CLASSIFYING DISASTER TWEETS USING NATURAL LANGUAGE PROCESSING

BY: KAZUKI SUSUKI, JONATHAN TRAN, JEAMPY KALAMBAYI

Georgia Gwinnett COLLEGE

ABSTRACT

For our Data Science and Analytics Capstone project, we developed a machine learning model to identify disaster-related tweets using natural language processing. Our goal is to create a machine learning model that can accurately determine if a tweet is about a disaster or not. Using a labeled dataset from Kaggle, we applied text cleaning, TF-IDF vectorization, and PCA for dimensionality reduction. We then trained a logistic regression classifier and evaluated its performance using accuracy and F1-score. We also tested other models like BERT and SVM and validated their performances to compare to PCA. The final model will be presented through interactive graphs and summaries on a visual dashboard to enhance understanding and usability.

TOOLS

Google Colab

- Data Cleaning
- Analysis
- Visualization
- Model Training

Github

- Team Collaboration
- Code Management
- Project File Storage

Jira

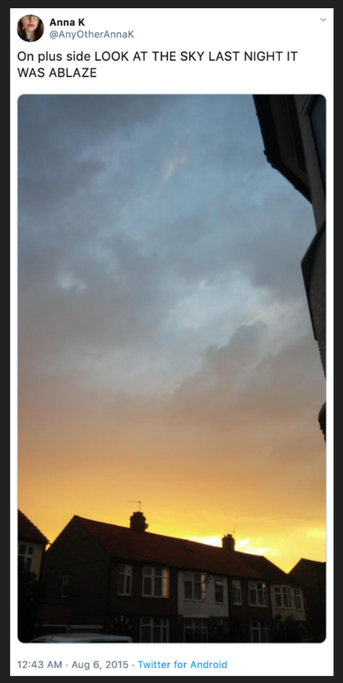
- Task Organization
- Project Management
- Scheduling

CHALLENGE DETAILS

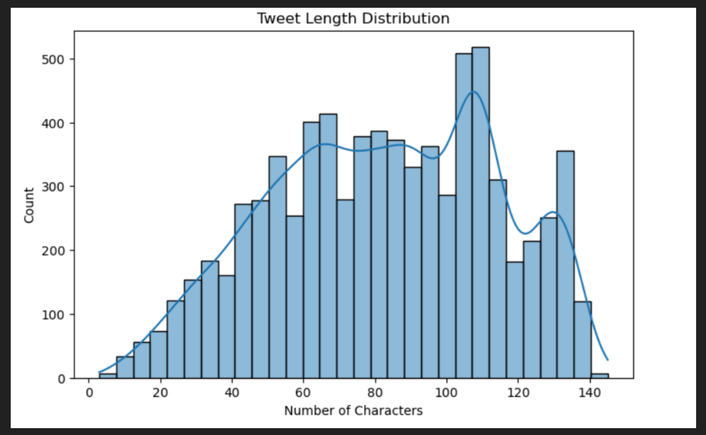
The challenge "Natural Language Processing with Disaster Tweets" requires training a machine learning model that can accurately predict if a tweet is about a disaster or not. Saying the sky is "ABLAZE" is metaphorical, which is not easy for a machine to process.

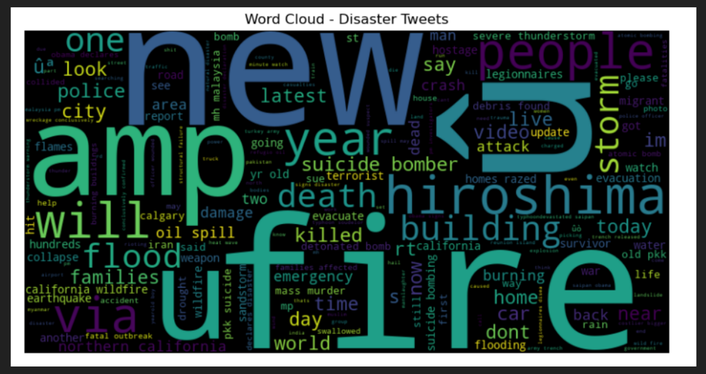
The model must be able to assign target values to each tweet

0 for Not Disaster
1 for Disaster



VISUALS





SVM Kernel Types:

RBF vs Linear vs Poly vs Sigmoid

```
Valid Accuracy: 0.80

Classification:
              precision    recall  f1-score   support

           0       0.80      0.86      0.83       874
           1       0.79      0.70      0.75       649

    accuracy                           0.80      1523
   macro avg       0.80      0.78      0.79      1523
weighted avg       0.80      0.80      0.79      1523


Confusion:
[[755 119]
 [192 457]]
```

```
Valid Accuracy: 0.79

Classification:
              precision    recall  f1-score   support

           0       0.81      0.83      0.82       874
           1       0.77      0.73      0.75       649

    accuracy                           0.79      1523
   macro avg       0.79      0.78      0.79      1523
weighted avg       0.79      0.79      0.79      1523


Confusion:
[[729 145]
 [172 477]]
```

```
Valid Accuracy: 0.73

Classification:
              precision    recall  f1-score   support

           0       0.70      0.95      0.80       874
           1       0.86      0.44      0.59       649

    accuracy                           0.73      1523
   macro avg       0.78      0.69      0.69      1523
weighted avg       0.77      0.73      0.71      1523


Confusion:
[[827  47]
 [361 288]]
```

```
Valid Accuracy: 0.80

Classification:
              precision    recall  f1-score   support

           0       0.81      0.84      0.82       874
           1       0.77      0.74      0.75       649

    accuracy                           0.80      1523
   macro avg       0.79      0.79      0.79      1523
weighted avg       0.79      0.80      0.79      1523


Confusion:
[[732 142]
 [170 479]]
```

BERT Accuracy:

```
              precision    recall  f1-score   support

           0       0.82      0.64      0.72       652
           1       0.63      0.81      0.71       490

    accuracy                           0.71      1142
   macro avg       0.72      0.73      0.71      1142
weighted avg       0.74      0.71      0.72      1142
```

SVM Accuracy:

```
Classification:
              precision    recall  f1-score   support

           0       0.80      0.86      0.83       874
           1       0.79      0.70      0.75       649

    accuracy                           0.80      1523
   macro avg       0.80      0.78      0.79      1523
weighted avg       0.80      0.80      0.79      1523
```

```
Confusion:
[[755 119]
 [192 457]]
```

PCA Accuracy:

Evaluate Model Performance

```
In [29]: accuracy = accuracy_score(val_labels, predictions)
         f1 = f1_score(val_labels, predictions)
         print(f"Validation Accuracy: {accuracy:.4f}")
         print(f"Validation F1 Score: {f1:.4f}")

Validation Accuracy: 0.7623
Validation F1 Score: 0.6885
```