to 5 times), indicating repetition whereas most words only appear once.

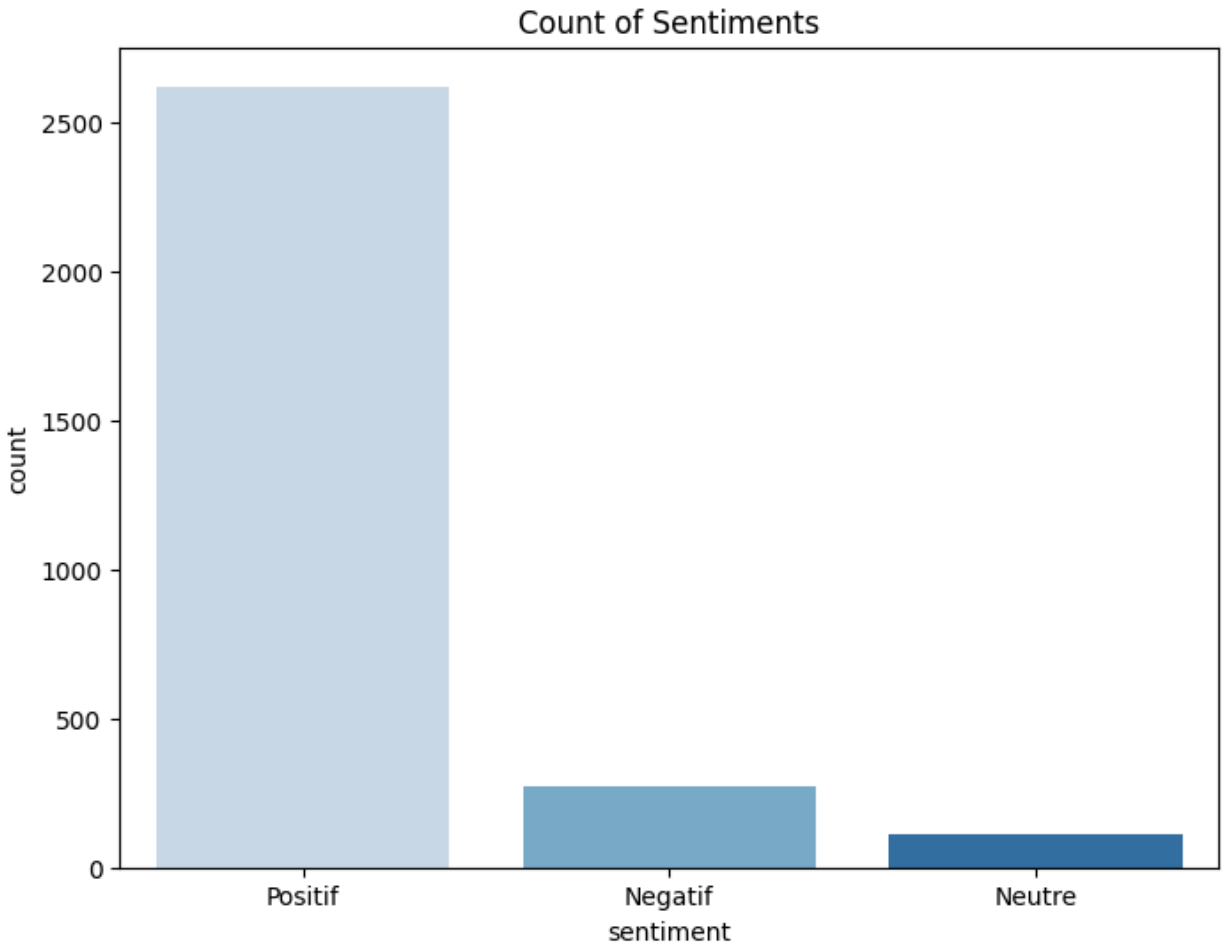

Figure 18: Count of sentiment

The image above shows the count of how often the sentiment labels appear in the dataset. From this graph we can see that the majority of the words have a positive sentiment, followed by the second most common being negative and lastly the least frequent is neutral.

There is a significant imbalance in the sentiment within the dataset which could impact how well the machine learning models perform. This could cause the machine learning models to perform better for the positive class and struggle with the negative and neutral class.

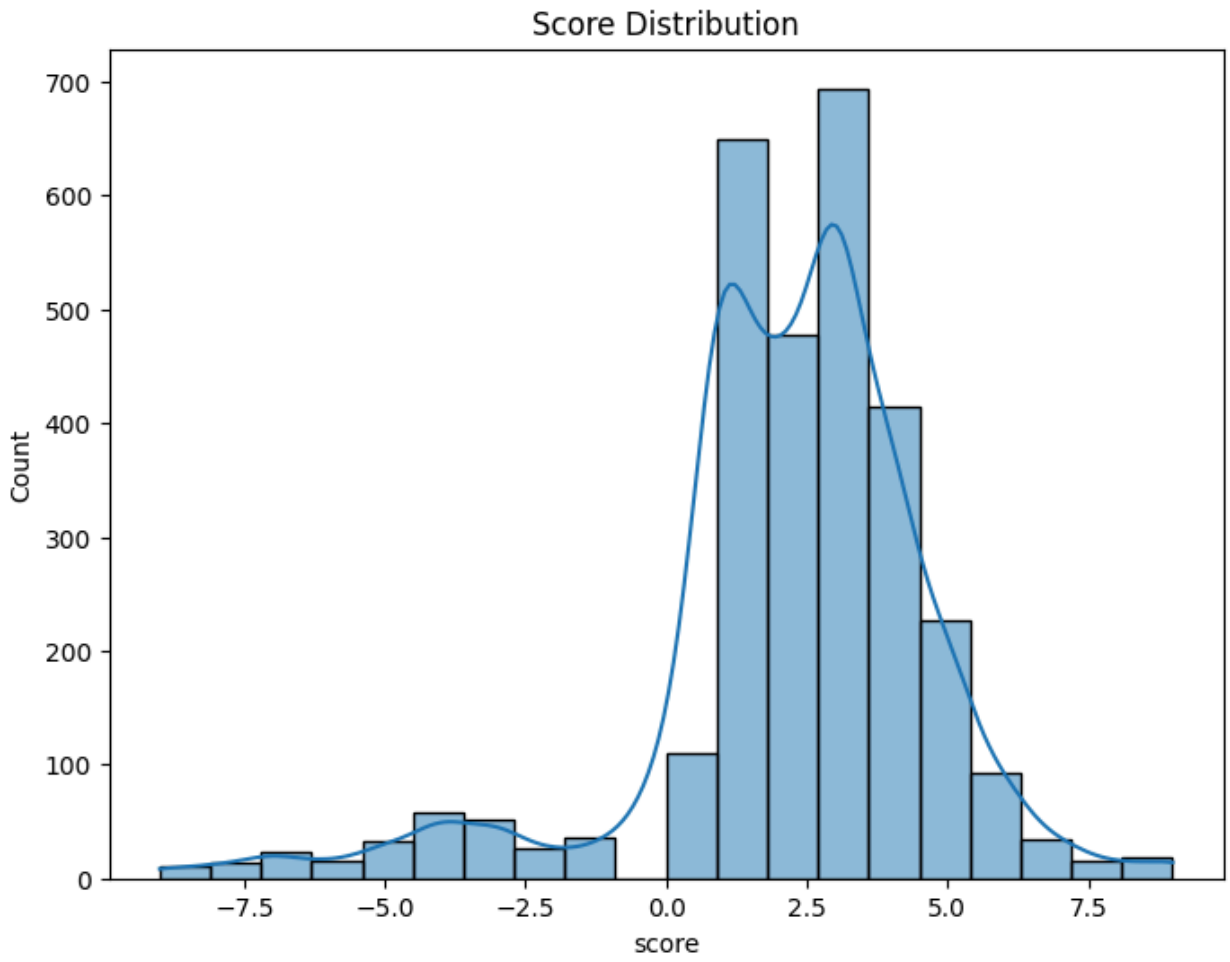

Figure 19: Distribution of Sentiment Scores

The graph above shows the distribution of sentiment scores within the dataset. The distribution is bimodal, meaning there are two peaks in data. The first peak is the largest peak near the score range of 2.0 to 2.5 where the count is 650 to 700. This suggests that there are many words in the dataset that have positive scores in this range. The second peak is around 4.0 to 4.5 which also has relatively high scores. There is a small cluster of negative scores to the left of 0 however this occurs less frequently than the positive scores. The overall distribution is positively skewed. The chart indicates a strong bias towards positive scores, which aligns with the higher count of positive sentiments in the dataset. The less frequent negative scores could also cause challenges for the machine learning models to accurately predict negative sentiment.

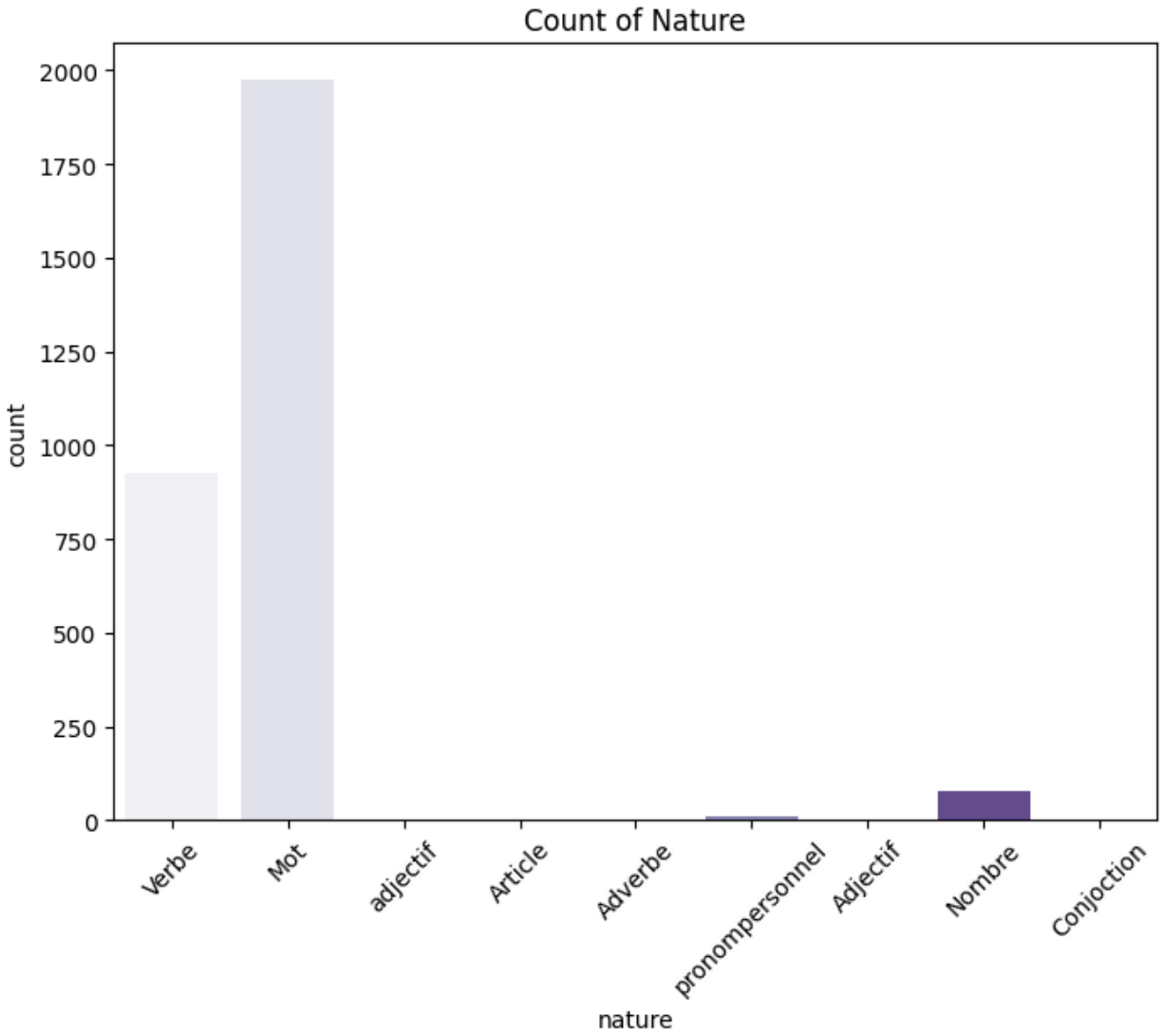

Figure 20: Count of Word Nature

The image above represents a bar chart showing the frequency of words of different nature in the dataset. The most frequent grammatical category is Mot which means word with nearly 2000 occurrences. The second most frequent is Verb which means verbs with a count of around 900 and the third most common is Nombre which means number. The chart indicates that there is a high frequency of words and verbs, while other grammatical categories like adjectives, numbers, and conjunctions are much less frequent.

```
Select the language you want to translate to:
1. Afrikaans
2. Zulu
3. Xhosa
c:\Python312\Lib\site-packages\urllib3\connectionpool.py:109
  warnings.warn(

Original sentence: I want to dance
Sentiment (Original): Neutral (Score: 0)
Translated into Afrikaans: Ek wil dans.
Sentiment (Translated): Neutral (Score: 0)
```

Figure 21: Language Translation Prompts

The image above describes how the language translation tool our team has built works with the machine learning model integration. Firstly the user is prompted to select a language being either Afrikaans, Zulu or Xhosa. The tool will then translate the original sentence into the language that has been chosen by the user. The tool will then analyse the sentence and classify the sentiment of the sentence, which in this case is neutral with a sentiment score of 0. After the translation, the tool will then run another sentiment analysis on the output which is also classified neutral with a score of 0.

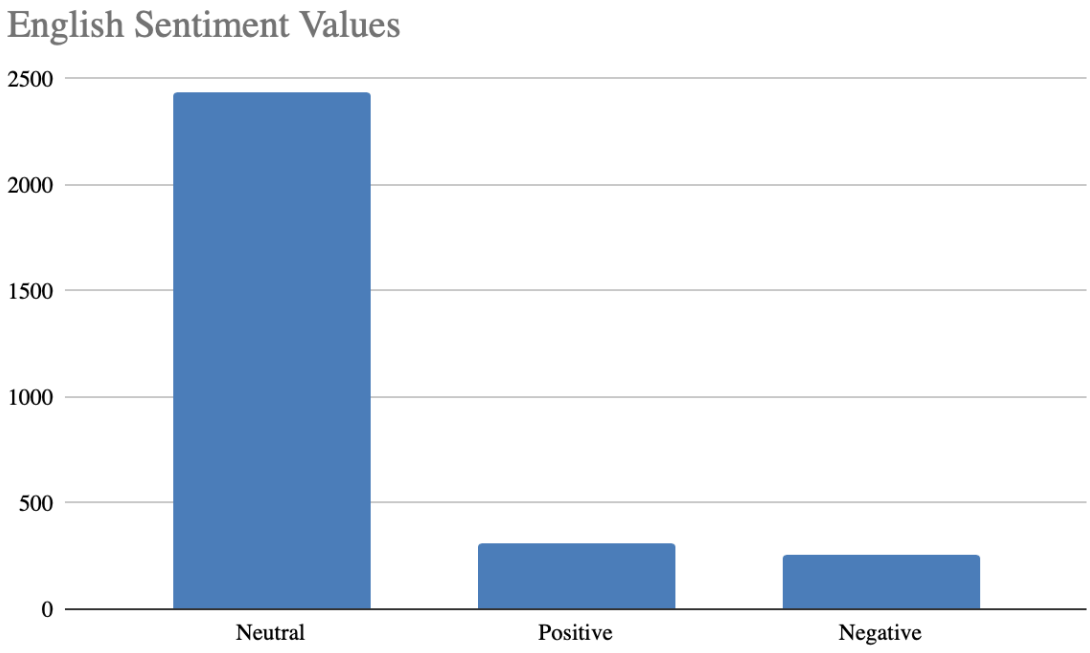

Figure 22: English Sentiment Values

The bar chart shows the distribution of sentiment values for English. The neutral category is the most frequent with a count of around 2500. The positive and negative sentiments each represent a significantly lower value at around 500. The distribution reveals an imbalance in sentiment, this could lead to the model classifying neutral values easily but struggling to classify positive and negative sentiments.

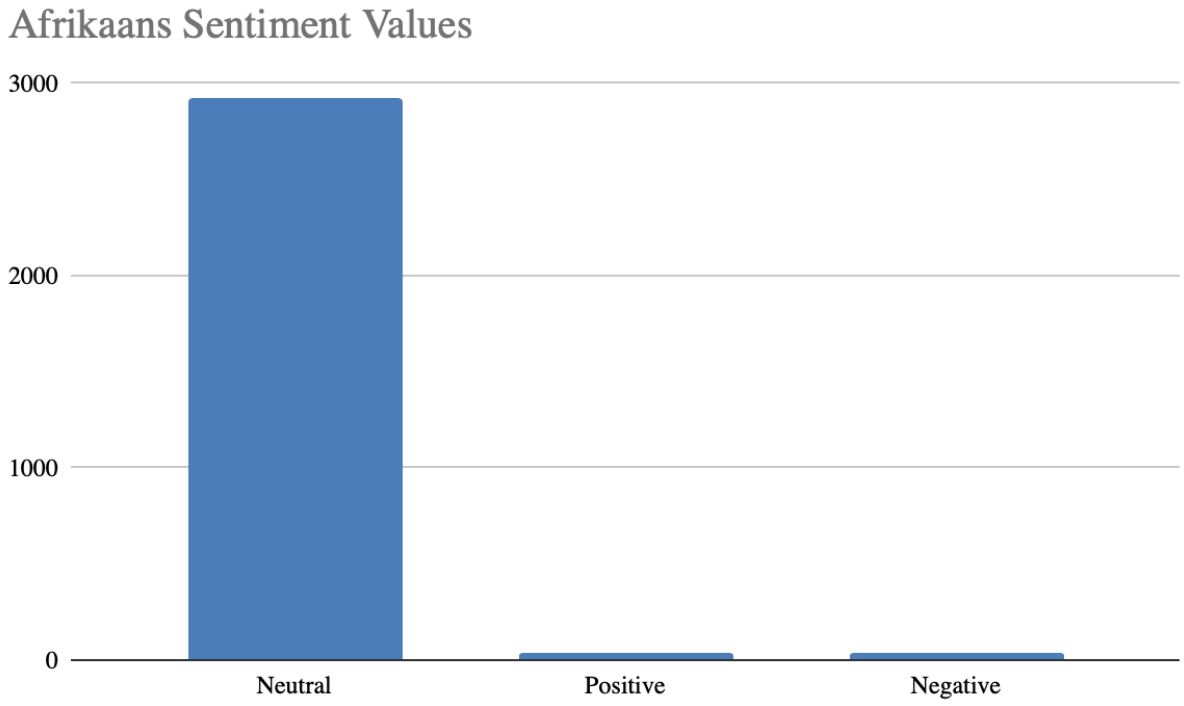

Figure 23: Afrikaans Sentiment Values

The bar chart shows the distribution of sentiment values for Afrikaans. The neutral category is the most frequent with a count of around 3000. The positive and negative sentiments each represent a significantly lower value close to 0. The distribution reveals an imbalance in sentiment, this could lead to the model classifying neutral values easily but struggling to classify positive and negative sentiments.

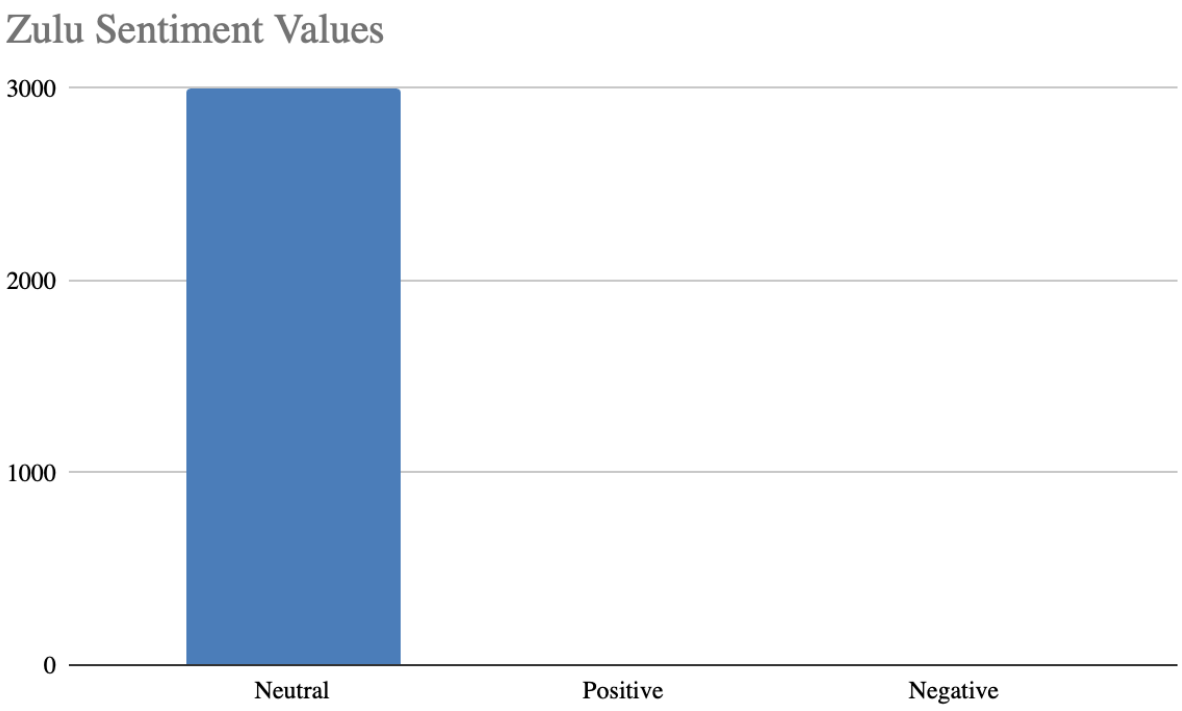

Figure 24: Zulu Sentiment Values

The bar chart shows the distribution of sentiment values for Zulu. The neutral category is the most frequent with a count of around 3000. The positive and negative sentiments each represent close to no values. The distribution reveals an imbalance in sentiment, this could lead to the model classifying neutral values easily but struggling to classify positive and negative sentiments.

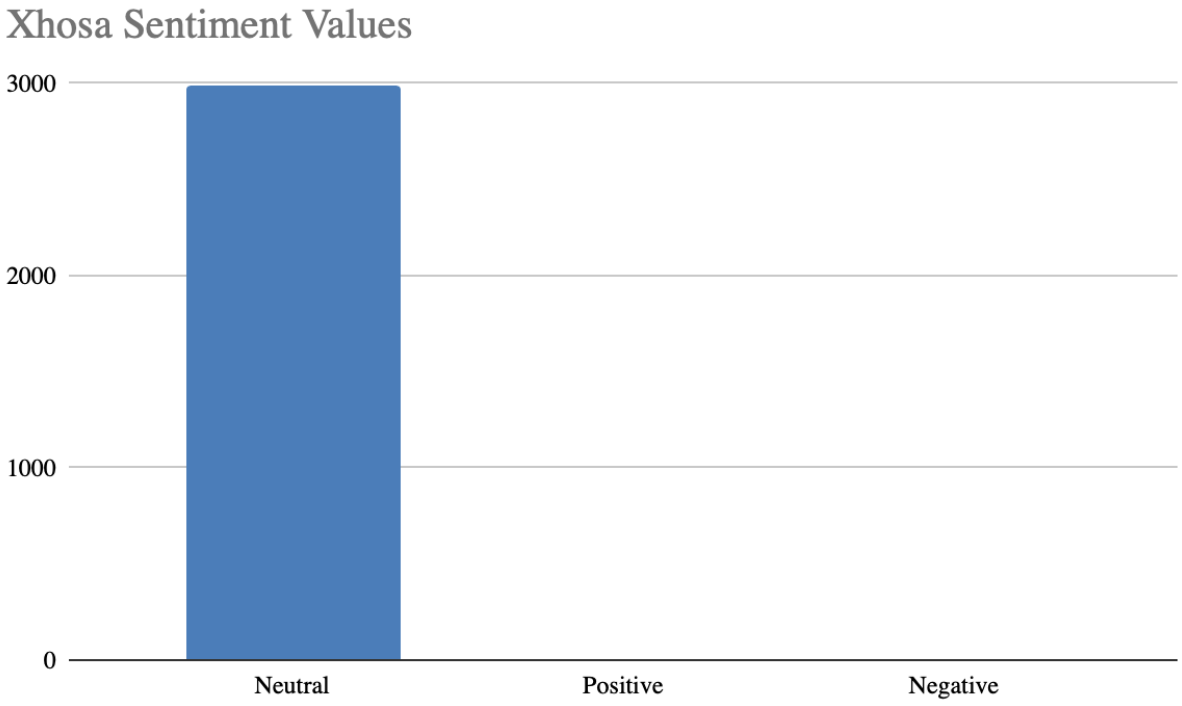

Figure 25: Zulu Sentiment Values

The bar chart shows the distribution of sentiment values for Xhosa. The neutral category is the most frequent with a count of around 3000. The positive and negative sentiments each represent close to no values. The distribution reveals an imbalance in sentiment, this could lead to the model classifying neutral values easily but struggling to classify positive and negative sentiments.

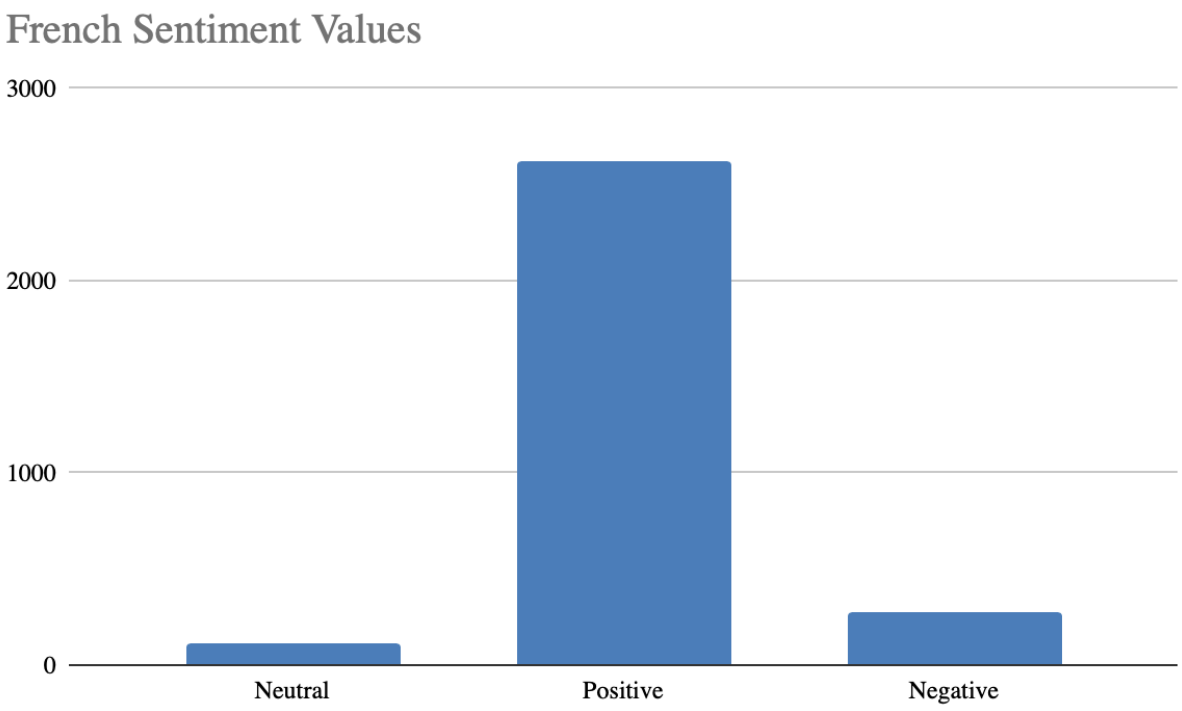

Figure 26: Zulu Sentiment Values

The bar chart shows the distribution of sentiment values for French. The positive category is the most frequent with a count of around 2500. The neutral and negative sentiments each represent a low value however negative has slightly more than neutral. The distribution reveals an imbalance in sentiment, this could lead to the model classifying positive values easily but struggling to classify neutral and negative sentiments.

Sentence translation and input translation

```
Select the language you want to translate to:
1. Afrikaans
2. Zulu
3. Xhosa

Translated 5 sentences into Zulu:
                                    Original English   Original Sentiment  \
0   On the top of the hill, the stars began to fade.  Positive (Score: 4)
1  In the middle of the forest, the moonlight ill...   Neutral (Score: 0)
2  She was reading her favorite book when a myste...  Positive (Score: 2)
3  As the music played softly, an unexpected visi...   Neutral (Score: 0)
4  He opened the door to find they exchanged stor...   Neutral (Score: 0)

                                          Translated Translated Sentiment
0  Esiqongweni segquma, izinkanyezi zaqala ukufip...   Neutral (Score: 0)
1  Maphakathi nehlathi, ukukhanya kwenyanga kwakh...   Neutral (Score: 0)
2  Wayefunda incwadi ayithandayo lapho kuzwakala ...   Neutral (Score: 0)
3  Njengoba umculo wawudlala kancane, isivakashi ...   Neutral (Score: 0)
4  Wavula umnyango ukuze athole ukuthi baxoxisana...   Neutral (Score: 0)
```

Figure 27: Sentence translation results

Explanation of code for this results:

sentence translation

Imports for Translation

1. import pandas as pd: This is used to handle data structures, particularly DataFrames, which are used to store and process the sentences.
2. import time: This helps manage retries for the translation process by allowing delays between retries if there are errors.
3. from deep_translator import MyMemoryTranslator: The MyMemoryTranslator class from the deep_translator library provides translation functionality, allowing for translation between specified languages.

Translation Steps

1. Define the translate_text function: This function accepts a text input and attempts to translate it using MyMemoryTranslator. It includes:

- Error Handling: If an error occurs (e.g., due to network issues), the function retries up to three times, with a 2-second delay between each attempt.
- Parameters: Takes the original text, the source_language, and target_language for translation, along with the retry count.
- Translation Process: The function instantiates the MyMemoryTranslator with the specified languages and returns the translated text.

2. Implement translate_sentences: This function handles translation for multiple sentences, where:
   - A loop iterates through a subset of sentences based on the user's input.
   - For each sentence, translate_text is called to translate the English sentence into the selected target language.
3. Language Selection in interactive_translation: This function prompts the user to select a language and number of sentences for translation. Based on user input, it calls translate_sentences to apply translation to the specified sentences.

Input translation (Any word or sentence can be used as input for translation)

```
Select the language you want to translate to:
1. Afrikaans
2. Zulu
3. Xhosa
c:\Python312\Lib\site-packages\urllib3\connectionpool.py:1099: Insecu
  warnings.warn(

Original sentence: This was the best module of the year
Sentiment (Original): Positive (Score: 9)
Translated into Zulu: Lena bekuyimodyuli ehamba phambili yonyaka
Sentiment (Translated): Neutral (Score: 0)
```

Figure 28: Input translation results

Imports for Translation

1. import requests: Used to handle HTTP requests. Here, it's specifically modified to bypass SSL verification during translation.
2. from deep_translator import MyMemoryTranslator: This class from deep_translator facilitates translation between the specified source and target languages.
3. from textblob import TextBlob: Enables sentiment analysis on both the original and translated sentences.

Translation Steps

1. translate_text function:
    - SSL Bypass Setup: Overrides requests.get to bypass SSL verification. The verify parameter is set to False, which allows connections even if SSL verification fails.
    - Translation Process: Instantiates MyMemoryTranslator with the specified source and target languages and performs the translation. After the translation, the original requests.get function is restored.
    - Retries: If translation fails, the function retries up to three times.
2. interactive_translation function:
    - Language Selection: Defines supported languages (Afrikaans, Zulu, Xhosa) and prompts the user to select a target language.
    - User Input: Prompts the user to input a sentence in English for translation.
    - Translation: Uses translate_text to translate the input sentence into the chosen target language.
3. Sentiment Analysis:
    - analyze_sentiment function: Performs sentiment analysis on both the original and translated sentences using TextBlob's polarity scoring, scaling it from -9 to +9 and labeling sentiment as Positive, Negative, or Neutral.

6. Discussion

Four machine learning models were utilised in the study namely, logistic regression, decision tree, random forest, and support vector machine (SVM). Each model demonstrated strength in being able to identify positive sentiment which is evident by the high precision and recall for this class. However, the models struggled to predict negative and neutral sentiment, this is most likely due to the fact that the dataset contained more data points that were labelled as positive, compared to negative and neutral, creating a class imbalance. The imbalance led to precision and recall being lower for negative prediction across all the models, underscoring the limitations in accurately predicting minority classes.

The logistic regression model was able to predict positive sentiment with 89% accuracy having high precision and recall. It struggled to predict negative sentiment having low recall of 0.08 meaning that it could only identify negative sentiment correctly 8% of the time. The low performance is shown in the confusion matrix, where a high number of negative predictions are misclassified.

The decision tree model performed well in predicting positive and neutral sentiment and was able to maintain a balanced precision and recall score, particularly for positive sentiment. Similar to logistic regression it struggled to accurately predict negative sentiment, with a low recall of 0.44 and a lower F1 score of 0.49. The imbalance in the dataset between positive, neutral, and negative contributed to the difficulty in accurately predicting negative sentiments.

The random forest model achieved a 90% accuracy rate with high precision and recall for positive and neutral classes. In a similar trend the random forest model had a low recall of 36% for negative sentiment. The confusion matrix again reveals many misclassifications related to negative sentiment, which further shows the models struggle to predict with minority classes.

SVM excelled in predicting positive sentiment, and had high precision and recall, and performed well with neutral sentiment. However, it had low recall for negative sentiment of 0.18, showing difficulty in identifying negative sentiment.

Across all models the main challenge was the class imbalance, specifically the over representation of positive sentiment and low level of representation for negative sentiment. The imbalance led to bias toward the positive class, which caused each model to struggle with accurate classification of negative, and to a lesser extent, neutral sentiment. The results suggest that adjustments should be made to the models, this could include resampling techniques or cost-sensitive learning to improve the models performance to handle underrepresented classes.
The dataset was heavily skewed towards positive sentiment, having 775 positive words, 41 neutral words, and 84 negative words. The skew towards positive sentiment impacts the generalisability of the models

7. Conclusion

In conclusion, this study showcased the transformative potential of AI in performing sentiment analysis and accurate translations across different languages. All four machine learning models (Logistic Regression, Decision Tree, Random Forest, and SVM) identified positive sentiment analysis strongly, with high accuracy and recall metrics. However, limitations were experienced when trying to predict negative and neutral sentiments due to a lack of representation of these scores. This showcased a class imbalance and a challenge in accurately predicting minority classes.

The results of the model performance in sentiment analysis highlight the importance of a well-represented dataset to ensure accurate model performance across all three levels of sentiment identification (positive, neutral and negative). This limitation could be mitigated with a dataset that equally populates positive, neutral and negative sentiment scores to increase performance in prediction across all sentiment classes.

This study identified the potential of AI-powered language applications, specifically in areas like South Africa where there are several underrepresented languages. It highlights the importance of improving sentiment analysis across a variety of languages. Future research calls for a deeper dive into how these tools can improve translations and sentiment analysis to accurately capture

diverse cultural nuances. More advanced models can be explored across more languages to further improve sentiment identification and inclusion.

8. Future Research

The current study lends itself to further research avenues within the multilingual sentiment analysis. A potential future research study could be where the lexicon is able to cover a wider variety of South African languages - this is done by expanding the lexicon. This in turn would ensure that it provides more accurate sentiment classification. Furthermore, another future research study could be whereby the lexicon could be integrated into real-time translation systems. This could then improve it as it is able to learn within dynamic contexts - such as in social media. By making use of transformers along with various deep learning models, the translation accuracy and sentiment prediction should improve, and this can be conducted within an experiment to verify the results. By investigating sentiment expressions across languages, numerous cultural nuances can allow for the precision and accuracy to produce better analyses. Lastly, by making use of a dataset that has been expanded with real-world multilingual words, could allow for a better contribution to make a better and more generalised tool for translation.

9. References

Bing, L. (2012). Sentiment analysis and opinion mining (synthesis lectures on human language technologies). In *University of Illinois: Chicago, IL, USA*.

Bouguesmia, M. T. (2020). Using AI in Translation, a Technological Leap, or a Translator's Nightmare. *ALTRALANG Journal*, *2*(02), 78-102.

Cambria, E., Schuller, B., Xia, Y., & ... (2013). New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*. https://ieeexplore.ieee.org/abstract/document/6468032/

Carrillo, E., González, M., Parrilla, R. and Tarrega, A., 2023. Classification trees as machine learning tool to explore consumers' purchasing decision pathway. A case-study on parent's perception of baby food jars. *Food Quality and Preference*, 109, p.104916. https://doi.org/10.1016/j.foodqual.2023.104916.

Chen, Y., & Skiena, S. (2014). Building sentiment lexicons for all major languages. In *Proceedings of the 52nd Annual Meeting of the ….* aclanthology.org. https://aclanthology.org/P14-2063.pdf

Hamed, B.A., Ibrahim, O.A.S. and Abd El-Hafeez, T., 2023. Optimizing classification efficiency with machine learning techniques for pattern matching. *Journal of Big Data*, 10(1), p.124. https://doi.org/10.1186/s40537-023-00804-6.

Rahnasto, I. and Hollestelle, M., 2024. Comparing discrete choice and machine learning models in predicting destination choice. *European Transport Research Review*, 16(1), p.43. https://doi.org/10.1186/s12544-024-00667-9.

Ramkumar, M., Alagarsamy, M., Balakumar, A. and Pradeep, S., 2023. Ensemble classifier fostered detection of arrhythmia using ECG data. *Medical & Biological Engineering & Computing*, 61(9), pp.2453–2466. https://doi.org/10.1007/s11517-023-02839-6.

Starbuck, C., 2023. Logistic Regression. In: *The Fundamentals of People Analytics*. [online] Cham: Springer International Publishing. pp.223–238. https://doi.org/10.1007/978-3-031-28674-2_12.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & ... (2011). Lexicon-based methods for sentiment analysis. *Computational ….* https://direct.mit.edu/coli/article-abstract/37/2/267/2105

Tanaka, H., & Rossi, I. (2024). Revolutionizing Communication: Machine Translation with AI-Language Agents. *Innovative Computer Sciences Journal*, *10*(1), 1− 8-1− 8.

Van Atteveldt, W., Van der Velden, M. A., & Boukes, M. (2021). The validity of sentiment analysis: Comparing manual annotation, crowd-coding, dictionary approaches, and machine learning algorithms. *Communication Methods and Measures*, *15*(2), 121-140.

Zhu, Y., Yin, X., Li, R. and Chen, C., 2021. Extracting Decision Tree from Trained Deep Reinforcement Learning in Traffic Signal Control. In: *2021 International Conference on Cyber-Physical Social Intelligence (ICCSI)*. [online] 2021 International Conference on Cyber-Physical Social Intelligence (ICCSI). Beijing, China: IEEE. pp.1–7. https://doi.org/10.1109/ICCSI53130.2021.9736263.