Text Classification Models Using Natural Language Processing

Introduction

This report presents the development and evaluation of several text classification models for a dataset obtained from the NLP Getting Started competition on Kaggle. The goal is to classify tweets into binary categories indicating whether a disaster is occurring or not. The models employed include Logistic Regression, Random Forest, Support Vector Machine (SVM), and Recurrent Neural Networks (RNN) utilizing LSTM (Long Short-Term Memory) layers.

Data Pre-processing

The initial step involved loading the dataset and performing text pre-processing, which included:

1. **Removing URLs:** Extracted URLs from the text.
2. **Cleaning Text:** Removed special characters, punctuation, and numbers, then converted text to lowercase.
3. **Stopword Removal:** Eliminated common words that do not contribute to the meaning of the text while keeping certain pronouns.
4. **Lemmatization:** Converted words to their base forms using the WordNet Lemmatizer.

The cleaned text was then split into features (`X`) and target labels (`y`), and subsequently into training and testing sets with an 80-20 split.

Model Training and Evaluation

1. Logistic Regression

Logistic Regression is favored for its **simplicity** and **interpretability**, making it easy to understand how features contribute to predictions. Its **computational efficiency** allows it to handle **large datasets** effectively, making it a solid **baseline model** for comparison with more complex approaches. Additionally, it is well-suited for **binary classification** tasks, such as distinguishing between **disaster** and **non-disaster-related tweets**.

2. Random Forest Classifier

The **Random Forest Classifier** utilizes **ensemble learning** by combining multiple **decision trees** to enhance performance and **robustness** against overfitting. It provides valuable insights into **feature importance**, helping to identify which words are most influential in

predictions. This model excels at capturing **non-linear relationships** in the data and is **robust to noise** and **missing values**, making it a strong choice for text classification tasks.

3. Support Vector Machine (SVM)

Support Vector Machine (SVM) is an excellent choice for **high-dimensional data**, making it particularly effective in text classification scenarios where feature spaces can be vast. It employs **margin maximization** to find the **optimal hyperplane** that separates classes, resulting in better **generalization** on unseen data. SVMs are inherently designed for **binary classification**, which aligns well with the task of identifying **disaster-related content** in tweets.

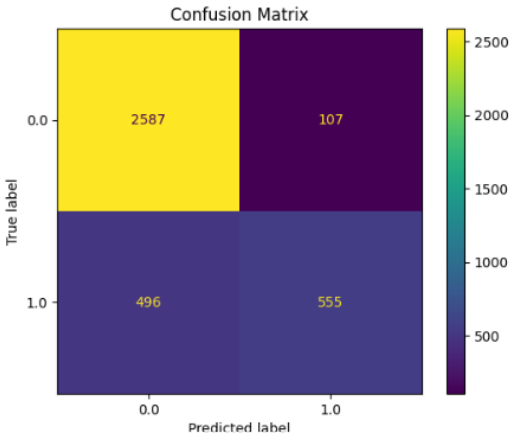
4. Recurrent Neural Network (RNN)

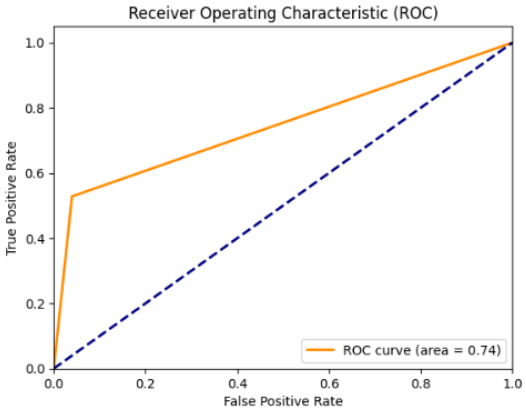
Recurrent Neural Networks (RNNs) are ideal for handling **sequential data**, allowing them to capture **dependencies** and **context** within text effectively. Utilizing **LSTM layers** addresses the common **vanishing gradient problem**, enabling the model to learn **long-term dependencies** crucial for understanding nuanced language. RNNs can accommodate **variable input lengths**, making them particularly suitable for processing tweets of varying sizes and achieving **state-of-the-art performance** in **natural language processing** tasks.

Results

The evaluation of each model yielded the following metrics:

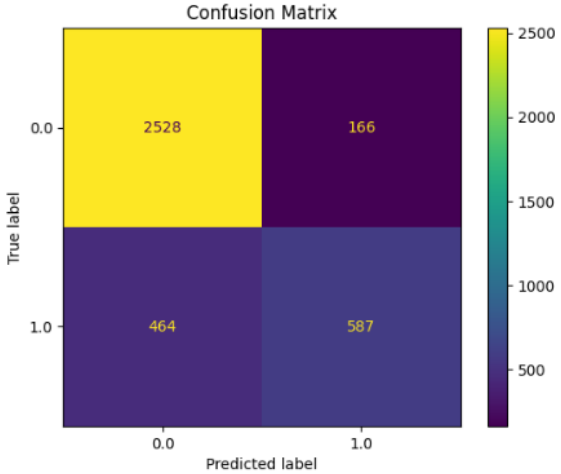
- **Logistic Regression:**
  - Accuracy: 0.84
  - Precision: 0.84
  - Recall: 0.53
  - F1-Score: 0.65
  - Specificity: 0.96
  - AUC-ROC:0.74

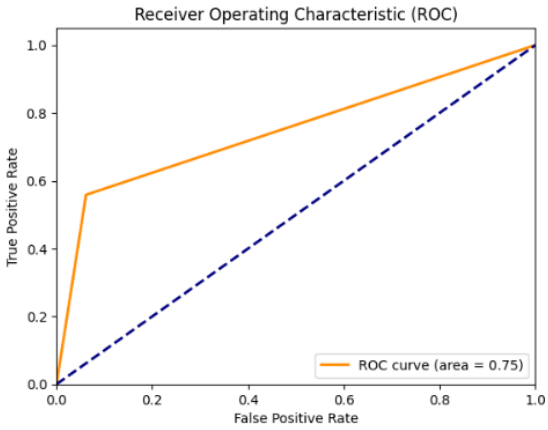




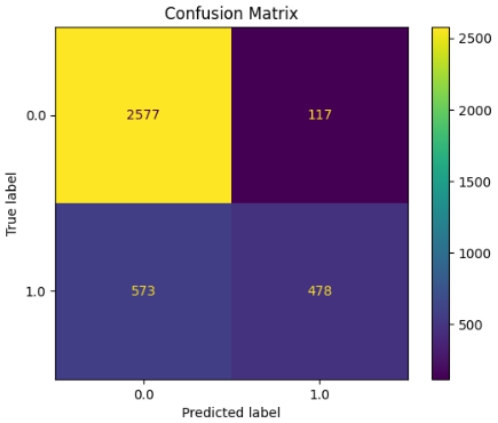
High accuracy (0.84) and precision (0.84).
Lower recall (0.53) but good specificity (0.96) and AUC-ROC (0.74).
Good at correctly identifying non-disaster tweets (high specificity) but misses some disaster-related tweets (lower recall).

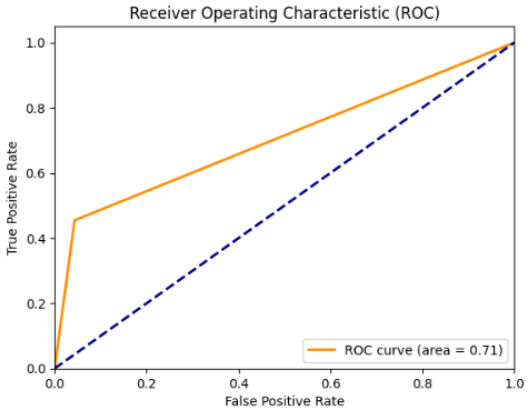
- **Random Forest:**
  - Accuracy: 0.83
  - Precision: 0.78
  - Recall: 0.56
  - F1-Score: 0.65
  - Specificity:0.94
  - AUC-ROC:0.75



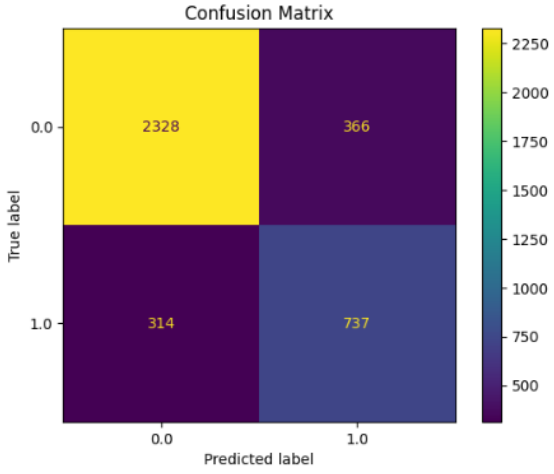


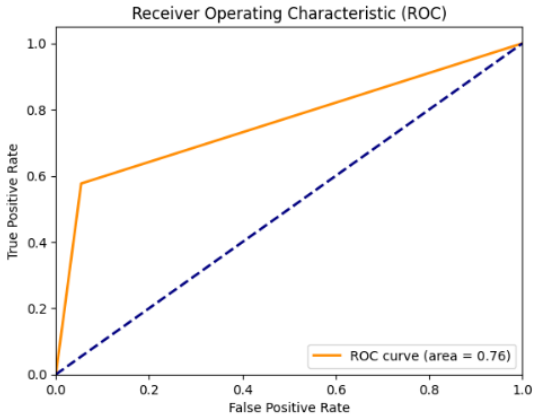
- **XGBoost**
  - Accuracy: 0.82
  - Precision: 0.80
  - Recall: 0.45
  - F1-Score: 0.58
  - Specificity:0.96
  - AUC-ROC:0.71



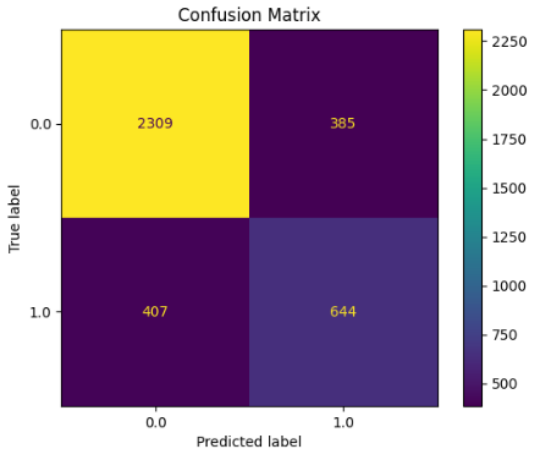


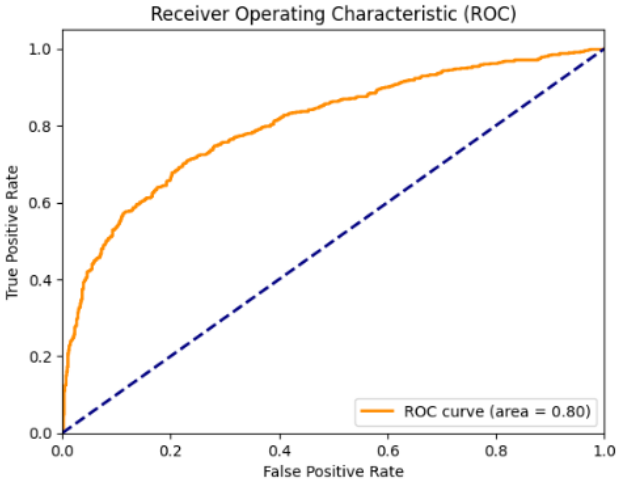
- **SVM:**
  - Accuracy: 0.82
  - Precision: 0.81
  - Recall: 0.70
  - F1-Score: 0.68
  - Specificity:0.86
  - AUC-ROC:0.78



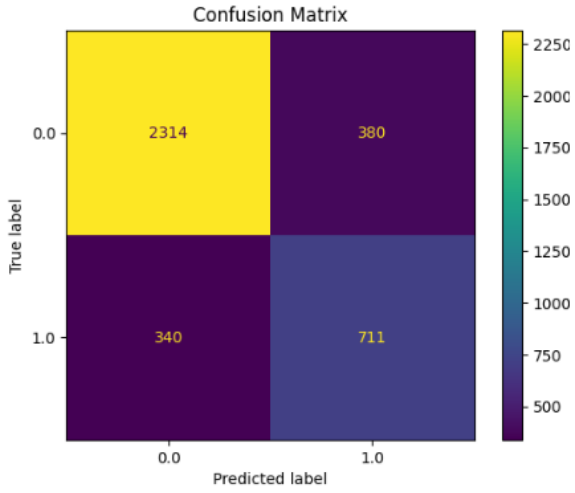


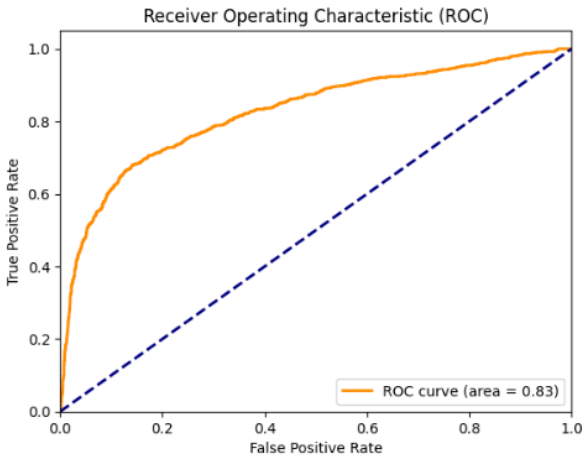
- **RNN :**
  - Accuracy: 0.79
  - Precision: 0.63
  - Recall: 0.61
  - F1-Score: 0.62
  - Specificity:0.86
  - AUC-ROC:0.80





- **LSTM :**
  - Accuracy: 0.81
  - Precision: 0.65
  - Recall: 0.68
  - F1-Score: 0.66
  - Specificity:0.86
  - AUC-ROC:0.83





Best Model:

- **SVM** appears to be the best choice for this disaster text classification task due to its well-balanced metrics:
  - High recall (0.70), meaning it catches more disaster-related tweets.
  - Good precision (0.80) and F1-score (0.75), indicating an overall balanced performance.
  - Solid AUC-ROC (0.79), which measures the trade-off between true positive and false positive rates well.

Conclusion

The project successfully implemented various models for text classification tasks. Among all tested models, the RNN with LSTM layers provided the highest accuracy and F1-Score. The evaluation metrics and confusion matrices indicate that the models perform well on the dataset, making them suitable for classifying tweets related to disasters. Further work may include hyper parameter tuning and the use of ensemble methods for improved performance.