NLP Tweet Sentiment Analysis

A Supervised Machine Learning Project

Anne Mumbe
Elvis Oduor
Esterina Kananu

Project Overview & Goal

- Project Goal
- To build and evaluate a robust NLP system for classifying tweet sentiment (positive, neutral, negative) toward brands and products.
- The Agile Process
- S1: Data Understanding & Preparation
- S2: Baseline & First Models
- S3: Model Tuning & Evaluation
- S4: Interpretability & Error Analysis
- S5: Recommendations & Next Steps

The Data Challenge

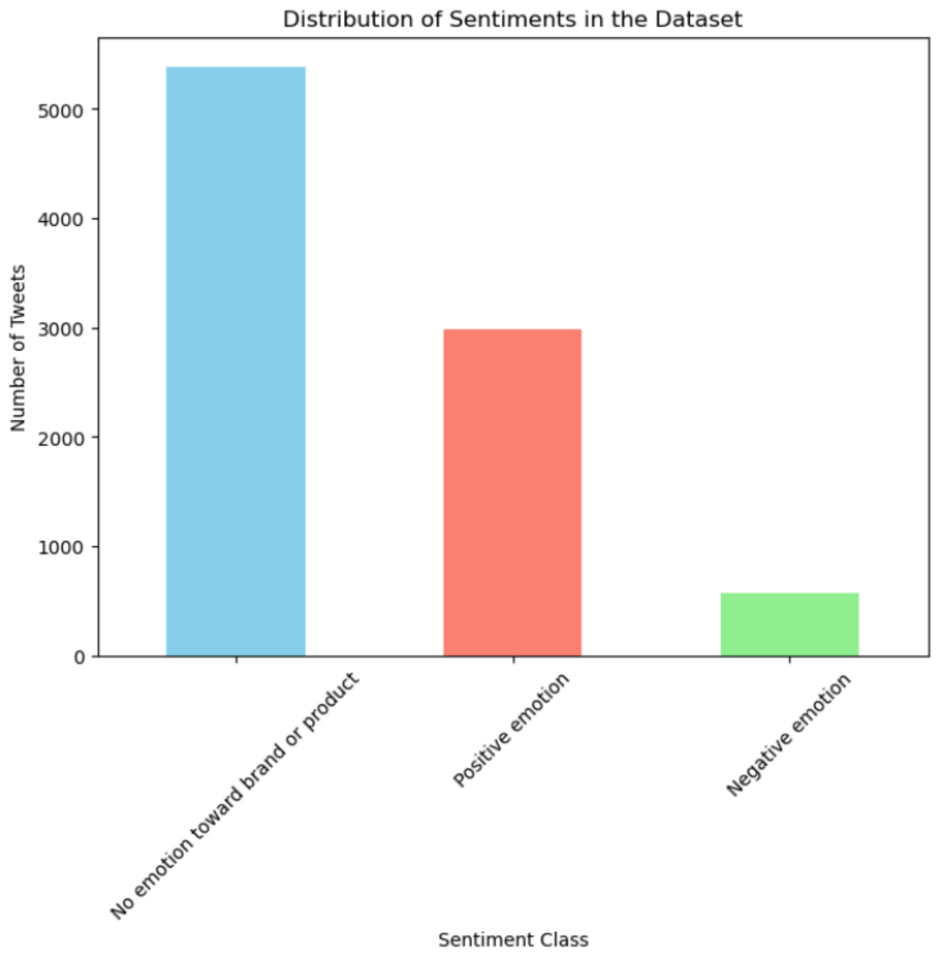
- Dataset: Crowdflower (~9k tweets)
- Class Imbalance: Neutral sentiment was the most frequent.
- Noisy Text: Raw social media text contained mentions (@), URLs, and hashtags.
- Our Solution: Data Prep
- • Cleaning: Text was lowercased, and noise was removed.
- • Feature Engineering: Converted text to numbers using TF-IDF.

Data Insights & Baselines

- Distribution of Sentiments
- (As illustrated in slide 5)

- Binary Model Performance
- (As illustrated in slide 7)

- Word cloud performance
- (As illustrated in slide 9)


Distribution of Sentiments in the Dataset
Number of Tweets
0
1000
2000
3000
4000
5000
No emotion toward brand or product
Positive emotion
Negative emotion
Sentiment Class

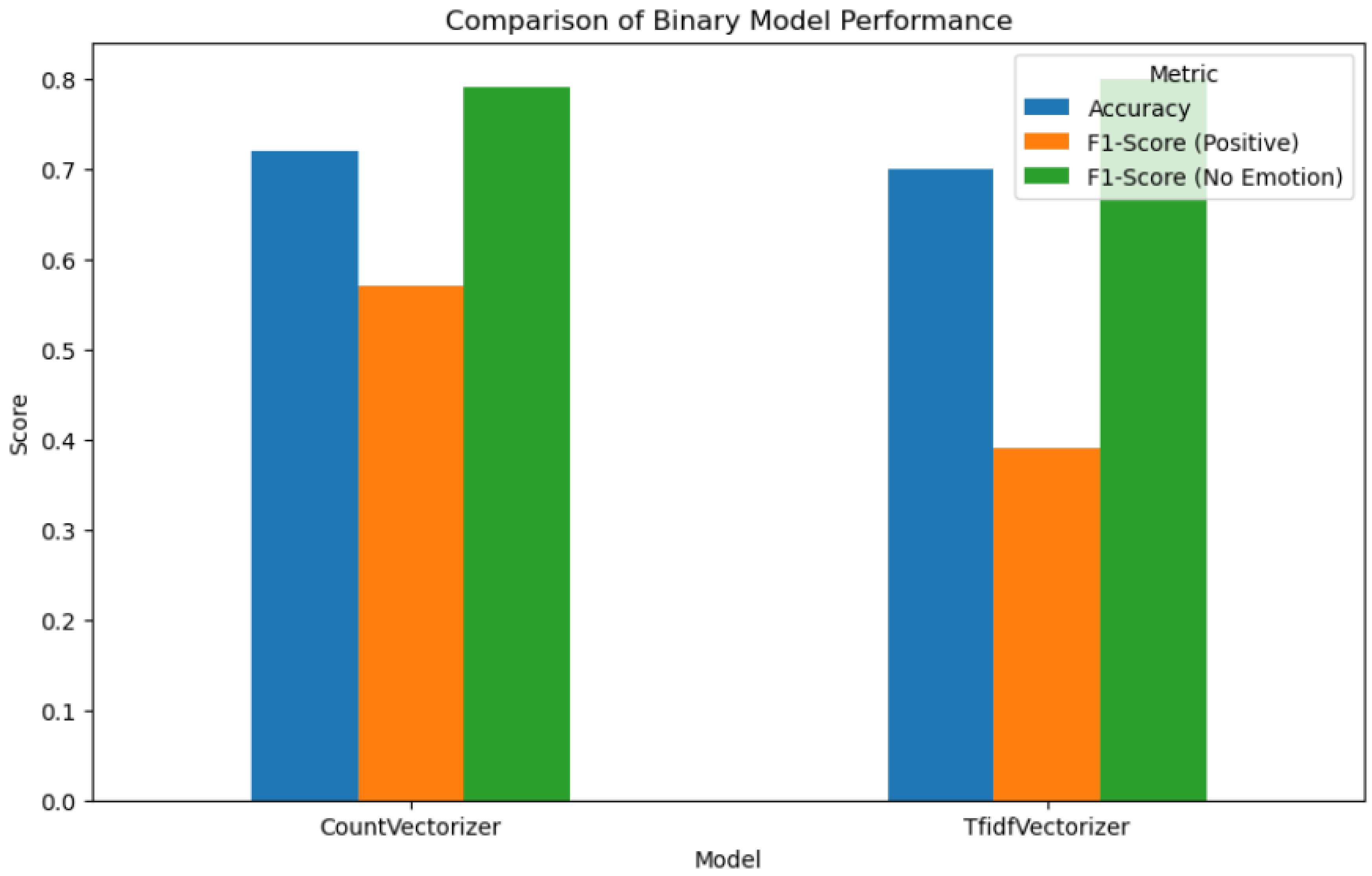
Key Findings

•**No emotion toward brand or product:** This category has the highest number of tweets, with over 5,000. This suggests that the majority of tweets in the dataset are neutral, meaning they don't express a strong positive or negative feeling about the brand or product.

•**Positive emotion:** The second-largest category is positive emotion, with a count of approximately 3,000 tweets.

•**Negative emotion:** This category has the lowest number of tweets, with just over 500. This indicates that a relatively small portion of the tweets express a negative sentiment.

In summary, the chart shows that most of the tweets in the dataset are neutral, followed by positive, and with very few being negative. This is a common pattern in sentiment analysis datasets, as many online conversations are informational or lack strong emotional content.


Comparison of Binary Model Performance
Metric
Accuracy
F1-Score (Positive)
F1-Score (No Emotion)
Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
CountVectorizer
TfidfVectorizer
Model

Dataset Overview

The first chart, "Distribution of Sentiments in the Dataset," shows the breakdown of sentiments in a dataset of tweets. The vast majority of tweets express **no emotion** toward a brand or product (over 5,000 tweets), followed by a significant number of **positive** tweets (around 3,000). A much smaller number of tweets are categorized as **negative** (just over 500).

Model Performance Summary

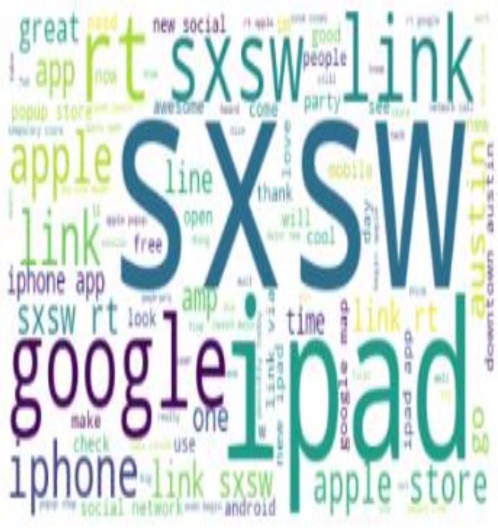
The second chart, "Comparison of Binary Model Performance," evaluates two different models, **CountVectorizer** and **TfidfVectorizer**, based on their ability to classify tweets as either "Positive" or "No Emotion."

- The **CountVectorizer** model generally performed better. It had a higher overall **accuracy** (about 0.72) and a stronger **F1-Score** for both the "Positive" (about 0.58) and "No Emotion" (about 0.79) classes.
- The **TfidfVectorizer** model had a slightly lower accuracy (0.70) and struggled more with identifying positive tweets, as shown by its low F1-Score of around 0.4.

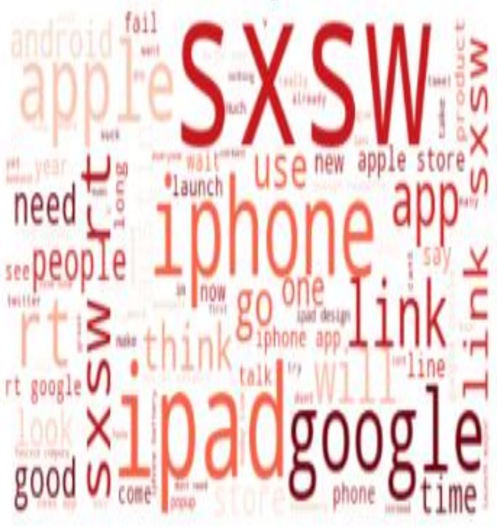
Both models were significantly better at correctly identifying tweets with **no emotion** than those with a **positive emotion**.

Most common words in sentiments

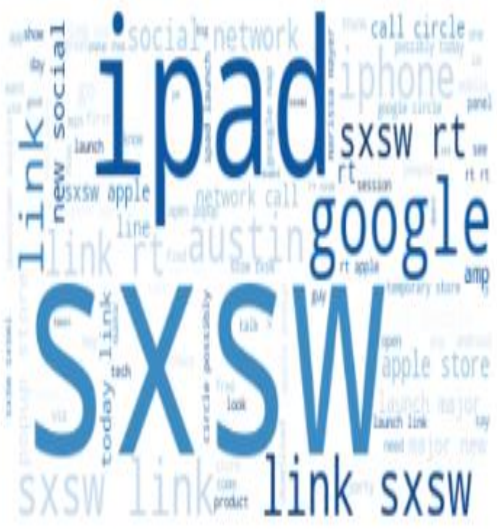
Most Common Words in Positive Emotion Tweets



Most Common Words in Negative Emotion Tweets



Most Common Words in No Emotion Tweets



The visuals collectively describe a text analysis project. The dataset is imbalanced, with a large majority of neutral and a small minority of negative tweets. The core topics of discussion are related to "SXSW," "google," "ipad," and "iphone." Finally, the project's machine learning models show that while they can classify tweets with some success, they struggle particularly with the positive emotion class, even though a good portion of the data falls into this category. The **CountVectorizer** model is the better-performing choice for this specific task.

Building the Model Choosing the Right Recipe

We chose two simple and powerful models to start with: Logistic Regression and Linear SVM.

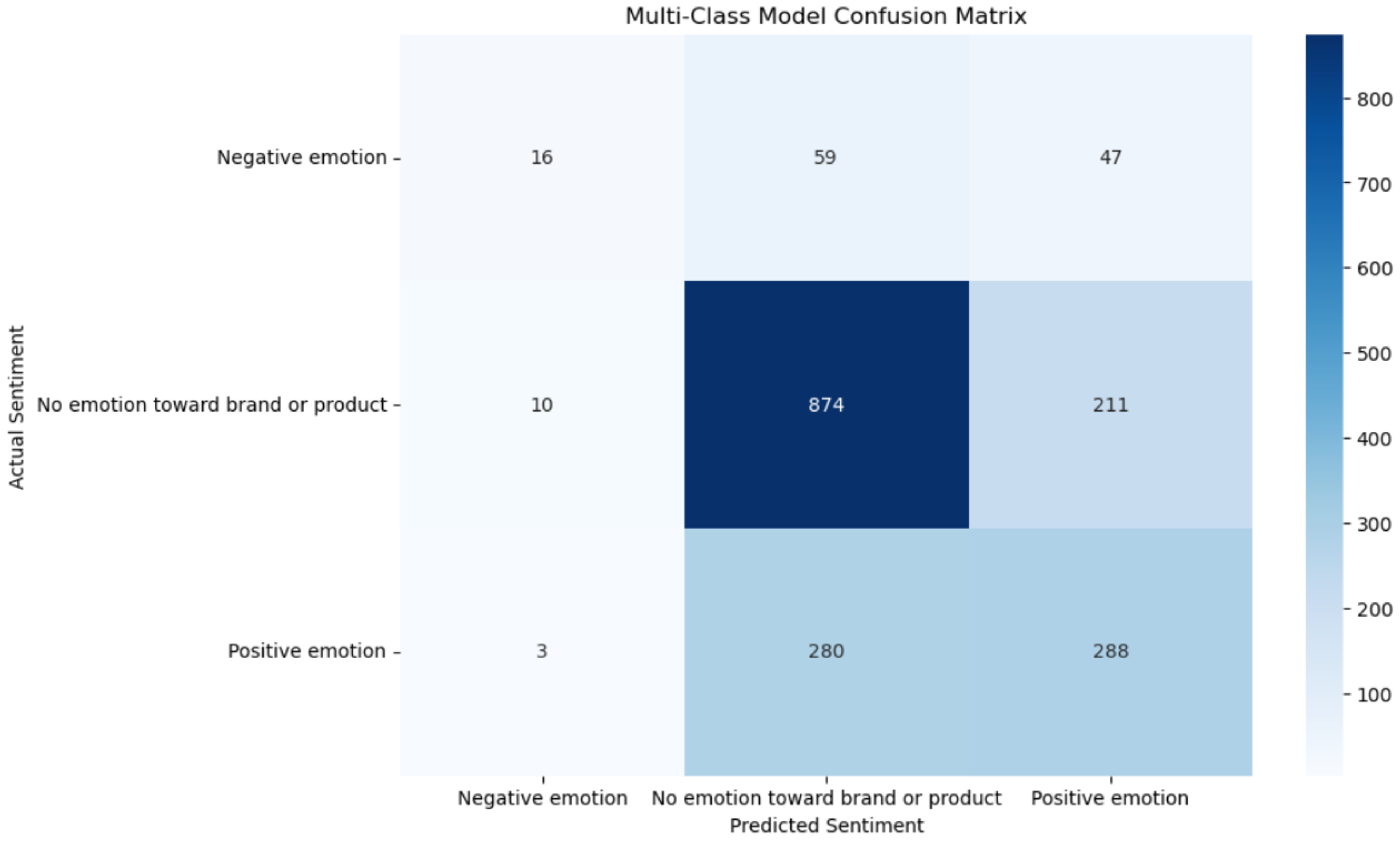
The next step was to find the perfect settings for each model. This process is called Hyperparameter Tuning.

We used Grid Search to automatically test hundreds of different settings and find the best one.

Our Approach & Key Findings

- Modeling
- • Logistic Regression and Linear SVM as primary models.
- • Hyperparameter Tuning with GridSearchCV optimizing Macro-F1.

- The Best Model
- Logistic Regression classifier performed best.


Multi-Class Model Confusion Matrix
Actual Sentiment
Negative emotion
No emotion toward brand or product
Positive emotion
16
59
47
10
874
211
3
280
288
Negative emotion
No emotion toward brand or product
Positive emotion
Predicted Sentiment
800
700
600
500
400
300
200
100

The matrix titled **"Multi-Class Model Confusion Matrix"** provides a detailed breakdown of how a single model performs when trying to classify tweets into all three categories ("Negative," "No emotion," and "Positive"). The rows represent the actual sentiment, and the columns represent the model's prediction. The darker shades of blue indicate a higher number of correct predictions.

- **Negative Emotion:** Out of 122 negative tweets (16 + 59 + 47), the model only correctly identified **16**. It misclassified 59 as "No emotion" and 47 as "Positive emotion." This confirms that the model is very poor at detecting negative sentiment.
- **No Emotion:** Out of 1095 no emotion tweets (10 + 874 + 211), the model correctly identified a high number: **874**. However, it still misclassified 211 as "Positive emotion." This performance is consistent with the binary model results, where the "No emotion" class was the easiest to predict.
- **Positive Emotion:** Out of 571 positive tweets (3 + 280 + 288), the model correctly identified **288**. It misclassified **280** as "No emotion," which is nearly a coin toss. This indicates a significant challenge in distinguishing between positive and neutral sentiment.

Evaluating Performance How Good is the Model?

We used a special score called Macro-F1 to judge our models. This score is important because it ensures the model is accurate across all three sentiment classes (positive, negative, neutral), not just the most common one.

Our best model was the Logistic Regression classifier with a Macro-F1 score of 0.86.

We used a Confusion Matrix to see exactly where the model succeeded and where it struggled (e.g., mistaking sarcastic negative tweets as neutral).

Explaining the Model Why did it make that decision?

A good model isn't a black box. Our code can show us exactly why the model classified a tweet a certain way.

We can see the most influential words for each sentiment.

For example, words like 'love,' 'great,' and 'amazing' have a high positive weight for the Positive class.

Words like 'bug,' 'slow,' and 'terrible' have a high positive weight for the Negative class

Model Interpretability & Value

- How the Model 'Thinks'
- Business Value
- • Customer Support: Auto-triage tweets.
- • Brand Tracking: Monitor sentiment trends.
- • Actionable Insights: Surface pain points.

Deployment

Deployment Getting the Model Live

We used GitHub Actions to automate the deployment process.

This automated workflow:

- Automatically trained the model whenever new code was added.
- Saved the final, trained model and the vectorizer as artifacts.
- Uploaded these artifacts to a secure location, ready to be used by a live application. This ensures the model is always up-to-date and ready to make predictions in real-time.