```python
#!/usr/bin/env python
# coding: utf-8

#Dataset-
#https://www.kaggle.com/datasets/yasserh/twitter-tweets-sentiment-dataset

# In[12]:


import pandas as pd
import numpy as np


# In[13]:


data=pd.read_csv("Tweets.csv")


# In[14]:


data


# In[15]:


data.info()


# In[16]:


data.describe()


# In[17]:


data["Labels"]=data["sentiment"].map({"negative":"Hate Speech","neutral":"Offensive","positive":"No any"})
data


# In[18]:


data=data[["text","Labels"]]
data


# In[19]:


import re
import nltk
from nltk.corpus import stopwords
stemmer = nltk.SnowballStemmer("english")


# In[51]:


#data cleaning
def clean_data(Text):
    Text = str(Text).lower()
    Text = re.sub("<.*?>&", '', Text)
    Text = re.sub("https//\!", '', Text)
    Text = [stemmer.stem(word) for word in Text.split(' ')]
    Text = ' '.join(Text)
    return Text


# In[52]:
```

```python
data["text"] = data["text"].apply(clean_data)


# In[53]:


data


# In[54]:


x = np.array(data["text"])
y = np.array(data["Labels"])


# In[57]:


#Train-Test classification
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split


# In[58]:


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


# In[60]:


x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.30,random_state=42)


# In[61]:


x_train


# In[62]:


#Building ML model
from sklearn.tree import DecisionTreeClassifier


# In[64]:


dt = DecisionTreeClassifier()
dt.fit(x_train,y_train)


# In[65]:


y_pred = dt.predict(x_test)


# In[70]:


#CM and Accuracy
from sklearn.metrics import confusion_matrix
CM = confusion_matrix(y_test,y_pred)


# In[71]:
```

```
CM


# In[75]:


import seaborn as sns
import matplotlib.pyplot as ply
get_ipython().run_line_magic('matplotlib', 'inline')


# In[80]:


sns.heatmap(Cm, annot=True, fmt=".1f")


# In[81]:


from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)  #63% accuracy that Hate speech or Offensive words are detected.


# In[91]:


sample = "Let's kill the people!"
sample = clean_data(sample)
sample


# In[92]:


data1 = cv.transform([sample]).toarray()
data1


# In[93]:


dt.predict(data1)
```