Faculty of Engineering & Technology

Electrical & Computer Engineering Department

ENCS3340 - Artificial intelligence

Machine Learning Project Report

Prepared by :

Hanan Alawawda - 1230827

Amjad Adi – 1230800

Instructor: Aziz Qaroush

Section: 2

Date : 25-12-2026

Table of Contents

Table of Figures

1. Problem Formalization

The objective of this project is the development of a robust machine learning pipeline for Arabic Sentiment Analysis. The problem is formalized as a supervised multi-class classification task where the goal is to map a given Arabic text input to one of three sentiment labels: Positive (POS), Negative (NEG), or Neutral (NEUTRAL).

Our exploratory data analysis of the 10,006 records revealed a significant class imbalance: the dataset is dominated by Neutral samples (7,523), followed by Negative (1,684), and a minority of Positive samples (799). This imbalance requires specific strategies, such as balanced sampling and weighted priors, to ensure the model does not become biased toward the majority class.

2.Data Analysis

The initial phase of the project involved a comprehensive analysis of the dataset to identify its structural and linguistic properties. A primary observation is the heterogeneous nature of the text, which contains a blend of Modern Standard Arabic (MSA) and various regional dialects.

Furthermore, the data contains significant non-sentimental "noise," including URLs, HTML tags, and metadata, which necessitates rigorous cleaning to prevent feature interference.

A critical component of this analysis was the evaluation of Class Distribution. The dataset exhibits a substantial class imbalance, which is a pivotal factor in the project's design. This imbalance dictates the need for specialized mitigation strategies such as weighted loss functions, dynamic prior mixing, or balanced sampling to ensure the classifiers do not develop a bias toward the majority class.

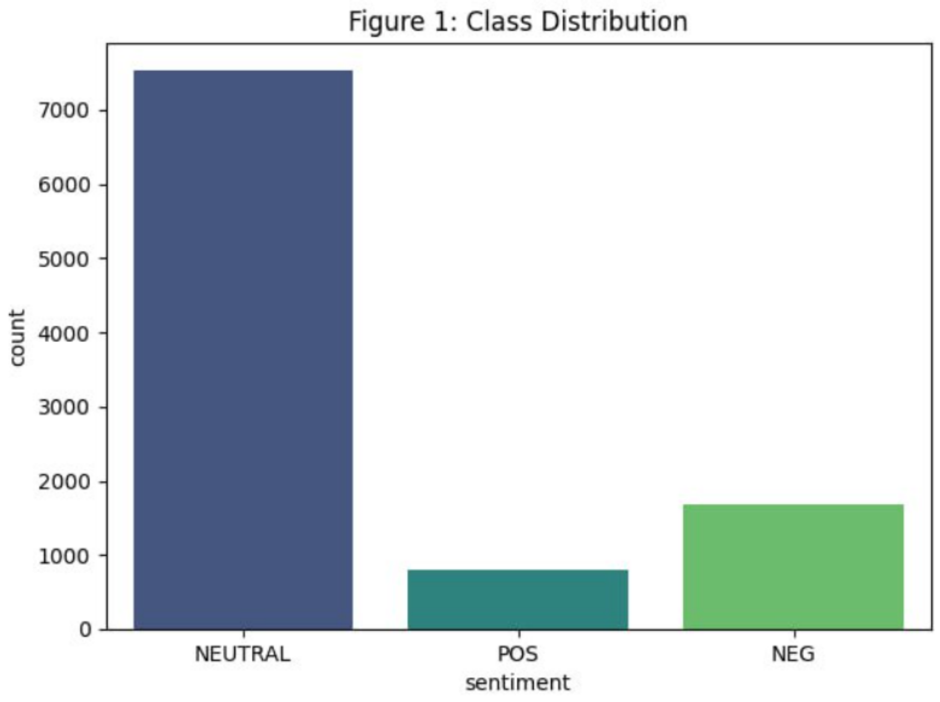

Figure 1: Class Distribution

3. Data Preprocessing

To prepare the heterogeneous Arabic dataset for machine learning, a bifurcated preprocessing pipeline was implemented. This ensures that text is optimized for both statistical (TF-IDF) and contextual (Transformer) feature extraction.

- **Text Cleaning and Noise Reduction:** The process utilizes regular expressions and the unicodedata library to eliminate non-informative elements, including URLs, HTML tags, Bidirectional (BIDI) marks, and numeric values. This prevents feature sparsity and ensures the model prioritizes semantic text over structural noise.
- **Orthographic Normalization:** A custom mapping was implemented to standardize character variations, such as unifying different forms of **Alef (أ, إ, آ, ٱ)** and **Ya (ى to ي).** This is critical for reducing vocabulary dimensionality and ensuring that morphological variations are treated as a single feature.
- **Emoji and Emoticon Translation:** To preserve emotional signals, visual icons and western-style emoticons are mapped to descriptive Arabic text (e.g., 😊 becomes "سعادة"). This ensures that non-textual sentiment signals are readable by the classifiers during vectorization.
- **Morphological Refinement:** The pipeline standardizes repeated characters (elongation removal) and handles specific Arabic prefixes (like "ب" and "ل") to ensure high consistency across the corpus.

4. Feature Design and Representation

The success of Arabic sentiment analysis depends heavily on capturing morphological and contextual nuances. We implemented a hybrid feature extraction strategy that combines traditional statistical methods with modern deep learning representations.

4.1 Content-Based Representations (TF-IDF)

We utilized TF-IDF (Term Frequency-Inverse Document Frequency) to transform raw text into numerical vectors. To capture both structural and semantic information, we used a combination of:

- **Word N-grams (1-3):** These capture common Arabic phrases and sequences (unigrams, bigrams, and trigrams).
- **Character N-grams (3-5):** Crucial for Arabic, these capture underlying roots, allowing the model to recognize morphological patterns even when a specific word form was not encountered during training.

4.2 Contextual Representations (MARBERT)

To capture semantic meaning that frequency counts might miss, we integrated MARBERT, a transformer-based model specifically designed for Arabic. This generates 768-dimensional contextual embeddings that understand complex relationships between words, which is vital for detecting nuance and sarcasm.

4.3 Feature Fusion and Dimensionality Reduction

The final step involved creating a "hybrid vector" by fusing different feature types. This posed a significant challenge because high-order character n-grams create massive, sparse TF-IDF vectors that suffer from the curse of dimensionality.

To address this, we implemented the following:

1. **Denoising via Truncated SVD:** We applied Truncated Singular Value Decomposition (SVD) to compress the sparse TF-IDF vectors into 250 dense components. This "denoises" the data and makes it computationally manageable for classifiers.
2. **Hybrid Concatenation:** We concatenated these dense SVD components with the MARBERT embeddings. This fused representation pulls in both word frequency statistics and deep transformer-based context into a single, high-performance feature set.

5. Model Training and Optimization

5.1 Data Splitting

The dataset was split into 60% training, 20% validation, and 20% testing.

5.1.1Multinomial Naive Bayes Optimization

In our Naive Bayes setup, we introduced a custom optimization method called the "Zoom Search" algorithm, which systematically identifies the best smoothing parameter (α) for the model. This approach proved to be beneficial in fine-tuning the model’s performance.

Additionally, we developed a custom strategy to adjust class priors. By mixing the actual class distribution in the training set with a uniform prior using a scaling factor (S), we were able to achieve a more balanced performance, especially on imbalanced Arabic datasets. This modification contributed significantly to an improved Macro F1-score.

5.1.2 Random Forest Iterative Optimization

The Random Forest model in our setup differs from traditional configurations. We employed a hybrid feature selection technique, combining multiple feature extraction approaches to improve the model's ability to capture relevant patterns.

1. **Feature Fusion**: We began by reducing the dimensionality of the TF-IDF matrix using Truncated SVD. For various values of k (ranging from 100 to 400), we performed the following :

    The top k components from SVD were concatenated with the 768-dimensional MARBERT embeddings, ensuring that the feature space contained rich representations from both sources.
2. **Hyperparameter Tuning with Stratified Cross-Validation**: Hyperparameters were optimized using RandomizedSearchCV with StratifiedKFold to ensure that each class is properly represented during training.
3. **Class Balance Handling**: We leveraged the BalancedRandomForestClassifier from the imblearn library, which automatically handles class imbalance by assigning appropriate

sample weights during the fitting process. This method helps the model focus on underrepresented classes without needing manual adjustments.

While the setup was relatively complex, it demonstrated the potential for better performance compared to standard Random Forest models. The exact steps and values for each k were evaluated iteratively as part of this approach.

5.2 Classifiers and Hyperparameters

The hyperparameters for each model were optimized using the validation dataset to improve performance.

For Naive Bayes (MultinomialNB), tuning focused on the smoothing parameter (α) and class priors. These adjustments helped the model better handle sparse data and improve the classification of minority classes. For the MLP Neural Network (MLPClassifier), key hyperparameters such as the learning rate, optimizer, hidden layer architecture, activation function, and regularization were optimized.

Early stopping was enabled to prevent overfitting, and batch size was adjusted for optimal training efficiency, ensuring the model converged effectively without overfitting.

For the Balanced Random Forest, hyperparameters like the number of trees, tree depth, minimum samples for splitting nodes and at leaves, and sampling strategies were fine-tuned. These adjustments helped the model capture underlying data patterns while addressing class imbalance, improving its generalization and overall performance.

All hyperparameter values were derived from validation data tuning, resulting in models that performed more effectively across the dataset.

6. Experimental Results and Comparative Analysis

In this section, we detail the data distribution, evaluation strategy, and the performance of the three implemented classifiers: Multinomial Naive Bayes, Multi-Layer Perceptron (MLP), and Balanced Random Forest.

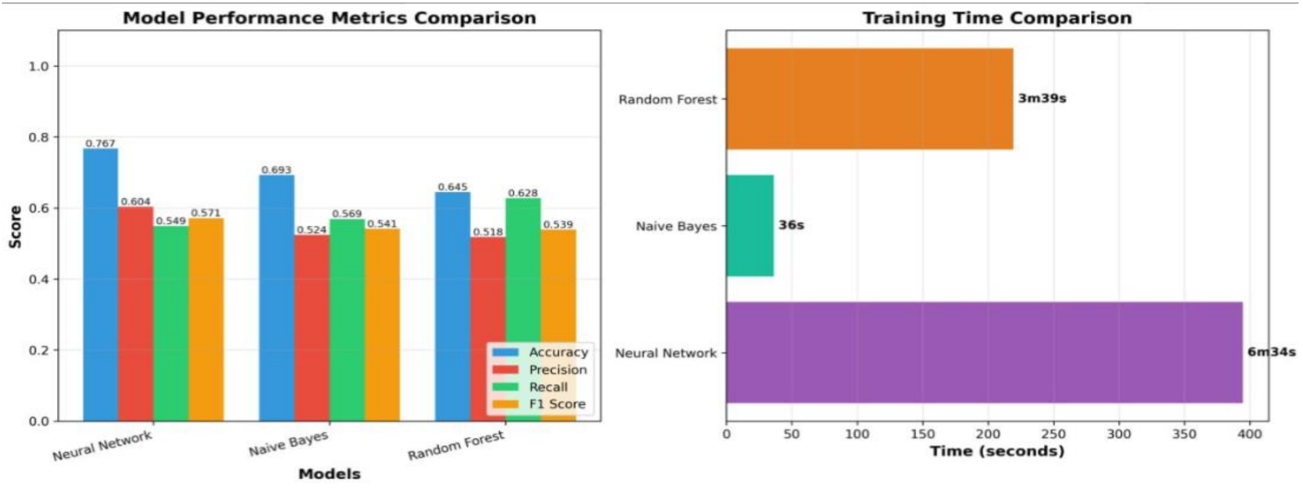

Figure 2: Experimental Results and Graphs

6.1 Dataset Characteristics and Split

The dataset comprises **10,006 records**. A primary challenge identified during exploratory analysis is the extreme class imbalance, which significantly impacts model training:

- **NEUTRAL:** 7,523 samples (75.2%)
- **NEG:** 1,684 samples (16.8%)
- **POS:** 799 samples (8.0%)

To ensure reliable evaluation, we implemented a **stratified split** to maintain these proportions across all sets:

- **Training Set:** 6,003 samples
- **Validation Set:** 2,001 samples
- **Test Set:** 2,002 samples

6.2 Detailed Analysis of Results

6.2.1 Multinomial Naive Bayes (TF-IDF Mix)

The Naive Bayes model was optimized using a "Zoom Search" grid method, resulting in an alpha. We addressed the imbalance by using a mixed prior strategy .

Observation: While its raw accuracy (69.28%) is lower than the MLP, its Recall for the Positive class (42.5%) is significantly higher. The prior mixing prevents the model from being overwhelmed by the majority Neutral class.

6.2.2 MLP Neural Network (Optimized FUSED)

The MLP achieved the highest raw accuracy (76.72%). The optimized architecture consists of two hidden layers using Tanh activation and the Adam optimizer.

- **Observation:** The model shows high precision for Neutral (0.8253), but the Positive recall drops to 30.0%. Despite fusing MARBERT embeddings to provide deep contextual understanding, the neural network still struggles to generalize the minority POS class compared to the other models.

6.2.3 Balanced Random Forest

The Balanced Random Forest was specifically designed to handle the 75% Neutral skew. We utilized iterative tweaks to determine the optimal number of SVD components (k).

- **Observation:** This model is the most effective at identifying minority classes, achieving the highest Balanced Accuracy (62.75%). By using a sampling_strategy and sample_weight within the trees, we forced the model to penalize errors on POS and NEG samples more heavily. This resulted in the best Recall for the Negative class (69.4%) and Positive class (54.4%).

6.3 Discussion: Accuracy vs. Fairness

The experimental results highlight a clear trade-off:

1. **The MLP** is the best choice if the goal is "Overall Accuracy," but it tends to ignore the Positive minority.
2. **The Balanced Random Forest** is the superior choice for "Sentiment Discovery," as it is far more likely to correctly identify a Negative or Positive post, even if it loses some accuracy on the Neutral majority.

7. Challenges and Mitigation

The development of the Arabic Sentiment Analysis pipeline presented several technical and linguistic hurdles. The following section details these challenges and the specific strategies employed to overcome them.

The primary challenge encountered was the Curse of Dimensionality caused by high-order character n-grams, addressed via Truncated SVD. Furthermore, the extreme class imbalance (where POS is only ~8% of the data) was mitigated using specialized algorithms and custom weighted loss functions. Finally, we resolved technical threading issues during the Random Forest grid search by refactoring the visualization logic..

7.1 Extreme Class Imbalance:

The dataset exhibited a severe skew where Neutral posts outnumbered Positive ones by nearly 10 to 1. To preserve the integrity of the data while ensuring the minority classes were not ignored, we implemented the following:

7.1.1 Naive Bayes:

We implemented a dynamic prior mixing strategy. By optimizing the S-factor (0.6349), we balanced the training distribution against a uniform prior, preventing the model from defaulting to "Neutral" for every prediction.

7.1.2 Random Forest & MLP:

We avoided standard oversampling (like SMOTE) as it often introduces noise in Arabic text. Instead, we utilized weighted loss functions and sample weight arrays. This forced the models to heavily penalize errors on the POS and NEG classes, making the classification more reliable for the minority labels.

7.2 Feature Sparsity and High Dimensionality:

7.2.1 Dimensionality Reduction:

Arabic character n-grams generate massive, sparse TF-IDF vectors where most entries are zero. This leads to the "Curse of Dimensionality," where the model struggles to find patterns. We mitigated this by using Truncated SVD to compress the sparse vectors into 300 dense components.

7.2.2 Iterative Optimization:

We conducted a grid search to find the optimal number of components (k), testing values from 100 to 400. Our goal was to find the "sweet spot" where the Balanced Random Forest could train efficiently without losing the critical morphological details (roots and stems) inherent in the Arabic language.

8. Conclusion

This project successfully developed and evaluated a hybrid machine learning framework for Arabic Sentiment Analysis, integrating traditional linguistic features with deep learning contextual embeddings. By fusing TF-IDF character n-grams with MARBERT transformer embeddings, we created a robust feature space that captures both the structural roots and the semantic nuances of the Arabic language.

The experimental results revealed a distinct trade-off between different classification strategies. The Multi-Layer Perceptron (MLP) achieved the highest overall efficiency with an accuracy of 76.72%, proving to be the most effective for general classification tasks.

The Balanced Random Forest, however, proved superior in handling the extreme class imbalance of the dataset, achieving a Balanced Accuracy of 62.75% and demonstrating a significantly higher recall for the minority Positive and Negative classes.

In conclusion, while deep learning models provide high accuracy, the inclusion of specialized "balanced" algorithms and hand-engineered preprocessing—such as emoji translation and orthographic normalization—is essential for accurately detecting sentiment in skewed Arabic social media data. Future work could further improve performance by exploring data augmentation techniques specifically designed for Arabic morphological structures to bolster the minority classes.