

```
In [5]: import pandas as pd
import numpy as np
```

```
In [6]: df=pd.read_csv('file.csv')
df
```

Out[6]:

	tweets	labels
0	ChatGPT: Optimizing Language Models for Dialog...	neutral
1	Try talking with ChatGPT, our new AI system wh...	good
2	ChatGPT: Optimizing Language Models for Dialog...	neutral
3	THRILLED to share that ChatGPT, our new model ...	good
4	As of 2 minutes ago, @OpenAI released their ne...	bad
...
219289	Other Software Projects Are Now Trying to Repl...	bad
219290	I asked #ChatGPT to write a #NYE Joke for SEOs...	good
219291	chatgpt is being disassembled until it can onl...	bad
219292	2023 predictions by #chatGPT. Nothing really s...	bad
219293	From ChatGPT, neat stuff https://t.co/qjjUF2Z2m0	neutral

219294 rows × 2 columns

```
In [7]: from sklearn.feature_extraction.text import CountVectorizer
```

```
In [8]: from sklearn.naive_bayes import BernoulliNB,MultinomialNB
```

```
In [9]: vectorizer1 = CountVectorizer(binary = True)
vectorizer2 = CountVectorizer(binary = False)
```

```
In [10]: x1 = vectorizer1.fit_transform(df.tweets)
x2 = vectorizer2.fit_transform(df.tweets)
```

```
In [11]: y = df.labels
```

```
In [12]: from sklearn.model_selection import train_test_split
xtrain1,xtest1,ytrain,ytest=train_test_split(x1,y,test_size=0.25,random_state=42)
```

```
In [13]: xtrain2,xtest2,ytrain,ytest = train_test_split(x2,y,test_size=0.25,random_state=42)
```

```
In [14]: bnb = BernoulliNB()
mnb = MultinomialNB()
```

```
In [15]: bnb.fit(xtrain1,ytrain)
```

Out[15]: BernoulliNB()

```
In [16]: mnb.fit(xtrain2,ytrain)
```

Out[16]: MultinomialNB()

```
In [17]: predictions = bnb.predict(xtest1)
```

```
In [18]: predictions1 = mnb.predict(xtest2)
```

```
In [19]: from sklearn.metrics import accuracy_score
```

```
In [20]: accuracy_score(ytest,predictions)
```

```
Out[20]: 0.6216255654457902
```

```
In [21]: accuracy_score(ytest,predictions1)
```

```
Out[21]: 0.7144681161535094
```

```
In [22]: from sklearn.feature_extraction.text import TfidfVectorizer
```

```
In [23]: vector1 = TfidfVectorizer(binary = False)
```

```
In [35]: x3 = vector1.fit_transform(df.tweets)
```

```
In [36]: from sklearn.model_selection import train_test_split
```

```
In [37]: xtrain2,xtest2,ytrain,ytest = train_test_split(x3,y,test_size=0.25,random_state=42)
```

```
In [47]: mnb2 = MultinomialNB()
```

```
In [48]: mnb.fit(xtrain2,ytrain)
```

```
Out[48]: MultinomialNB()
```

```
In [49]: predictions1 = mnb.predict(xtest2)
```

```
In [50]: from sklearn.metrics import accuracy_score
```

```
In [51]: accuracy_score(ytest,predictions1)
```

```
Out[51]: 0.7238435721581788
```

```
In [ ]: #conclusion:: The above data shows better accuracy score for MultinomialNB Model with frequency based vecto
```

```
In [53]: import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
def remove_stopwords(text):
    stop_words = set(stopwords.words('english'))
    words = text.split()
    filtered_words = [word for word in words if word.lower() not in stop_words]
    return ' '.join(filtered_words)
df['tweets'] = df['tweets'].apply(remove_stopwords)
x=df.iloc[:,0]
y=df.iloc[:,1]
x1=vectorizer1.fit_transform(x)
x2=vectorizer2.fit_transform(x)
x3=vector1.fit_transform(x)
xtrain1,xtest1,ytrain,ytest=train_test_split(x1,y,test_size=0.25,random_state=1)
xtrain2,xtest2,ytrain,ytest=train_test_split(x2,y,test_size=0.25,random_state=1)
xtrain3,xtest3,ytrain,ytest=train_test_split(x3,y,test_size=0.25,random_state=1)
bnb.fit(xtrain1,ytrain)
mnb.fit(xtrain2,ytrain)
mnb2.fit(xtrain3,ytrain)
pred1=bnb.predict(xtest1)
pred2=mnb.predict(xtest2)
pred3=mnb2.predict(xtest3)
print('Accuracy score for Binary-based Vectors is',accuracy_score(pred1,ytest))
print('Accuracy score for Frequency-based vectors is',accuracy_score(pred2,ytest))
print('Accuracy score for tfidf score based vectors is',accuracy_score(pred3,ytest))
```

```
[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\PC-19\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!
```

```
Accuracy score for Binary-based Vectors is 0.6446811615350941
Accuracy score for Frequency-based vectors is 0.7238435721581788
Accuracy score for tfidf score based vectors is 0.6569750474244856
```

```
In [ ]: # conclusion- The above data shows better accuracy score for MultinomialNB Model with frequency based vecto
```