In [1]:

```
import pandas as pd
df=pd.read_csv('spam_ham_dataset.csv')
df.head()
df.describe()
```

Out[1]:

	Unnamed: 0	label_num
count	5171.000000	5171.000000
mean	2585.000000	0.289886
std	1492.883452	0.453753
min	0.000000	0.000000
25%	1292.500000	0.000000
50%	2585.000000	0.000000
75%	3877.500000	1.000000
max	5170.000000	1.000000

In [2]:

```
df.head()
```

Out[2]:

	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 [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5171 entries, 0 to 5170
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  5171 non-null   int64 
 1   label       5171 non-null   object
 2   text        5171 non-null   object
 3   label_num   5171 non-null   int64 
dtypes: int64(2), object(2)
memory usage: 161.7+ KB
```

In [5]:

```
df.isnull().sum().sum()
```

Out[5]:

```
0
```

In [6]:

```
df.columns
```

Out[6]:

```
Index(['Unnamed: 0', 'label', 'text', 'label_num'], dtype='object')
```

In [37]:

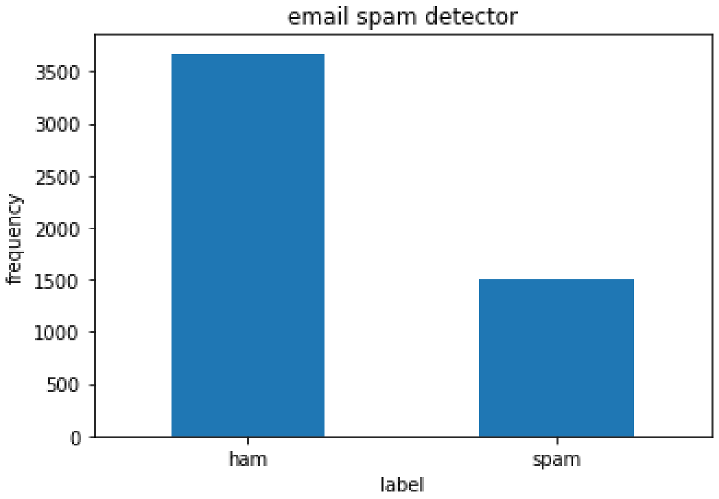
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [9]:

```
Labels=["spam","not spam"]
count_class=pd.value_counts(df['label'],sort=True)
count_class.plot(kind='bar',rot=0)
plt.title("email spam detector")
plt.xlabel("label")
plt.ylabel("frequency")
```

Out[9]:

```
Text(0, 0.5, 'frequency')
```



In [10]:

```
df.label_num.value_counts()
```

Out[10]:

```
0    3672
1    1499
Name: label_num, dtype: int64
```

In [11]:

```
df.label.value_counts()
```

Out[11]:

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

In [12]:

```
spam=df[df['label_num']==1]
not_spam=df[df['label_num']==0]
```

In [13]:

```
spam.shape
```

Out[13]:

```
(1499, 4)
```

In [14]:

```
not_spam.shape
```

Out[14]:

```
(3672, 4)
```

In [19]:

```
df1=df.drop(['label'],axis='columns')
```

In [20]:

```
df1
```

Out[20]:

	Unnamed: 0	text	label_num
0	605	Subject: enron methanol ; meter # : 988291\r\n...	0
1	2349	Subject: hpl nom for january 9 , 2001\r\n(see...	0
2	3624	Subject: neon retreat\r\nho ho ho , we ' re ar...	0
3	4685	Subject: photoshop , windows , office . cheap ...	1
4	2030	Subject: re : indian springs\r\nthis deal is t...	0
...	...	...	...
5166	1518	Subject: put the 10 on the ft\r\nthe transport...	0
5167	404	Subject: 3 / 4 / 2000 and following noms\r\nhp...	0
5168	2933	Subject: calpine daily gas nomination\r\n>\r\n...	0
5169	1409	Subject: industrial worksheets for august 2000...	0
5170	4807	Subject: important online banking alert\r\ndea...	1

5171 rows × 3 columns

In [68]:

```
inputs=df1.drop(['label_num','Unnamed: 0'],axis='columns')
inputs.head()
```

Out[68]:

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

In [65]:

```
target=df1.label_num
#temp=df.text
#temp
```

Out[65]:

```
0       Subject: enron methanol ; meter # : 988291\r\n...
1       Subject: hpl nom for january 9 , 2001\r\n( see...
2       Subject: neon retreat\r\nho ho ho , we ' re ar...
3       Subject: photoshop , windows , office . cheap ...
4       Subject: re : indian springs\r\nthis deal is t...
```

```
                                   ...
5166    Subject: put the 10 on the ft\r\nthe transport...
5167    Subject: 3 / 4 / 2000 and following noms\r\nhp...
5168    Subject: calpine daily gas nomination\r\n>\r\n...
5169    Subject: industrial worksheets for august 2000...
5170    Subject: important online banking alert\r\ndea...
Name: text, Length: 5171, dtype: object
```

In [24]:

```
inputs.columns[inputs.isna().any()]
```

Out[24]:

```
Index([], dtype='object')
```

In [89]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(df.text,df.label_num,test_size=0.25)
```

In [90]:

```
from sklearn.feature_extraction.text import CountVectorizer
```

In [91]:

```
v=CountVectorizer()
x_train_count=v.fit_transform(x_train.values)
x_train_count.toarray()[:2]
#x_train_count.head()
```

Out[91]:

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 1, 0, ..., 0, 0, 0]], dtype=int64)
```

In [92]:

```
from sklearn.naive_bayes import MultinomialNB
model=MultinomialNB()
model.fit(x_train_count,y_train)
```

Out[92]:

```
MultinomialNB()
```

In [93]:

```
x_test_count=v.transform(x_test)
model.score(x_test_count,y_test)
```

Out[93]:

```
0.974477958236659
```

In [ ]:

```

```