```
import pandas as pd
import numpy as np
import nltk
import re
import string
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import BernoulliNB
from sklearn.tree import DecisionTreeClassifier
from nltk.util import pr
from nltk.corpus import stopwords

data=pd.read_csv("HateSpeechData.csv")
print(data.head())

nltk.download('stopwords')
stemmer=nltk.SnowballStemmer("english")
stopword = stopwords.words('english')

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

data=data[["tweet","labels"]]
print(data.head())
```

```
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)

x = np.array(data["tweet"])
y = np.array(data["labels"])

cv = CountVectorizer()
X = cv.fit_transform(x) # Fit the Data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)

clf = DecisionTreeClassifier()
clf.fit(X_train,y_train)
```

```
sample="bitch"

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

print(clf.predict(data))
```