VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“JNANA SANGAMA”, BELAGAVI - 590 018

Assignment report on

“Hate speech detection using Machine Learning”

submitted by

Swaroop - 4SF19IS111

for the subject of

Artificial Intelligence and Machine Learning

Under the Guidance of

Mrs. Madhura N Hegde

Assistant Professor,

Department of Information Science Engineering
At

SAHYADRI
College of Engineering & Management
Mangaluru – 575 007

Introduction

Hate speech detection is the model which identifies and detects hateful and offensive speech being poured on the internet. Social media is a place for many people to make hateful and offensive comments about others. So hate speech detection has become an important solution to problems in today's online world. As we understood the main goal to build this project, let's start with building the **Hate Speech detection project in python**.

Procedure

Before moving into the implementation part directly, let us get an insight into the steps in building a **Hate Speech detection project with Python**.

- Set up the development environment
- Understand the data
- Import the required libraries
- Preprocess the data
- Split the data
- Build the model
- Evaluate the results

Setting up the development environment

The first major step is to set up the development environment for building a **Hate Speech detection project with Python**. For developing a **Hate Speech detection project** you should have the system with Jupyter notebook software installed. Else, you can also use Google Colab https://colab.research.google.com/ for developing this project.

Understanding the data

The **dataset** for building our **hate speech detection model** is available on www.kaggle.com. The **dataset** consists of **Twitter hate speech detection data**, used to research hate-speech detection. The text in the data is classified as hate speech, offensive language, and neither. Due to the nature of the study, it's important to note that this dataset contains text that can be considered racist, sexist, homophobic, or generally offensive.

There are 7 columns in the hate speech detection dataset. They are index, count, hate_speech, offensive_language, neither, class and tweet. The description of the column is as follows.

index – This column has the index value

count– It has the number of users who coded each tweet

hate_speech – This column has the number of users who judged the tweet to be hate speech

offensive_language – It has the number of users who judged the tweet to be offensive

neither – This has the number of users who judged the tweet to be neither offensive nor non-offensive

class – it has a class label for the majority of the users, in which 0 denotes hate speech, 1 means offensive language and 2 denotes neither of them.

tweet – This column has the text tweet.

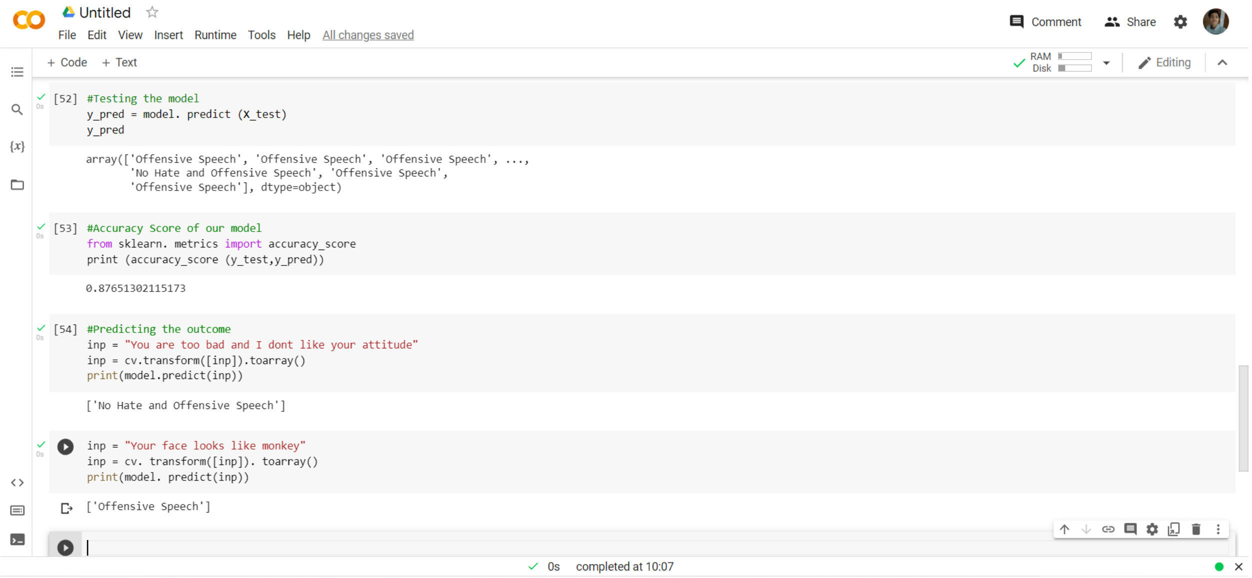
Importing the required libraries

After analyzing the data our next step is to import the required libraries for our project. Some of the libraries we use in this project are **pandas**, **numpy**, **scikit learn**, and **nltk**.

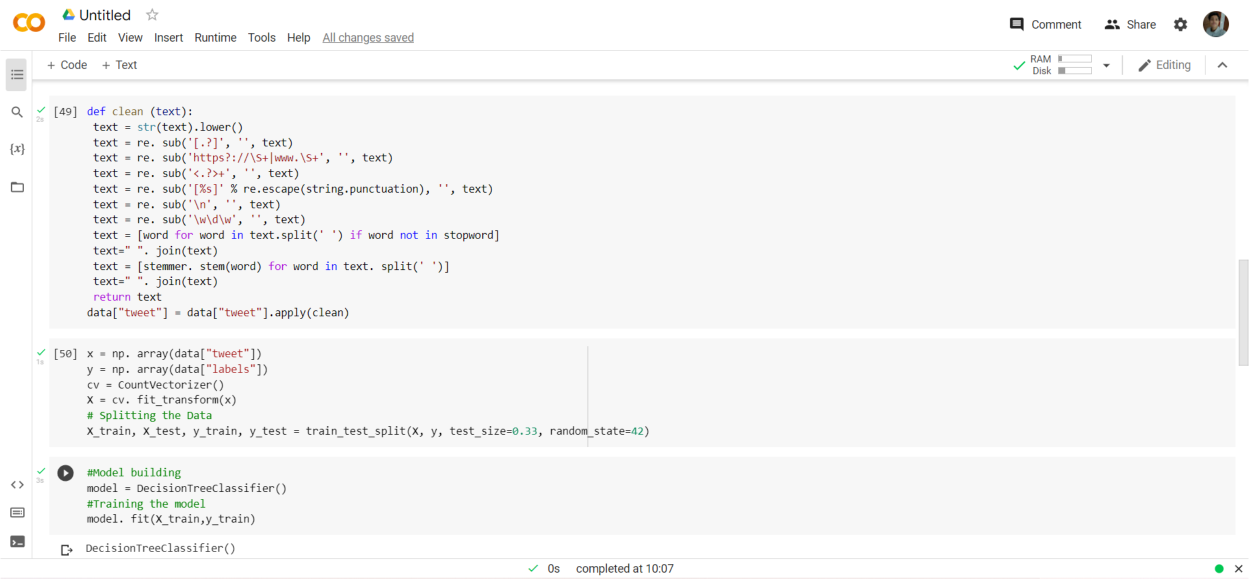
Links:

Github Link:
https://github.com/Swaroop-Acharya/AIML-mini-project

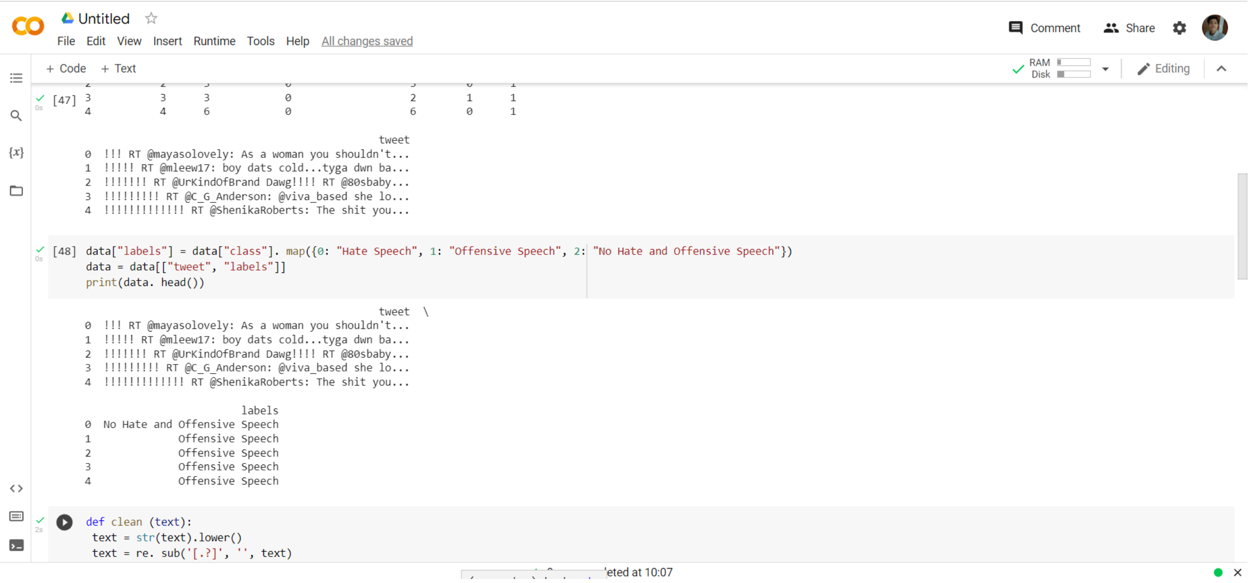
Result:



```
[52] #Testing the model
     y_pred = model. predict (X_test)
     y_pred

     array(['Offensive Speech', 'Offensive Speech', 'Offensive Speech', ...,
            'No Hate and Offensive Speech', 'Offensive Speech',
            'Offensive Speech'], dtype=object)

[53] #Accuracy Score of our model
     from sklearn. metrics import accuracy_score
     print (accuracy_score (y_test,y_pred))

     0.87651302115173

[54] #Predicting the outcome
     inp = "You are too bad and I dont like your attitude"
     inp = cv.transform([inp]).toarray()
     print(model.predict(inp))

     ['No Hate and Offensive Speech']

     inp = "Your face looks like monkey"
     inp = cv. transform([inp]). toarray()
     print(model. predict(inp))

     ['Offensive Speech']
```



```
[49] def clean (text):
      text = str(text).lower()
      text = re. sub('[.?]', '', text)
      text = re. sub('https?://\S+|www.\S+', '', text)
      text = re. sub('<.?>+', '', text)
      text = re. sub('[%s]' % re.escape(string.punctuation), '', text)
      text = re. sub('\n', '', text)
      text = re. sub('\w\d\w', '', text)
      text = [word for word in text.split(' ') if word not in stopword]
      text=" ". join(text)
      text = [stemmer. stem(word) for word in text. split(' ')]
      text=" ". join(text)
      return text
     data["tweet"] = data["tweet"].apply(clean)

[50] x = np. array(data["tweet"])
     y = np. array(data["labels"])
     cv = CountVectorizer()
     X = cv. fit_transform(x)
     # Splitting the Data
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)

     #Model building
     model = DecisionTreeClassifier()
     #Training the model
     model. fit(X_train,y_train)

     DecisionTreeClassifier()
```

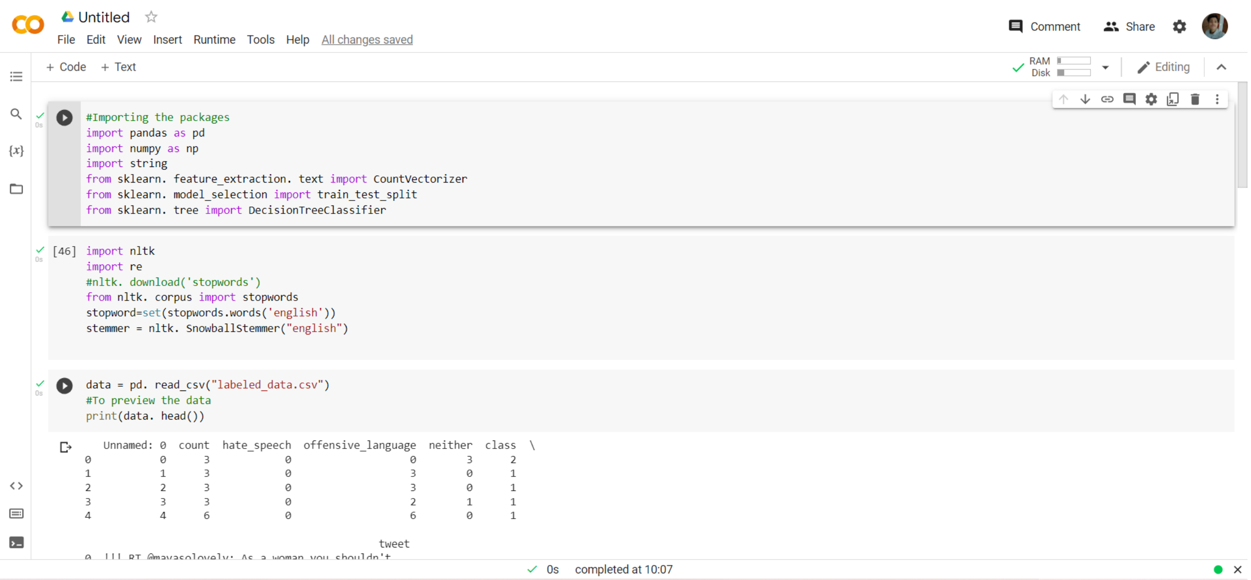


```
[47] 3           3     3         0                  2       1     1
     4           4     6         0                  6       0     1

                                              tweet
     0  !!! RT @mayasolovely: As a woman you shouldn't...
     1  !!!!! RT @mleew17: boy dats cold...tyga dwn ba...
     2  !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
     3  !!!!!!!!! RT @C_G_Anderson: @viva_based she lo...
     4  !!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...
```

```
[48] data["labels"] = data["class"]. map({0: "Hate Speech", 1: "Offensive Speech", 2: "No Hate and Offensive Speech"})
     data = data[["tweet", "labels"]]
     print(data. head())
```

```
                                              tweet  \
     0  !!! RT @mayasolovely: As a woman you shouldn't...
     1  !!!!! RT @mleew17: boy dats cold...tyga dwn ba...
     2  !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
     3  !!!!!!!!! RT @C_G_Anderson: @viva_based she lo...
     4  !!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...

                              labels
     0  No Hate and Offensive Speech
     1              Offensive Speech
     2              Offensive Speech
     3              Offensive Speech
     4              Offensive Speech
```

```
def clean (text):
 text = str(text).lower()
 text = re. sub('[.?]', '', text)
```



```
#Importing the packages
import pandas as pd
import numpy as np
import string
from sklearn. feature_extraction. text import CountVectorizer
from sklearn. model_selection import train_test_split
from sklearn. tree import DecisionTreeClassifier
```

```
[46] import nltk
     import re
     #nltk. download('stopwords')
     from nltk. corpus import stopwords
     stopword=set(stopwords.words('english'))
     stemmer = nltk. SnowballStemmer("english")
```

```
data = pd. read_csv("labeled_data.csv")
#To preview the data
print(data. head())
```

```
   Unnamed: 0  count  hate_speech  offensive_language  neither  class  \
0           0      3            0                   0        3      2
1           1      3            0                   3        0      1
2           2      3            0                   3        0      1
3           3      3            0                   2        1      1
4           4      6            0                   6        0      1

                                              tweet
```

Impact:

There are several potential impacts of doing a project on hate speech detection using machine learning. Some of these impacts may include:

Improved safety and well-being: By detecting and mitigating hate speech, machine learning can help create a safer and more inclusive online environment for all users.

Greater accountability: Machine learning can help identify and track individuals or groups who engage in hate speech, which can help hold them accountable for their actions.

Enhanced trust and credibility: By detecting and addressing hate speech, organizations and platforms can demonstrate their commitment to creating a safe and welcoming environment, which can enhance trust and credibility with users.

Increased efficiency: Automating the detection of hate speech using machine learning can help organizations and platforms more efficiently identify and address hateful content.

However, it is important to consider the potential ethical implications of using machine learning for hate speech detection, as it can be difficult to define and identify accurately, and there is the potential for false positives and false negatives to occur. It is important to carefully consider the potential impacts and take steps to mitigate any negative consequences.
and stars. These measurements can be affected by various factors, including atmospheric conditions, the observer's location, and the time of year.

Conclusion:

Hate speech detection using machine learning involves using algorithms to identify and classify text or speech that contains hateful or discriminatory language. This can be challenging because hate speech often involves the use of subtle cues and can be context-dependent. It can also be difficult to define and identify accurately due to the subjectivity of the concept.

To develop a machine learning model for hate speech detection, you will need a large dataset of labeled examples of hate speech and non-hate speech. You can then use this dataset to train a machine learning model to classify text or speech as either hate speech or non-hate speech.

There are various techniques that can be used for this task, including natural language processing (NLP) techniques such as sentiment analysis, and machine learning techniques such as support vector machines (SVMs) and deep learning networks.

It is important to note that hate speech detection using machine learning is a complex and ongoing research area, and there is no one-size-fits-all solution. It is also important to consider the ethical implications of using machine learning for hate speech detection, as it can be difficult to define and identify accurately, and there is the potential for false positives and false negatives to occur.