In [2]:

```
import pandas as pd
d = pd.read_csv('https://raw.githubusercontent.com/mohitgupta-omg/Kaggle-SMS-Spam-Col
```

In [3]:

```
d.head()
```

Out[3]:

	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

In [7]:

```
d.drop(["Unnamed: 2","Unnamed: 3","Unnamed: 4"],axis=1,inplace=True)
d.columns = ['labels','text']
d.head()
```

Out[7]:

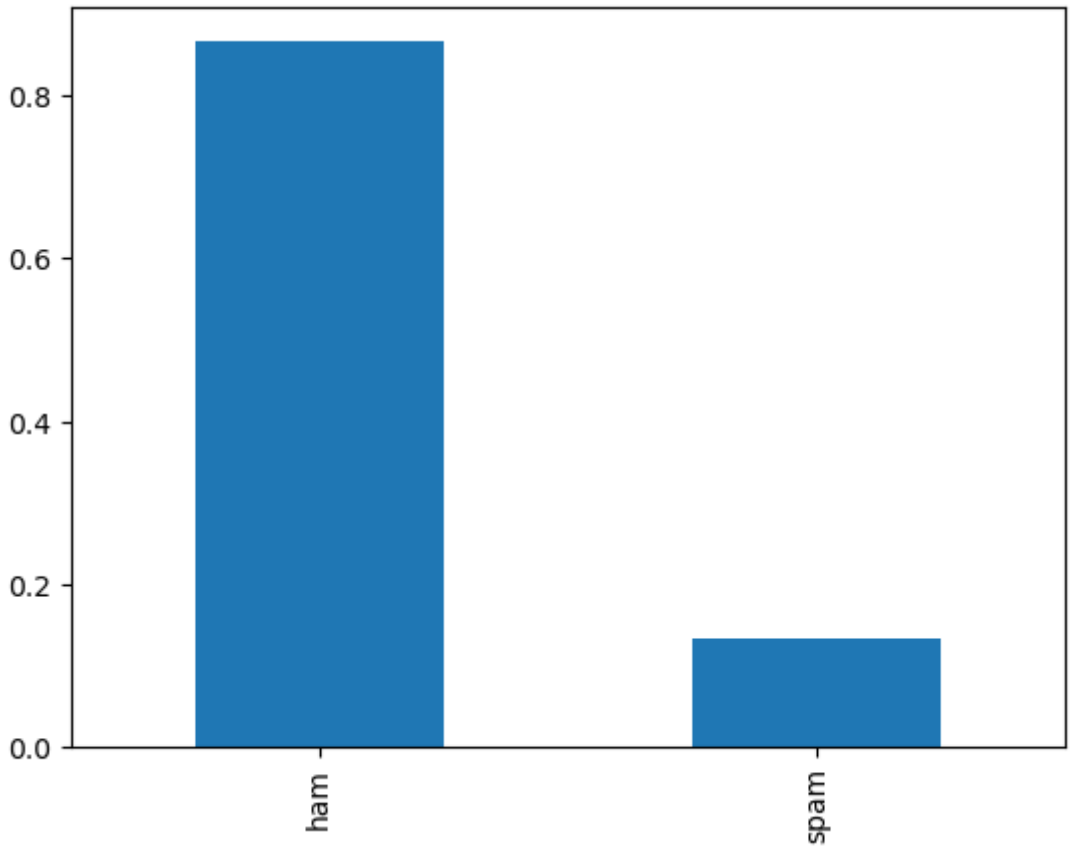
	labels	text
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

In [12]:

```
d['labels'].value_counts(normalize=True).plot.bar()
```

Out[12]:

```
<Axes: >
```



In [18]:

```
import nltk
nltk.download("all")
```

```
[nltk_data]    | Downloading package wordnet2021 to
[nltk_data]    |     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]    | Downloading package wordnet2022 to
[nltk_data]    |     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\wordnet2022.zip.
[nltk_data]    | Downloading package wordnet31 to
[nltk_data]    |     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]    | Downloading package wordnet_ic to
[nltk_data]    |     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\wordnet_ic.zip.
[nltk_data]    | Downloading package words to
[nltk_data]    |     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\words.zip.
[nltk_data]    | Downloading package ycoe to
[nltk_data]    |     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]    |   Unzipping corpora\ycoe.zip.
[nltk_data]    |
[nltk_data]  Done downloading collection all
```

Out[18]:

In [22]:

```
import re
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer

le = WordNetLemmatizer()
corp = []

t = list(d["text"])

for i in range(len(t)):
    r = re.sub('[^a-zA-Z]',' ',t[i])
    r=r.lower()
    r = r.split()
    r = [word for word in r if word not in stopwords.words("english")]
    r = [le.lemmatize(word) for word in r]
    r = ' '.join(r)
    corp.append(r)

d["text"] = corp
d.tail()
```

Out[22]:

	labels	text
5567	spam	nd time tried contact u u pound prize claim ea...
5568	ham	b going esplanade fr home
5569	ham	pity mood suggestion
5570	ham	guy bitching acted like interested buying some...
5571	ham	rofl true name

In [25]:

```
from sklearn.model_selection import train_test_split

x = d['text']
y = d['labels']

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.33,random_state=123)

print("Training Data: ",x_train.shape)
print("Testing Data: ",x_test.shape)
```

```
Training Data:  (3733,)
Testing Data:  (1839,)
```

In [29]:

```
from sklearn.feature_extraction.text import CountVectorizer
c = CountVectorizer()
x_c = c.fit_transform(x_train)
x_c.shape
```

Out[29]:

```
(3733, 5685)
```

In [32]:

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(x_c,y_train)
x_te_c = c.transform(x_test)
predictions = lr.predict(x_te_c)
predictions
```

Out[32]:

```
array(['ham', 'spam', 'ham', ..., 'ham', 'ham', 'spam'], dtype=object)
```

In [33]:

```
lr.score(x_te_c,y_test)
```

Out[33]:

```
0.9820554649265906
```

In [34]:

```
from sklearn import metrics
df = pd.DataFrame(metrics.confusion_matrix(y_test,predictions), index=['ham','spam'],

print(df)
```

```
       ham  spam
ham   1600     2
spam    31   206
```

In [ ]:

```

```