ABSTRACT :

At any given second, over a million tweets are uploaded through social media platforms like X or THREADS. We know that social media provides both positive and negative aspects. To address the negatives, we propose a simple technique for identifying hate speech. Our approach aims to filter out statements that contain hate or offensive content, ensuring a healthier communication environment. Leveraging data science, our project detects hate speech and assesses whether a tweet is harmful.

OBJECTIVE :

The objective of this project is to build a hate speech detection model that can automatically classify text data into categories such as hate speech, offensive language, and normal speech. By achieving this, we aim to contribute to the efforts of creating safer online communities and platforms. Thus it aims to classify textual content into non-hate or hate speech.

INTRODUCTION :

In this digital era, where 6,000 tweets are posted every second, the total amounts to approximately 360,000 tweets per minute. Imagine the daunting task of manually monitoring each tweet—assuming we employ a person who takes an average of 6 seconds to read each one. If one person were to monitor each tweet, they would need to handle 36,000 tweets simultaneously. Clearly, this becomes an overwhelming challenge. To address this issue, our project, Hate Speech Detection, comes into play. By collecting relevant data, we can automate the process of identifying hate speech. This automation ensures smoother communication by filtering out offensive words and harmful content.

METHODOLOGY :

1). DATA COLLECTION:

Data is the King. Thus its an essential to get the data from an authenticate place which covers all the things. Thus for that we find the data and download it from Kaggle, which is in csv formate. And then we fetch the data from csv file with the help of PANDAS lib.

2). DATA CLEANING:

The main job of Data Scientist is to clean the data in a way so that an ML algorithm can understand the data well. While the fact is most of time of Data Scientist spent time to clean data, here we had practise that. We used re and nltk lib to perform elimination processes of the unanted words and signs. Which make the given data at the stage where one can say the dataset is at the cleanest form. { For this one can check clean_data function }

3). SUPERVISED LEARNING :

Yes this is truth, hear we had used supervised learning algorithm. To make that possible we do following things:

1). Text Preprocessing: Using CountVectorizer to convert text data into a numerical format suitable for machine learning algorithms. This step involves tokenizing the text, converting it to lowercase, removing punctuation, and building a vocabulary of known words.

2). Data Splitting: Splitting dataset into training and testing sets using train_test_split. This step is crucial for evaluating the model's performance on unseen data.

3). Model Selection and Training: Choosing a decision tree classifier (DecisionTreeClassifier) as model and then fitting it to the training data using the fit method.

4). Model Evaluation: Using the trained model to make predictions on the test dataset (x_test) and then evaluating its performance using a confusion matrix (confusion_matrix). The confusion matrix helps you understand how well the model is performing in terms of true positives, true negatives, false positives, and false negatives.

Overall, this process represents a supervised learning approach, specifically classification, within the realm of data science. It involves preprocessing the data, splitting it into training and testing sets, selecting and training a model, and evaluating the model's performance.

4). ACCURACY & VISUALISING :

In the code snippet you provided, you're utilizing `seaborn` and `matplotlib` libraries to visualize the confusion matrix (`cm`) using a heatmap. Additionally, you're calculating the accuracy score of your model using `accuracy_score` from `sklearn.metrics`.

Here's what each part does:

1. **Visualizing Confusion Matrix**:
   - `sns.heatmap(cm, annot=True, fmt='.1f', cmap='YlGnBu')`: This line generates a heatmap visualization of the confusion matrix (`cm`). The `annot=True` parameter adds annotations to the heatmap, displaying the numeric values of the confusion matrix. `fmt='.1f'` specifies the format of the annotations (one decimal place). `cmap='YlGnBu'` sets the color map for the heatmap. This line is usually used to visually analyze the performance of the classifier by examining true positives, true negatives, false positives, and false negatives.
   - `plt.show()`: This line displays the heatmap.
2. **Calculating Accuracy Score**:
   - `accuracy_score(y_test, y_pred)`: This line calculates the accuracy score of your model by comparing the predicted labels (`y_pred`) with the actual labels

(`y_test`). It measures the proportion of correctly predicted labels out of all labels. The higher the accuracy score, the better the performance of the classifier.

Both the visualization of the confusion matrix and the accuracy score calculation are essential steps in evaluating the performance of a classification model. The confusion matrix provides insights into the model's behavior across different classes, while the accuracy score gives a single metric summarizing the overall performance.

5). SAMPLE:

```
- While concluding the project at last, we had provide code snippet
where you can easily give sample on run properly and also to check
where it is running or not ?
```

CODE:

Below is the task / code which you can do following for success

```
#import libraries

import numpy as np
import pandas as pd

dataset = pd.read_csv('hate_speech.csv')

# to check wheather given data has null or not and if then how much ?

dataset.isnull().sum()

Unnamed: 0            0
count                 0
hate_speech           0
offensive_language    0
neither               0
class                 0
tweet                 0
dtype: int64

dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24783 entries, 0 to 24782
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Unnamed: 0          24783 non-null  int64
 1   count               24783 non-null  int64
 2   hate_speech         24783 non-null  int64
 3   offensive_language  24783 non-null  int64
 4   neither             24783 non-null  int64
 5   class               24783 non-null  int64
 6   tweet               24783 non-null  object
dtypes: int64(6), object(1)
memory usage: 1.3+ MB
```

```
dataset.describe()
```

```
         Unnamed: 0         count   hate_speech  offensive_language  \
count  24783.000000  24783.000000  24783.000000        24783.000000
mean   12681.192027      3.243473      0.280515            2.413711
std     7299.553863      0.883060      0.631851            1.399459
min        0.000000      3.000000      0.000000            0.000000
25%     6372.500000      3.000000      0.000000            2.000000
50%    12703.000000      3.000000      0.000000            3.000000
75%    18995.500000      3.000000      0.000000            3.000000
max    25296.000000      9.000000      7.000000            9.000000

            neither         class
count  24783.000000  24783.000000
mean       0.549247      1.110277
std        1.113299      0.462089
min        0.000000      0.000000
25%        0.000000      1.000000
50%        0.000000      1.000000
75%        0.000000      1.000000
max        9.000000      2.000000
```

```
dataset['labels'] = dataset['class'].map({0:'Hate Speech',
1:'offencive language', 2:'no hate or offensive languages'})

data = dataset[['tweet','labels']]

data
```

```
                                                   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...
...                                                  ...
24778  you's a muthaf***in lie &#8220;@LifeAsKing: @2...
24779  you've gone and broke the wrong heart baby, an...
24780  young buck wanna eat!!.. dat nigguh like I ain...
24781              youu got wild bitches tellin you lies
24782  ~~Ruffled | Ntac Eileen Dahlia - Beautiful col...

                               labels
0      no hate or offensive languages
1                  offencive language
2                  offencive language
```

```
3                    offencive language
4                    offencive language
...                                 ...
24778                offencive language
24779  no hate or offensive languages
24780                offencive language
24781                offencive language
24782  no hate or offensive languages

[24783 rows x 2 columns]
```

```
import re
import nltk

# removal of stop words and stemming the words

from nltk.corpus import stopwords

stop_words_set = set(stopwords.words('english'))  # Renamed the
variable to stop_words_set

stemmer = nltk.SnowballStemmer('english')

import re
import string

def clean_data(text):
    text = str(text).lower()
    text = re.sub(r'https?://\S+|www\.\S+', '', text)
    text = re.sub(r'\[.*?\]', '', text)
    text = re.sub(r'<.*?>+', '', text)
    text = re.sub(r'[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub(r'\n', '', text)
    text = re.sub(r'\w*\d\W', '', text)

    # STOP WORDS REMOVAL
    stop_words = set(stop_words_set)  # Changed variable name to
stop_words_set
    text = ' '.join([word for word in text.split() if word not in
stop_words])

    # STEMMING
    text = [stemmer.stem(word) for word in text.split()]
    text = ' '.join(text)

    return text


data['tweet'] = data['tweet'].apply(clean_data)
```

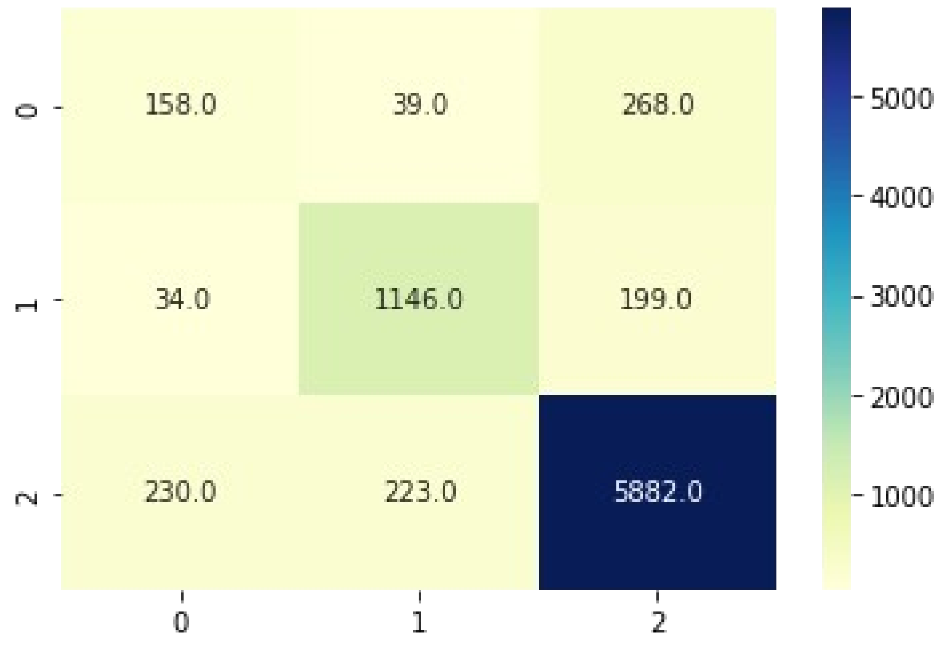
```
/tmp/ipykernel_14400/3370867953.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  data['tweet'] = data['tweet'].apply(clean_data)

data

                                                   tweet  \
0      rt mayasolov woman shouldnt complain clean hou...
1      rt boy dat coldtyga dwn bad cuffin dat hoe 1st...
2      rt urkindofbrand dawg rt 80sbaby4lif ever fuck...
3                 rt cganderson vivabas look like tranni
4      rt shenikarobert shit hear might true might fa...
...                                                  ...
24778  yous muthafin lie 8220lifeask 20pearl coreyema...
24779  youv gone broke wrong heart babi drove redneck...
24780  young buck wanna eat dat nigguh like aint fuck...
24781                     youu got wild bitch tellin lie
24782  ruffl ntac eileen dahlia beauti color combin p...

                               labels
0      no hate or offensive languages
1                  offencive language
2                  offencive language
3                  offencive language
4                  offencive language
...                               ...
24778              offencive language
24779  no hate or offensive languages
24780              offencive language
24781              offencive language
24782  no hate or offensive languages

[24783 rows x 2 columns]

X = np.array(data['tweet'])
Y = np.array(data['labels'])

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split

cv = CountVectorizer()
x = cv.fit_transform(X)

x_train, x_test, y_train, y_test = train_test_split(x, Y,
test_size=0.33, random_state=42)
```

```
#model

from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier()
dt.fit(x_train, y_train)
y_pred = dt.predict(x_test)

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

cm
```

```
array([[ 158,   39,  268],
       [  34, 1146,  199],
       [ 230,  223, 5882]])
```

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.heatmap(cm, annot=True, fmt='.1f', cmap='YlGnBu')
plt.show()
```

```
/home/poojandoshi/.local/lib/python3.10/site-packages/matplotlib/
projections/__init__.py:63: UserWarning: Unable to import Axes3D. This
may be due to multiple versions of Matplotlib being installed (e.g. as
a system package and as a pip package). As a result, the 3D projection
is not available.
  warnings.warn("Unable to import Axes3D. This may be due to multiple
versions of "
```



```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)

0.8785915148551168

sample = input()
sample = clean_data(sample)
# print(sample)

# Transform the preprocessed sample text using CountVectorizer
sample_vectorized = cv.transform([sample]).toarray()
print(sample_vectorized)

# Convert the transformed sample into an array
# sample_array = sample_vectorized.toarray()

# print(sample_array)
# Predict the label for the sample
predicted_label = dt.predict(sample_vectorized)

print("Predicted label:", predicted_label)
# print(accuracy_score(y_pred=sample_vectorized))

xxx
[[0 0 0 ... 0 0 0]]
Predicted label: ['no hate or offensive languages']
```

CONCLUSION:

So now at last this ends but as a purpose of project. We know there are around 10K official languages in world. There are total many languages where hates are spread rapidly online. Thus to make it train to do simmiliar with other languages, means other then english becasue total umber of tweets other then english are also massive.

Thus we can say that with the help of data science we are now able to automate thee hate detection tweets.