



DISASTER TWEET CLASSIFICATION

Presented by Annelize Krause

Login

THE PROBLEM

Thousands of social media posts per minute. Some report real disasters, but most don't. Can I build a classification model to tell them apart?





WHY IT MATTERS

Correctly identifying disaster posts in real time could decrease emergency response times and create better outcomes.

THE DATASET

Use Kaggle dataset of 7,613 pre-labelled tweets to train and evaluate a prediction model.

Binary target:

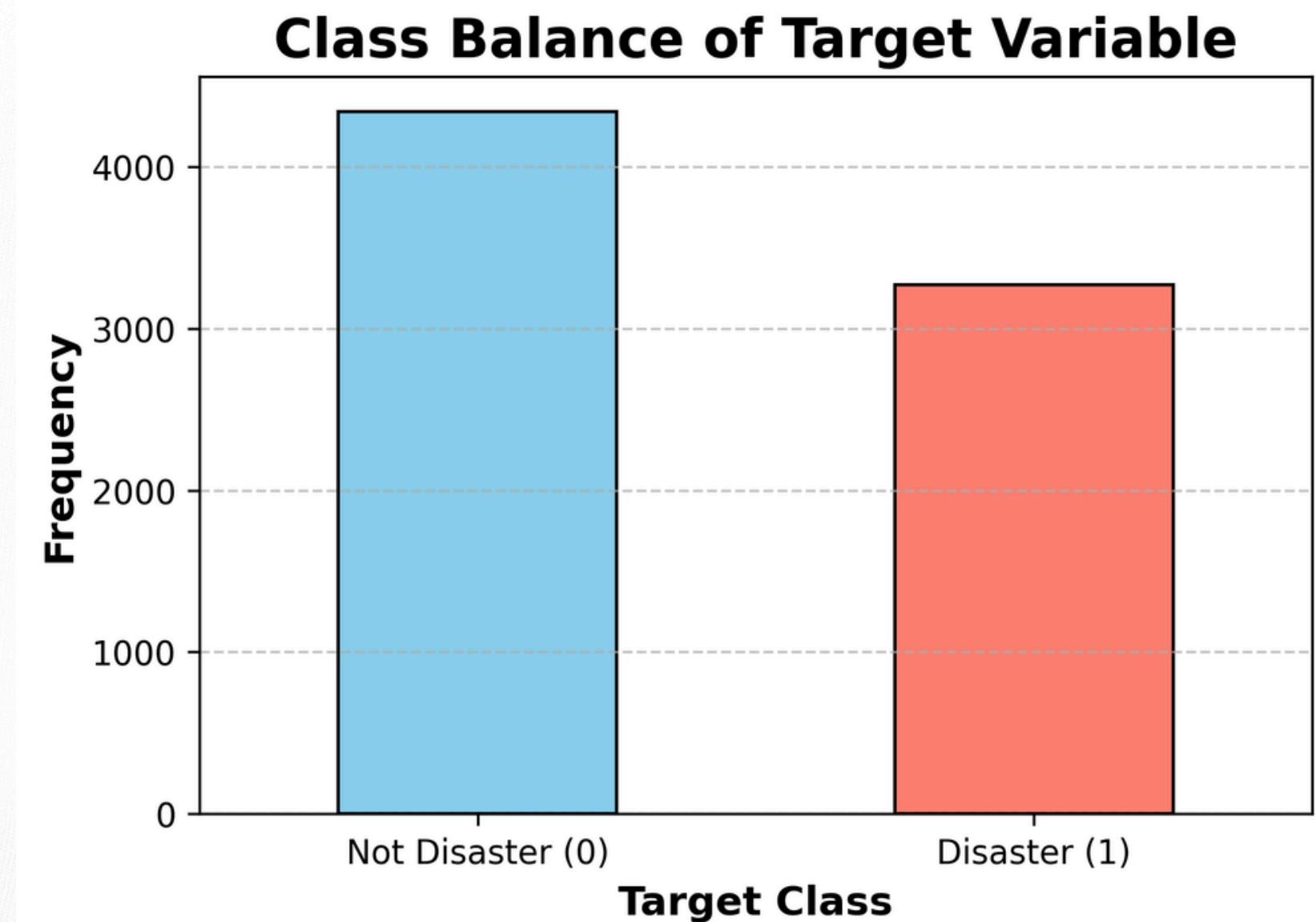
0 = Not Disaster

1 = Disaster

Distribution:

57% = Not disaster

43% = Disaster



TEXT PREPROCESSING

Tweets are noisy and preprocessing removes noise while preserving meaning:

- Lowercase all text
- Removed URLs, @mentions, special characters, numbers
- Kept hashtag words (removed `#` symbol only)
- Removed stop words using NLTK
- Lemmatized words to their base form



Mom's Spaghetti
@momspaghetti

#RockyFire Update => California Hwy. 20 closed in both directions due to Lake County fire - #CAfire
#wildfires

1:17 PM · Jul 16, 2023

57 Retweets 86 Quote Tweets 1K Likes

...



Mom's Spaghetti
@momspaghetti

rockyfire update california hwy closed direction due lake county fire cafire wildfire

1:17 PM · Jul 16, 2023

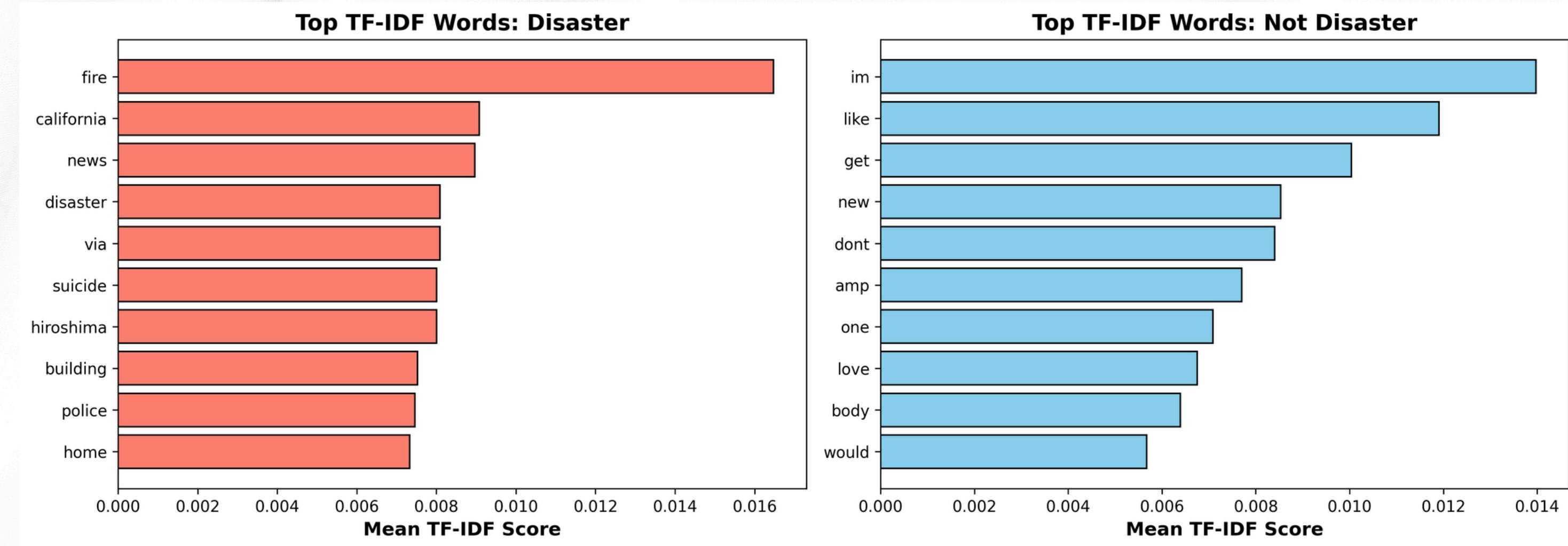
57 Retweets 86 Quote Tweets 1K Likes

...

FEATURE EXTRACTION

TF-IDF converts text into numbers a model can learn from by:

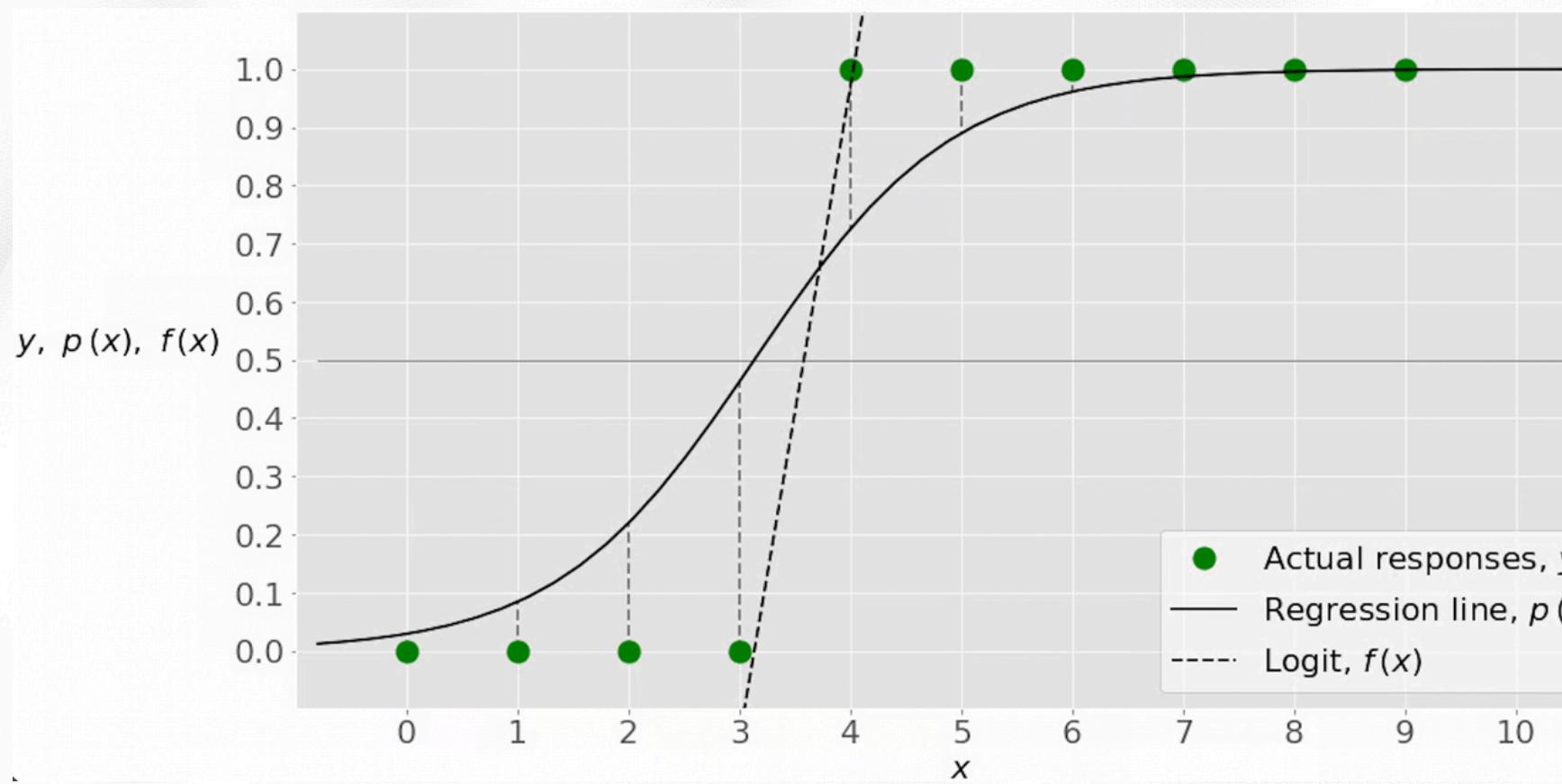
- scoring words by how important they are to a specific tweet relative to the whole dataset
- common words get low scores, distinctive words get high scores
- result: 11,708 features from the training vocabulary



MODEL SELECTION

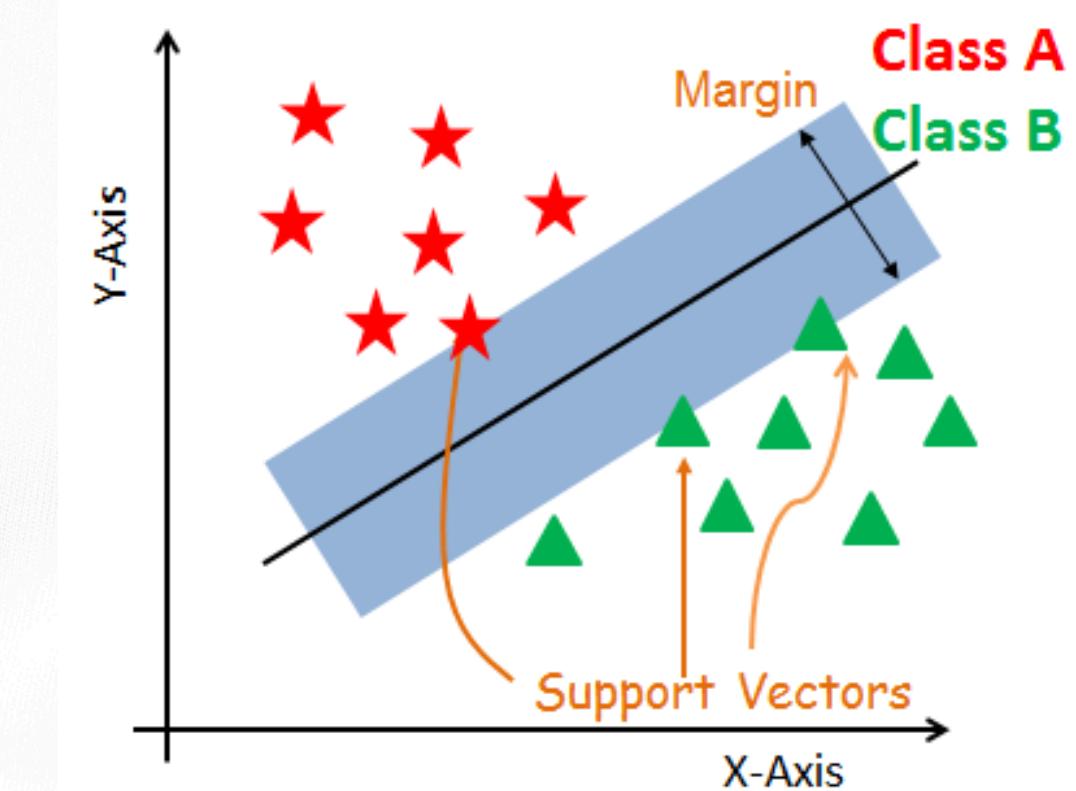
Logistic Regression

- Uses a sigmoid function to estimate probability of each class
- Well suited for binary classification (disaster / not disaster)
- Works well with high-dimensional sparse data like TF-IDF vectors
- Outputs probabilities, making predictions easy to interpret



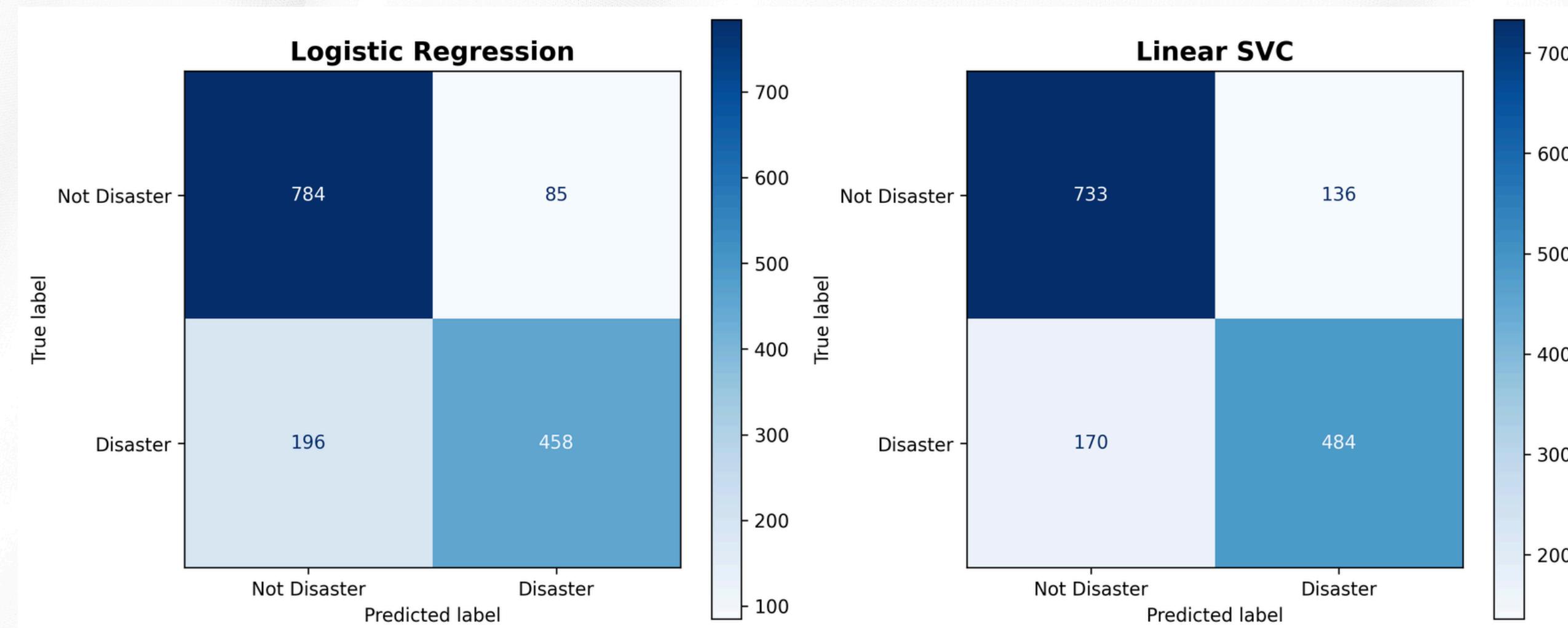
Linear SVC

- Finds decision boundary that maximises margin between classes
- Strong performance on text classification tasks
- Different approach to same problem helps validate results



BASELINE RESULTS

- Both models reach ~80-82% accuracy, but the confusion matrices reveal a precision-recall trade-off:
 - Logistic Regression** is more precise 84% with lower recall 70%.
 - Linear SVC** is less precise 78% catching more actual disasters 74%.



HYPERPARAMETER TUNING

- Can we improve the 30% miss rate by adjusting the model's settings?
- Used GridSearchCV to test 24 combinations across vocabulary size, word groupings, and regularisation strength
- Modest improvement, meaning the baseline was already near optimal for this model type.

Metric	Baseline	Tuned
Accuracy	82%	83%
Disaster Recall	0.70	0.72
Disaster F1	0.77	0.78

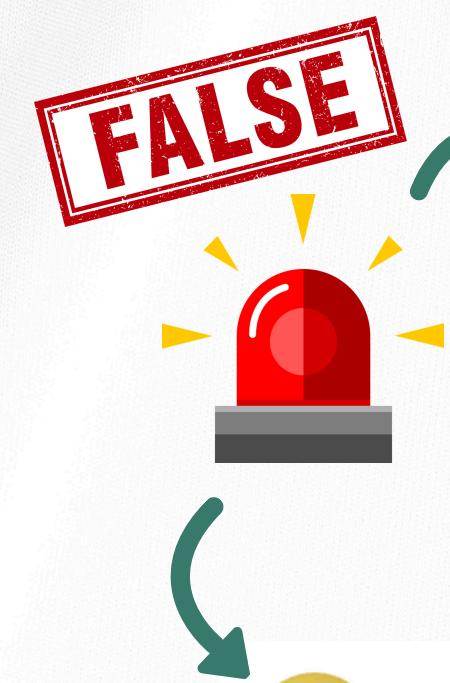
HITS & MISSES

What works well:

- Explicit disaster keywords
- Factual, news-like language

What does it fail on:

- Figurative language
- Sarcasm and slang
- Short or vague tweets with little context



Mom's Spaghetti
@momspaghetti

if firefighters acted like cops they'd drive around shooting a flamethrower at burning buildings

6:56 PM · Apr 9, 2024

156 Retweets 250 Quote Tweets 10.5K Likes



...



Mom's Spaghetti
@momspaghetti

@soonergrunt better than tornado!

10:14 AM · Oct 22, 2024

5 Retweets 3 Quote Tweets 248 Likes



...

KEY TAKEAWAYS

- A simple Logistic Regression + TF-IDF pipeline achieves 83% accuracy on disaster tweet classification
- Text preprocessing choices have real trade-offs (removing stop words can lose negation)
- Precision vs recall trade-off depends on the application — for disaster detection, recall should be prioritised
- The main limitation is the bag-of-words representation, which cannot capture context or figurative language

LIMITATIONS & IMPROVEMENTS

Limitations:

- TF-IDF ignores word order and context
- Small dataset (7,613 samples)
- Class imbalance affects disaster recall
- Preprocessing removes potentially meaningful signals (stop words, punctuation)

Future improvements:

- Word embeddings (Word2Vec, GloVe) for semantic features
- Transformer models (BERT, BERTweet) for context-aware classification
- Class balancing techniques (SMOTE, class weighting)



**DOES ANYONE
HAVE QUESTIONS?**



THANK YOU