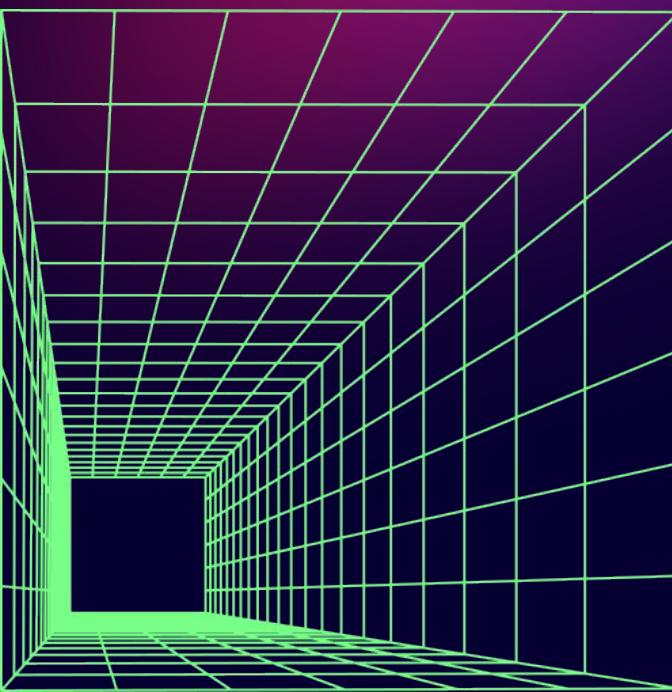


SENSITIVE TEXT

By Mubarak Abiola Keshiro

BACKGROUND & INTRODUCTION



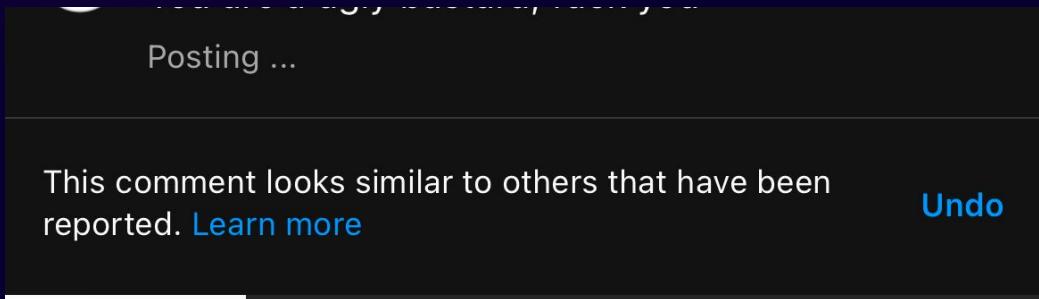
The primary objective was to develop a sentiment analysis model capable of assessing the sentiment (negative or non negative) expressed in imputed text.

THE PROBLEM AND WHY

The major aim is to give users access to a tool or application where they can see the way (sentiment) in which others may perceive their sentences and statements.

WHY?

On social media, when you post something, the application believes might be harmful in most cases it usually just deletes, restricts or suspends your post or your account with little to no explanation .



HOW AND PROCESS

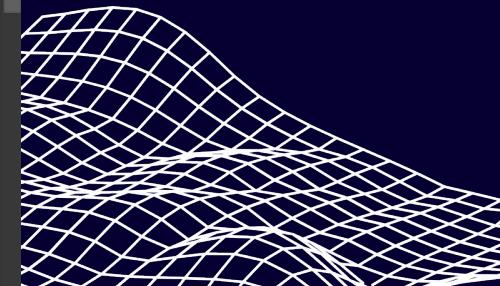
To achieve this;

- I used python and its libraries, Google colab and Flask for Implementation
- I used Natural Language processing and Logistic regression.
- I used a publicly available dataset of 500 thousand tweets polarized into positive and negative
- I trained the model at first with the method above and achieved a ~79% accuracy

```
report = classification_report(y_test, y_pred)

print(f'Accuracy: {accuracy}')
print(report)
```

	precision	recall	f1-score	support
bad	0.80	0.79	0.79	50055
good	0.79	0.80	0.80	49945
accuracy			0.79	100000
macro avg	0.79	0.79	0.79	100000
weighted avg	0.79	0.79	0.79	100000



X

+ Code + Text

✓ RAM Disk ▾ ▲

```
[32] while(1) :  
    user_input = input("Enter a sentence: ")  
    user_input_tfidf = tfidf_vectorizer.transform([user_input])  
    predicted_sentiment = model.predict(user_input_tfidf)  
  
    if predicted_sentiment[0] == 'good':  
        print("That was not a negative sentence!")  
    else:  
        print("That was a negative sentence. 😞")
```

Enter a sentence: i like him
That was not a negative sentence!
Enter a sentence: i hate him
That was a negative sentence. 😞
Enter a sentence: hi
That was not a negative sentence!
Enter a sentence: i dont know
That was a negative sentence. 😞
Enter a sentence: i dislike her
That was a negative sentence. 😞
Enter a sentence: she likes to hurt people
That was a negative sentence. 😞
Enter a sentence: she likes to kill people
That was not a negative sentence!
Enter a sentence: she likes to harass people
That was not a negative sentence!

PROCESS

I wanted to improve the accuracy by trying other models;

```
print(f'Accuracy Naive Bayes: {accuracy_nb}')
print(report_nb)
```

Accuracy Naive Bayes: 0.7890434791303174

	precision	recall	f1-score	support
bad	0.75	0.86	0.80	99910
good	0.83	0.72	0.77	100117
accuracy			0.79	200027
macro avg	0.79	0.79	0.79	200027
weighted avg	0.79	0.79	0.79	200027

NAIVE BAYES- 78 PERCENT

PROCESS

I wanted to improve the accuracy by trying other models;

A screenshot of a Jupyter Notebook interface. At the top, there is a code cell with the following content:

```
print(f'Accuracy using BERT: {accuracy}')
```

Below the code cell, several files are being downloaded, each shown with a progress bar:

- tokenizer_config.json: 100% [28.0/28.0] [00:00<00:00, 330B/s]
- vocab.txt: 100% [232k/232k] [00:00<00:00, 1.41MB/s]
- tokenizer.json: 100% [466k/466k] [00:00<00:00, 5.72MB/s]
- config.json: 100% [570/570] [00:00<00:00, 8.28kB/s]
- model.safetensors: 100% [440M/440M] [00:24<00:00, 24.8MB/s]

At the bottom of the notebook, there is a warning message:

of BertForSequenceClassification were not initialized from the model checkpoint
probably TRAIN this model on a down-stream task to be able to use it for prediction

[View runtime logs](#)

On the left side of the screen, there is a message about session crashing:

four session crashed after using all available RAM. If you are interested in access to high-RAM runtimes, you may want to check out [High-RAM](#).

BERT -NEEDED GPU

PROCESS

I went back to logistic regression and Improved my dataset and increased the test size and achieved 81 percent accuracy

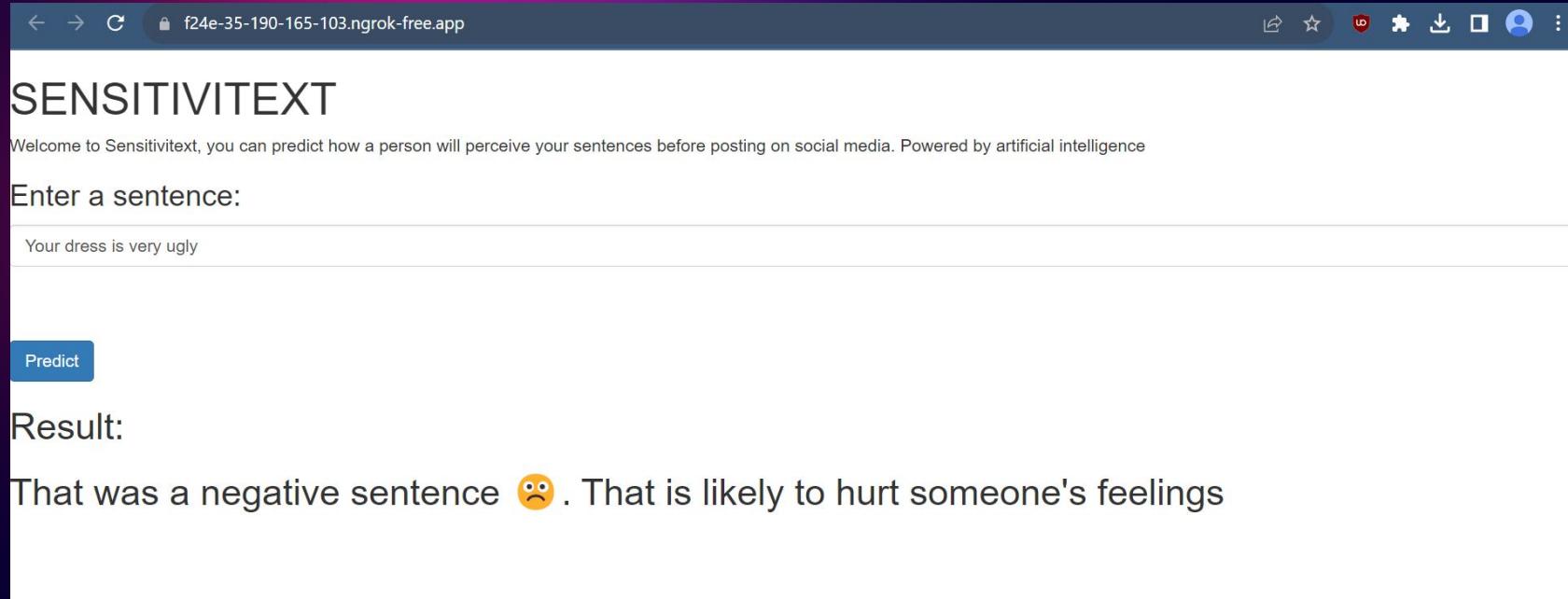
```
print(f' Accuracy: {accuracy}')
print(report)

→ Accuracy: 0.8100406445129907
      precision    recall   f1-score   support
      bad         0.81     0.82     0.81    99910
      good        0.81     0.80     0.81   100117

      accuracy           0.81    200027
      macro avg       0.81     0.81     0.81    200027
      weighted avg    0.81     0.81     0.81    200027
```

```
[12] !pip install pyngrok
```

RESULTS & DEMO

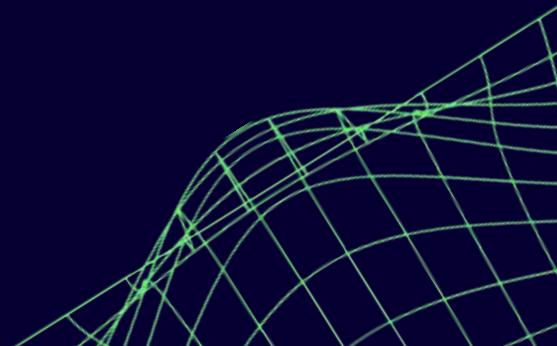


The screenshot shows a web browser window with the URL `f24e-35-190-165-103.ngrok-free.app`. The page title is "SENSITIVITEXT". A sub-header states: "Welcome to Sensitivitext, you can predict how a person will perceive your sentences before posting on social media. Powered by artificial intelligence". Below this, a text input field contains the sentence "Your dress is very ugly". A blue "Predict" button is visible. To the right, under the heading "Result:", the text "That was a negative sentence 😞 . That is likely to hurt someone's feelings" is displayed.

I used Flask and ngrok to implement a front end of the application

RESULTS & DEMO

{VIDEO OR LIVE DEMO }



QUESTIONS?

REFERENCES

- Hosmer, D. W. (2013). Applied Logistic Regression. Google Books. URL:
https://books.google.ca/books?hl=en&lr=&id=64JYAwAAQBAJ&oi=fnd&pg=PR13&dq=logistic+regression+model&ots=DtkT2XbqhQ&sig=hurRvZxkuP5HwkcTmlu_2L6gvy4#v=onepage&q=logistic%20regression%20model&f=false
- LearnDataSci. (n.d.). Term Frequency - Inverse Document Frequency (TF-IDF). Retrieved from
[https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/#:~:text=Term%20Frequency%20%2D%20Inverse%20Document%20Frequency%20\(TF%2DIDF\)%20is,%2C%20relative%20to%20a%20corpus.](https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/#:~:text=Term%20Frequency%20%2D%20Inverse%20Document%20Frequency%20(TF%2DIDF)%20is,%2C%20relative%20to%20a%20corpus.)
- NLTK. (n.d.). Natural Language Toolkit (NLTK). Retrieved from <https://www.nltk.org/>
- Towards Data Science. (n.d.). How I Achieved 90% Accuracy on a Text Classification Problem with Zero Preprocessing. Retrieved from
<https://towardsdatascience.com/how-i-achieved-90-accuracy-on-a-text-classification-problem-with-zero-preprocessing-6acfa96e8d2e>
- GeeksforGeeks. (n.d.). How to Run Flask App on Google Colab. Retrieved from
<https://www.geeksforgeeks.org/how-to-run-flask-app-on-google-colab/>