In [21]:

```
import pandas as pd
df=pd.read_csv("Training_Essay_Data.csv")
df.head()
```

Out[21]:

	text	generated
0	Car-free cities have become a subject of incre...	1.0
1	Car Free Cities Car-free cities, a concept ga...	1.0
2	A Sustainable Urban Future Car-free cities ...	1.0
3	Pioneering Sustainable Urban Living In an e...	1.0
4	The Path to Sustainable Urban Living In an ...	1.0

In [22]:

```
df.shape
```

Out[22]:

```
(29151, 2)
```

In [23]:

```
df=df.dropna() #removing NaN values
df.shape
```

Out[23]:

```
(18520, 2)
```

In [24]:

```
df=df.drop(range(520))
```

In [26]:

```
df.shape
```

Out[26]:

```
(18000, 2)
```

In [70]:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import BernoulliNB, MultinomialNB
vectorizer1=CountVectorizer(binary=True)
vectorizer2=CountVectorizer(binary=False)
x1=vectorizer1.fit_transform(df.text)
x2=vectorizer2.fit_transform(df.text)
y=df.generated
y.shape
```

Out[70]:

```
(18000,)
```

In [71]:

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x1,y,test_size=0.25,random_state=41)
model1=BernoulliNB()
model1.fit(xtrain,ytrain)
```

Out[71]:

```
BernoulliNB()
```

In [72]:

```
x2train,x2test,y2train,y2test=train_test_split(x2,y,test_size=0.25,random_state=41)
model2=MultinomialNB()
model2.fit(x2train,y2train)
```

Out[72]:

```
MultinomialNB()
```

In [73]:

```
pred=model1.predict(xtest)
```

In [77]:

```
pred2=model2.predict(x2test)
```

In [78]:

```
from sklearn.metrics import accuracy_score , confusion_matrix
a=accuracy_score(ytest,pred)
print("accuracy score for BernoulliNB with Countvectorizer is ",a)
c=confusion_matrix(ytest,pred)
print("confusion matrix for BernoulliNB is \n",c)
m=accuracy_score(y2test,pred2)
print("accuracy score for MultinomialNB with countvectorizer is ",m)
cm=confusion_matrix(y2test,pred2)
print("confusion matrix for MultinomialNB is \n",cm)
```

```
accuracy score for BernoulliNB with Countvectorizer is  0.9784444444444444
confusion matrix for BernoulliNB is
 [[4001   18]
 [  79  402]]
accuracy score for MultinomialNB with countvectorizer is  0.9764444444444444
confusion matrix for MultinomialNB is
 [[3977   42]
 [  64  417]]
```

```
interpret the confusion matrix c: here total 18 datapoints were actually from class 0 but classified to be
class 1
and 79 data points were of class 1 but missclassified to be class 0
```

In [33]:

```
#using MultinomialNB model with Tfidf vectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer2=TfidfVectorizer(stop_words='english')
x2=vectorizer2.fit_transform(df.text)
```

In [34]:

```
y=df.generated
y.shape
xtrain,xtest,ytrain,ytest=train_test_split(x2,y,test_size=0.25,random_state=30)
```

In [35]:

```
from sklearn.naive_bayes import  MultinomialNB
model=MultinomialNB()
model.fit(xtrain,ytrain)
```

Out[35]:

```
MultinomialNB()
```

In [36]:

```
pred=model.predict(xtest)
```

In [37]:

```
b=accuracy_score(ytest,pred)
print("accuracy score for MultinomialNB with tfidf is ",b)
```

```
accuracy score for MultinomialNB with tfidf is  0.9251111111111111
```

by looking at the accuracy scores of all the three models the highest accuracy score (0.97844)is of model1 i.e bernoulliNB model using countvectorizer

In [81]:

```
#to save the best model for given dataset use joblib
import joblib
joblib.dump(model1,'bernoulli_nb_model.pkl') #model is saved
```

Out[81]:

```
['bernoulli_nb_model.pkl']
```

In [82]:

```
Propermodel = joblib.load('bernoulli_nb_model.pkl')
#reloaded anytime and can be use for predictions
```