Multi-Lingual Spam Detection

Sri Sai Singala
University of New Haven
ssing39@unh.newhaven.edu

Bharat Veer Bootla
University of New Haven
bboot1@unh.newhaven.edu

Rahulsai Somepalli
University of New Haven
rsome1@unh.newhaven.edu

Abstract

In today's tech driven era, the major source of communication has become messaging or SMS. Some of these messages that we receive may be spam and it becomes difficult to distinguish between spam and non-spam messages. In this study, we trained several machine learning models on the multilingual SMS spam data. We go from classical ML models to using pre-trained models for multilingual data. The results show that pre-trained models heavily outperform the classical ML models.

1 Introduction

In our fast-paced digital world, emails and text messages have become our go-to for staying connected. However, this surge has brought forth a parallel rise in spam-related challenges, necessitating the development of intelligent automated detection systems ([1], [2], [3]). As we hustle through our digital lives, dealing with spam has become a real puzzle. It is not just about irritation, it is about keeping our information and devices safe ([1]). Recent research has honed in on leveraging supervised learning techniques to enhance the efficiency of spam detection, especially in the context of emails ([1], [2]). Beyond the inconvenience, the potential for malicious links and information theft requires the need for advanced techniques to tackle this issue ([2]). However, existing approaches show limitations in feature extraction and selection, particularly when applied with messages in languages beyond English ([2], [4]).

The previous works have proposed novel models that leverage pre-trained Transformers and Ensemble Learning for enhanced detection capabilities ([3]). These advanced techniques address the challenges associated with effective feature representation and the dynamic nature of multilingual content. Moreover, the recent work extends to evaluating the performance of traditional machine learning techniques, such as logistic regression and neural networks, in the context of SMS spam filtering ([5]). Another work introduces a method to distinguish between harmless and potentially harmful messages, emphasizing the importance of user education and proactive protective measures ([6]). We aim to use all the necessary techniques to come up with a model that accurately determines and distinguishes spam and non spam messages.

2 Proposed Methodology

For detection of spam messages in multi-lingual texts we apply several models and pick the best one based on the performance. The methodology is to use a step-by-step process to solve the problem as described in the following steps:

- Collecting and gathering the Multi-Lingual Dataset.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

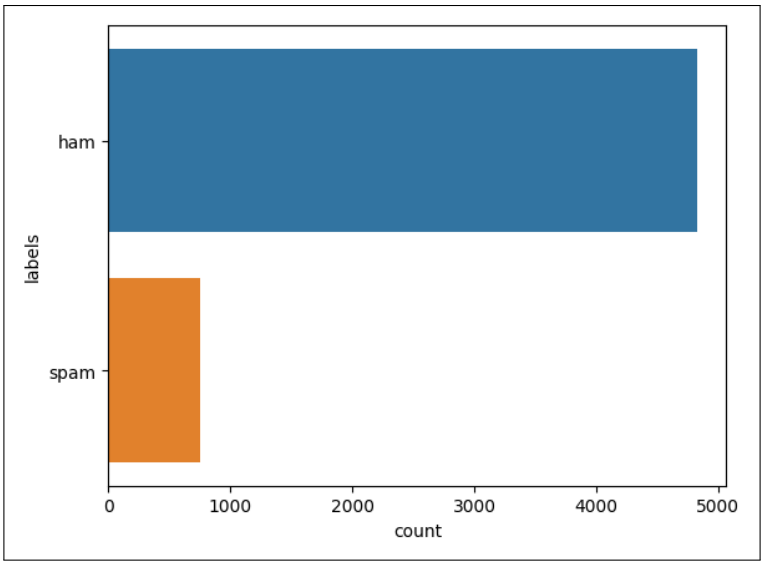

Figure 1: Countplot of spam and ham labels

- Performing analysis of the collected data to gain insights into it.
- Pre-processing the data to handle missing data in the dataset and then clean and transform the data into a format suitable for training models. Here we also select the features that contribute to model performance.
- Selecting appropriate classical Machine Learning models and pre-trained models for training
- Separating the traning and testing data in a standard ratio to train and test the models respectively.
- The selected models will then be trained on the training data.
- Assessing the performance of the models on the test set using appropriate evaluation metrics.

3 Technical Details

3.1 Dataset

The Multi-Lingual Spam SMS dataset is taken from

`https://huggingface.co/datasets/dbarbedillo/SMS_Spam_Multilingual_Collection_Dataset`

. The dataset consists of 5572 rows. These originally contained SMS messages in English language and were further translated to 20 other languages. The texts were machine translated into Hindi, German and French and were further translated into Spanish, Chinese, Arabic, Bengali, Russian, Portuguese, Indonesian, Urdu, Japanese, Punjabi, Javanese, Turkish, Korean, Marathi, Ukrainian, Swedish, and Norwegian. One categorical column of the dataset named labels showed if the SMS messages were spam or non-spam (ham).

3.2 Data Analysis

The dataset contains 23 columns and 5572 rows out of which 21 columns contain texts of different languages. Figure 3.2 shows the countplot of the type of labels. It can be observed that Only approximately 20% of the data contains spam sms messages.

3.3 Data Pre-Processing

The dataset is pre-processed to be further utilized for training. The initial steps of processing include:

- The dataset is checked for null values and the rows containing any null values are dropped. The remaining data will be processed as shown in the next steps.
- Regular Expression library is used in removing the unwanted special characters. Numbers and money signs are kept as they are important factor in spam messages.
- If the translated text contains English letters or words, they are converted to lowercase.
- The punctuation marks are removed from the text using string library.
- Finally, any extra space from the messages is removed.
- The unwanted column present in the dataset is dropped and the labels column is encoded using LabelEncoder from scikit-learn's preprocessing library. It converts the categorical labels column to numerical by encoding ham to 0 and spam to 1.

4 Classical ML Models

4.1 Further Processing

Text Preprocessing To train classical Machine Learning models on the dataset, further preprocessing is performed using the natural language toolkit library. First, the text is tokenized and then the stop-words are removed from the text messages. These stop-words are specific to languages. The obtained text is then joined which completes the processing. Although our modeling works for all languages, for our study, we will consider 4 languages. We perform our analysis on English which is the main one, French, Arabic and Spanish languages.

Train and Test Data The dataset is divided into train and test sets where 80% of the data goes into training set and the remaining 20% of the data accounts for test set. The 4 languages are taken into consideration for training and testing on these models. These texts of all the selected languages are are vertically concatenated to obtain all language texts in one X_train column. So, instead of 4400, the length of the train data now becomes 17600.

Vectorization Vectorization is performed on the train and test data to convert text into numbers. The CountVectorizer class from scikit-learn is used for converting a collection of text documents to a matrix of token counts. It is a part of scikit-learn's feature extraction module. The resulting matrix is a sparse matrix where each row corresponds to a document, and each column corresponds to a unique word in the corpus. The values in the matrix represent the counts of each word in the respective documents.

Principal Component Analysis As the vectorized matrix is too sparse, the complex data is to be simplified for further modeling. This is done using Principal Component Analysis which identifies the directions, called principal components, in the data where the most variation or information is present.

4.2 Training the Models

The training data was trained on 3 models which are described below:

Model 1: SVC Model The SVC or Support Vector classifier model uses Support Vector Machines or SVM for classification tasks. SVC tries to find a hyperplane that best separates the data into different classes. It does this by identifying support vectors, which are the data points closest to the decision boundary or the hyperplane.

Model 2: Naive Bayes Model The Naive Bayes model is grounded by Bayes' theorem, a fundamental concept in probability theory. For classification of messages, Naive Bayes calculates the probability of a particular instance belonging to spam or ham. It does so by leveraging prior probabilities and likelihoods estimated from the training data.

Model 3: XGBoost Model XGBoost or eXtreme Gradient Boost is an ensemble learning method that builds a strong predictive model by combining the outputs of multiple weak models, typically decision trees. The gradient boosting aspect involves iteratively refining the model by emphasizing the correction of errors made by the previous models in the ensemble.

4.3 Results

The accuracy obtained on testing the test data on the SVC model is 0.7729. This may sound better but as the data is imbalanced, accuracy might not be the best metric to evaluate the model. Hence, confusion matrix used which clearly gives the number of true positives, false negatives, false positives and false negatives respectively in the matrix. These numbers were

$$\begin{bmatrix} 3243 & 513 \\ 487 & 161 \end{bmatrix}$$

It can be observed that the true negatives are very less, which means the model is not able to recognize spam messages properly. The False Positives and False Negatives are very high. Thus, it can be said that the model is not effective.

The accuracy of Naive Bayes Model on test data is 0.7525 which is less than that of SVC model. The resulting confusion matrix is

$$\begin{bmatrix} 3265 & 491 \\ 599 & 49 \end{bmatrix}$$

It can be observed that the number of True Negatives identified is lower when compared to SVC. The performance of Naive Bayes on this data is very low.

The final model, XGBoost showed an accuracy of 0.7895, higher than all the models. But, the classification matrix, as shown below, makes a clear comparison.

$$\begin{bmatrix} 3457 & 299 \\ 628 & 20 \end{bmatrix}$$

Even though the Model 3 has the best accuracy, it has performed the worst in comparison to the other two models. The number of true negatives identified are only 20 which shows that the model is least efficient.

Thus, it can be concluded that the classical ML Models are not sufficient to perform classification on Multi lingual messages.

5 Pre-Trained Models

As the classical Machine Learning models did not perform well on the multi-lingual data, we switched to the pre-trained models. The details of the working is described below.

5.1 Further Pre-Processing

Train and Test data The train test split is done before embedding the data because the train data will only contain English text as the encoding for all languages produces similar embeddings. The English sms are then tested along with the same rows of other languages so that no train data is repeated in testing. The ratio of train and test data is same as before, 80:20.

Converting Texts to Embeddings Tradition embedding schemes like TFIDF or Bag of words could have been used but in those techniques, the model would only support one language. Hence, we use a Multilingual model which is basically a pre-trained deep neural network model that produces same vectors for similar meaning of text in different languages. Sentence Transformers are applied on the SMS messages. Sentence Transformers are used to create word embeddings for multilingual data. distiluse-base-multilingual-cased-v2 is a pre-trained model which is a distilled version of the

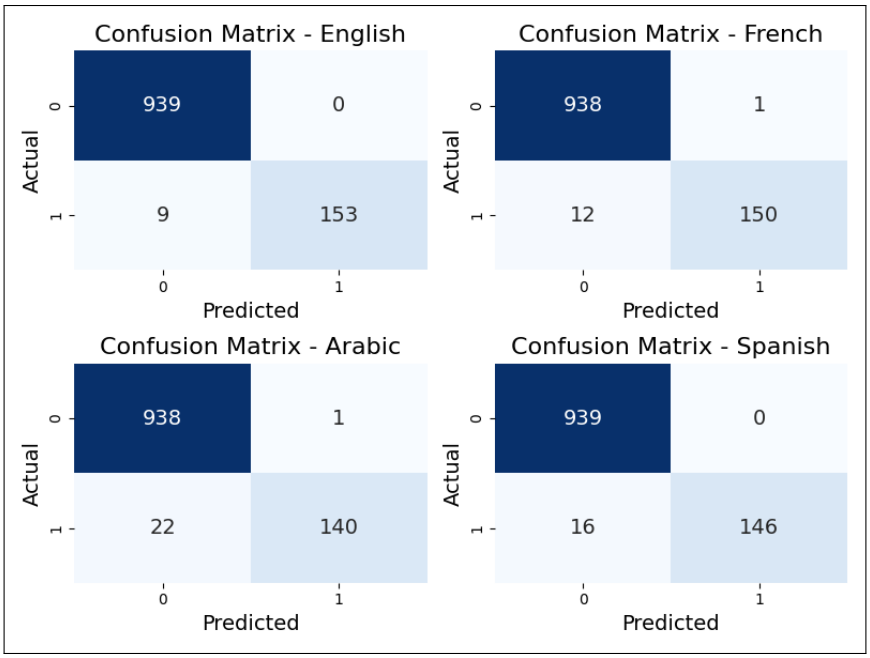

Figure 2: Confusion Matrices of SVC test data

Universal Sentence Encoder (USE) for multilingual sentence embeddings. The Universal Sentence Encoder is a pre-trained model developed by Google that converts sentences or short texts into fixed-size vector representations, capturing semantic meanings and contextual information. We use the distiluse-base-multilingual-cased-v2 model for the multilingual SMS data. It maps sentences and paragraphs to a 512 dimensional dense vector space. It can be noted that the train data here will only be 4400 as only English sentences are used in training. For test data, separate embeddings are created for each of the English, French, Arabic and Spanish languages as we will be testing them separately.

5.2 Training and Evaluation

Model 1: SVC Model The word embeddings generated by the sentence transformers as the train data are used to train a Support Vector Model. The accuracy obtained on the English test data is 99.18% which is much higher than the accuracy of classical ML SVC Model. The confusion matrix of the model on the english data can be seen in figure 5.2. The accuracy obtained on the French test data is 98.82% which is much higher than the accuracy of classical ML SVC Model. The accuracy of Arabic test data is 97.91% and that of Spanish test data is 98.55%. The confusion matrices of all the languages on the SVC model shows that there are only a few false positives and false negatives and a high number of true rates are achieved.

For comparing the performance of different languages on the model, a count plot of the types of predictions is drawn for all the languages as shown in Figure 5.2.

Model 2: Naive Bayes Model The word embeddings generated by the sentence transformers as the train data are used to train a Naive Bayes Model. The accuracy obtained on the English test data is 98.09% which is much higher than the classical Naive Bayes Model, but the accuracy is less than the previous pre-trained SVC model. The accuracy of French data is 97.18%, Arabic data is 97.28% and that of Spanish test data is 97.55%. The confusion matrices of all the languages on the Naive Bayes model is shown in figure 5.2 conveys that there are only a few false positives and false negatives and a high number of true rates are achieved. It can also be seen that the postive rates are slightly less than the pre-trained SVC model.

For comparing the performance of different languages on the model, a count plot of the types of predictions is drawn for all the languages as shown in Figure 5.2.

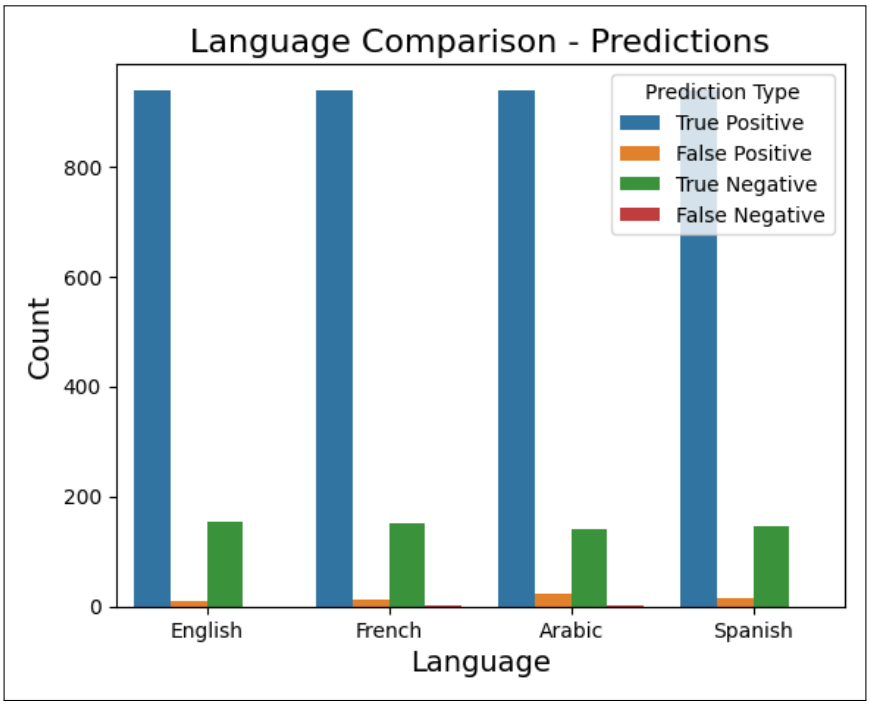

Figure 3: Countplot of performance of SVC on test data

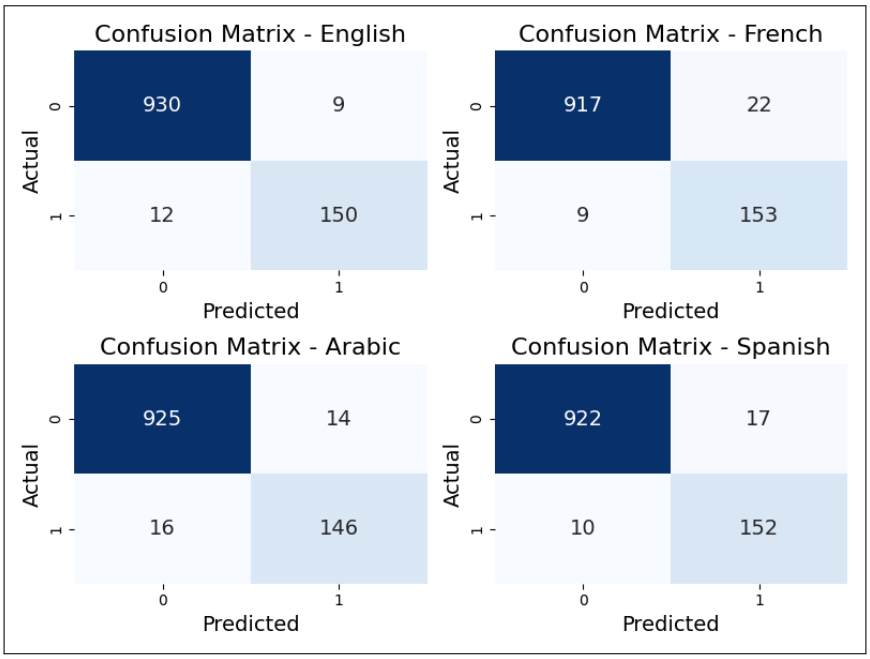

Figure 4: Confusion Matrices of Naive Bayes test data

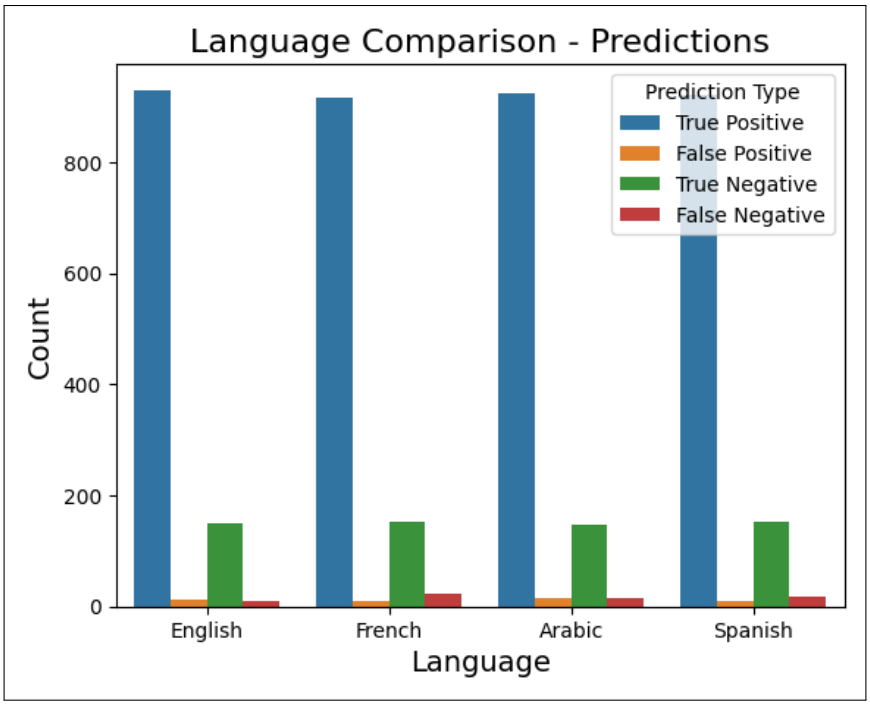

Figure 5: Countplot of performance of Naive Bayes on test data

Model 3: XGBoost Model The word embeddings generated by the sentence transformers as the train data are used to train an XGBoost Model. The accuracy obtained on the English test data is 98.64% which is much higher than the classical Naive Bayes Model, but the accuracy is less than the previous pre-trained SVC model and slightly higher than pre-trained Bayes model. The accuracy of French data is 97.73%, Arabic data is 96.55% and that of Spanish test data is 97.09%. The confusion matrices of all the languages on the Naive Bayes model is shown in figure 5.2 conveys that there are only a few false positives and false negatives and a high number of true rates are achieved. It can also be seen that the postive rates are slightly less than the pre-trained SVC model.

For comparing the performance of different languages on the model, a count plot of the types of predictions is drawn for all the languages as shown in Figure 5.2.

5.3 Comaprison of the three pre-trained models

From the figure 5.3 it can be observed that SVC has the least (almost 0) False negatives, while Bayes Model has the highest number of false negatives. XGBoost counts for the most number of False Positives. Hence, it can be said that the Support Vector Classifier model performed the best followed by the XGBoost model while the Naive Bayes model performed the least.

6 Comparison

It is evident from the work that the pre-trained models performed the best in comparison to the classical models. The count plot of the comparison of the models is as shown in Figure 6. It can clearly be observed from the chart that the Classical ML Models without Pre-trained transformers have performed the worst. The Classical ML Models have high number of False Positives and False Negatives and low count of True Negatives. The Models with pre-trained encoding gave very high accuracies and required training on only one language. With Classical ML Models, all the interested languages data was trained but the accuracies reported are very low.

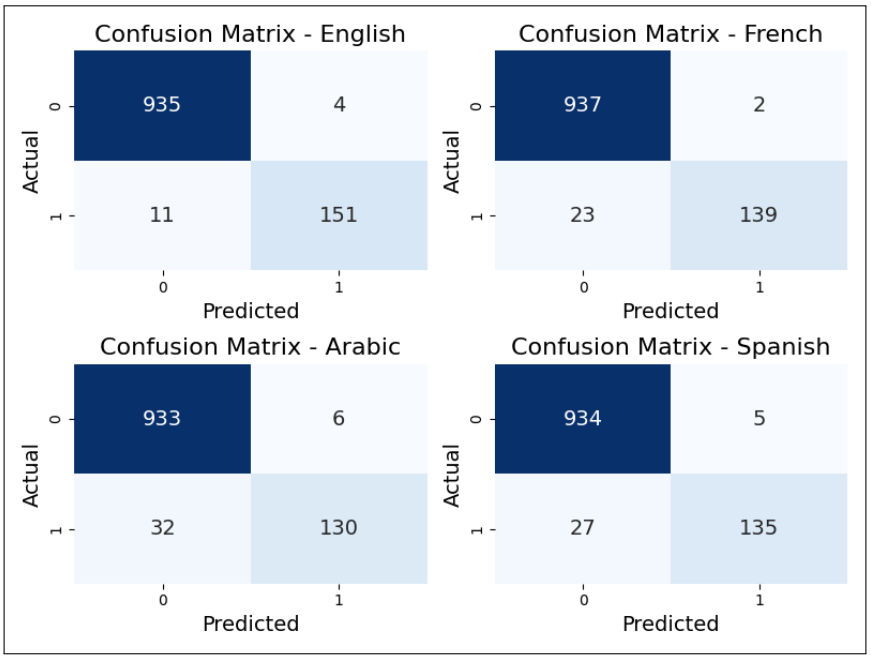

Figure 6: Confusion Matrices of Naive Bayes test data

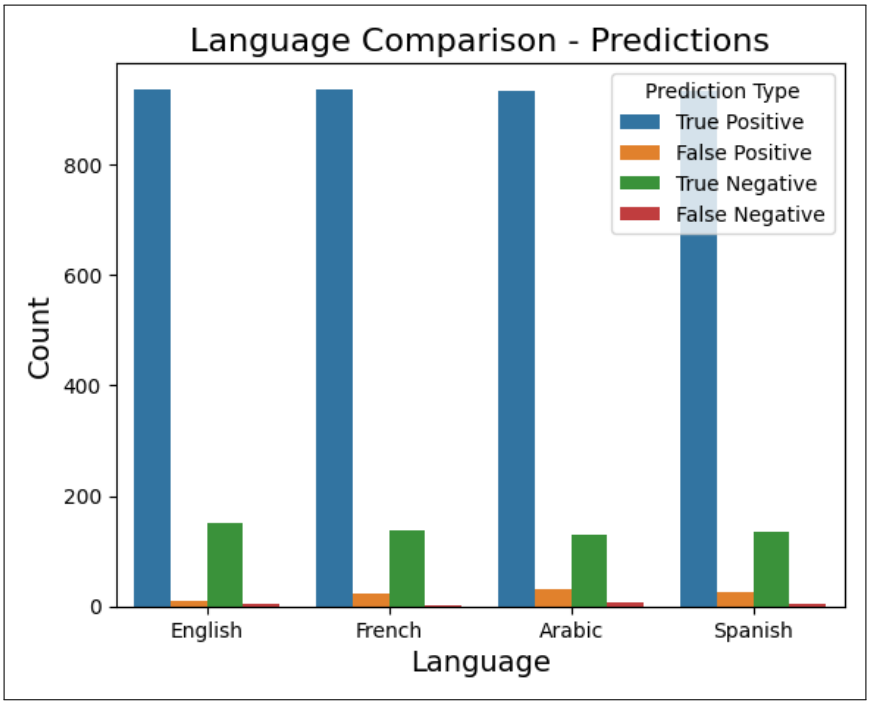

Figure 7: Countplot of performance of Naive Bayes on test data

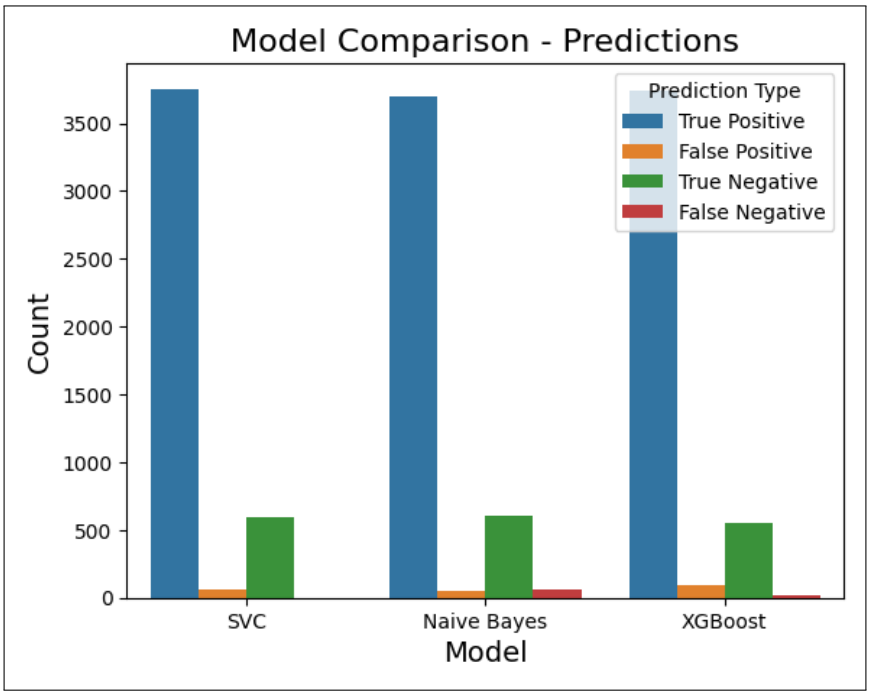

Figure 8: Countplot for Comparison of pre-trained models

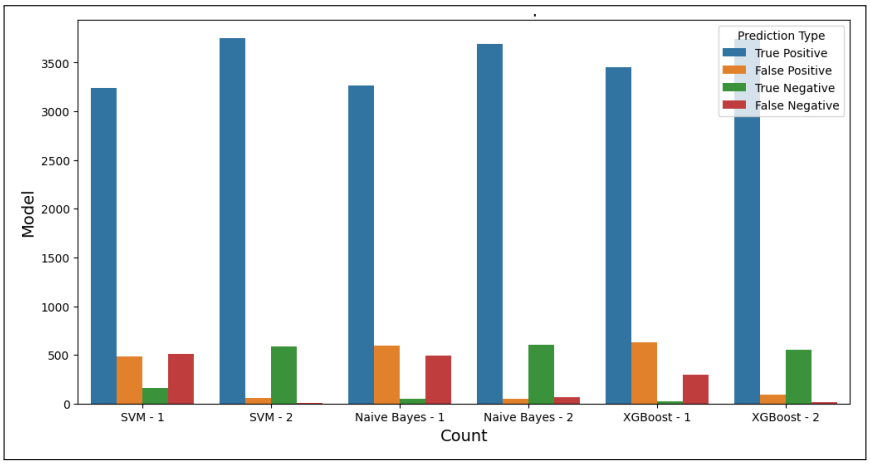

Figure 9: Countplot for Comparison of classical and pre-trained models

7 Discussion

The pre-trained machine learning models heavily outperformed the classical machine learning models. This shows that multilingual data cannot be handled in a regular manner and requires effective modelling to give close to accurate results. The best model using the pre-trained DistilUSE encoder is SVC which gave more than 97% accuracy on the individual languages while the mean accuracy of the SVC on all languages is more than 98.5%. It can be concluded from our study that from all the trained models the SVC model with DistilUSE encoder is the most efficient for multilingual spam detection. As a part of future work, the models can be tested on the other languages in the dataset along with training the data on different models.

References

[1] Rawat, A., Behera, S., & Rajaram, V. (2022). Email Spam Classification Using Supervised Learning in Different Languages. *2022 International Conference on Computer, Power and Communications (ICCPC), 294–298*. doi:10.1109/ICCPC55978.2022.10072054

[2] E, R., K, S., & Sharma, A. (2022). Multi-lingual Spam SMS detection using a hybrid deep learning technique. *2022 IEEE Silchar Subsection Conference (SILCON), 1–6*. doi:10.1109/SILCON55242.2022.10028936

[3] Ghourabi, A., & Alohaly, M. (2023). Enhancing Spam Message Classification and Detection Using Transformer-Based Embedding and Ensemble Learning. Sensors, 23(8). doi:10.3390/s23083861

[4] Dewi, D. A. C., Shaufiah, & Asror, I. (2018). Analysis and implementation of cross lingual short message service spam filtering using graph-based k-nearest neighbor. *Journal of Physics: Conference Series, 971(1), 012042*. doi:10.1088/1742-6596/971/1/012042

[5] Alzahrani, A., & Rawat, D. B. (2019). Comparative Study of Machine Learning Algorithms for SMS Spam Detection. 2019 SoutheastCon, 1–6. doi:10.1109/SoutheastCon42311.2019.9020530

[6] Santoso, I. (2022, April). Short Message Service Spam Detection Using BERT. *In International Conference on Big Data Engineering and Technology (pp. 37-45)*. Cham: Springer International Publishing.