In [3]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
```

In [24]:

In [25]:

Out[25]:

	Unnamed: 0	label	text	label_num
0	605	ham	Subject: enron methanol ; meter # : 988291\r\n...	0
1	2349	ham	Subject: hpl nom for january 9 , 2001\r\n(see...	0
2	3624	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...	0
3	4685	spam	Subject: photoshop , windows , office . cheap ...	1
4	2030	ham	Subject: re : indian springs\r\nthis deal is t...	0

In [26]:

Out[26]:

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	5171.0	2585.000000	1492.883452	0.0	1292.5	2585.0	3877.5	5170.0
label_num	5171.0	0.289886	0.453753	0.0	0.0	0.0	1.0	1.0

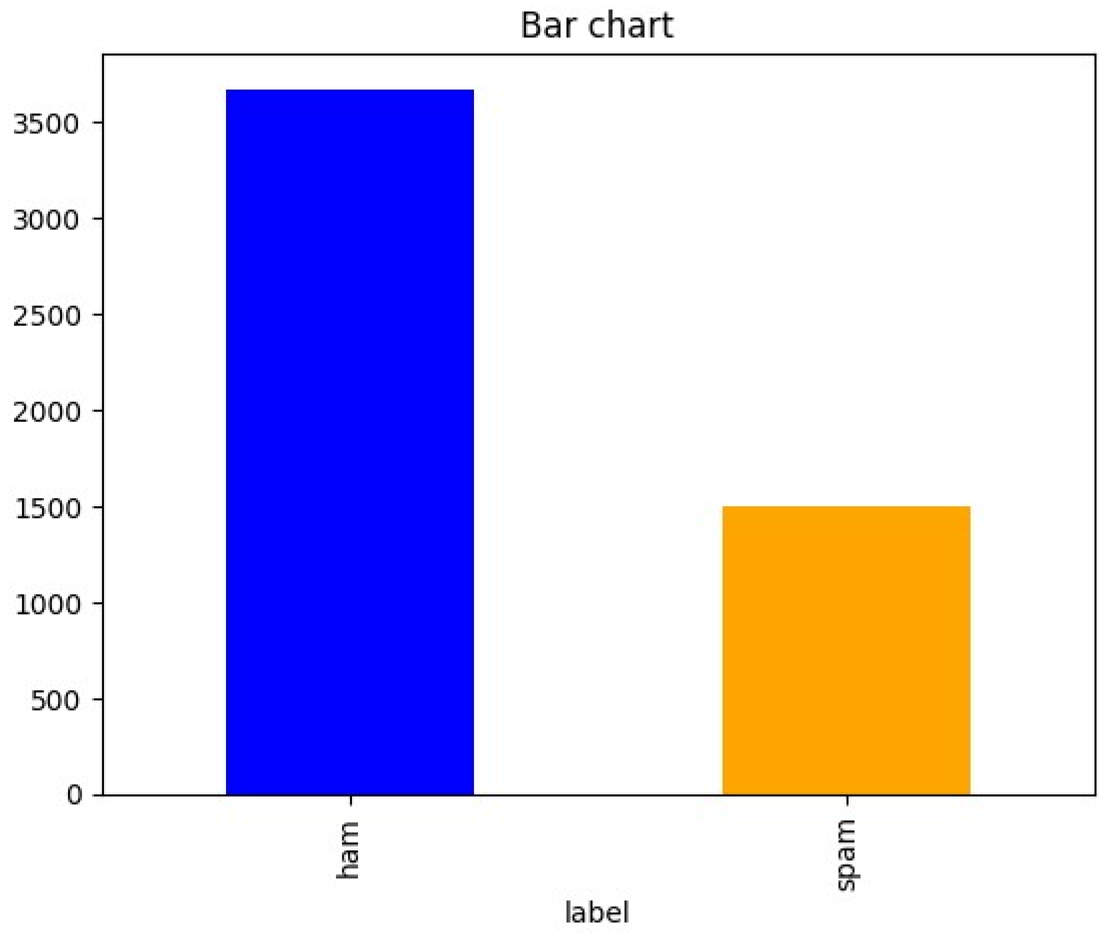
In [27]:

Out[27]:

```
label
ham     3672
spam    1499
Name: count, dtype: int64
```

In [28]:

```
count_Class=pd.value_counts(df['label'], sort= True)
count_Class.plot(kind= 'bar', color= ["blue", "orange"])
plt.title('Bar chart')
```



In [29]:

In [30]:

In [31]:

In [32]:

In [33]:

```
frequency_matrix = Vectorizer.transform(X_train.values).toarray()
```

Out[33]:

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

In [34]:

In [35]:
```
model =  MultinomialNB()
```

Out[35]:
```
▾ MultinomialNB
MultinomialNB()
```

In [36]:

In [37]:

```
0.9835748792270531
```

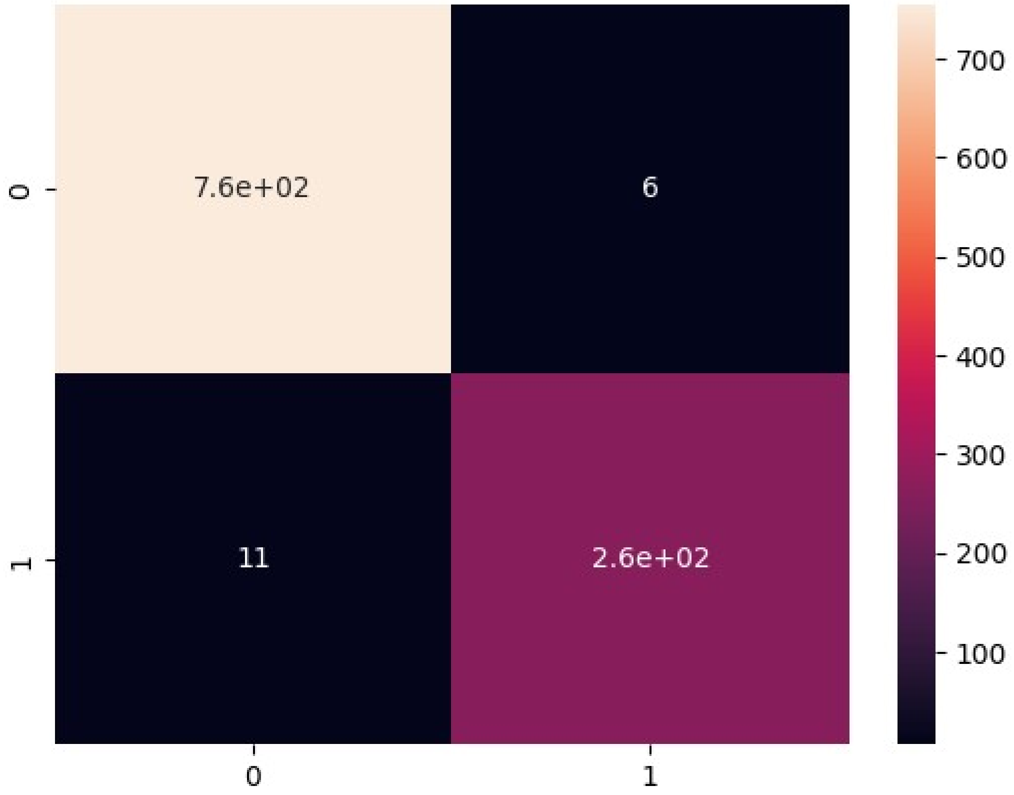
In [38]:

```
              precision    recall  f1-score   support

         ham       0.99      0.99      0.99       761
        spam       0.98      0.96      0.97       274

    accuracy                           0.98      1035
   macro avg       0.98      0.98      0.98      1035
weighted avg       0.98      0.98      0.98      1035
```

In [39]:
```
import seaborn as sns
print(confusion_matrix(y_test, y_pred))
```

```
[[755   6]
 [ 11 263]]
```

Out[39]:
```
<Axes: >
```



7.6e+02
6
11
2.6e+02
0
1
0
1
700
600
500
400
300
200
100