```
import pandas as pd
import numpy as np
```

```
data = pd.read_csv("spam.csv",encoding = 'cp1252')
```

```
data.head()
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

```
data[['v1' , 'v2' ]]
```

	v1	v2
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...
5568	ham	Will Ì_ b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other s...
5570	ham	The guy did some bitching but I acted like i'd...
5571	ham	Rofl. Its true to its name

5572 rows × 2 columns

```
data['v1'] = data['v1'].apply(lambda x:0 if x == 'ham' else 1)
```

```
data
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	0	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	0	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	0	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	0	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN
...	...	...	...	...	...
5567	1	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	0	Will Ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	0	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	0	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN

```
#Pre PRocessing
def process(x):
    temp=[]
    document=nlp(x.lower())
    print(document)
    for i in document:
        if i.is_stop!= True and i.is_punct!=True:
            print(i)
```

```
            temp.append(i.lemma_)
            print(temp)
        else:
            pass
            return (' '.join(temp))
```

```
data.head()
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN

```
from  sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(analyzer='word',stop_words='english')
text_vector = vectorizer.fit_transform(data['v2'].values.tolist())
print(text_vector)
```

```
  (0, 8026)     0.19609779550499865
  (0, 1051)     0.3509649021061901
  (0, 3494)     0.16470488207184114
  (0, 1994)     0.2964965675440533
  (0, 1701)     0.33503393550839805
  (0, 4349)     0.2964965675440533
  (0, 8227)     0.23740046706740073
  (0, 3534)     0.19387320529717864
  (0, 1703)     0.2964965675440533
  (0, 1271)     0.2625103008882829
  (0, 2271)     0.27179815735762314
  (0, 5741)     0.2745089285415426
  (0, 4224)     0.3509649021061901
  (1, 5369)     0.5465881710238072
  (1, 8134)     0.4316010362639011
  (1, 4192)     0.5236458071582338
  (1, 4385)     0.4082988561907181
  (1, 5343)     0.27211951321382544
  (2, 77)       0.23759715224911548
  (2, 1128)     0.1707825659976717
  (2, 6062)     0.1707825659976717
  (2, 7701)     0.12576907263059747
  (2, 7028)     0.1989696587085652
  (2, 6010)     0.1808417865094903
  (2, 6115)     0.16914304332607796
  :     :
  (5567, 5118)  0.2445888397614688
  (5567, 8202)  0.19074118816829963
  (5567, 2000)  0.185955090206136
  (5567, 5894)  0.19532744699307247
  (5567, 6062)  0.23098372602432177
  (5568, 2907)  0.6005703500933404
  (5568, 3252)  0.5182632994409236
  (5568, 8390)  0.37764633472218584
  (5568, 3463)  0.33726519867912935
  (5568, 3789)  0.3381624442072128
  (5569, 7168)  0.6095307789831879
  (5569, 5673)  0.6095307789831879
  (5569, 4992)  0.5068968918274174
  (5570, 1500)  0.42660925054744336
  (5570, 900)   0.40724464263367516
  (5570, 4040)  0.35477601883872634
  (5570, 3587)  0.30410983535074937
  (5570, 1737)  0.35477601883872634
  (5570, 3373)  0.3451921871853967
  (5570, 2532)  0.23146710969423193
  (5570, 4485)  0.20020413973165185
  (5570, 8071)  0.23479081568562485
  (5570, 3265)  0.19999603918651723
  (5571, 6323)  0.7930026248542038
  (5571, 7656)  0.6092182178615007
```

```
#Splitting the dataset
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(text_vector.toarray(),data['v1'],test_size=0.2,random_state=20)
```

```
len(x_train)
```

```
4457
```

```
#Data Modelling
from sklearn.naive_bayes import BernoulliNB


modelB = BernoulliNB()
modelB.fit(x_train,y_train)
print(modelB.score(x_train,y_train))
```

```
0.9847431007404084
```

```
y_predictedB = modelB.predict(x_test)


from sklearn.metrics import accuracy_score
print(accuracy_score(y_test,y_predictedB))
```

```
0.9829596412556054
```

```
#Best Model is BernoulliNB with 98% accuracy
```