This report contains the second phase of my disaster response tweet classification project, where the objective is to classify tweets for urgency or event type during disasters with transformer-based models. In this article, I will discuss how I systematically experiment, evaluate models, and make changes to improve classifications. The TAMU High Performance Computing (HPC) cluster was used for all experiments.

Experimental Setup and Model Development.
In Update 1, I important experiments with recurrent and convolutional networks on a preprocessed disaster tweet dataset. In this update, I compare transformer-based architectures with a machine learning baseline. The 7,613 tweets used in the training data consists of 4,342 non disaster tweets and 3,271 disaster tweets. I split the data into 6090 training samples and 1523 validation samples using stratified sampling to maintain class distribution.

While doing text preprocessing, I lowered the case of the text. Then removed the URLs, hashtags, and mentions. I did the removal of special characters too. This provided clean, standardized input to the baseline as well as transformer model.

Hyperparameter Tuning for DistilBERT.
To tune hyperparameters for the DistilBERT model, I tried four different combinations with different learning rates (2e-5, 3e-5, 5e-5) and batch sizes (16, 32) while keeping the other two parameters i.e. epochs as 3 and weight decay as 0.01. We implement early stopping with patience of 2 epochs to avoid overfitting. The results are summarized below.

Configuration	Learning Rate	Batch Size	Accuracy	Precision	Recall	F1 Score
1	2e-5	16	0.8391	0.8851	0.7187	0.7932
2	3e-5	16	0.8424	0.8557	0.7615	0.8058
3	5e-5	16	0.8403	0.8117	0.8028	0.8143
4	2e-5	32	0.8444	0.8347	0.7951	0.8144

Among the four configurations, Configuration 1 (with a learning rate of 2e-5 and a batch size of 16) achieved the highest precision of 88.51% in predicting breast cancer. The analyst must trade off precision for recall. In other words, the configuration that works best for you depends whether you want to minimize false positives or false negatives.

Architecture Comparison: DistilBERT vs. BERT.
I also trained a full BERT-base-uncased model using the best hyperparameters (learning rate 2e-5, batch size 16, 3 epochs). Thus, to compare transformer models. The final model comparison results are.

Model	Accuracy	Precision	Recall	F1 Score
Baseline (TF-IDF + LR)	0.8162	0.8375	0.7095	0.7682
DistilBERT (Best)	0.8391	0.8851	0.7187	0.7932
BERT	0.8510	0.8588	0.7813	0.8183

The BERT model attained the highest accuracy of 85.10% along with the highest F1 score of 0.8183 in our analysis and is better than both baseline and DistilBERT. Still, DistilBERT gets the highest precision of 88.51%, which shows it is better suited when false positives must be reduced. BERT had higher recall value (78.13%), making it effective in discovering the tweet of disaster.

Evaluation Metrics and Baseline Comparison.
I adopted metrics consisting of accuracy, precision, recall, F1 score, and ROC-AUC for robust evaluation. The basic model, TF-IDF with logistic regression, got a ROC-AUC value of 0.8659 and F1 score 0.7682. The transformer models performed better than baseline. BERT increased the F1 score by roughly 5 percentage points, whereas DistilBERT boosted precision by nearly 5 percentage points over the baseline.

A visualization of these metrics was created and saved as model_comparison_visualization.png.

Challenges Encountered.
I faced permission issues while trying to get certain python packages to install on the HPC cluster, during experimentation. I solved this problem using the available environment modules. Also, special care was taken to manage GPU memory while training the transformer model, so, batch sizes were adjusted and gradients were accumulated.

Plans for Final Deliverables.
For the final submission, I plan to.

Try out ensemble methods that combine BERT and DistilBERT.

Try using methods like SMOTE or class weighting to manage class imbalance.

Explore additional architectures like RoBERTa or DeBERTa.

Make a complete final paper that summarizes methodology, experiments, and results.

Create a presentation with key visualizations.

Generate predictions on the test set for submission.

Conclusion.
By tuning the hyper parameters and comparing the architecture systematically, I showed the transformer-based models (DistilBERT and BERT) to outperform traditional TF-IDF + Logistic Regression baseline for classification of disaster tweet. BERT had the overall best performance with an F1 score of 0.8183 and an accuracy of 85.10. The results will serve as the backbone of the last phase which will focus on refining the models and packaging the outputs.