Comparing traditional classification methods to Pre-trained DistilBERT base uncased(hugging face) transformer model for contextual disaster tweets classification

Achyuth Balaji
Department of Computer Science
University of Windsor
Windsor, Canada
bachyuth@uwindsor.ca

Aliasgar Khozem Palgharwala
Department of Computer Science
University of Windsor
Windsor, Canada
palghar@uwindsor.ca

Jay Hemalkumar Shah
Department of Computer Science
University of Windsor
Windsor, Canada
shah4v@uwindsor.ca

***Abstract*—Social media has recently risen to prominence as one of the main forms of communication. Among these platforms, Twitter is very popular where people tend to update and share various incidents in near real-time. When these updates are related to any disastrous incidents government agencies can use this data to analyze the ongoing situation and take necessary steps. As it is very hard to classify such a huge amount of data text data in a short period of time it becomes important to prepare models that are able to accurately classify text as "related to natural disastrous" or "not related to natural disastrous" based on the context of the tweet. In our paper, we have tried to do a comparative analysis of various classification methods and their performance.**

I. Introduction

With 100 million daily active users and 500 million tweets posted each day, Twitter, a social networking website introduced in 2006, is without a question one of the most well-known social media platforms available today. Twitter is a social media platform with the main goal of connecting users and enabling them to express their ideas to a large audience. Users can follow people or businesses who post content they enjoy reading, learn about the greatest news and events happening right now, or just use Twitter to connect with friends[1].

Twitter has gained popularity due to how simple it is to read, write, and collect data that is constantly published. Twitter data (also know as tweets) is a rich source of information on a large set of topics. This data can be used to find trends related to a specific keyword, measure brand sentiment or gather feedback about new products and services[2].

Twitter users can change their statuses and broadcast their thoughts, feelings, and observations. Recent studies in the areas of marketing, social sciences, natural language processing (NLP), opinion mining, and predictive analysis have all focused on Twitter[3].

The connective action theory, which ties a live event with user reactions, can be applied to Twitter data to forecast and monitor events[4]. The immediacy of utilizing the network to track news in real-time is one of the main reasons why people claim they use Twitter and the most typical kind of news that is followed, even though individuals end up following news in general on Twitter and use the social network "to pass the time"[5]. When it comes to sharing information on natural catastrophes like fires, floods, hurricanes, and earthquakes, Twitter can be utilised as a substitute platform. Recent studies have also shown that Twitter can serve as a source of information for disseminating knowledge on ecological events with clearly defined temporal patterns[6].

Natural Language Processing (NLP) is an area within artificial intelligence and can be explained as the process to read, analyze, and understand large amount of text data. The goal of Natural Language Processing (NLP) is to comprehend human language through reading texts and gain insightful knowledge from it. The usage of Natural Language Processing (NLP) will only become more crucial and is thought to be the future for interpreting unstructured text given the enormous volume of user-generated information that is produced every day through various social media platforms[7].

Determining the contextual meaning of keywords in tweets has become crucial as the use of social media grows. It might not be the best strategy to categorize tweets solely based on keywords. To address this issue, we tested different classification techniques against the DistilBERT-base-uncased transformer model in the Hugging face transformers library. Tweet classification for distaste/offensive content has been chosen as the use case to show the effectiveness of our pipeline. We can alert the relevant government agencies, as well as the people in that area if we can locate the actual harmful tweets.

II. Related Works

Before conducting this experiment, we have researched several papers related Natural Language Processing for Twitter data, disaster tweet recognition, crisis classification and contextual crisis embedding and hate speech detection using transformers.

The paper titled “Identification of Disaster-related tweets using Natural Language Processing” written by Shriya Goswami and Debaditya Raychaudhuri has tried to analyze and perform text mining on Twitter text datasets. It has tried to classify tweets into 2 categories – disaster-related and not disaster-related. It can be safely concluded from the experimental findings of this paper that Twitter data can be satisfactorily classified into categories with a good accuracy rate. A shortcoming of this experiment is that if we try to decrease the frequency of the appearance of terms in the entire corpus beyond the lower threshold of 0.5-1% the performance might suffer. Another room for improvement is that the accuracy can be improved by further lowering the False Positive and False Negative values[8].

A review article ”Detection of informative tweets in crisis events” written by Anna Kruspe, Jens Kersten, and Friederike Klan gave an overview of current methods to detect tweets pertaining to disaster events. There are three main ways to approach this problem: Filtering tweets by characteristics such as location and contained keywords or hashtags, crowdsourcing, and machine-learning-based methods. Each of these has its advantages and disadvantages, but machine learning appears to be the current main avenue of research with big improvements in the past few years. To train and test these 330 models, various data sets spanning one or multiple events have been created and are available online. However, these usually only provide ids of the tweets, which leads to changes in the data sets over time[9].

In another review artticle titled ”Using Twitter Data to Monitor Natural Disaster Social Dynamics: A Recurrent Neural Network Approach with Word Embeddings and Kernel Density Estimation”, a methodology that uses Twitter as a social sensor is proposed. This is accomplished by employing an information sequential extraction procedure known as Named Entity Recognition (NER), which aims to describe mentioned entities, such as places, persons, and organizations. The methodology considers the semantic, morphological, and contextual information about each word composing a tweet and its surrounding context, thus allowing to properly identify a named place (toponym). To achieve this, words are tokenized and transformed into word embeddings to represent them as vectors with rich syntactic and semantic relationships that are established by neighboring words. To ensure that high classification accuracy of the sequential data is achieved without heavily relying on handcrafted feature extraction techniques, a Recurrent Neural Network variant, i.e., a Bidirectional Long Short-Term Memory (biLSTM) network, is used[10].

For work related to transformer methods and Natural Language Processing, we studied the paper ”Hate Speech Detection in Twitter using Transformer Methods”. In this paper, they have explored several transformer-based methods for hate speech detection. They have evaluated the effectiveness of our method using six state-of-the-art metrics. The results showed that the DistilBERT, a distilled version of BERT, outperforms all transformer-based baseline methods and the attention-based LSTM explored in this study. They, therefore, conclude that the proposed method can be used to learn effective information for the classification of hate speech in resource-constrained environments because it is computationally inexpensive. In addition, transformers facilitate transfer learning, allowing them to be used where training data is limited[11].

Additionally, we studied an article titled ”Exploring the Potential of Twitter Data and Natural Language Processing Techniques to Understand the Usage of Parks in Stockholm”, which helped us understand the true potential of using Natural Language Processing on Social networking data such as Twitter. In this report a modern approach for exploring how the usage of parks can be understood, using NLP techniques on social media data, have been demonstrated. The results show that it is possible to understand what attitudes and activities are associate with visiting parks using NLP techniques on social media data[12].

Furthermore, the paper ”CRISISBERT: A ROBUST TRANSFORMER FOR CRISIS CLASSIFICATION AND CONTEXTUAL CRISIS EMBEDDING” discusses a transformer-based crisis classification model CrisisBERT, and a contextual crisis-related tweet embedding model Crisis2Vec. The authors conduct tests with three datasets and two crisis categorization tasks to test the effectiveness and robustness of the suggested models. Through increasing from 6 classes to 36 classes with only 51.4% more data points, experimental results show that CrisisBERT enhances the state-of-the-art for both detection and recognition classes. Finally, their tests using two classification models demonstrate that Crisis2Vec improves classification performance in comparison to Word2Vec embeddings, which are frequently employed in earlier publications[33].

Lastly, ”A Comparison of Pre-Trained Language Models for Multi-Class Text Classification in the Financial Domain” the authors looked on challenges involving multi-class text classification in the finance industry. They evaluated the effectiveness of various generic PLMs using both private and public datasets of actual financial documents. They then evaluated FinBERT, a PLM designed specifically for the financial industry, for its added value. On our financial document classification assignment, they discovered that FinBERT was unable to outperform the generic PLMs. They looked at whether FinBERT’s performance may be enhanced by a unique lexicon. Their research demonstrates that it did not[13].

Studying all of these previous works helped up understand the potential and value of social media data and the need for contextual analysis of this data using Natural Language Processing and the use of transformers and pretrained models compared to traditinal classification methods.

III. Methodology

In this experiment, we have first cleaned and vectorized the dataset to apply various classification methods on the dataset. Lastly, we have used the DistilBERT base model (uncased) from Hugging face transformers library and compared its results to the classic classification methods.

A. Dataset

The dataset is constructed of 7613 rows of data with 5 columns, namely, ID, keyword, location, text(tweet text), and target. We have considered the values from columns text and target for our experiment. The text column contains the tweet itself while the target column contains values 0 or 1 where 0 indicates that the tweet is not related to a real disaster and 1 indicates that the tweet is related to a real disaster. In this dataset, 57.03% of tweets are not related to a real disaster while 42.97% of tweets are related to real disasters.

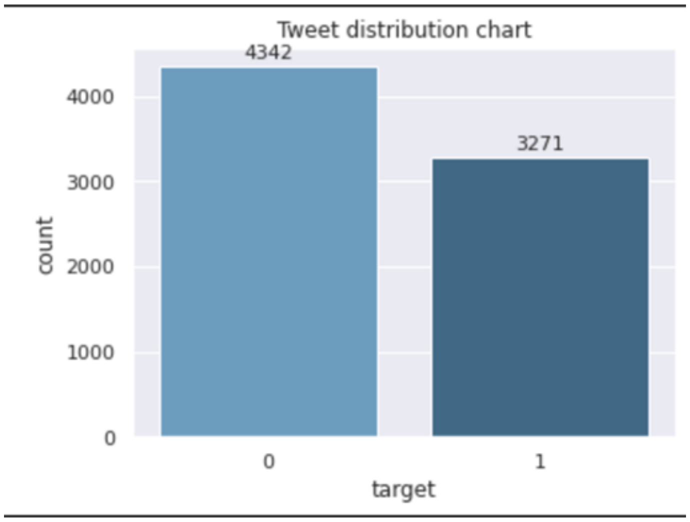
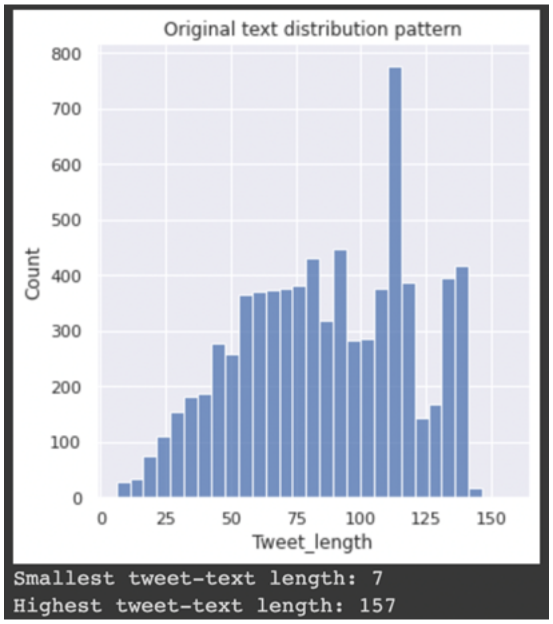


Fig. 1. Tweet Distribution Chart

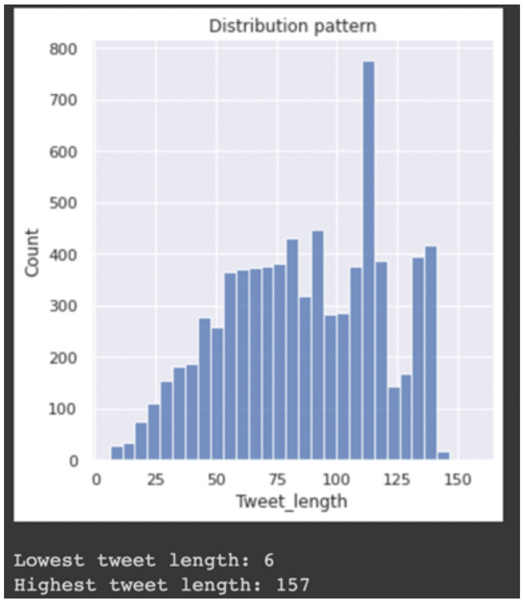
To be able to use the data in the text column for various machine learning models, we need to apply data pre-processing where the data is subject to a data cleaning pipeline. The data cleaning pipeline performs the following steps:

- Step 1: Text is converted to lower case.
- Step 2: All the HTML tags are removed with the help of the beautiful soup library.
- Step 3: All the HTTP and HTTPS text is removed.
- Step 4: Remove unwanted spaces(multiple spaces/tab spaces) in the text.
- Step 5: All the special characters are removed.
- Step 6: General abbreviations are expanded such as I've to I have.
- Step 7: Removed stop words.



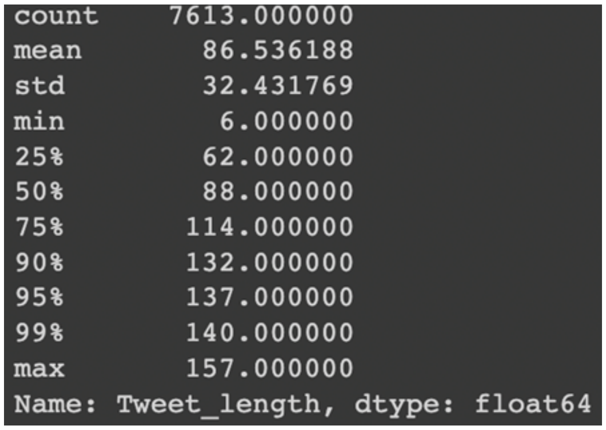
```
Smallest tweet-text length: 7
Highest tweet-text length: 157
```

Fig. 2. Original text length distribution pattern



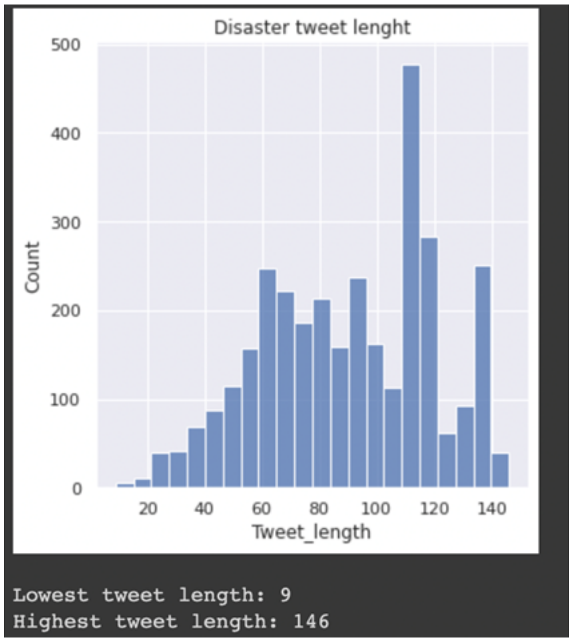
```
Lowest tweet length: 6
Highest tweet length: 157
```

Fig. 3. Cleaned text length distribution pattern



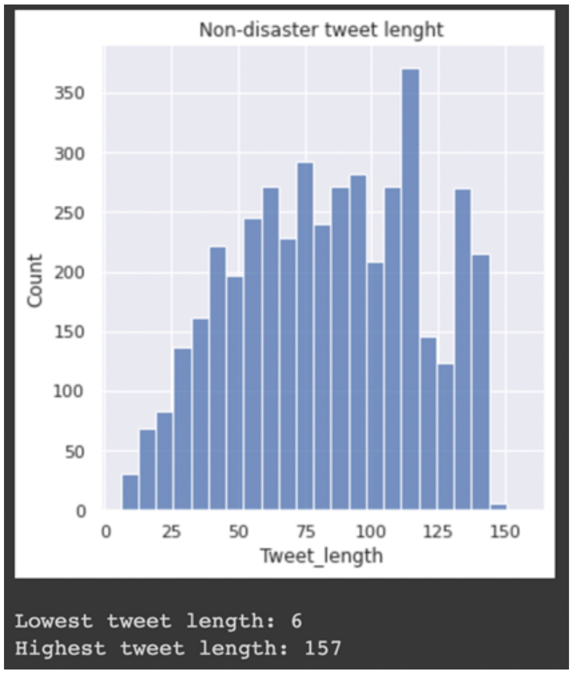
```
count    7613.000000
mean       86.536188
std        32.431769
min         6.000000
25%        62.000000
50%        88.000000
75%       114.000000
90%       132.000000
95%       137.000000
99%       140.000000
max       157.000000
Name: Tweet_length, dtype: float64
```

Fig. 4.



```
Lowest tweet length: 9
Highest tweet length: 146
```

Fig. 5. Disaster tweet length distribution charts.



```
Lowest tweet length: 6
Highest tweet length: 157
```

Fig. 6. Non-disaster tweet length distribution charts.

For training the models, only the cleaned text and output variable target are considered. The data is split into training and testing parts. Testing parts contain 30% of the data. The rest of the 70% is used for the training part. The number of rows in training data is 5329 and the number of rows in testing data is 2284. Input variables in the training and testing data are subjected to the vectorization process using TF-IDF vectorize method. These vectors are used for the classification process.

B. Data Vectorization

A vector is a set of real numbers that can be used to process natural language text and extract relevant information from the given word in a sentence using machine learning and deep learning techniques. The process of converting words into numbers are called Vectorization[14].

The TF-IDF vectorization converts textual data into numerical vectors while taking into account the frequency of each word in the document, the total number of words in the document, the total number of documents, and the number of documents containing each unique word[15].

C. Classification Methods:

a) SVM: A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. An SVM model can classify new text after being given sets of labelled training data for each category[16].

They offer two key advantages over more recent algorithms like neural networks: greater speed and improved performance with fewer samples (in the thousands). As a result, the approach is excellent for our text classification issues, where we have a dataset with a few thousand rows[16].

b) Logistic Regression: Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability[17].

A logistic regression model can be compared to a linear regression model, but a logistic regression utilizes a cost function that is more sophisticated and can be described as either the "sigmoid function" or the "logistic function" rather than a linear function[17].

c) XG Boost: XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. Artificial neural networks frequently outperform all other algorithms or frameworks in prediction issues involving unstructured data (pictures, text, etc.). However, decision tree-based algorithms are currently thought to be best-in-class for small- to medium-sized structured/tabular data[18].

d) Multinomial naïve base: Naïve Bayes is a probabilistic approach for constructing the data classification models. It deals with probability as the "likely" that data belongs to a certain class and is often used as an alternative to decision tree forests and distance-based K-Means clustering. The Gaussian and Multinomial models of the naïve Bayes exist[19].

D. Hugging face transformer

Hugging Face is a large open-source community that quickly became an enticing hub for pre-trained deep learning models, mainly aimed at NLP. Their core mode of operation for natural language processing revolves around the use of Transformers[20]. The Transformers library written in Python exposes a well-furnished API to leverage a plethora of deep learning architectures for state-of-the-art NLP tasks. All available models come with a set of pre-trained weights that one can fine-tune for their specific use[20].

The hugging face transformers perform the following steps:

- First step involves text tokenization. In this step, cleaned tweet text is converted to numerical vectors so that model can understand the sequence of words. In this process the words in sentences are converted are broken down into tokens, these token sequences are converted to tensor numerical tokens. In the tokenization process, certain special tokens are used such as [CLS] and [SEP] where CLS denotes starting of the sentence and SEP denotes the ending of the sentence. In order to tokenize "DistilBERT base model (uncased)" is used which is comparatively smaller than the full-fledged BERT model yet very fast and effective. Once the text is tokenized it is stored in list which is of the form ["input_ids", "attention_mask", "labels"]. (attention_mask helps the model to understand which tokens need to be focused on and which can be ignored). Optional step:- Embeddings can be extracted from the last hidden layer by feedings in the tensor inputs to the model and these embeddings can be used for the prediction of the class.
- In second step, Auto-trained classification model(AutoModelForSequenceClassification) with classification header is used for finetuning the training model. The training parameters such as the number of epochs, learning rate, number of label classes, etc are passed to the model as parameters. Once the model is initialized with the necessary parameters training procedure starts executing and the model gets trained. Then classes are predicted for the test data.
- Lastly, Accuracy is computed and the best model is finalized based on the efficiency of the models.

a) DistilBERT base model (uncased): DistilBERT is a transformers model that is smaller and faster than BERT and was pretrained on the same corpus using the BERT base model as a teacher. This means that it was pretrained using only the raw texts, with no human labelling of them in any way (which explains why it can use a large amount of data that is readily accessible to the public), and an automatic process to automatically generate inputs and labels from those texts using the BERT base model[21]. It was pretrained with three goals in mind, specifically:

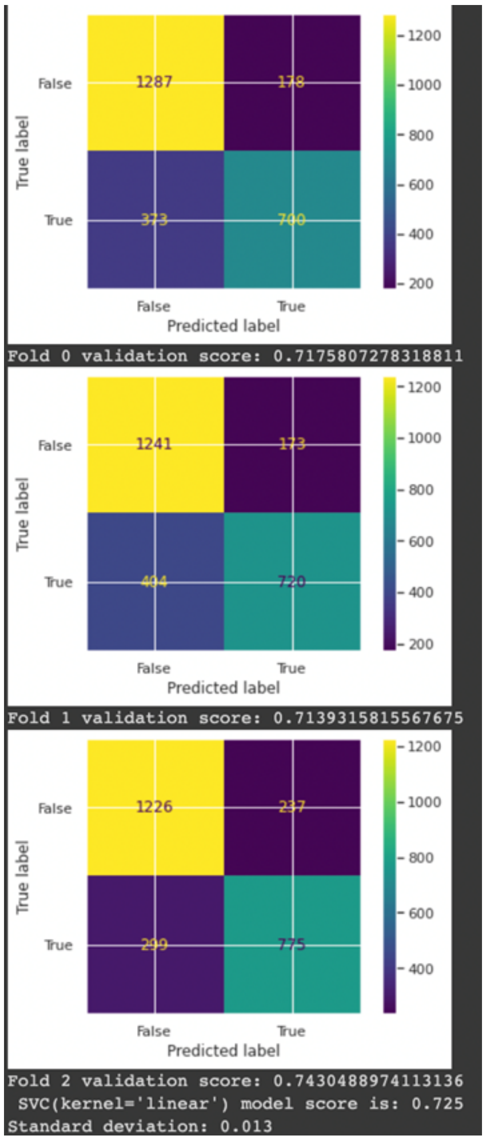
- Distillation loss: The model was trained to output probabilities identical to those of the BERT base model[21].
- Masked language modeling (MLM): This is a part of the BERT basic model's initial training loss. When given a sentence, the model randomly selects 15% of the input words to be masked, after which it must predict the words that were masked. This contrasts with conventional recurrent neural networks (RNNs), which typically perceive the words sequentially, and autoregressive models like GPT, which internally conceal the next tokens. This makes it possible for the model to learn a two-way representation of the statement[21].
- Cosine embedding loss: The model was trained to produce hidden states that were as similar to the BERT base model as possible[21].

By doing so, the model acquires the same internal representation of the English language as its instructor model while performing downstream or inference tasks more quickly.

IV. Experiments and Results

In this experiment, we have used 4 methods from the basket of classic machine learning classification techniques. As the results were not satisfactory, we have also used transformer models and finetuned the language models to suit our use case. Machine learning models are coupled with a k-fold (chosen k value is 3) method used to determine the model efficiency. To extract textual features in numerical vector form TF-IDF[22] method was used. We have used SK-learn TF-IDF [23] library and parameter setting are as follows max_features=1000,ngram_range=(1,3). As our dataset is not equally distributed (imbalanced) we used F1-score to determine our model efficiency[24].

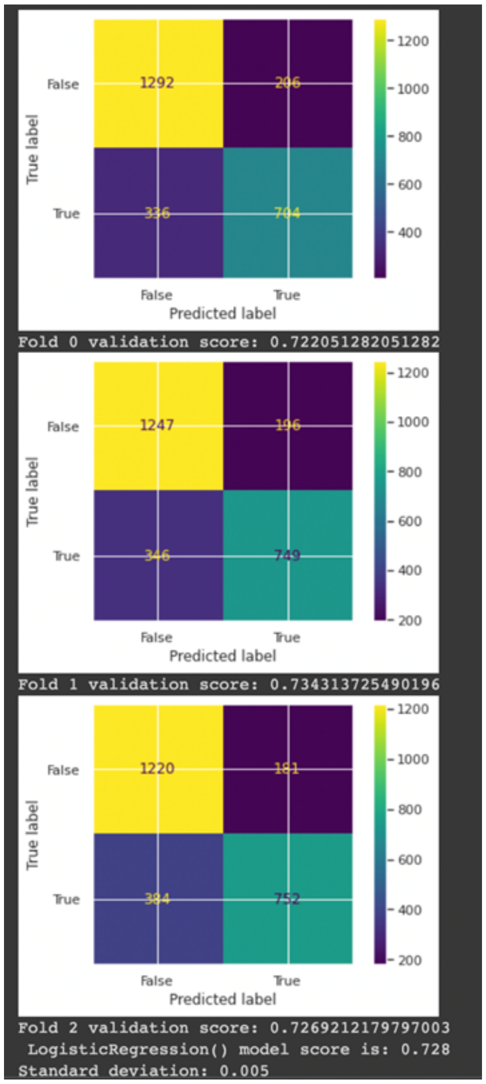
a) SVM: We conducted experimentation using sklearn-SVM library [25]. Additionally, the model's kernel functions were changed to see the results. The best result was obtained for the Linear kernel. (As our problem is binary text classification generally linear kernel performs better[26]). The model's average f1-score was 72.5%. The results of the 3 folds along with the confusion matrix[27] are as follows:



```
Fold 0 validation score: 0.7175807278318811
Fold 1 validation score: 0.7139315815567675
Fold 2 validation score: 0.7430488974113136
 SVC(kernel='linear') model score is: 0.725
Standard deviation: 0.013
```

Fig. 7. SVM Results

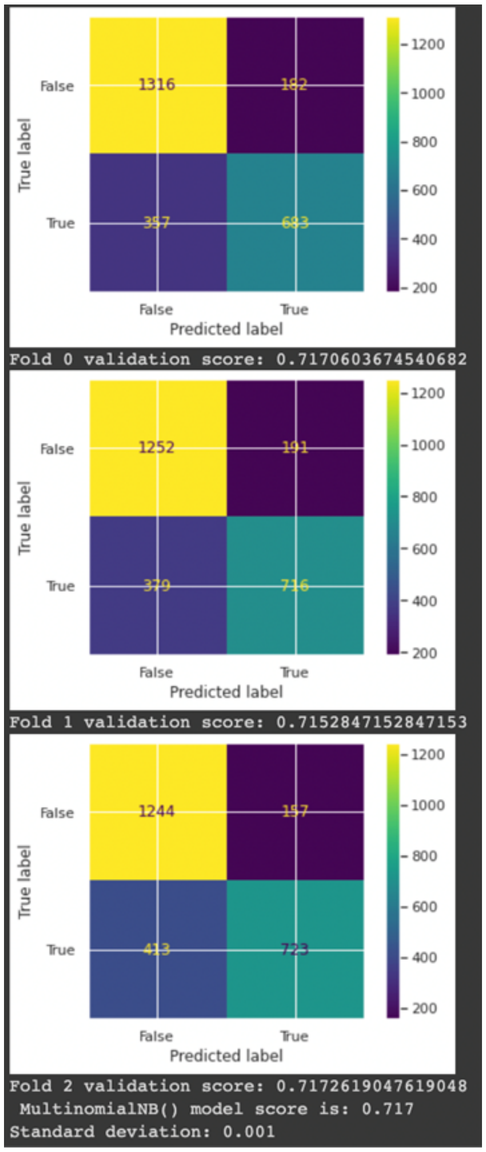
b) Logistic Regression: We have used learn logistic regression[28] model with the default configuration. The model's average f1-score was 72.8%. The results of 3 folds along with the confusion matrix are as follows:



```
Fold 0 validation score: 0.722051282051282
Fold 1 validation score: 0.734313725490196
Fold 2 validation score: 0.7269212179797003
 LogisticRegression() model score is: 0.728
Standard deviation: 0.005
```

Fig. 8. Logistic Regression Results

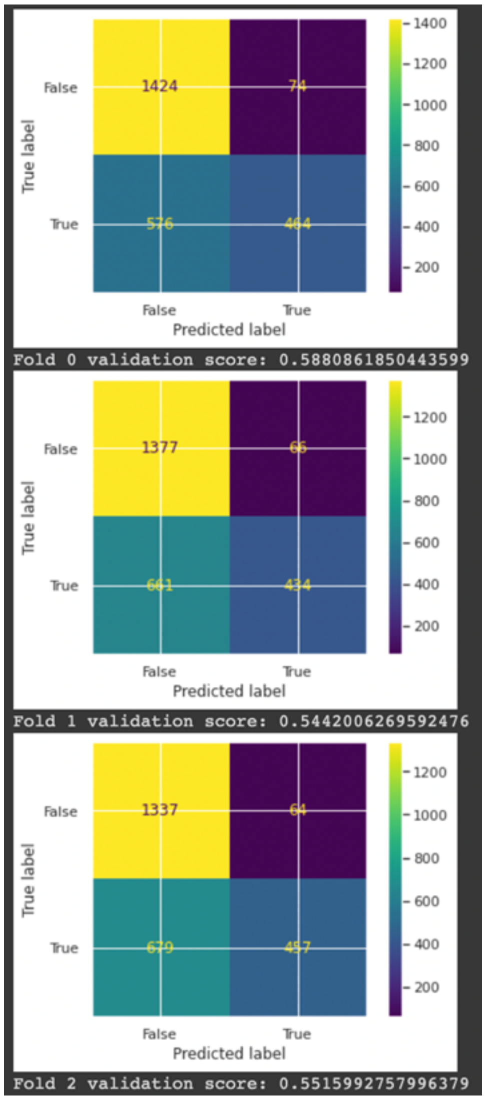
c) MultinomialNB: We have used the sklearn Multinomial Naïve based [29] model with the default configuration. The model's average f1-score was 71.7%. The results of the 3 folds along with the confusion matrix are as follows:



```
Fold 0 validation score: 0.7170603674540682
Fold 1 validation score: 0.7152847152847153
Fold 2 validation score: 0.7172619047619048
 MultinomialNB() model score is: 0.717
Standard deviation: 0.001
```

Fig. 9. MultinomialNB Results

d) XG-boost: XG-boost:- We have used the XGboost python package for the XG-boost model[30]. Models hyper parameter is tuned as follows n_jobs=4,maximize=True, feval=f1. The model's average f1-score was 56%. The results of the 3 folds along with the confusion matrix are as follows:



```
Fold 0 validation score: 0.5880861850443599
Fold 1 validation score: 0.5442006269592476
Fold 2 validation score: 0.5515992757996379
```

Fig. 10. XG-boost Results

e) Transformer model: Pre-trained DistilBERT-base-uncased(hugging face)[31] is used to tokenize and extract embeddings from the textual content. Embeddings 2-d plot looks as follows:

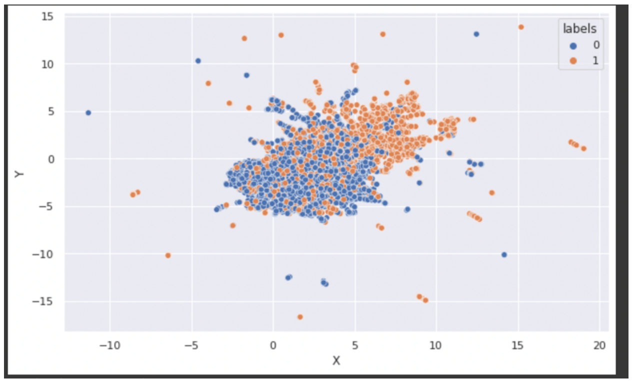


Fig. 11. Plot for transformer model

Text can be just classified using these embeddings without any fine-tuning. Just for extermination purposes, we tried that with the logistic regression model (as it was the best performing model for chosen data in classical Machine learning models) on these embeddings. The f1-score of the model was 75.4%.

We used AutoModelForSequenceClassification [32] transformer library for loading pre-trained DistilBERT model. This library takes 2 parameters the first parameter is the model path and the second parameter is the number of classes which is 2 for our data(i.e 0 and 1). We have used the trainer library to train to fine-tune the model hyperparameter and train the new model. We tuned the hyperparameters as follows: we chose 3 epochs, the learning rate was set to 2e-5 and weight decay was set to 0.01.

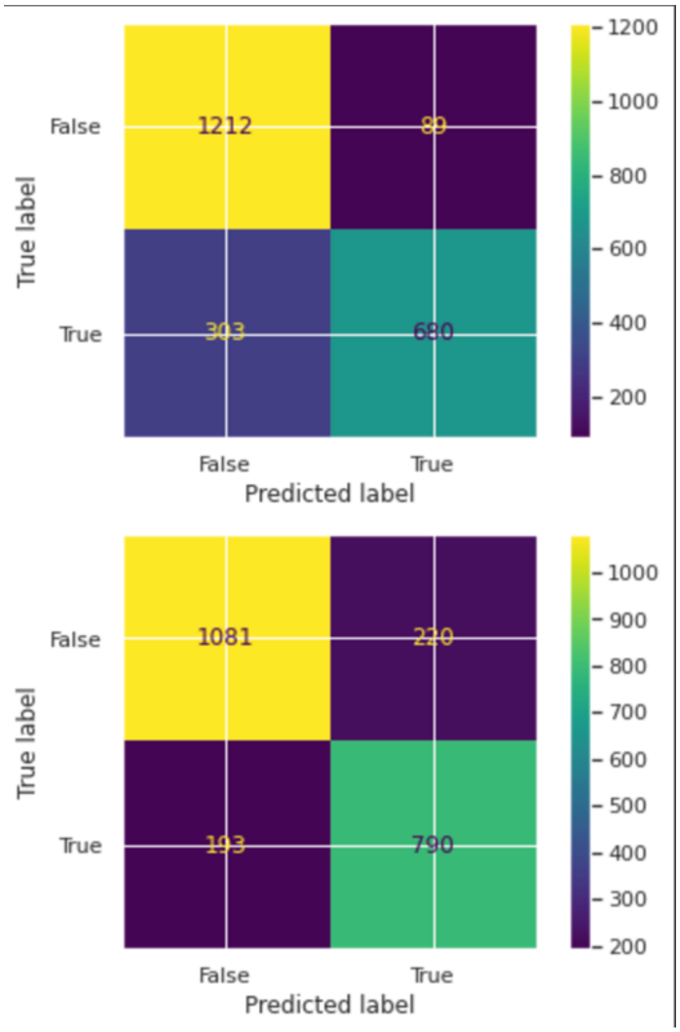


Fig. 12. Result for transformer model

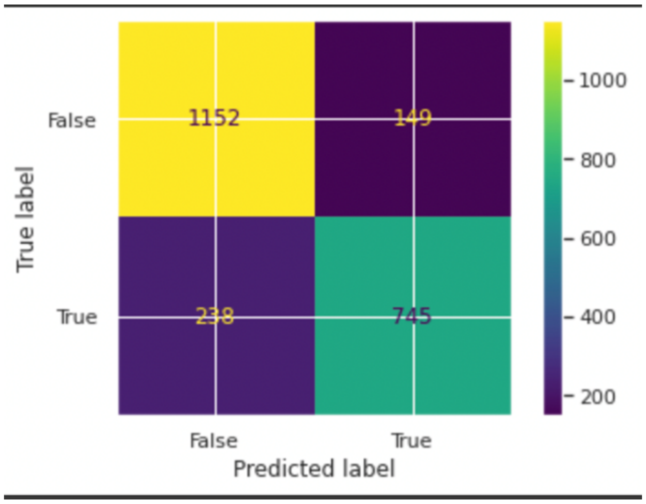
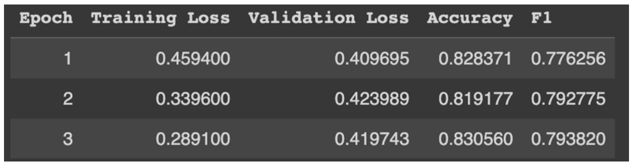


Fig. 13. Result for transformer model



Epoch	Training Loss	Validation Loss	Accuracy	F1
1	0.459400	0.409695	0.828371	0.776256
2	0.339600	0.423989	0.819177	0.792775
3	0.289100	0.419743	0.830560	0.793820

Fig. 14. Result for transformer model

The average model f1-score is 79%. The training tie was close to 70 mins on cpu.

V. Conclusion and Future Work

Using the power of NLP to understand the context of tweets accurately during a disastrous situation will help us to save lives. In order to achieve this goal of classifying tweets, our pipeline was able to do it with close to 80% accuracy using the transfer learning method with help of the DistilBERT base model. As we are using a pre-trained model we can extract knowledge from the vast training knowledge of the model and use it in our use case to save computation time. Though we do this still the computation time is comparatively high when executing on a CPU device. In the classical machine learning methods, Logistic regression and SVM performance was very similar. They had f1 scores of 72.8% and 72.5% respectively. Classification is done on embeddings generated by the DistilBERT model also performed relatively better than SVM and logistic regression with an f1 score of 75.4%. The embeddings from different pre-trained models can be used as a baseline to compare performance on the chosen datasets.

In the future, we are planning to train the datasets on various other pre-trained models and compare the results. We are also planning to build a full-fledged pipeline to handle any textual data classification problems. Additionally, we are also planning to decrease the computation time by using TPU(Tensor Processing Unit) and GPU as computation time on these devices would be less when compared to the CPU.

References

[1] Forsey, C. (2021, July 26). What is Twitter and how does it work? HubSpot Blog. Retrieved August 14, 2022, from https://blog.hubspot.com/marketing/what-is-twitter

[2] Shreyas, P. (2020, June 6). Tweet analytics using NLP. Medium. Retrieved August 14, 2022, from https://medium.com/analytics-vidhya/tweet-analytics-using-nlp-f83b9f7f7349

[3] Kursuncu, U.; Gaur, M.; Lokala, U.; Thirunarayan, K.; Sheth, A.; Arpinar, I.B. Predictive Analysis on Twitter: Techniques and Applications. In Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining; Springer: Cham, Switzerland, 2019; pp. 67–104.

[4] Pond, P.; Lewis, J. Riots and Twitter: Connective politics, social media and framing discourses in the digital public sphere. Inf. Commun. Soc. 2019, 22, 213–231.

[5] Rosenstiel, T. (2015, September 2). Twitter and news: How people use twitter to get news. American Press Institute. Retrieved August 14, 2022, from https://www.americanpressinstitute.org/publications/reports/survey-research/how-people-use-twitter-news/single-page/#twitter-and-breaking-news

[6] Hart, A.G.; Carpenter, W.S.; Hlustik-Smith, E.; Reed, M.; Goodenough, A.E. Testing the potential of Twitter mining methods for data acquisition: Evaluating novel opportunities for ecological research in multiple taxa. Methods Ecol. Evol. 2018, 9, 2194–2205.

[7] Lee, A. (2019, December 11). Why NLP is important and it'll be the future - our future. Medium. Retrieved August 14, 2022, from https://towardsdatascience.com/why-nlp-is-important-and-itll-be-the-future-our-future-59d7b1600dda

[8] Goswami, Shriya and Raychaudhuri, Debaditya, Identification of Disaster-Related Tweets Using Natural Language Processing: International Conference on Recent Trends in Artificial Intelligence, IOT, Smart Cities & Applications (ICAISC-2020) (May 26, 2020). Available at SSRN: https://ssrn.com/abstract=3610676 or http://dx.doi.org/10.2139/ssrn.3610676

[9] A. Kruspe, J. Kersten, F. Klan, 'Review article: Detection of actionable tweets in crisis events', Natural Hazards and Earth System Sciences, . 21, . 6, . 1825–1845, 2021.

[10] Hernandez-Suarez, Aldo & Sanchez-Perez, Gabriel Toscano, Karina & Perez-Meana, Hector Portillo-Portillo, Jose & Luis, Victor García Villalba, Luis. (2019). Using Twitter Data to Monitor Natural Disaster Social Dynamics: A Recurrent Neural Network Approach with Word Embeddings and Kernel Density Estimation. Sensors. 19. 1746. 10.3390/s19071746.

[11] Raymond T Mutanga, Nalindren Naicker and Oludayo O Olugbara, "Hate Speech Detection in Twitter using Transformer Methods" International Journal of Advanced Computer Science and Applications(IJACSA), 11(9), 2020. http://dx.doi.org/10.14569/IJACSA.2020.0110972

[12] Norsten, T. (2020). Exploring the Potential of Twitter Data and Natural Language Processing Techniques to Understand the Usage of Parks in Stockholm.

[13] Arslan, Y., Allix, K., Veiber, L., Lothritz, C., Bissyandé, T.F., Klein, J., Goujon, A. (2021). A Comparison of Pre-Trained Language Models for Multi-Class Text Classification in the Financial Domain. Companion Proceedings of the Web Conference 2021.

[14] Prabhu. (2019, November 21). Understanding NLP word embeddings - text vectorization. Medium. Retrieved August 14, 2022, from https://towardsdatascience.com/understanding-nlp-word-embeddings-text-vectorization-1a23744f7223

[15] Kilmen Bulut (2022, Jan. 16). Okan Bulut: Text Vectorization Using Python: TF-IDF. Retrieved from https://okan.cloud/posts/2022-01-16-text-vectorization-using-python-tf-idf/

[16] Stecanella, B. (2017, June 22). Support Vector Machines (SVM) algorithm explained. MonkeyLearn Blog. Retrieved August 14, 2022, from https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/

[17] Pant, A. (2019, January 22). Introduction to logistic regression. Medium. Retrieved August 14, 2022, from https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148

[18] Morde, V. (2019, April 8). XGBoost algorithm: Long may she reign! Medium. Retrieved August 14, 2022, from https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

[19] Ratz, A. V. (2022, April 8). Multinomial NAVE Bayes' for documents classification and Natural Language Processing (NLP). Medium. Retrieved August 14, 2022, from https://towardsdatascience.com/multinomial-na%C3%AFve-bayes-for-documents-classification-and-natural-language-processing-nlp-e08cc848ce6

[20] Ratz, A. V. (2022, April 8). Multinomial NAVE Bayes' for documents classification and Natural Language Processing (NLP). Medium. Retrieved August 14, 2022, from https://towardsdatascience.com/multinomial-na%C3%AFve-bayes-for-documents-classification-and-natural-language-processing-nlp-e08cc848ce6

[21] Distilbert-base-uncased · hugging face. distilbert-base-uncased · Hugging Face. (n.d.). Retrieved August 14, 2022, from https://huggingface.co/distilbert-base-uncased

[22] J. Ramos, 'Using TF-IDF to determine word relevance in document queries', 01 2003.

[23] Sklearn.feature_extraction.text.TfidfVectorizer. scikit. (n.d.). Retrieved August 14, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[24] Huilgol, P. (2019, August 24). Accuracy vs. F1-score. Medium. Retrieved August 14, 2022, from https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2

[25] Sklearn.svm.SVC. scikit. (n.d.). Retrieved August 15, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[26] Seven most popular SVM kernels. Dataaspirant. (2020, December 17). Retrieved August 15, 2022, from https://dataaspirant.com/svm-kernels/

[27] Sklearn.metrics.confusion_matrix. scikit. (n.d.). Retrieved August 15, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

[28] Sklearn.linear_model.logisticregression. scikit. (n.d.). Retrieved August 15, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[29] Sklearn.naive_bayes.multinomialnb. scikit. (n.d.). Retrieved August 15, 2022, from

[30] Python API reference. Python API Reference - xgboost 1.6.1 documentation. (n.d.). Retrieved August 15, 2022, from https://xgboost.readthedocs.io/en/stable/python/python_api.html

[31] Distilbert-base-uncased · hugging face. distilbert-base-uncased · Hugging Face. (n.d.). Retrieved August 15, 2022, from https://huggingface.co/distilbert-base-uncased

[32] Auto classes. Auto Classes. (n.d.). Retrieved August 15, 2022, from https://huggingface.co/docs/transformers/model_doc/auto

[33] Singhal, Trisha Liu, Junhua Blessing, Lucienne Wood, Kristin Lim, Kwan Hui. (2020). CrisisBERT: Robust Transformer for Crisis Classification and Contextual Crisis Embedding.