PANIMALAR ENGINEERING COLLEGE

Department of Computer Science and Engineering

CS8811 PROJECT WORK

Offensive Language Detection Using Machine Learning Classifiers

Project Guide:
Dr. L. Jabasheela

Presented By:
Silas Dhayanand S - 211419104252

Batch Number: C24

Introduction

- Offensive language detection is an important application of machine learning that aims to automatically identify and flag text that contains offensive content.
- This project involves building a machine learning classifier that can accurately detect offensive language in text.
- The classifier will be trained on a dataset of labeled examples, where each example is labeled as offensive or non-offensive.
- In recent years, offensive language detection has become increasingly important in social media platforms, online forums, and other digital environments where users can post content anonymously. The goal is to identify and remove harmful content to create a safer and more welcoming online community for all users.

S.No	TITLE	JOURNAL NAME, AUTHOR AND YEAR	OBJECTIVE AND METHODOLOGY
1.	Automated hate speech detection and the problem of offensive language	Proceedings of the 11th International AAAI Conference on Web and Social Media, Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017)	It involves a combination of data collection, annotation, feature extraction, classifier training, error analysis, and interpretation. This approach provides a framework for developing automated systems for detecting hate speech and offensive language
2.	A survey on automatic detection of hate speech in text	ACM Computing Surveys, Fortuna, P., Nunes, C., & Marques, T. (2018)	It is a comprehensive survey of the state-of-the-art in automatic detection of hate speech in text. The survey aims to provide an overview of the methods and techniques used for detecting hate speech, including the types of data sources, features, classifiers, and evaluation methods.
3.	A review on automatic hate speech detection using machine learning techniques	IEEE Access, Hossain, M. S., & Muhammad, G. (2019)	This paper involves a comprehensive review and analysis of the current state-of-the-art techniques and challenges in automatic hate speech detection using machine learning.
4.	Abusive language detection in online user content	Proceedings of the 25th International Conference on World Wide WebNobata, C., Tetreault, J., Thomas, A., & Mehdad, Y. (2016)	It involves using a combination of annotated data, preprocessing techniques, and machine learning algorithms to identify and classify abusive language in online user content.
5.	Predicting the type and target of offensive posts in social media	Transactions of the Association for Computational Linguistics, S., Farra, N., & Kumar, R. (2020)	The authors achieved high accuracy in predicting both the type and target of offensive language in social media posts. They also conducted an error analysis to provide insights for future improvements.

Problem Statement

- The need for this project arises from the increasing prevalence of offensive language in online platforms.
- Offensive language can include hate speech, cyberbullying, and other forms of harmful content.
- The development of an accurate and reliable machine learning classifier for offensive language detection. The classifier should be able to account for variations in language use and capture subtle nuances that distinguish offensive from non-offensive language.
- Overall, the objective of this project is to build a machine learning classifier that can effectively detect offensive language in text and contribute to the creation of a safer online environment.

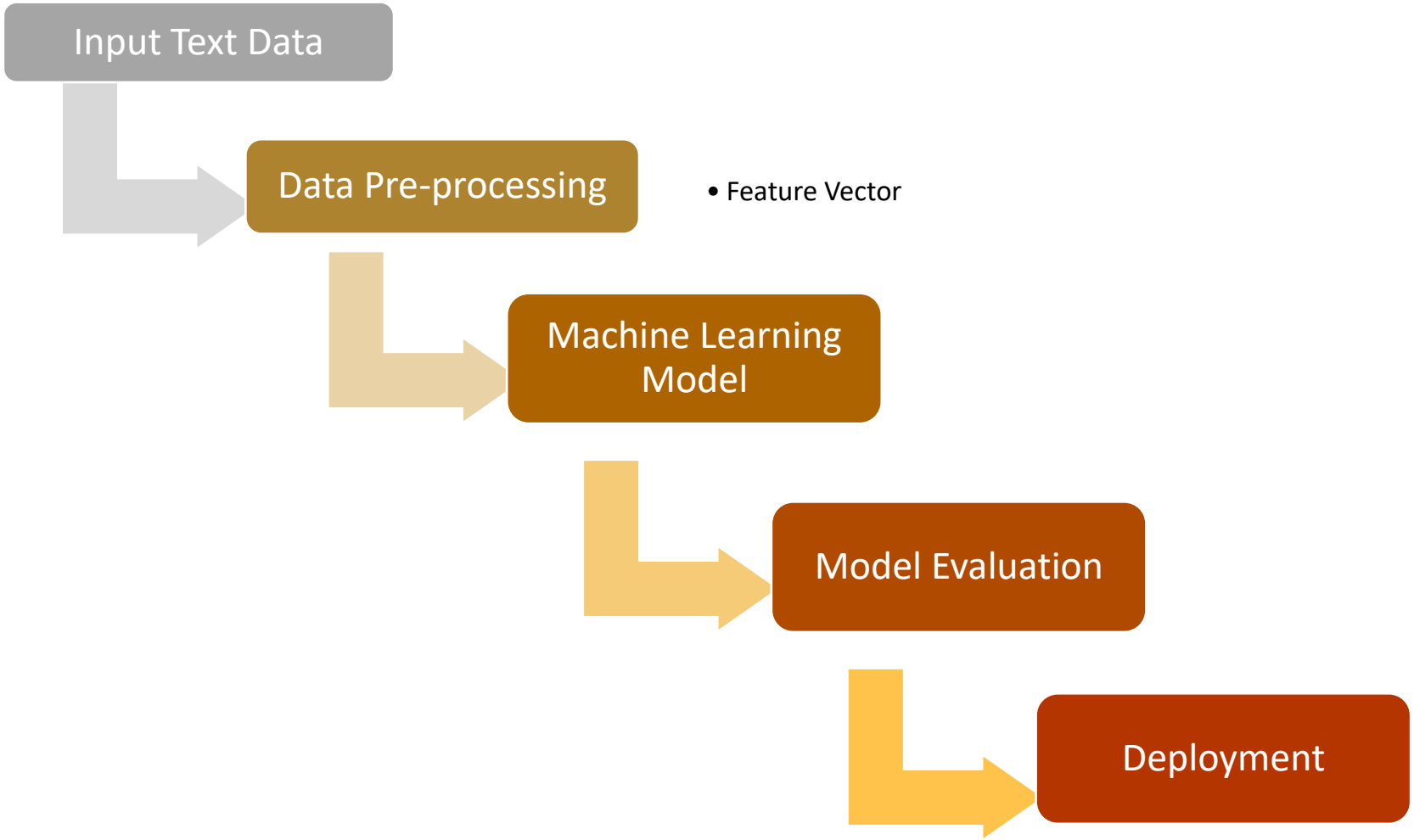
Technology Stack

- Python
- Scikit-learn
- Pandas
- Joblib

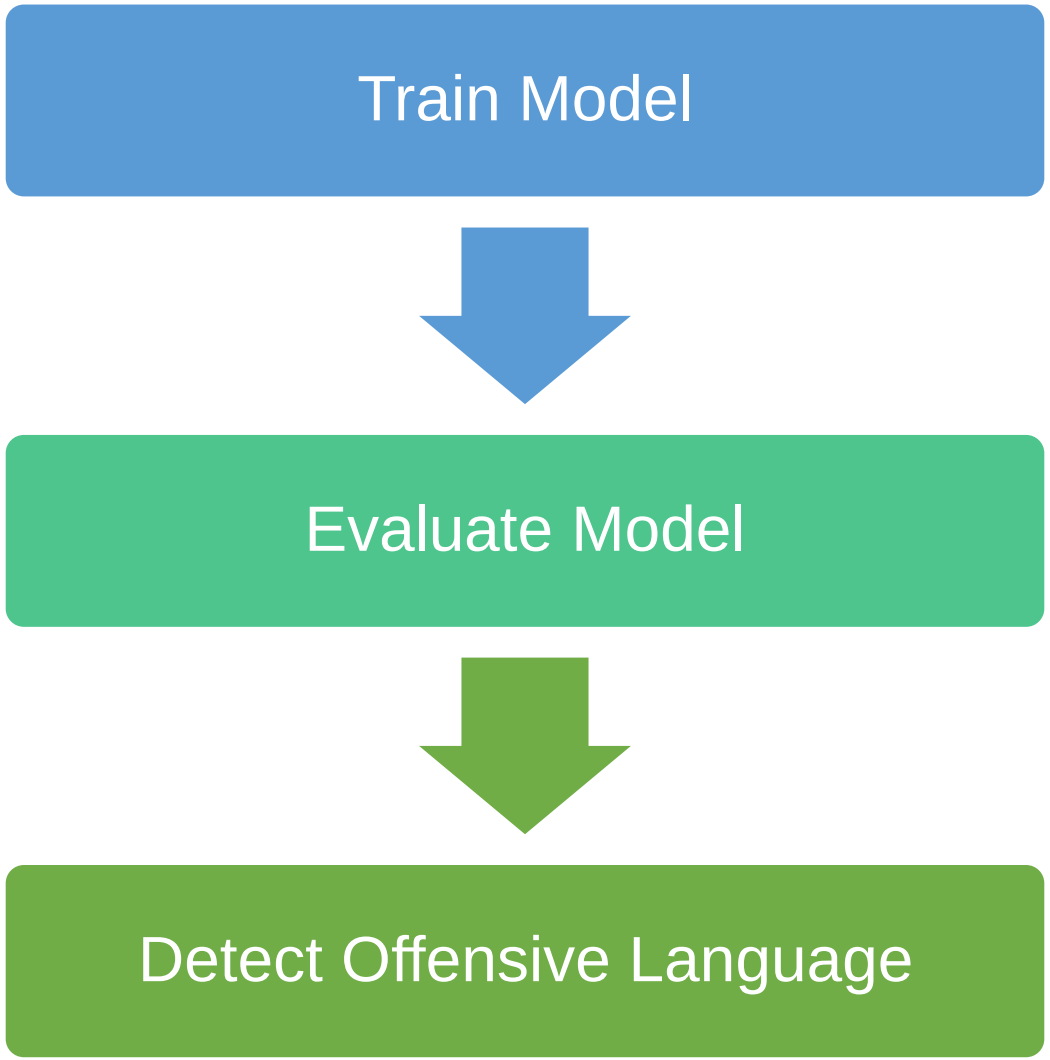
System Architecture

- Data Pre-processing: The input text data is pre-processed to convert it into a format that can be used for training the machine learning classifier.
- Feature Extraction: The pre-processed text data is then transformed into a feature vector representation that can be used for training the machine learning classifier.
- Machine Learning Model: A machine learning model is trained on the feature vectors to predict whether a given text example is offensive or non-offensive.
- Model Evaluation: The trained machine learning model is evaluated using metrics such as accuracy, precision, recall, and F1-score.
- Deployment: Once the machine learning model has been trained and evaluated, it can be deployed in a production environment where it can automatically detect offensive language in text.

System Design



Use CaseDiagram



Module Description

- Train Dataset: The test dataset contains offensive and non-offensive text examples so that the ML model can test and train itself to be more accurate
- Test Dataset: The test dataset also contains offensive and non-offensive text that the ML model can test it’s classifying proficiency after learning from the train dataset
- Python Script: The python script contains the code that makes prediction about the text and measures it’s accuracy using precision, recall, F1 score

Training

- Before using the module to detect offensive language we must first train it using a dataset containing of offensive and non-offensive text.
- The dataset must contain preferably a large number of offensive and non-offensive sentences that are labelled appropriately.
- In-order to get maximum accuracy and precision the machine learning model must be trained using a large and extremely varied number of examples

Testing

- After undergoing training using a dataset that contains a large amount of extremely varied offensive and non-offensive text examples the test dataset is used.

- After training the model, the performance of the model is evaluated on the testing set.

- The accuracy and classification report are used to evaluate the model

Python Script

- The python script is used to carry out the training and testing of the machine learning model on the datasets.

- TfidfVectorizer(): It is used to convert the text data in the training and testing datasets to numerical data that can be used for machine learning.

- MultinomialNB(): It creates a new instance of the MultinomialNB algorithm and train it using the numerical data and labels from the training dataset.

Results

- With our training datasets and test datasets, the Naïve Bayes algorithm based classifier gets an overall accuracy score of 76%.

- Naive Bayes is a popular choice for offensive language detection due to its speed, efficiency, and ability to work well with small datasets.

- Naive Bayes is a popular choice for offensive language detection due to its speed, efficiency, and ability to work well with small datasets.

Screenshots

Train Datasets:

	text	label
2	What time is it?	non-offensive
3	I love spending time with my family.	non-offensive
4	The flowers in the garden are beautiful.	non-offensive
5	This pizza tastes really good.	non-offensive
6	The movie we saw last night was entertaining.	non-offensive
7	You are so brilliant.	non-offensive
8	I hope you live.	non-offensive
9	I'm so happy right now.	non-offensive
10	I want to hug you.	non-offensive
11	I'm gonna encourage you to give it your best!	non-offensive
12	You're a soldier.	non-offensive
13	Go back home!	non-offensive
14	You're a very distinguished person	non-offensive
15	I hate people who love pizza.	offensive
16	You're the worst!	offensive
17	You're looking ugly today.	offensive
18	You're so disgusting.	offensive
19	Shut up!	offensive
20	I hate your dress.	offensive
21	This is ugly!	offensive
22	This cake is malicious.	offensive
23	You're the worst!	offensive
24	This is so much shit.	offensive
25	You have a terrible smile.	offensive
26	You have a beautiful smile.	non-offensive
27	You're such a coward	offensive
28	This is a disgusting tweet.	offensive
29	I love you.	non-offensive
30	You're such a hateful person.	offensive
31	You're such a competent person.	non-offensive
32	You have a terrible smile.	offensive
33	You're such a hero	non-offensive
34	You're very useless .	offensive
35	The weather is nice today.	non-offensive
36	I had a great time at the park.	non-offensive
37	The book I'm reading is really interesting.	non-offensive
38	The cat is sleeping on the couch.	non-offensive
39	Congratulations on your new job.	non-offensive

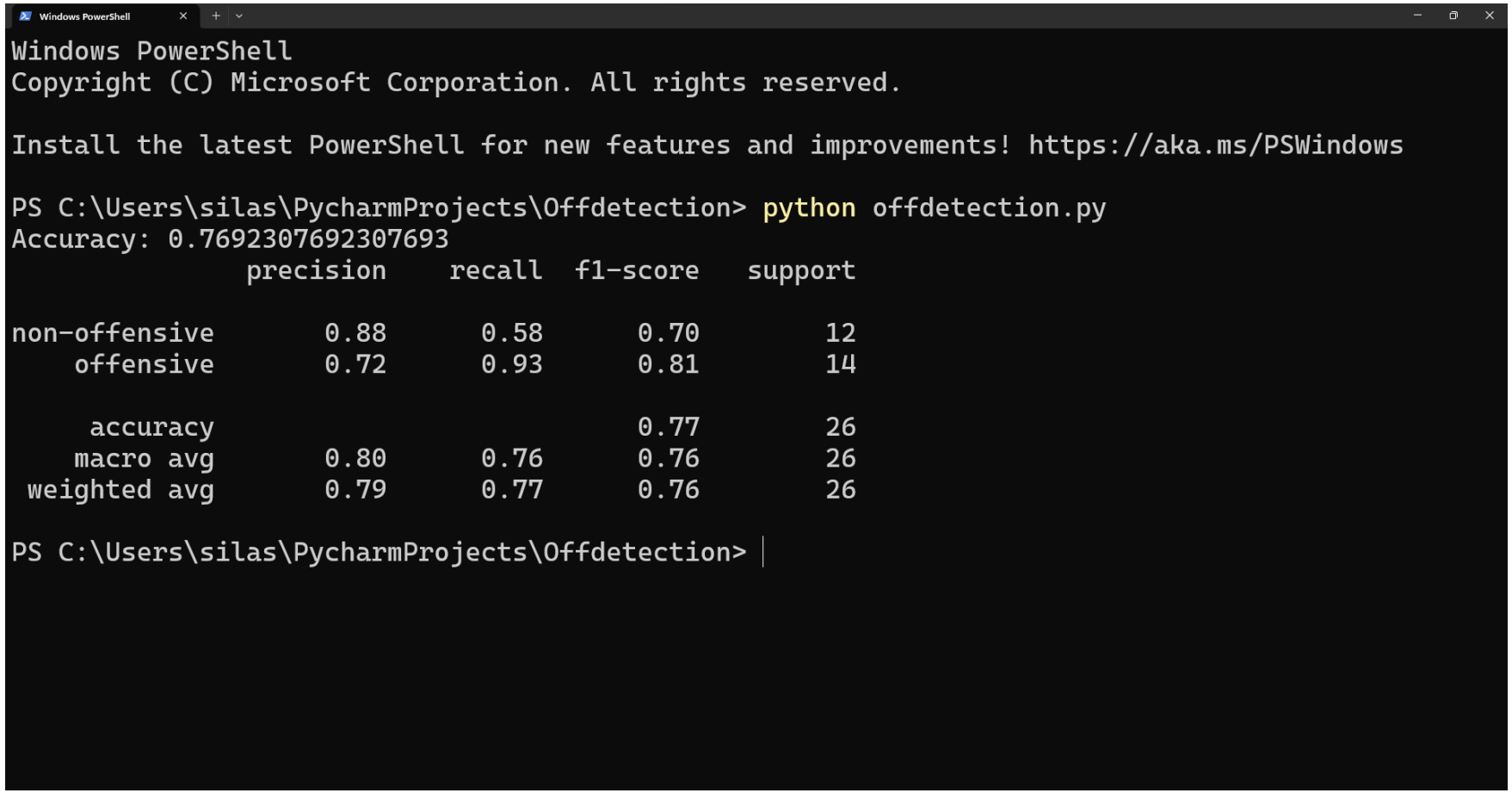
Test Dataset:

	text	label
2	This is a disgusting tweet.	offensive
3	I love you.	non-offensive
4	You are so brilliant.	non-offensive
5	I hate people who love pizza.	offensive
6	I hope you live.	non-offensive
7	You're the worst!	offensive
8	I'm so happy right now.	non-offensive
9	You're such a hateful person.	offensive
10	You're looking ugly today.	offensive
11	I want to hug you.	non-offensive
12	You're very useless .	offensive
13	You're so disgusting.	offensive
14	Shut up!	offensive
15	I hate your dress.	offensive
16	What a genius!	non-offensive
17	This is ugly!	offensive
18	You're such a competent person.	non-offensive
19	I'm gonna encourage you to give it your best!	non-offensive
20	You're a soldier.	non-offensive
21	This cake is malicious.	offensive
22	You're the worst!	offensive
23	Go back home!	non-offensive
24	This is so much shit.	offensive
25	You're a very distinguished person	non-offensive
26	You have a terrible smile.	offensive
27	You're such a hero	non-offensive
28		

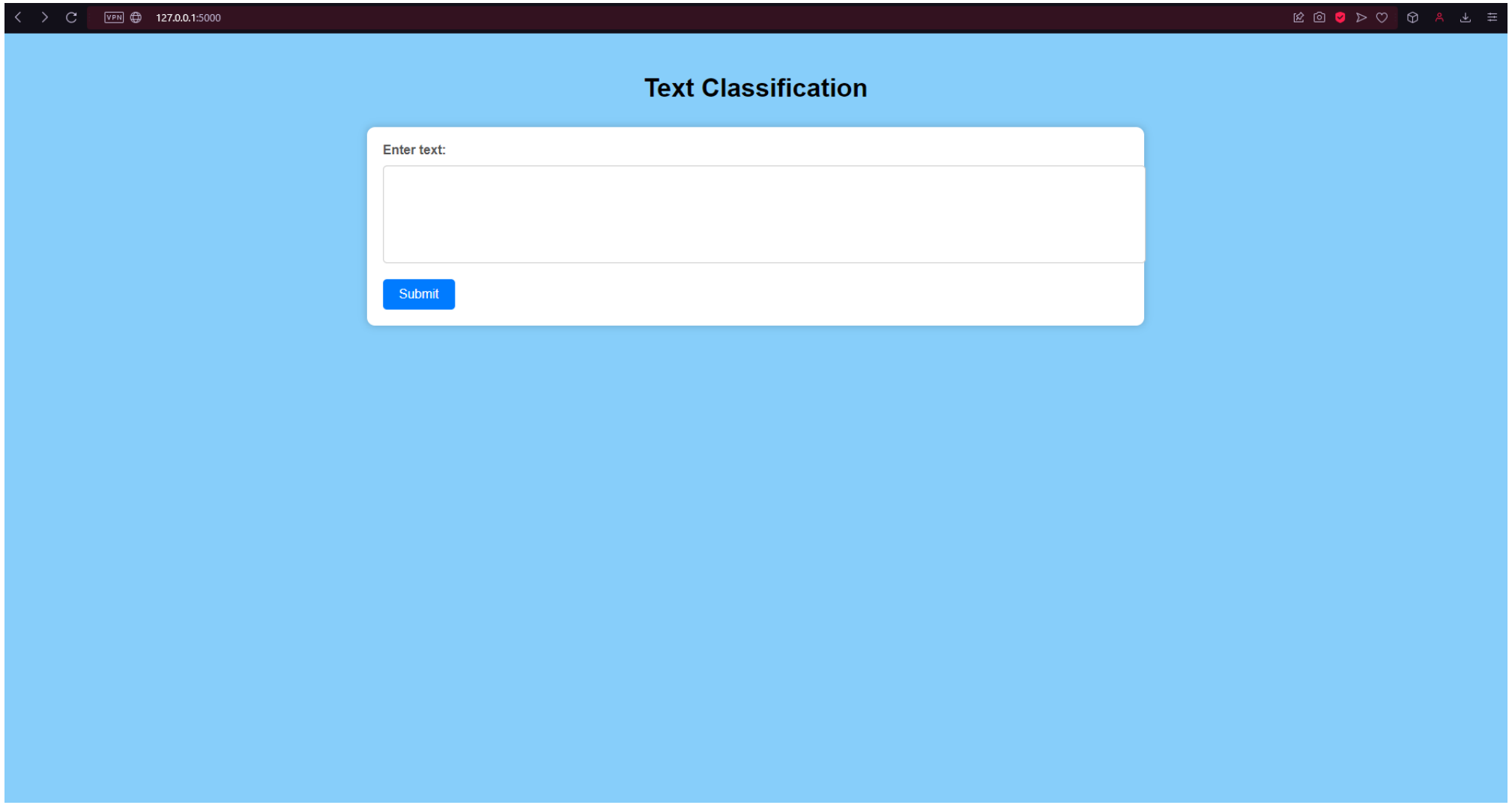
Source Code:

```python
import pandas as pd
import joblib
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report

# Load the vectorizer and classifier from disk
vectorizer = joblib.load('vectorizer.joblib')
clf = joblib.load('classifier.joblib')

# Load training dataset
train_df = pd.read_csv('train_dataset.csv')

# Load test dataset
test_df = pd.read_csv('test_dataset.csv')

# Convert text data to numerical data
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(train_df['text'])
y_train = train_df['label']
X_test = vectorizer.transform(test_df['text'])
y_test = test_df['label']

# Train a Multinomial Naive Bayes classifier
clf = MultinomialNB()
clf.fit(X_train, y_train)

# Make predictions on testing set
y_pred = clf.predict(X_test)

# Evaluate performance
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)
print(classification_report(y_test, y_pred))

# Save the classifier and vectorizer to disk
joblib.dump(clf, 'classifier.joblib')
joblib.dump(vectorizer, 'vectorizer.joblib')
```

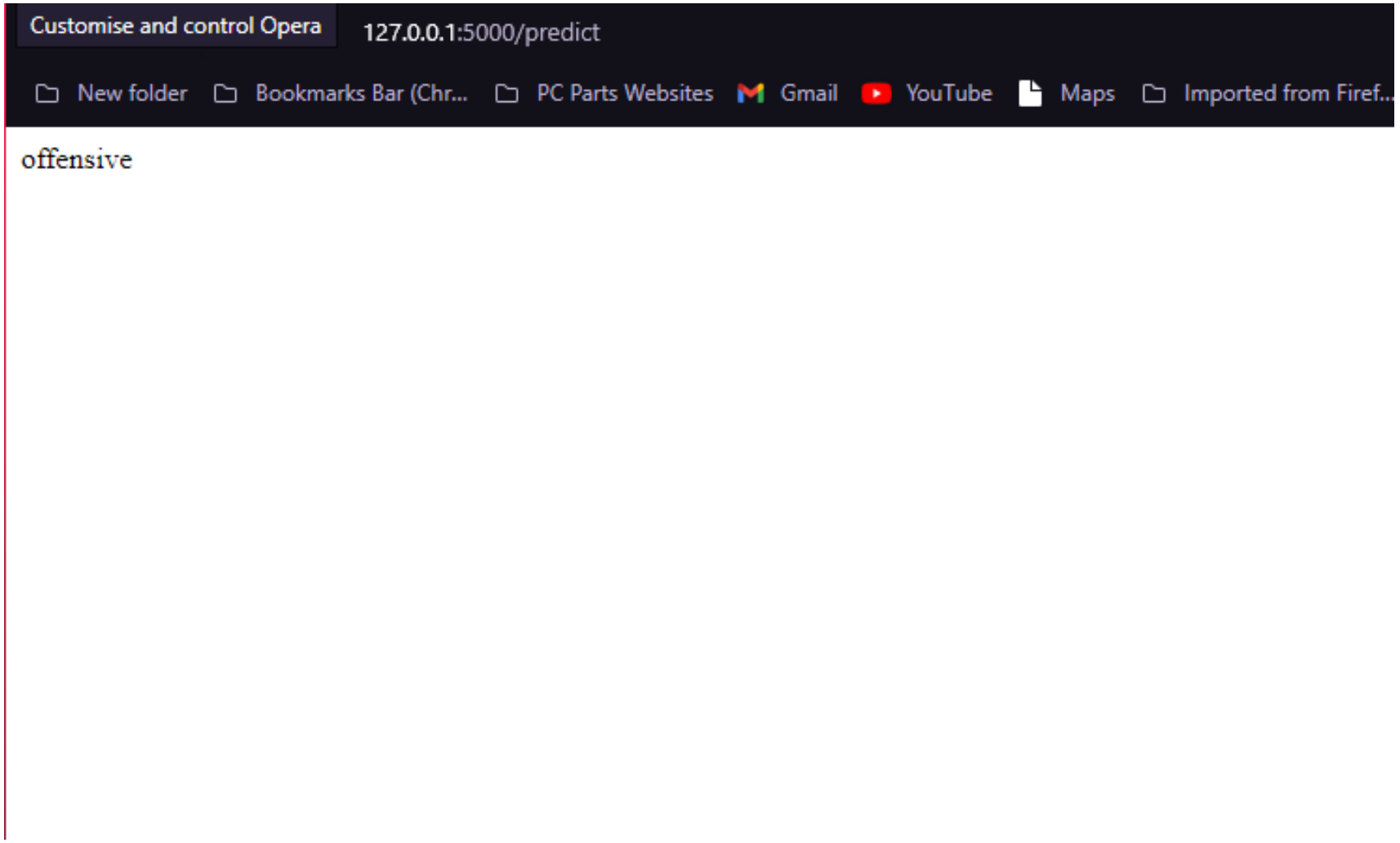
Output:


Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\silas\PycharmProjects\Offdetection> python offdetection.py
Accuracy: 0.7692307692307693
precision recall f1-score support
non-offensive 0.88 0.58 0.70 12
offensive 0.72 0.93 0.81 14
accuracy 0.77 26
macro avg 0.80 0.76 0.76 26
weighted avg 0.79 0.77 0.76 26
PS C:\Users\silas\PycharmProjects\Offdetection>

Web App:



Web App Output:



Conclusion

- This project is a feasible and effective solution for identifying and classifying offensive language in text data.
- The use of the CountVectorizer and Multinomial Naive Bayes algorithm ensures high accuracy in identifying offensive language in text data.
- Its implementation can contribute to a safer and more inclusive online environment, which is essential in today's digital age.

References

- [1] "Offensive Language Detection: A Review" by V. Bansal, R. Bhatia, and A. Rana (2020)
- [2] "A Survey on Offensive Language Detection Techniques" by N. Farhan and T. Kim (2020)
- [3] "Hate Speech and Offensive Language Detection: A Comprehensive Review" by A. Singh and A. Singh (2021)
- [4] "Survey of Methods for Offensive Language Detection" by N. Akhtar and W. Hu (2020)
- [5] "Offensive Language Detection: A Systematic Literature Review" by C. Rojas, L. M. Sanchez, and M. M. Crespo (2021)