Using Text Classification to Determine a Person's Myers Briggs Personality Type Indicator

UC Berkeley - AIML Professional Certification Capstone Project

Executive Summary

Project Overview and Goals:

The goal of this project is to identify an effective way to determine a person's Myers-Briggs Type Indicator (MBTI) based on their text posts. We will be training and tuning several classification models to accurately classify social media posts according to MBTI types. Models will be able to predict, from future/unseen posts, the MBTI type of their writer. We will then evaluate and compare the models' performances to identify the best one and further scrutinize it to find the most effective features (words) that enhance performance in this classification task. Insights will be drawn from this model by conducting a global analysis, using various libraries to identify the most important words/features used for making accurate predictions. We will also be locally analyzing this model and evaluating its class prediction process for individual posts. Lastly, we will draw insights from our analyses and recommend areas to research and courses to undertake for future work in determining MBTI from text posts.

Findings:

The best model for determining a person's Myers-Briggs Type Indicator (MBTI) from text posts is the Support Vector Classifier (SVC) model, using Lemmatization + TF-IDF, with an accuracy score of 0.5735, a recall of 0.5735, and an F1 score of 0.572964. Its performance is followed by the Random Forest model, Logistic Regression model, and the Naive Bayes model. This decision is based on comparing the finetuned models' accuracy, recall, and F1 scores (results summary below). The SVC model has the best overall performance metrics, including accuracy, recall, and F1 score. In terms of errors, the Logistic Regression model has a balanced number of false positives (FP) and false negatives (FN), the Naive Bayes model has more FPs than FNs, the Decision Tree model has a similar count of FP and FN, and the SVC model maintains a slightly higher number of FPs compared to FNs.

Results and Conclusion:

In this project, we explored two different approaches to determine a person's Myers-Briggs Type Indicator (MBTI) from text posts: SentenceTransformer embeddings and Lemmatization + TF-IDF vectorization.

The Lemmatization + TF-IDF approach outperformed the SentenceTransformer approach across all models. The Support Vector Classifier (SVC) using Lemmatization + TF-IDF achieved the highest accuracy score of 0.5735, recall of 0.5735, and F1 score of 0.572964. This was followed by the Decision Tree and Random Forest models, which also showed strong performance with accuracy scores of 0.5710 and 0.5665, respectively.

model	speed	best_params	accuracy	recall	f1	confusion_matrix
Lemmatization + TF-IDF Logistic Regression	1.895791	{'penalty': 'l1', 'solver': 'saga'}	0.5535	0.5535	0.548725	[[444, 543], [350, 663]]
Lemmatization + TF-IDF SVC	7.588776	{'kernel': 'rbf'}	0.5735	0.5735	0.572964	[[531, 456], [397, 616]]
Lemmatization + TF-IDF Decision Tree	2.613449	{'max_depth': None, 'min_samples_split': 2}	0.5710	0.5710	0.570961	[[576, 411], [447, 566]]
Lemmatization + TF-IDF Naive Bayes	0.028872	{'alpha': 10}	0.5670	0.5670	0.566561	[[593, 394], [472, 541]]
Lemmatization + TF-IDF RandomForest	80.851408	{'max_depth': None, 'min_samples_split': 10, 'n_estimators': 100}	0.5665	0.5665	0.566046	[[527, 460], [407, 606]]

In contrast, the SentenceTransformer approach did not perform as well. The highest accuracy was observed with the SVC model, achieving a score of 0.5525, while the Random Forest model achieved an accuracy of 0.5515. These results indicate that the Lemmatization + TF-IDF approach captures the textual features more effectively for this particular classification task.

model	speed	best_params	accuracy	recall	f1	confusion_matrix
SentenceTransformer Logistic Regression	10.590147	{'penalty': 'l1', 'solver': 'liblinear'}	0.5290	0.5290	0.528958	[[534, 453], [489, 524]]
SentenceTransformer SVC	32.965018	{'kernel': 'rbf'}	0.5525	0.5525	0.552454	[[534, 453], [442, 571]]
SentenceTransformer Decision Tree	15.421539	{'max_depth': None, 'min_samples_split': 10}	0.5330	0.5330	0.533000	[[533, 454], [480, 533]]
SentenceTransformer Naive Bayes	0.173762	{}	0.5335	0.5335	0.530365	[[609, 378], [555, 458]]
SentenceTransformer RandomForest	345.428539	{'max_depth': None, 'min_samples_split': 10, 'n_estimators': 200}	0.5515	0.5515	0.551431	[[531, 456], [441, 572]]

While the Lemmatization + TF-IDF approach provides a more effective feature representation for MBTI classification from text posts compared to SentenceTransformer, neither approach achieved sufficiently high accuracy to be used reliably in real-life applications. The highest accuracy of 0.5735 indicates that there is still significant room for improvement in predicting MBTI types from text.

Next Steps and Recommendations:

- Leverage Deep Learning Models:
  - Neural Networks: CNNs or RNNs could capture more complex text patterns.
  - Transformers: Models like BERT or GPT, known for state-of-the-art NLP performance, could enhance results.
- Hybrid Approaches:
  - Combine TF-IDF with neural embeddings for richer text representation.
- Data Augmentation:
  - Increase dataset size and diversity with techniques like paraphrasing and back-translation. This is especially applicable for the MTBI types for which we have fewer data such as ESTP, ESFP, ESFJ, and ESTJ.
- Ensemble Methods:
  - Use ensemble techniques like stacking or boosting to improve robustness and accuracy.
- Feature Engineering:
  - Explore additional features like sentiment analysis, topic modeling, and linguistic features.

By implementing these advanced techniques, we can aim for more accurate and reliable MBTI prediction models suitable for real-life applications.

Rationale

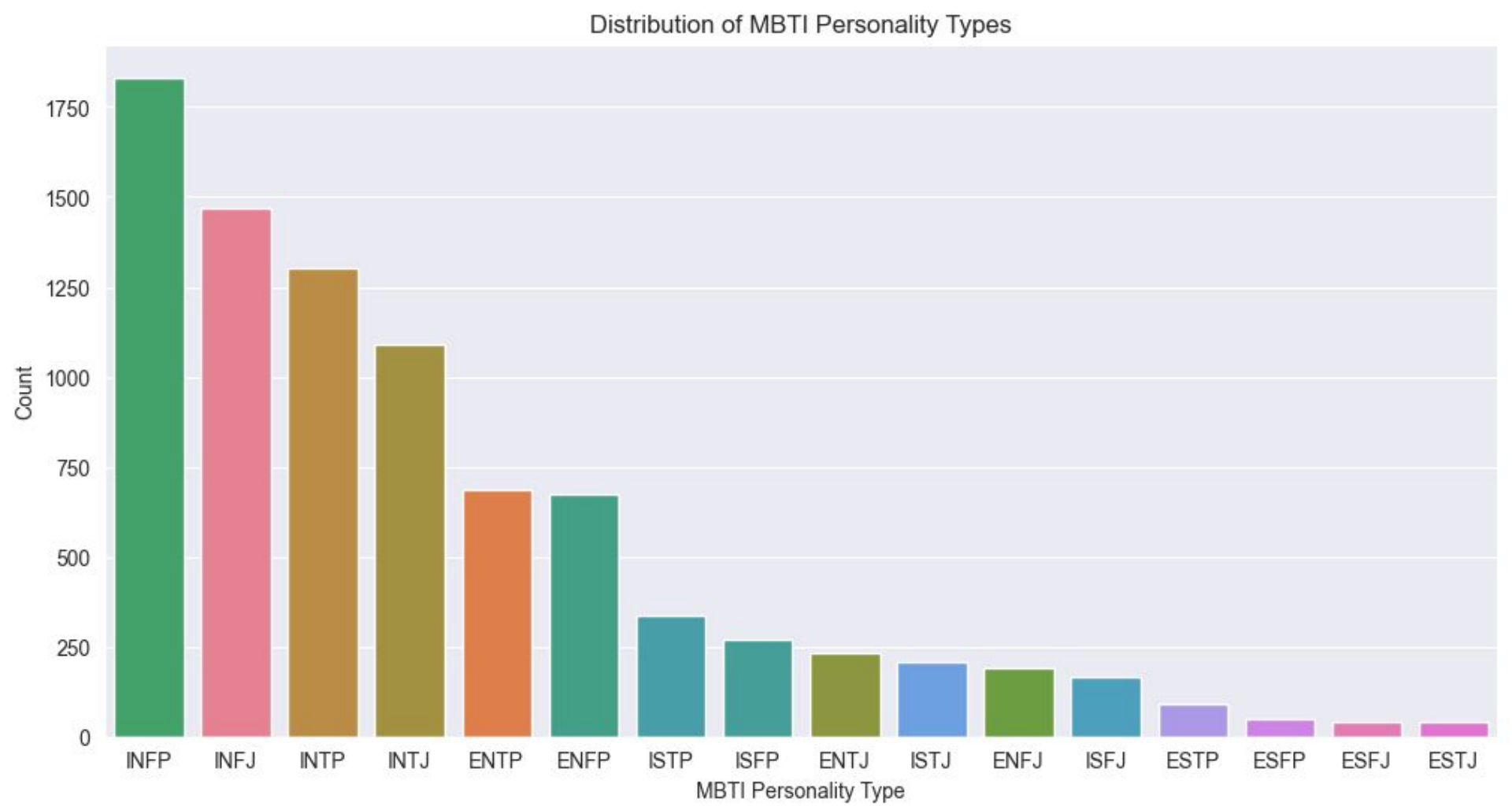
Understanding a person's Myers-Briggs Type Indicator (MBTI) from their text posts provides numerous compelling benefits and applications. Utilizing this capability allows organizations and researchers to gain deeper insights into individual personality traits, which can be applied in various domains. This opens up a wide range of opportunities across multiple fields. From personalized marketing and enhanced user experiences to improved team dynamics and advanced psychological research, the potential applications are extensive and impactful. By recognizing and addressing the diverse personality traits of individuals, organizations can achieve higher engagement, satisfaction, and success.

Data Sources

Dataset
The dataset was sourced from Kaggle (https://www.kaggle.com/datasets/datasnaek/mbti-type), containing over 8600 rows of MBTI types and their corresponding social media posts. Each entry provided a rich text-based profile for analysis.

Exploratory Data Analysis:
Initial exploration involved understanding the data distribution and identifying the unique MBTI types. Visualizations highlighted the imbalance among different personality types, guiding subsequent preprocessing steps.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8675 entries, 0 to 8674
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   TYPE    8675 non-null   object
 1   POSTS   8675 non-null   object
dtypes: object(2)
memory usage: 135.7+ KB
```


Distribution of MBTI Personality Types
Count
1750
1500
1250
1000
750
500
250
0
INFP
INFJ
INTP
INTJ
ENTP
ENFP
ISTP
ISFP
ENTJ
ISTJ
ENFJ
ISFJ
ESTP
ESFP
ESFJ
ESTJ
MBTI Personality Type

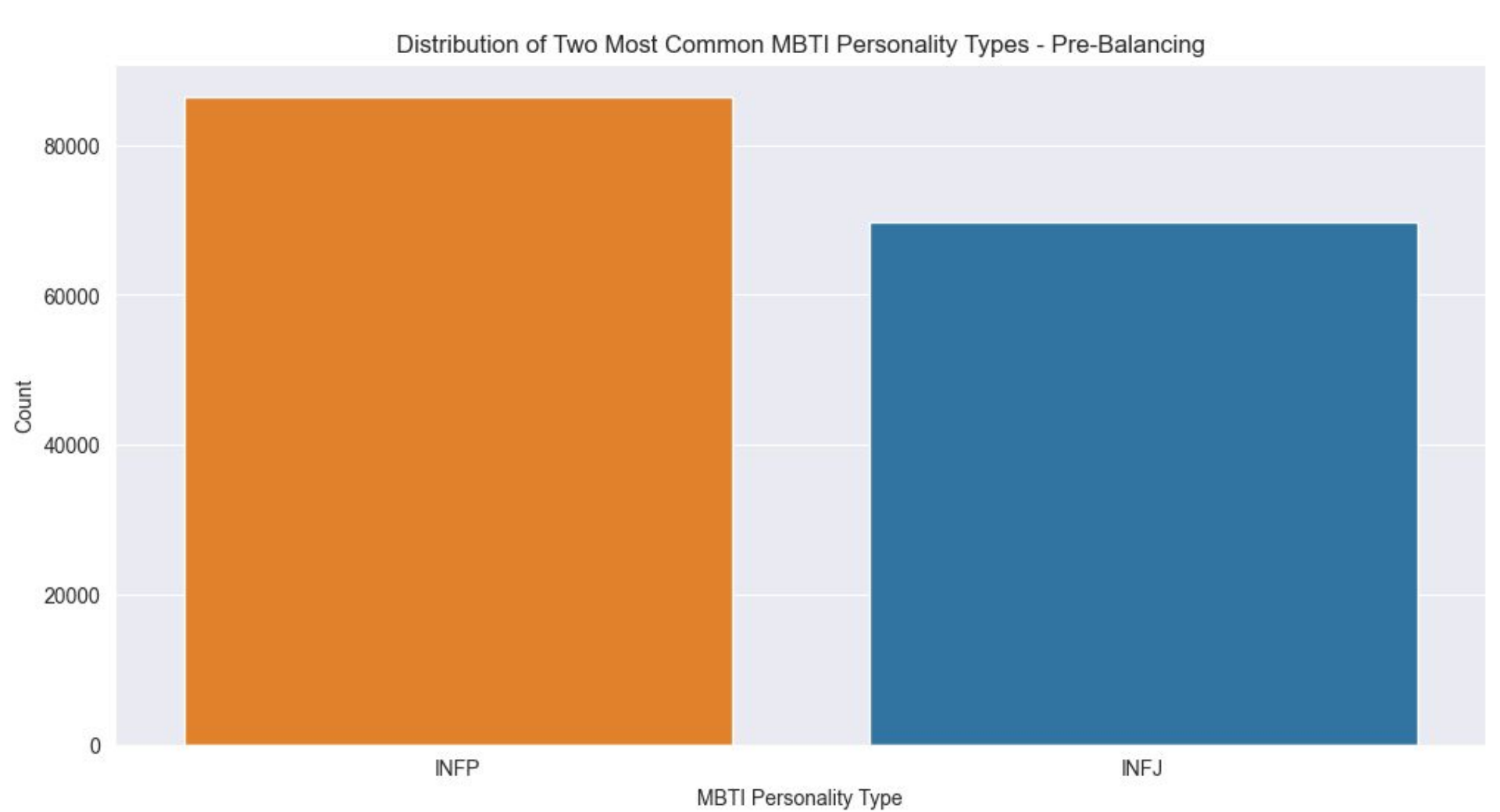
```
Number of unique values in TYPE column: 16

Unique values in TYPE column: ['INFJ' 'ENTP' 'INTP' 'INTJ' 'ENTJ' 'ENFJ' 'INFP' 'ENFP' 'ISFP' 'ISTP'
 'ISFJ' 'ISTJ' 'ESTP' 'ESFP' 'ESTJ' 'ESFJ']
```

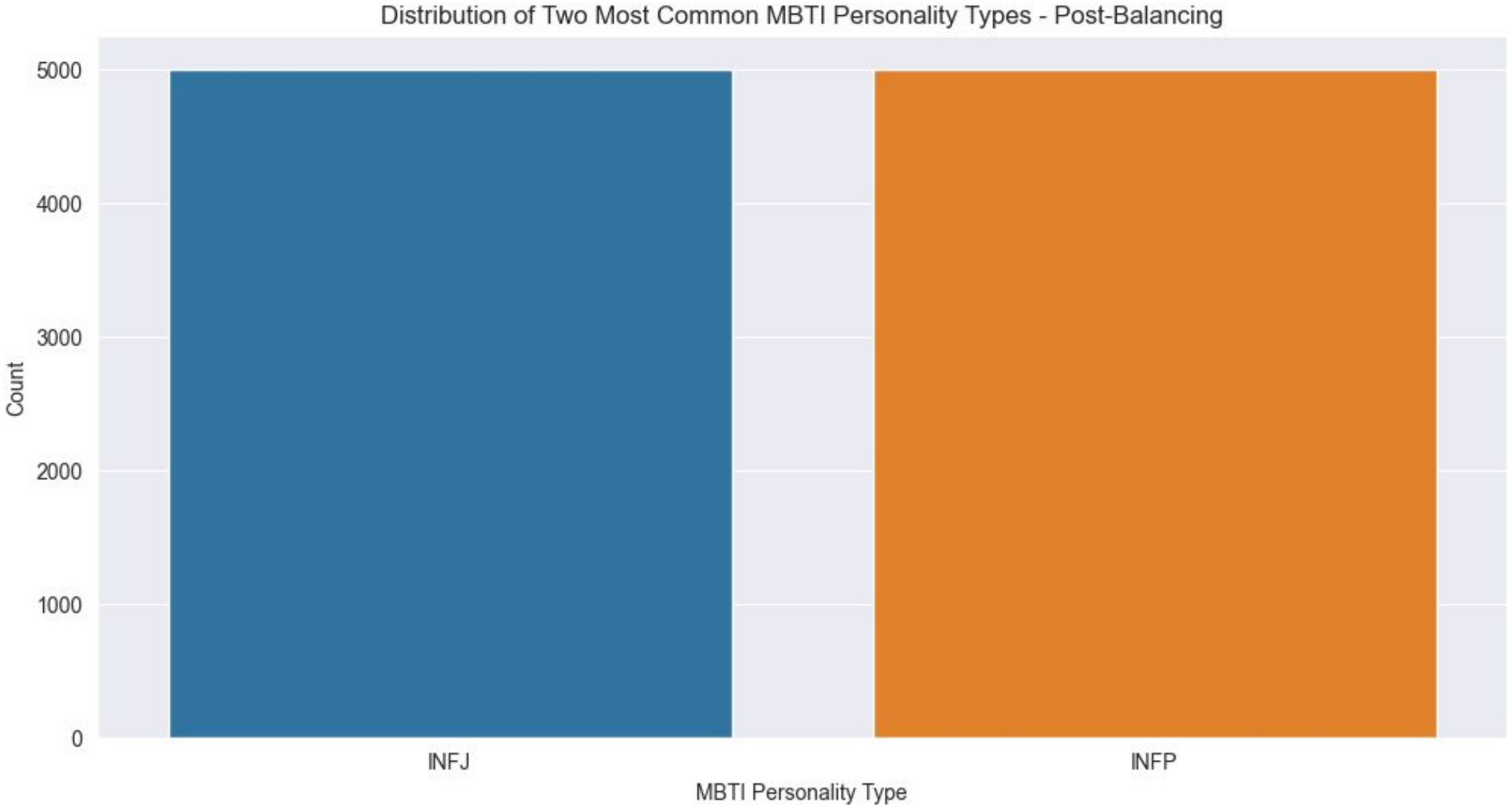
Cleaning and Preparation:

Data cleaning involved converting text to lowercase, parsing posts into separate rows, and removing URLs, special characters, and numbers. This ensured that the textual data was uniform and ready for analysis.

Preprocessing:

The dataset was filtered to focus on INFP and INFJ types, balancing the data to improve model performance. Text data was then prepared using two methods: SentenceTransformer for generating embeddings and TF-IDF vectorization for capturing term frequency and importance.


Distribution of Two Most Common MBTI Personality Types - Pre-Balancing
80000
60000
40000
20000
0
Count
INFP
INFJ
MBTI Personality Type


Distribution of Two Most Common MBTI Personality Types - Post-Balancing
5000
4000
3000
2000
1000
0
Count
INFJ
INFP
MBTI Personality Type

Final Dataset:

The final dataset was balanced and preprocessed, containing clean text posts with associated MBTI types. This prepared data was then used for model training and evaluation.

Methodology

Holdout cross-validation was employed, where models were trained on a training set and validated on a test set. Additionally, GridSearchCV was utilized to evaluate models using accuracy score and to fine-tune each model's hyperparameters to optimize this metric. Accuracy is an appropriate measure because we have a balanced dataset; it calculates the proportion of correctly predicted observations out of the total observations, given by:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

It was decided to evaluate two approaches towards feature representation, namely SentenceTransformer and TfidfVectorizer.

SentenceTransformer provides semantic understanding and context awareness, such as sentiment analysis or detecting nuanced similarities between texts.

TfidfVectorizer, on the other hand, measures term importance and frequency. Before vectorization with TfidfVectorizer, lemmatization was performed on the dataset for normalization, accuracy, and performance purposes. This was not necessary for the SentenceTransformer approach as it processes sentences as a whole as opposed to individual words in the case of TfidfVectorizer.

Evaluating both approaches allows for a comprehensive assessment of feature representations, helping to identify the most effective approach for improving model accuracy and performance across various text classification tasks. This dual evaluation ensures that models are robust, contextually aware, and capable of capturing both semantic meaning and term importance.

For both the SentenceTransformer and TfidVectorizer approached five models were trained, fine-tuned, and compared to determine the best model(s) for this task. The following models were used:

Logistic Regression:
This model was tuned using GridSearchCV, evaluating different penalties and solvers to optimize performance on the training set.

- SentenceTransformer approach: Optimized hyperparameters - {'penalty': 'l1', 'solver': 'liblinear'}
- Lemmatization + TFIDF approach: Optimized hyperparameters - {'penalty': 'l1', 'solver': 'saga'}

SVC:
Support Vector Classifier was applied with linear and RBF kernels, aiming to find the best separating hyperplane for the MBTI types.

- SentenceTransformer approach: Optimized hyperparameters - {'kernel': 'rbf'}
- Lemmatization + TFIDF approach: Optimized hyperparameters - {'kernel': 'rbf'}

Decision Tree:
Decision Tree models were evaluated with varying depths and minimum samples splits to determine the best structure for classification.

- SentenceTransformer approach: Optimized hyperparameters - {'max_depth': 50, 'min_samples_split': 10}
- Lemmatization + TFIDF approach: Optimized hyperparameters -{'max_depth': None, 'min_samples_split': 2}

Naive Bayes:
Both Gaussian and Multinomial Naive Bayes models were tested, focusing on different assumptions about the distribution of features.

- SentenceTransformer approach: Optimized hyperparameters - {} (GaussianNB was used due to negative values in embeddings)
- Lemmatization + TFIDF approach: Optimized hyperparameters - {'alpha': 10} (MultinomialNB was used)

Random Forest:
Random Forest models were trained with various numbers of estimators and depths, leveraging ensemble learning to enhance predictive accuracy.

- SentenceTransformer approach: Optimized hyperparameters - {'max_depth': 50, 'min_samples_split': 10, 'n_estimators': 200}
- Lemmatization + TFIDF approach: Optimized hyperparameters - {'max_depth': None, 'min_samples_split': 10, 'n_estimators': 100}

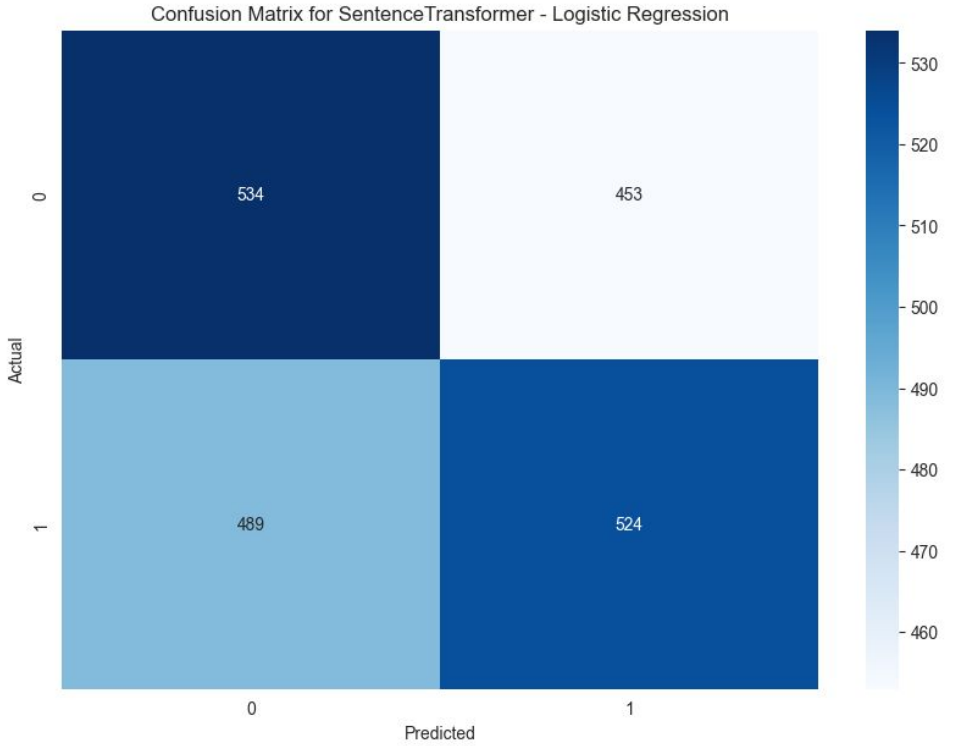
Model Evaluation and Results

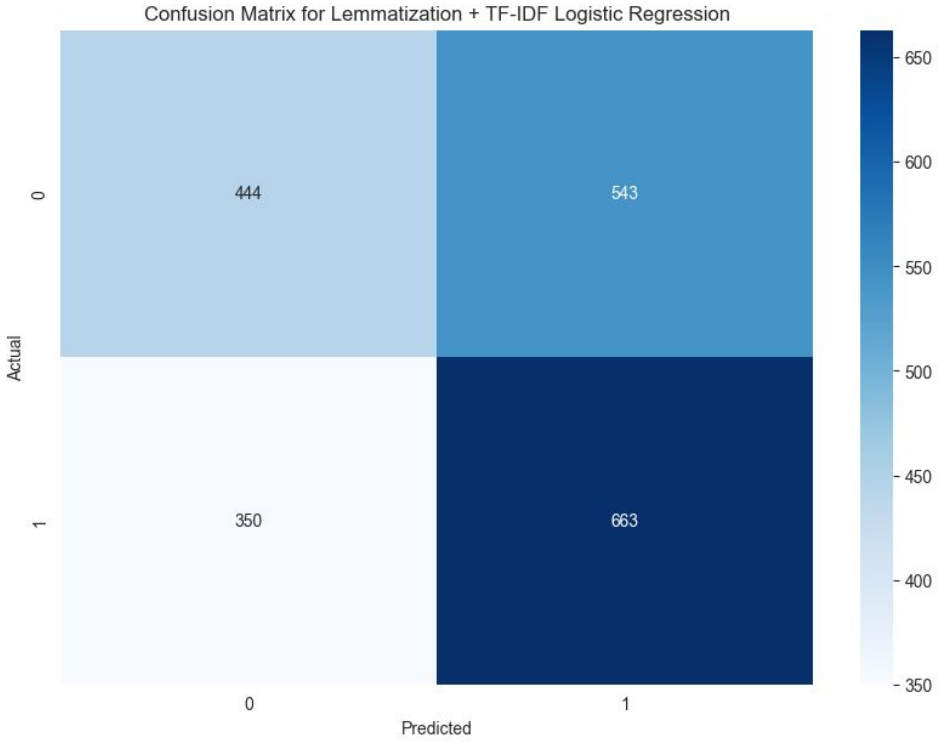
The models were evaluated based on accuracy, recall, F1 score, and confusion matrices. Results showed that the Random Forest and SVC models provided the highest accuracy and balanced performance. Confusion matrices for each model illustrated their ability to correctly classify INFP and INFJ types, with detailed classification reports highlighting the precision and recall for each class. The use of both SentenceTransformer embeddings and TF-IDF vectorization provided a comprehensive evaluation, ensuring that the models captured both semantic meaning and term importance.

Model performance will be visualized using confusion matrices, which display the counts of each error type made by the model during the classification task. In these plots, 0 represents INFP and 1 represents INFJ.

Logistic Regression:

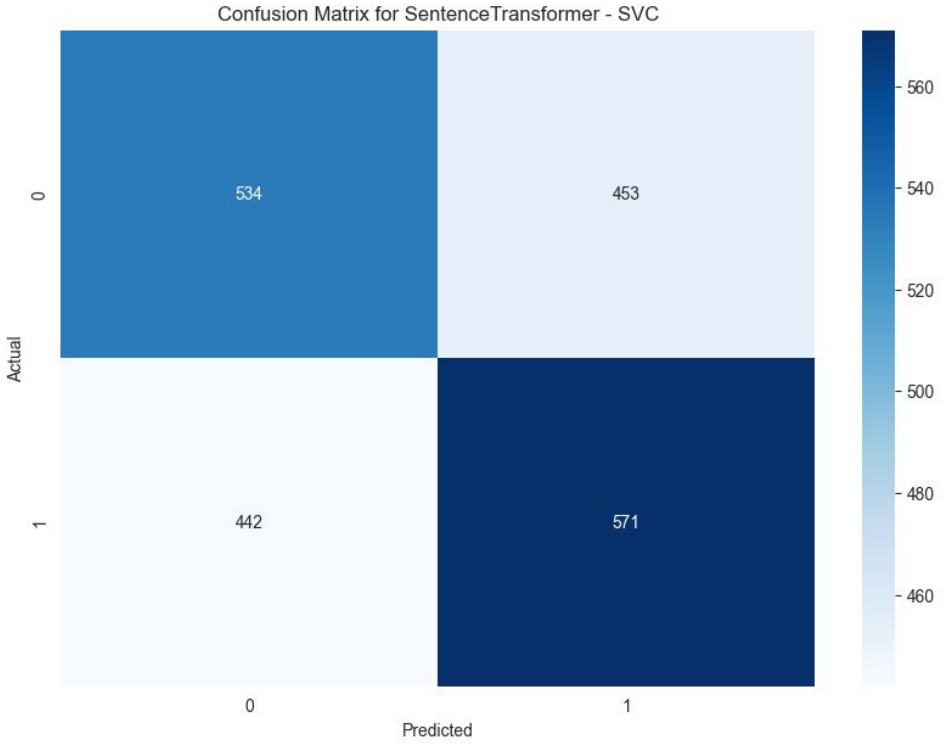
SentenceTransformer approach: Speed = 5.541228, Accuracy = 0.5290, Recall = 0.5290, F1 Score = 0.528958
Lemmatization + TF-IDF approach: Speed = 1.895791, Accuracy = 0.5535, Recall = 0.5535, F1 Score = 0.548725

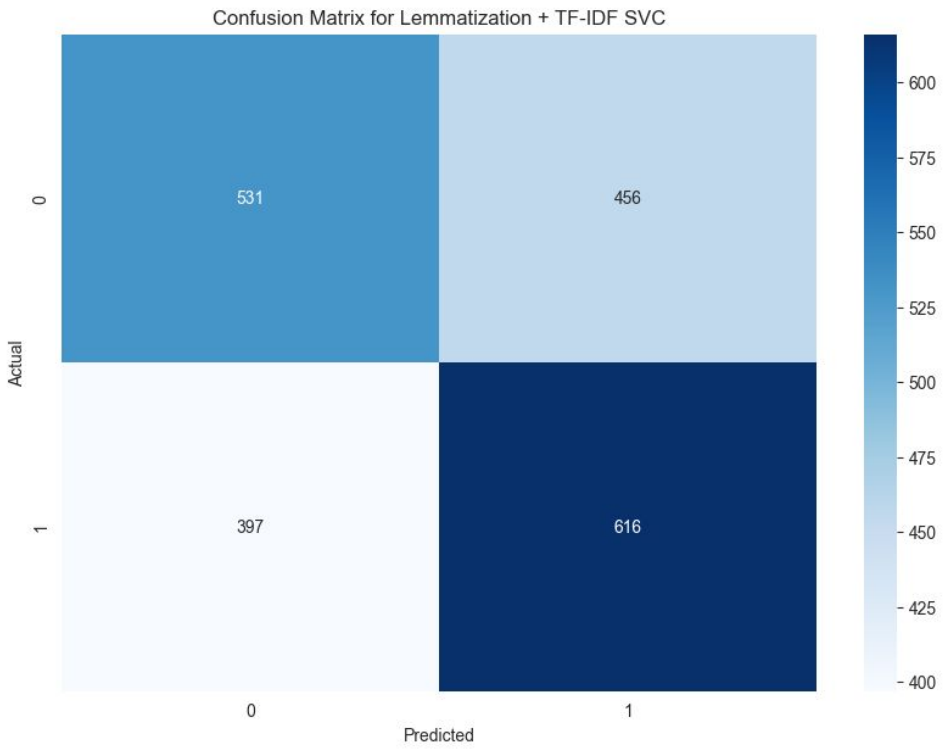




SVC:

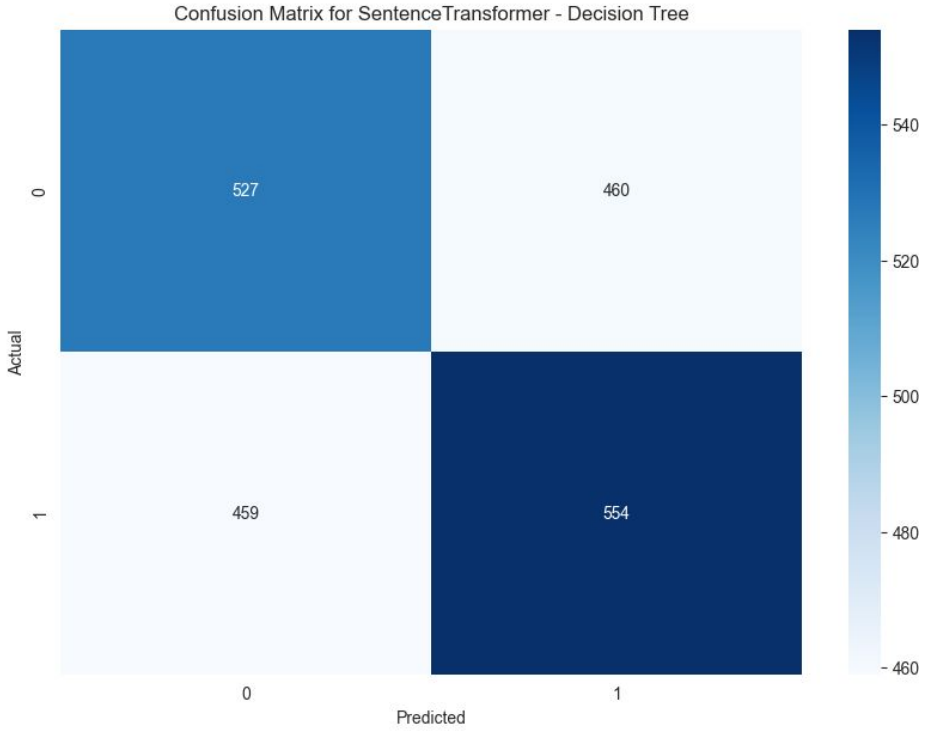
SentenceTransformer approach: Speed = 33.494465, Accuracy = 0.5525, Recall = 0.5525, F1 Score = 0.552454
Lemmatization + TF-IDF approach: Speed = 7.588776, Accuracy = 0.5735, Recall = 0.5735, F1 Score = 0.572964

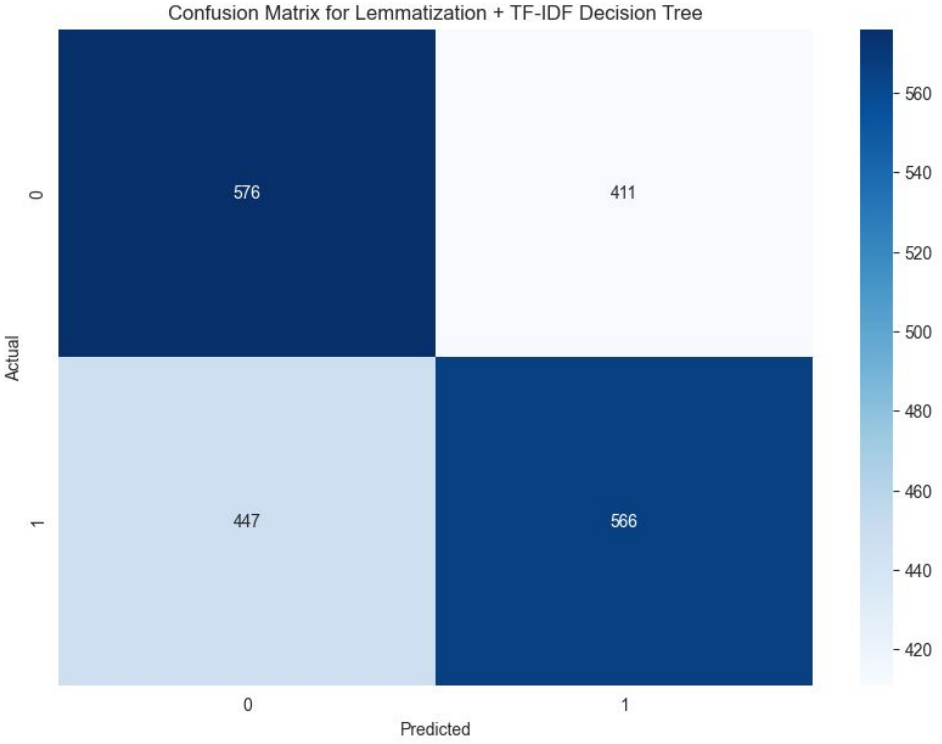




Decision Tree:

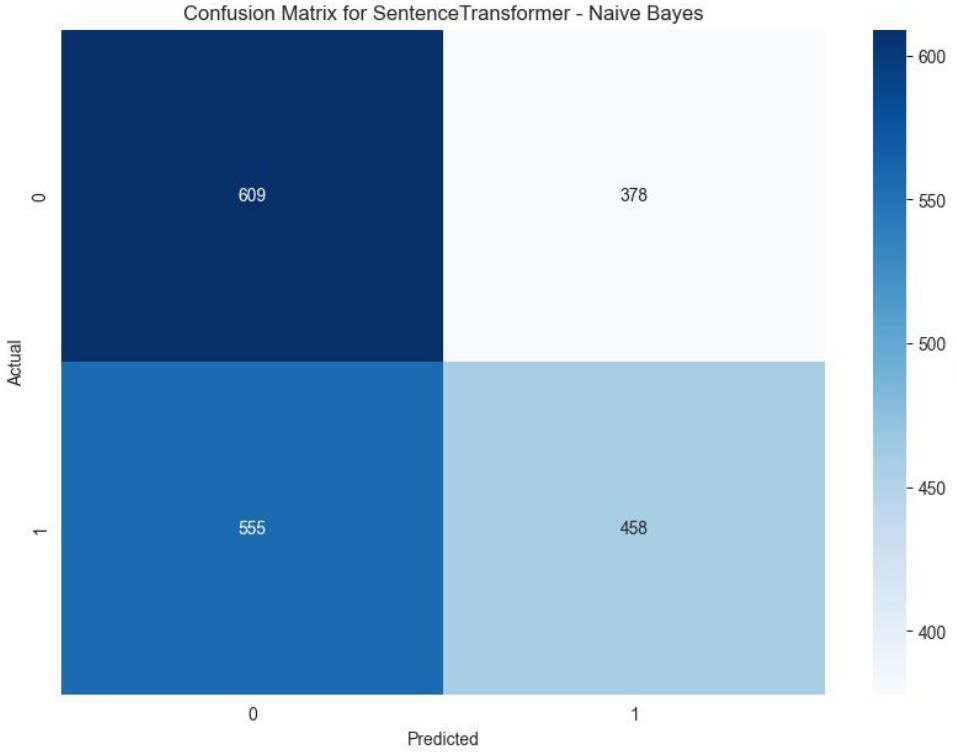
SentenceTransformer approach: Speed = 15.428351, Accuracy = 0.5405, Recall = 0.5405, F1 Score = 0.540497
Lemmatization + TF-IDF approach: Speed = 2.613449, Accuracy = 0.5710, Recall = 0.5710, F1 Score = 0.570961

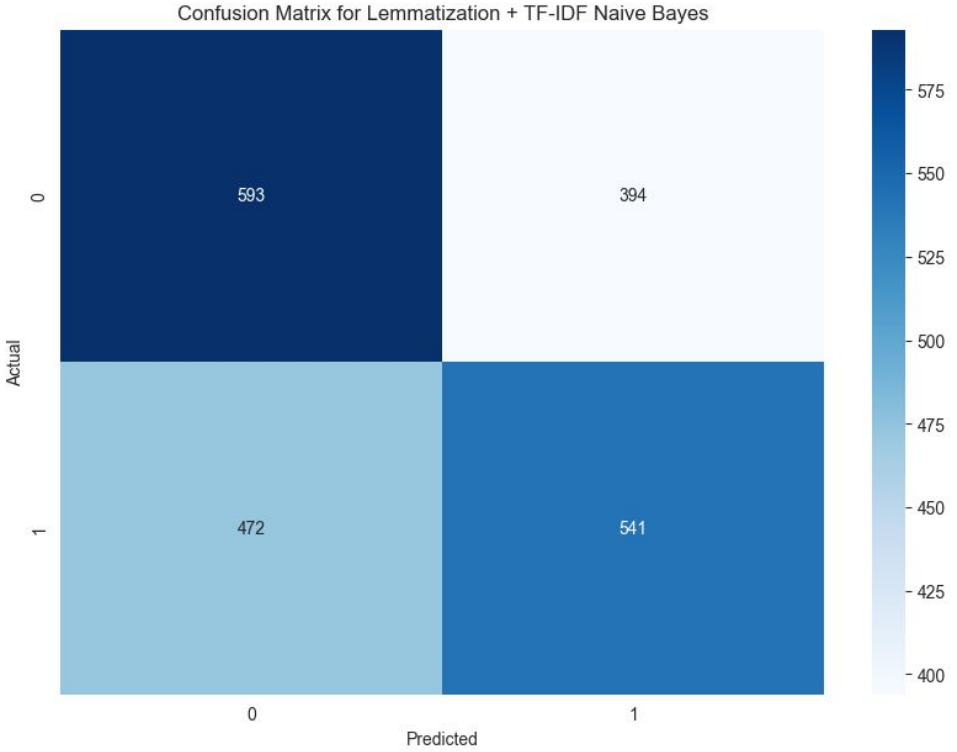




Naive Bayes:

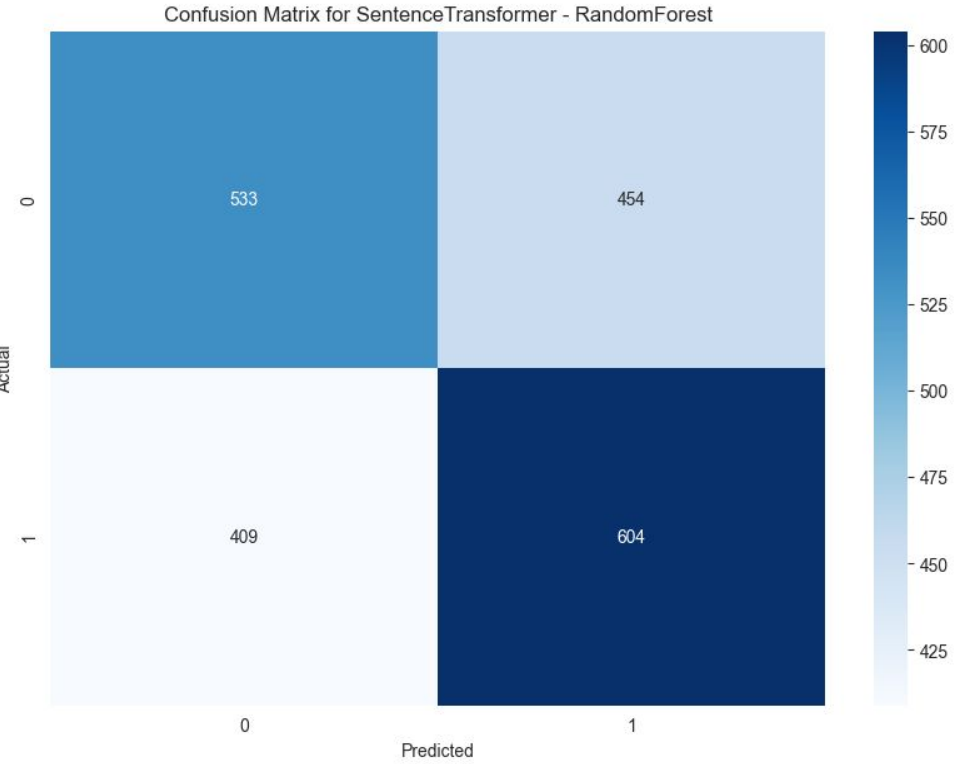
SentenceTransformer approach: Speed = 0.168367, Accuracy = 0.5335, Recall = 0.5335, F1 Score = 0.530365
Lemmatization + TF-IDF approach: Speed = 0.028872, Accuracy = 0.5670, Recall = 0.5670, F1 Score = 0.566561

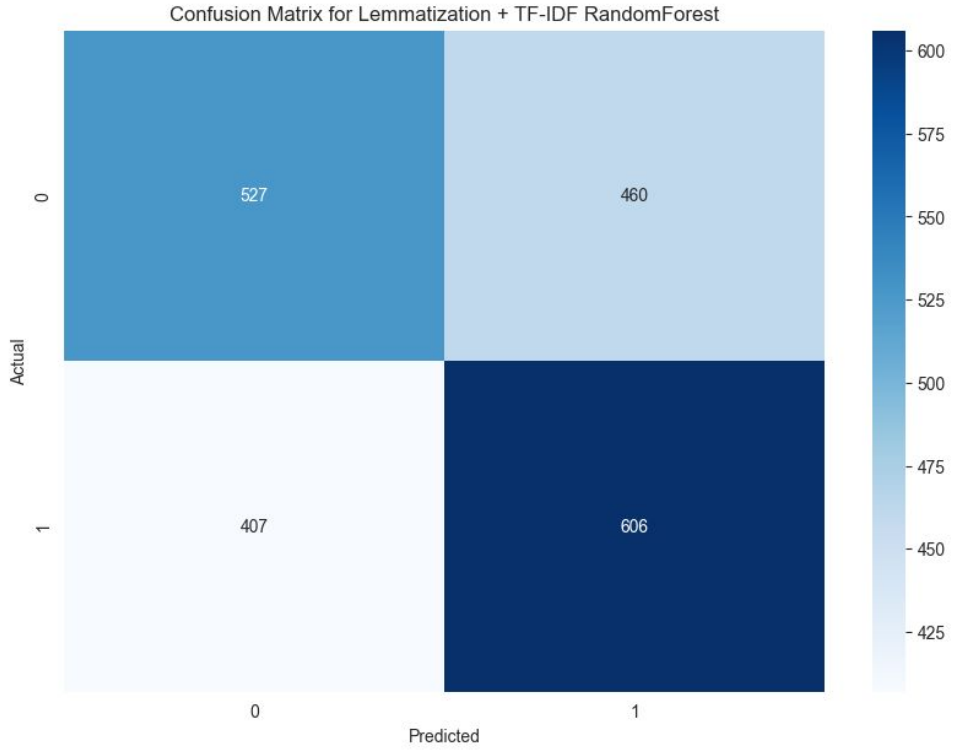




Random Forest:

SentenceTransformer approach: Speed = 348.468205, Accuracy = 0.5685, Recall = 0.5685, F1 Score = 0.568155
Lemmatization + TF-IDF approach: Speed = 80.851408, Accuracy = 0.5665, Recall = 0.5665, F1 Score = 0.566046





A comprehensive analysis and assessment of the top-performing model are provided in the Results and Conclusions section of the Executive Summary.

This completes the presentation. Thank you!