Moving on to applying models. To select only the best model from all possible combinations of embedders and classifiers, we implement stratified cross-validation. It splits our training set into four folds, each of which takes turn being used as a validation set to evaluate models that have been trained on the other ones. The ratio of disaster and non-disaster tweets in the training set also remains the same across all folds, as shown by the arrows to the pie chart.

Now, before the Tweets are classified, they are to be vectorized by embedders. Tweets are embedded directly in Bag-of-Words and TF-IDF. The dimension of their embedding vectors is the same as the vocabulary size. For Glove and Word2Vec embedders, a pre-trained model must be downloaded before words can be vectorized and then aggregated to encode the Tweets. Their vector dimensions are hard-coded, being 50 for Glove and 300 for Word2Vec. The same thing applies to the Sentence BERT embedder, but not only can we embed the Tweets directly with a fine-tuned model, but it is also possible to continue training it so that more disaster-relevant contexts can be captured in its 384-dimensional vectors.

The next step is to initialize the classifiers. For Logistic Regression we use Ridge regularization and set the inverse of regularization strength to 1. For multilayer perceptron we use relu activation, only one hidden layer with 100 neurons, and set the learning rate to 0.001. In Random Forest we build 100 Decision Trees with Bootstrapping and use Gini Impurity as their split criteria.

Now, let us look at the results. From the bar plot we can see that the highest cross-validation score is 0.975, which belongs to the combination of Sentence-BERT and Logistic Regression. However, when we apply the chosen model to the test set, it achieves only a F0.5 Score of 0.654. But surprisingly, our model achieved a F1-Score of 0.792 on Kaggle.

We conducted an analysis and found six problems regarding the low performance. There was human error during labeling test set, no hyperparameter tuning for the Logistic Regression Classifier, only one training epoch for the BERT Embedder with inaccurate and incomplete training samples. There is also a possiblity of insignificant added potential indicators, or overfitted data or inadequately cleaned Tweets. Therefore, we suggest labeling test set be done with a gold standard model, using Random and Grid Search to optimize hyperparameters, increasing the number of training epochs as well as preprocessing samples before training the Sentence-BERT embedder, treating potential indicators as hyperparameters during fine-tuning, and finally applying stronger regularization and stricter data cleaning.

That concludes our presentation. Thank you for listening. We welcome any questions.