NLP: SMS Spam Classifier

In this task, you will work with a dataset of text messages that are labelled as Spam or Ham based on their content. Your task will be to use NLP techniques to clean and pre-process this dataset so that it becomes ready to train a Machine Learning model which can learn to classify texts as spam or not spam.

Note: Execute the below cell to import required basic packages

In [1]:
```
!mkdir .ans
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
import pandas as pd
pd.set_option('display.max_colwidth', 100)
```

```
mkdir: cannot create directory ‘.ans’: File exists

[nltk_data] Downloading package punkt to /home/tarun/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /home/tarun/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /home/tarun/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /home/tarun/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!

Defaulting to user installation because normal site-packages is not writeable
Processing ./smsspam-0.1-py3-none-any.whl
smsspam is already installed with the same version as the provided wheel. Use
--force-reinstall to force an installation of the wheel.
DEPRECATION: distro-info 0.23ubuntu1 has a non-standard version number. pip 2
4.1 will enforce this behaviour change. A possible replacement is to upgrade
to a newer version of distro-info or contact the author to suggest that they
release a version with a conforming version number. Discussion can be found a
t https://github.com/pypa/pip/issues/12063 (https://github.com/pypa/pip/issue
s/12063)
DEPRECATION: python-debian 0.1.36ubuntu1 has a non-standard version number. p
ip 24.1 will enforce this behaviour change. A possible replacement is to upgr
ade to a newer version of python-debian or contact the author to suggest that
they release a version with a conforming version number. Discussion can be fo
und at https://github.com/pypa/pip/issues/12063 (https://github.com/pypa/pip/
issues/12063)

[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python3 -m pip install --upgrade pip
```

In [2]:
```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import LinearSVC

from sklearn.metrics import accuracy_score, classification_report
from collections import Counter
import warnings

warnings.filterwarnings('ignore')
```

Task 1: Data Loading

- Using Pandas, read the data `SMSSpamCollection.tsv` and store it in the variable `dataset` . Use **sep** argument equal to *"\t"* and **header** equal to *None*.
- Rename the first column to **label** and the second column to **content**.
- Explore the dataset and remove any missing values if present. Assign a copy of the final dataframe to the variable `dataset_q1` for testing.

In [3]:
```
dataset = pd.read_csv("SMSSpamCollection.tsv", sep="\t", header=None)
dataset.head()
```

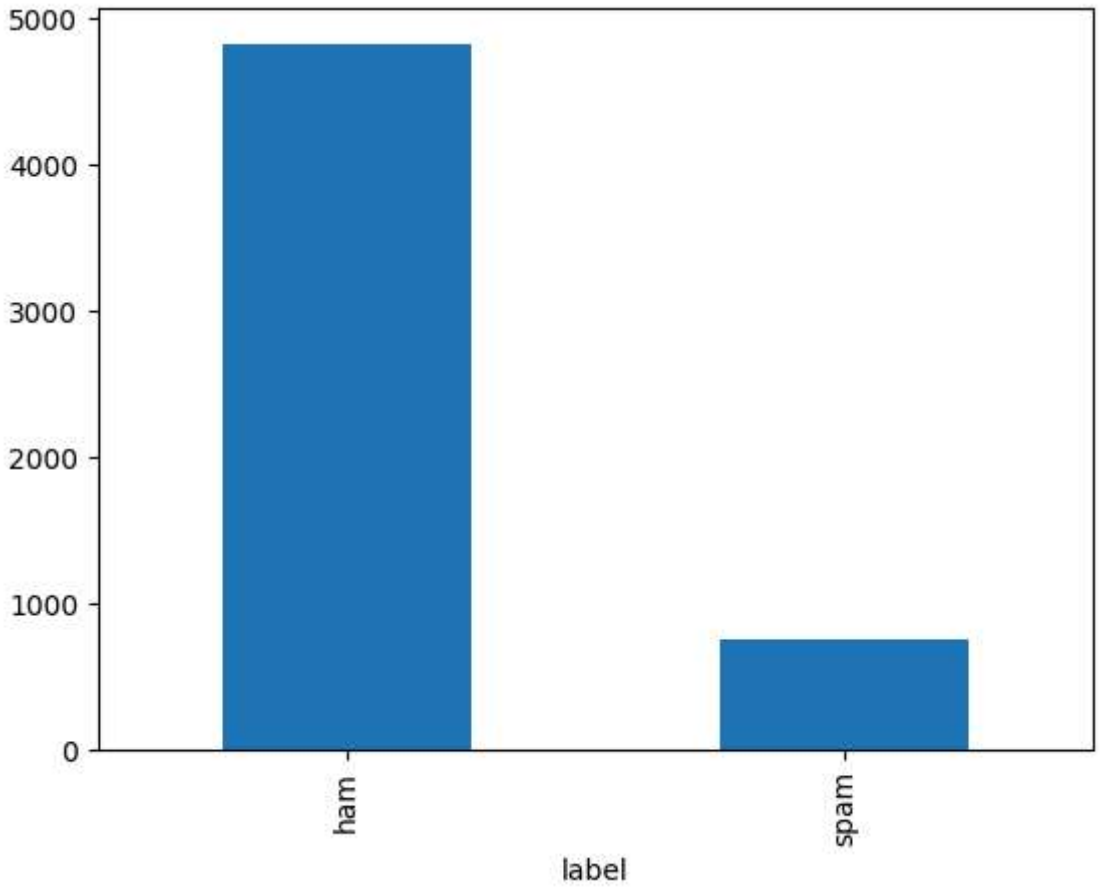
Out[3]:

	0	1
0	ham	I've been searching for the right words to thank you for this breather. I promise i wont take yo...
1	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive ...
2	ham	Nah I don't think he goes to usf, he lives around here though
3	ham	Even my brother is not like to speak with me. They treat me like aids patent.
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!

In [4]:
```
dataset.columns = ['label', 'content']
```

In [5]:
```
dataset['label'].value_counts().plot(kind='bar')
```

Out[5]: <Axes: xlabel='label'>



In [7]:
```
dataset_q1
```

Out[7]:

	label	content
0	ham	I've been searching for the right words to thank you for this breather. I promise i wont take yo...
1	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive ...
2	ham	Nah I don't think he goes to usf, he lives around here though
3	ham	Even my brother is not like to speak with me. They treat me like aids patent.
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!
...	...	...
5563	spam	This is the 2nd time we have tried 2 contact u. U have won the £750 Pound prize. 2 claim is easy...
5564	ham	Will ü b going to esplanade fr home?
5565	ham	Pity, * was in mood for that. So...any other suggestions?
5566	ham	The guy did some bitching but I acted like i'd be interested in buying something else next week ...
5567	ham	Rofl. Its true to its name

5568 rows × 2 columns

In [ ]:
```
```

Task 2: Removing Punctuation

- Import **string** package and view the punctuations present in *string.punctuation*.
- Complete the function `remove_punct` which takes in a line of text as input and returns the text after removing every character in text which is present in *string.punctuation*.
- Create a new column called **content_clean** and store the cleaned text returned from the function after removing the punctuations inside this column.
  - *Hint: Use lambda function*
- Assign a copy of the cleaned dataframe to the variable `dataset_q2` for testing.

Expected Output:

	label	content	content_clean
0	ham	I've been searching for the right words to thank you for this breather. I promise i wont take yo...	Ive been searching for the right words to thank you for this breather I promise i wont take your...
1	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive ...	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005 Text FA to 87121 to receive e...
2	ham	Nah I don't think he goes to usf, he lives around here though	Nah I dont think he goes to usf he lives around here though
3	ham	Even my brother is not like to speak with me. They treat me like aids patent.	Even my brother is not like to speak with me They treat me like aids patent
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL

In [9]:
```
import string
string.punctuation
```

Out[9]:
```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [10]:
```
def remove_punct(text):
    text_nopunct = "".join([char for char in text if char not in string.punctu
    return text_nopunct

dataset['content_clean'] = dataset['content'].apply(lambda x: remove_punct(x))

dataset.head()
```

Out[10]:

	label	content	content_clean
0	ham	I've been searching for the right words to thank you for this breather. I promise i wont take yo...	Ive been searching for the right words to thank you for this breather I promise i wont take your...
1	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive ...	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005 Text FA to 87121 to receive e...
2	ham	Nah I don't think he goes to usf, he lives around here though	Nah I dont think he goes to usf he lives around here though
3	ham	Even my brother is not like to speak with me. They treat me like aids patent.	Even my brother is not like to speak with me They treat me like aids patent
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL

In [ ]:
```
```

Task 3: Tokenization

- From **nltk.tokenize** import **word_tokenize** function.
- Apply the function on the **content_clean** column created in the previous task and store the result in a new column **content_tokenized**. Make sure to convert the text to lowercase before passing it to the word_tokenize function.
- Assign a copy of the final dataframe to the variable `dataset_q3` for testing.

Expected Output:

	label	content	content_clean	content_tokenized
0	ham	I've been searching for the right words to thank you for this breather. I promise i wont take yo...	Ive been searching for the right words to thank you for this breather I promise i wont take your...	[ive, been, searching, for, the, right, words, to, thank, you, for, this, breather, i, promise, ...
1	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive ...	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005 Text FA to 87121 to receive e...	[free, entry, in, 2, a, wkly, comp, to, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, to...
2	ham	Nah I don't think he goes to usf, he lives around here though	Nah I dont think he goes to usf he lives around here though	[nah, i, dont, think, he, goes, to, usf, he, lives, around, here, though]
3	ham	Even my brother is not like to speak with me. They treat me like aids patent.	Even my brother is not like to speak with me They treat me like aids patent	[even, my, brother, is, not, like, to, speak, with, me, they, treat, me, like, aids, patent]
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]

In [13]:
```
from nltk.tokenize import word_tokenize
```

In [14]:
```
dataset['content_tokenized'] = dataset['content_clean'].apply(lambda x: word_t
```

In [15]:
```
dataset.head()
```

Out[15]:

	label	content	content_clean	content_tokenized
0	ham	I've been searching for the right words to thank you for this breather. I promise i wont take yo...	Ive been searching for the right words to thank you for this breather I promise i wont take your...	[ive, been, searching, for, the, right, words, to, thank, you, for, this, breather, i, promise, ...
1	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive ...	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005 Text FA to 87121 to receive e...	[free, entry, in, 2, a, wkly, comp, to, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, to...
2	ham	Nah I don't think he goes to usf, he lives around here though	Nah I dont think he goes to usf he lives around here though	[nah, i, dont, think, he, goes, to, usf, he, lives, around, here, though]
3	ham	Even my brother is not like to speak with me. They treat me like aids patent.	Even my brother is not like to speak with me They treat me like aids patent	[even, my, brother, is, not, like, to, speak, with, me, they, treat, me, like, aids, patent]
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]

In [ ]:
```

```

Task 4: Removing Stopwords

- Fetch the common stopwords present in English from *nltk.corpus.stopwords.words* and pass *'english'* as the argument. View the result and understand the common English stopwords present.
- Next, complete the function `remove_stopwords` which takes in a tokenized list as input and returns a list after removing the words that are present in the stopwords.
- Apply the function on the **content_tokenized** column created in the previous task and store the result in a new column **content_nostop**.
- Assign a copy of the resultant dataframe to the variable `dataset_q4` for testing.

Expected Output:

label	content	content_clean	content_tokenized	content_nostop

In [18]:
```
import nltk

stopword = nltk.corpus.stopwords.words('english')
```

In [19]:
```
def remove_stopwords(tokenized_list):
    text = [word for word in tokenized_list if word not in stopword]
    return text

dataset['content_nostop'] = dataset['content_tokenized'].apply(lambda x: remov

dataset.head()
```

Out[19]:

	label	content	content_clean	content_tokenized	content_nostop
0	ham	I've been searching for the right words to thank you for this breather. I promise i wont take yo...	Ive been searching for the right words to thank you for this breather I promise i wont take your...	[ive, been, searching, for, the, right, words, to, thank, you, for, this, breather, i, promise, ...	[ive, searching, right, words, thank, breather, promise, wont, take, help, granted, fulfil, prom...
1	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive ...	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005 Text FA to 87121 to receive e...	[free, entry, in, 2, a, wkly, comp, to, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, to...	[free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receiv...
2	ham	Nah I don't think he goes to usf, he lives around here though	Nah I dont think he goes to usf he lives around here though	[nah, i, dont, think, he, goes, to, usf, he, lives, around, here, though]	[nah, dont, think, goes, usf, lives, around, though]
3	ham	Even my brother is not like to speak with me. They treat me like aids patent.	Even my brother is not like to speak with me They treat me like aids patent	[even, my, brother, is, not, like, to, speak, with, me, they, treat, me, like, aids, patent]	[even, brother, like, speak, treat, like, aids, patent]
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]	[date, sunday]

In [ ]:
```
```

Task 5: Stemming

- Extract the columns **label**, **content** and **content_nostop** from the previous task's dataframe and use it for this task.
- Use **Porter Stemmer** for performing the Stemming operation. Complete the function `stemming` which takes in a tokenized cleaned text list as input and returns the stemmed version of every word of the list as output. Return type is also a list.
- Apply the function on the **content_nostop** column and store the result in a new column called **content_stemmed**.
- Assign a copy of the resultant dataframe to the variable `dataset_q5` for testing.

Expected Output:

	label	content	content_nostop	content_stemmed
0	ham	I've been searching for the right words to thank you for this breather. I promise i wont take yo...	[ive, searching, right, words, thank, breather, promise, wont, take, help, granted, fulfil, prom...	[ive, search, right, word, thank, breather, promis, wont, take, help, grant, fulfil, promis, won...
1	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive ...	[free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receiv...	[free, entri, 2, wkli, comp, win, fa, cup, final, tkt, 21st, may, 2005, text, fa, 87121, receiv,...
2	ham	Nah I don't think he goes to usf, he lives around here though	[nah, dont, think, goes, usf, lives, around, though]	[nah, dont, think, goe, usf, live, around, though]
3	ham	Even my brother is not like to speak with me. They treat me like aids patent.	[even, brother, like, speak, treat, like, aids, patent]	[even, brother, like, speak, treat, like, aid, patent]
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	[date, sunday]	[date, sunday]

In [22]:
```
dataset = dataset.iloc[:,[0,1,-1]]
dataset.head()
```

Out[22]:

	label	content	content_nostop
0	ham	I've been searching for the right words to thank you for this breather. I promise i wont take yo...	[ive, searching, right, words, thank, breather, promise, wont, take, help, granted, fulfil, prom...
1	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive ...	[free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receiv...
2	ham	Nah I don't think he goes to usf, he lives around here though	[nah, dont, think, goes, usf, lives, around, though]
3	ham	Even my brother is not like to speak with me. They treat me like aids patent.	[even, brother, like, speak, treat, like, aids, patent]
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	[date, sunday]

In [23]:
```
ps = nltk.PorterStemmer()
```

In [24]:
```
def stemming(tokenized_text):
    text = [ps.stem(word) for word in tokenized_text]
    return text

dataset['content_stemmed'] = dataset['content_nostop'].apply(lambda x: stemmin

dataset.head()
```

Out[24]:

	label	content	content_nostop	content_stemmed
0	ham	I've been searching for the right words to thank you for this breather. I promise i wont take yo...	[ive, searching, right, words, thank, breather, promise, wont, take, help, granted, fulfil, prom...	[ive, search, right, word, thank, breather, promis, wont, take, help, grant, fulfil, promis, won...
1	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive ...	[free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receiv...	[free, entri, 2, wkli, comp, win, fa, cup, final, tkt, 21st, may, 2005, text, fa, 87121, receiv,...
2	ham	Nah I don't think he goes to usf, he lives around here though	[nah, dont, think, goes, usf, lives, around, though]	[nah, dont, think, goe, usf, live, around, though]
3	ham	Even my brother is not like to speak with me. They treat me like aids patent.	[even, brother, like, speak, treat, like, aids, patent]	[even, brother, like, speak, treat, like, aid, patent]
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	[date, sunday]	[date, sunday]

Task 6: Lemmatizing

- Use **Word Net Lemmatizer** for performing the Lemmatizing operation. Complete the function `lemmatizing` which takes in a tokenized cleaned text list as input and returns the lemmatized version of every word of the list as output. Return type is also a list.
- Apply the function on the **content_nostop** column and store the result in a new column called **content_lemmatized**.
- Assign a copy of the resultant dataframe to the variable `dataset_q6` for testing.

Expected Output:

	label	content	content_nostop	content_stemmed	content_lemmatized
0	ham	I've been searching for the right words to thank you for this breather. I promise i wont take yo...	[ive, searching, right, words, thank, breather, promise, wont, take, help, granted, fulfil, prom...	[ive, search, right, word, thank, breather, promis, wont, take, help, grant, fulfil, promis, won...	[ive, searching, right, word, thank, breather, promise, wont, take, help, granted, fulfil, promi...
1	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive ...	[free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receiv...	[free, entri, 2, wkli, comp, win, fa, cup, final, tkt, 21st, may, 2005, text, fa, 87121, receiv,...	[free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receiv...
2	ham	Nah I don't think he goes to usf, he lives around here though	[nah, dont, think, goes, usf, lives, around, though]	[nah, dont, think, goe, usf, live, around, though]	[nah, dont, think, go, usf, life, around, though]
3	ham	Even my brother is not like to speak with me. They treat me like aids patent.	[even, brother, like, speak, treat, like, aids, patent]	[even, brother, like, speak, treat, like, aid, patent]	[even, brother, like, speak, treat, like, aid, patent]
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	[date, sunday]	[date, sunday]	[date, sunday]

In [27]:
```
wn = nltk.WordNetLemmatizer()
```

In [28]:
```
def lemmatizing(tokenized_text):
    text = [wn.lemmatize(word) for word in tokenized_text]
    return text

dataset['content_lemmatized'] = dataset['content_nostop'].apply(lambda x: lemm
dataset.head()
```

Out[28]:

	label	content	content_nostop	content_stemmed	content_lemmatized
0	ham	I've been searching for the right words to thank you for this breather. I promise i wont take yo...	[ive, searching, right, words, thank, breather, promise, wont, take, help, granted, fulfil, prom...	[ive, search, right, word, thank, breather, promis, wont, take, help, grant, fulfil, promis, won...	[ive, searching, right, word, thank, breather, promise, wont, take, help, granted, fulfil, promi...
1	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive ...	[free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receiv...	[free, entri, 2, wkli, comp, win, fa, cup, final, tkt, 21st, may, 2005, text, fa, 87121, receiv,...	[free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receiv...
2	ham	Nah I don't think he goes to usf, he lives around here though	[nah, dont, think, goes, usf, lives, around, though]	[nah, dont, think, goe, usf, live, around, though]	[nah, dont, think, go, usf, life, around, though]
3	ham	Even my brother is not like to speak with me. They treat me like aids patent.	[even, brother, like, speak, treat, like, aids, patent]	[even, brother, like, speak, treat, like, aid, patent]	[even, brother, like, speak, treat, like, aid, patent]
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	[date, sunday]	[date, sunday]	[date, sunday]

Task 7: N-gram Vectorizing

- Create a new column called **cleaned content** using the **content_lemmatized** column created in the previous task. The column should join the lemmatized words back together to form a text.
- Create a sample of the dataset containing only the first **20** rows. Store it in the variable `data_sample`.
- Create a **CountVectorizer** object having the parameter *ngram_range* equal to *(2,2)*, indicating a **Bigram**. Assign the object to the variable `bigram_vect`.
- Fit and transform the `bigram_vect` object on the **cleaned_content** column created earlier and store the result in the variable `result_vect`.
- Assign the feature names created using the Count Vectorizer object to the variable `feature_names` as a list and a copy of the dataset used for this task to the variable `dataset_q7` for testing.

- *Hint: Use either get_feature_names() or get_feature_names_out() method of the Count*

In [31]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

In [32]:
```
dataset['cleaned_content'] = dataset['content_lemmatized'].apply(lambda x: " "
```

In [33]:
```
dataset.head()
```

Out[33]:

	label	content	content_nostop	content_stemmed	content_lemmatized	cleaned_content
0	ham	I've been searching for the right words to thank you for this breather. I promise i wont take yo...	[ive, searching, right, words, thank, breather, promise, wont, take, help, granted, fulfil, prom...	[ive, search, right, word, thank, breather, promis, wont, take, help, grant, fulfil, promis, won...	[ive, searching, right, word, thank, breather, promise, wont, take, help, granted, fulfil, promi...	ive searching right word thank breather promise wont take help granted fulfil promise wonderful ...
1	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive ...	[free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receiv...	[free, entri, 2, wkli, comp, win, fa, cup, final, tkt, 21st, may, 2005, text, fa, 87121, receiv,...	[free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receiv...	free entry 2 wkly comp win fa cup final tkts 21st may 2005 text fa 87121 receive entry questions...
2	ham	Nah I don't think he goes to usf, he lives around here though	[nah, dont, think, goes, usf, lives, around, though]	[nah, dont, think, goe, usf, live, around, though]	[nah, dont, think, go, usf, life, around, though]	nah dont think go usf life around though
3	ham	Even my brother is not like to speak with me. They treat me like aids patent.	[even, brother, like, speak, treat, like, aids, patent]	[even, brother, like, speak, treat, like, aid, patent]	[even, brother, like, speak, treat, like, aid, patent]	even brother like speak treat like aid patent
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	[date, sunday]	[date, sunday]	[date, sunday]	date sunday

In [34]:
```
data_sample = dataset[0:20]

bigram_vect = CountVectorizer(ngram_range=(2,2))
result_vect = bigram_vect.fit_transform(data_sample['cleaned_content'])
print(result_vect.shape)
print(bigram_vect.get_feature_names_out())
feature_names = list(bigram_vect.get_feature_names_out())
```

```
(20, 211)
['09061701461 claim' '100 20000' '100000 prize' '11 month' '12 hour'
 '150pday 6days' '16 tsandcs' '20000 pound' '2005 text' '21st may'
 '4txtú120 poboxox36504w45wq' '6days 16' '81010 tc' '87077 eg'
 '87077 trywales' '87121 receive' '87575 cost' '900 prize' 'aft finish'
 'aid patent' 'anymore tonight' 'apply 08452810075over18s' 'apply reply'
 'ard smth' 'around though' 'blessing time' 'breather promise'
 'brother like' 'call 09061701461' 'call mobile' 'caller press'
 'callertune caller' 'camera free' 'cash 100' 'chance win' 'claim 81010'
 'claim call' 'claim code' 'click httpwap' 'click wap' 'co free'
 'code kl341' 'colour mobile' 'comp win' 'copy friend' 'cost 150pday'
 'credit click' 'cried enough' 'csh11 send' 'cup final'
 'customer selected' 'da stock' 'date sunday' 'dont miss' 'dont think'
 'dont want' 'eg england' 'eh remember' 'england 87077'
 'england macedonia' 'enough today' 'entitled update' 'entry questionstd'
 'entry wkly' 'even brother' 'fa 87121' 'fa cup' 'feel that' 'final tkts'
 'fine that' 'finish lunch' 'finish ur' 'first lar' 'free 08002986030'
 'free call' 'free entry' 'free membership' 'friend callertune'
 'fulfil promise' 'go str' 'go usf' 'goalsteam news' 'going try' 'gon na'
 'granted fulfil' 'ha ha' 'ha joking' 'help granted' 'hl info' 'home soon'
 'httpwap xxxmobilemovieclubcomnqjkgighjjgcbl' 'im gon' 'ive cried'
 'ive searching' 'jackpot txt' 'kim watching' 'kl341 valid' 'lar da'
 'latest colour' 'lccltd pobox' 'life around' 'like aid' 'like speak'
 'link next' 'lor ard' 'lor finish' 'lunch already' 'lunch go'
 'macedonia dont' 'make wet' 'may 2005' 'melle melle' 'melle oru'
 'membership 100000' 'message click' 'minnaminunginte nurungu'
 'miss goalsteam' 'mobile 11' 'mobile camera' 'mobile update'
 'month entitled' 'month ha' 'na home' 'nah dont' 'name yes'
 'national team' 'naughty make' 'network customer' 'news txt' 'next txt'
 'nurungu vettam' 'oh kim' 'oru minnaminunginte' 'pay first' 'per request'
 'pobox 4403ldnw1a7rw18' 'poboxox36504w45wq 16' 'pound txt' 'press copy'
 'prize jackpot' 'prize reward' 'promise wonderful' 'promise wont'
 'questionstd txt' 'ratetcs apply' 'receive entry' 'receivea 900'
 'remember spell' 'reply hl' 'request melle' 'reward claim' 'right word'
 'scotland 4txtú120' 'searching right' 'selected receivea' 'send 87575'
 'seriously spell' 'set callertune' 'six chance' 'smth lor' 'soon dont'
 'speak treat' 'spell name' 'stock comin' 'str lor' 'stuff anymore'
 'take help' 'talk stuff' 'tc wwwdbuknet' 'team 87077' 'text fa'
 'thank breather' 'that way' 'think go' 'tkts 21st' 'tonight ive'
 'treat like' 'try month' 'trywales scotland' 'tsandcs apply' 'txt csh11'
 'txt message' 'txt ratetcs' 'txt ur' 'txt word' 'update co'
 'update latest' 'ur lunch' 'ur national' 'urgent week' 'use credit'
 'usf life' 'valid 12' 'valued network' 'vettam set' 'want talk'
 'wap link' 'way feel' 'way gota' 'week free' 'win cash' 'win fa'
 'winner valued' 'wkly comp' 'wonderful blessing' 'wont take' 'word claim'
 'word thank' 'wwwdbuknet lccltd' 'xxxmobilemovieclub use' 'yes naughty']
```

In [35]:
```
feature_names
```

Out[35]:
```
['09061701461 claim',
 '100 20000',
 '100000 prize',
 '11 month',
 '12 hour',
 '150pday 6days',
 '16 tsandcs',
 '20000 pound',
 '2005 text',
 '21st may',
 '4txtú120 poboxox36504w45wq',
 '6days 16',
 '81010 tc',
 '87077 eg',
 '87077 trywales',
 '87121 receive',
 '87575 cost',
 '900 prize',
 'aft finish',
```

Task 8: Creating Vectorized DataFrame

- Convert the `result_vect` object created in the previous task to an array using the toarray() method and further convert the array to a dataframe. Assign the dataframe to the variable `result_vect_df` .
- Assign the column names of the dataframe using the `feature_names` variable created in the previous task.
- View the vectorized dataframe and check its shape.

In [38]:
```
result_vect_df = pd.DataFrame(result_vect.toarray())
result_vect_df.columns = bigram_vect.get_feature_names_out()
result_vect_df
```

Out[38]:

	09061701461 claim	100 20000	100000 prize	11 month	12 hour	150pday 6days	16 tsandcs	20000 pound	2005 text	21st may	...	win fa	w v
0	0	0	0	0	0	0	0	0	0	0	...	0	
1	0	0	0	0	0	0	0	0	1	1	...	1	
2	0	0	0	0	0	0	0	0	0	0	...	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	
5	0	0	0	0	0	0	0	0	0	0	...	0	
6	1	0	0	0	1	0	0	0	0	0	...	0	
7	0	0	0	1	0	0	0	0	0	0	...	0	
8	0	0	0	0	0	0	0	0	0	0	...	0	
9	0	1	0	0	0	1	1	1	0	0	...	0	
10	0	0	1	0	0	0	0	0	0	0	...	0	
11	0	0	0	0	0	0	0	0	0	0	...	0	
12	0	0	0	0	0	0	0	0	0	0	...	0	
13	0	0	0	0	0	0	0	0	0	0	...	0	
14	0	0	0	0	0	0	0	0	0	0	...	0	
15	0	0	0	0	0	0	0	0	0	0	...	0	
16	0	0	0	0	0	0	0	0	0	0	...	0	
17	0	0	0	0	0	0	0	0	0	0	...	0	
18	0	0	0	0	0	0	0	0	0	0	...	0	
19	0	0	0	0	0	0	0	0	0	0	...	0	

20 rows × 211 columns

Task 9 : Text Classification Model Building

- Prepare the data and labels; store them in `X` & `y` respectively.
- Create an instance of TF-IDF Vectorization with max features set to 5000 in variable `tfidf_vectorizer`.
- Fit and transform the data extracted and store it in `X_tfidf`.
- Split the dataset into training and testing named X_train, X_test, y_train, y_test with the newly transformed data and the labels with a test size of 20% and random state set to 42.
- Initialize SVM classifier with seed value of `42` stored in variable `svm`.

- Fit the training data to the classifier and gather the predictions against the testing data and store it in `y_pred` .

In [40]:
```
# Prepare data and labels
X = dataset['cleaned_content']

y = dataset['label']

# TF-IDF Vectorization
tfidf_vectorizer = TfidfVectorizer(max_features=5000)
X_tfidf = tfidf_vectorizer.fit_transform(X)

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X_tfidf, y, test_size=0.2,

# Initialize SVM classifier
svm = LinearSVC(random_state=42)

# Train the classifier
svm.fit(X_train, y_train)
```

Out[40]:
```
LinearSVC(random_state=42)
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

Task 10 : Model Evaluation

- Fit the training data to the classifier and gather the predictions against the testing data and store it in `y_pred` .
- Evaluate the score for predictions against the testing data and store the output in `accuracy` .
- Get the classification report and of the predictions and the testing data as a dictionary and store it in `report` .

In [41]:
```
# Predictions
y_pred = svm.predict(X_test)

# Evaluation
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")

# Classification report
report = classification_report(y_test, y_pred, output_dict=True)
print(report)
```

```
Accuracy: 0.9775583482944344
{'ham': {'precision': 0.9786150712830958, 'recall': 0.9958549222797928, 'f1-s
core': 0.9871597329224449, 'support': 965.0}, 'spam': {'precision': 0.9696969
696969697, 'recall': 0.8590604026845637, 'f1-score': 0.9110320284697508, 'sup
port': 149.0}, 'accuracy': 0.9775583482944344, 'macro avg': {'precision': 0.9
741560204900328, 'recall': 0.9274576624821782, 'f1-score': 0.949095880696097
8, 'support': 1114.0}, 'weighted avg': {'precision': 0.9774222551822584, 'rec
all': 0.9775583482944344, 'f1-score': 0.9769774816087542, 'support': 1114.0}}
```

In [ ]:

Optional: Spam Classification using Deep Learning

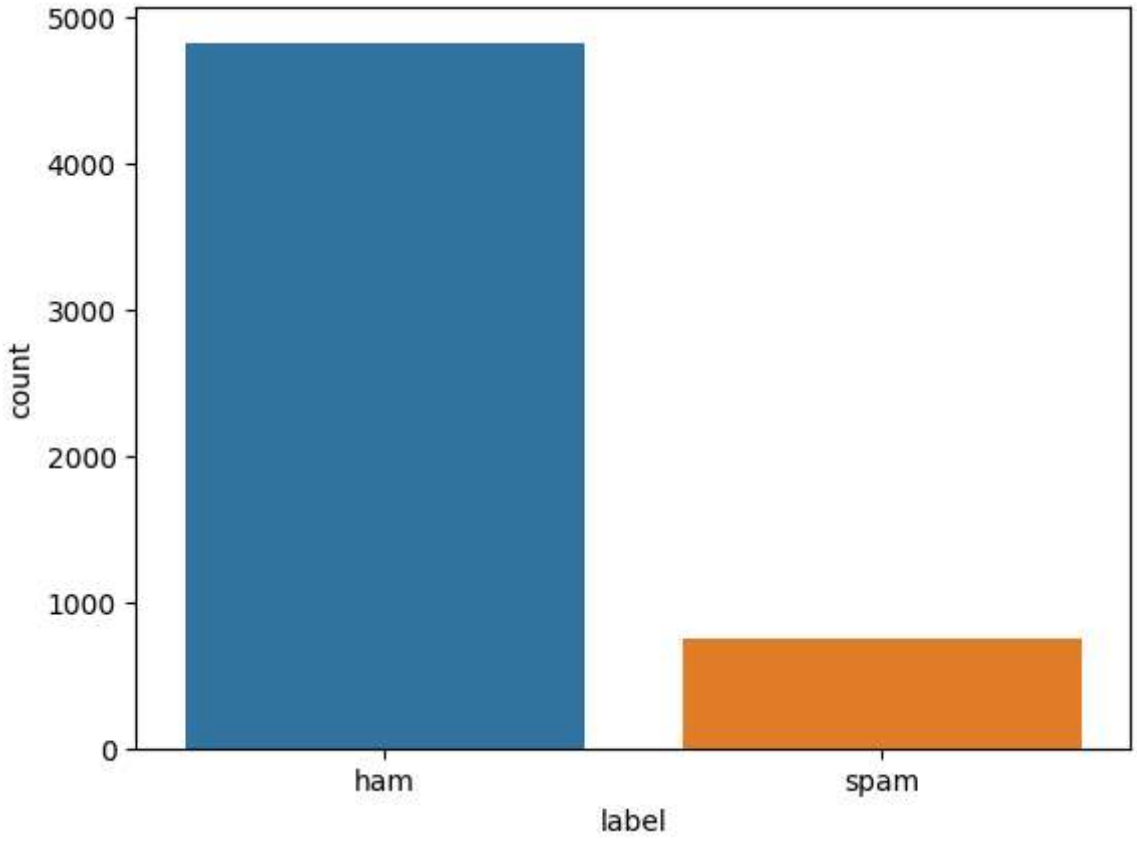
In [42]:
```
# Importing necessary libraries for EDA
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import string
import nltk
from nltk.corpus import stopwords
from wordcloud import WordCloud
nltk.download('stopwords')

# Importing libraries necessary for Model Building and Training
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.model_selection import train_test_split
from keras.callbacks import EarlyStopping, ReduceLROnPlateau

import warnings
warnings.filterwarnings('ignore')
```

```
[nltk_data] Downloading package stopwords to /home/tarun/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
2024-09-27 13:25:04.278888: I tensorflow/core/platform/cpu_feature_guard.cc:1
82] This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2, in other operations, rebuild Tens
orFlow with the appropriate compiler flags.
```

In [43]:
```
dataset.head()
data = dataset[['label','cleaned_content']]
```

Visualizing the Class Imbalance

In [44]:
```
sns.countplot(x='label', data=dataset)
plt.show()
```



We can clearly see that number of samples of Ham is much more than that of Spam which implies that the dataset we are using is imbalanced.

Downsampling to Balance the Dataset

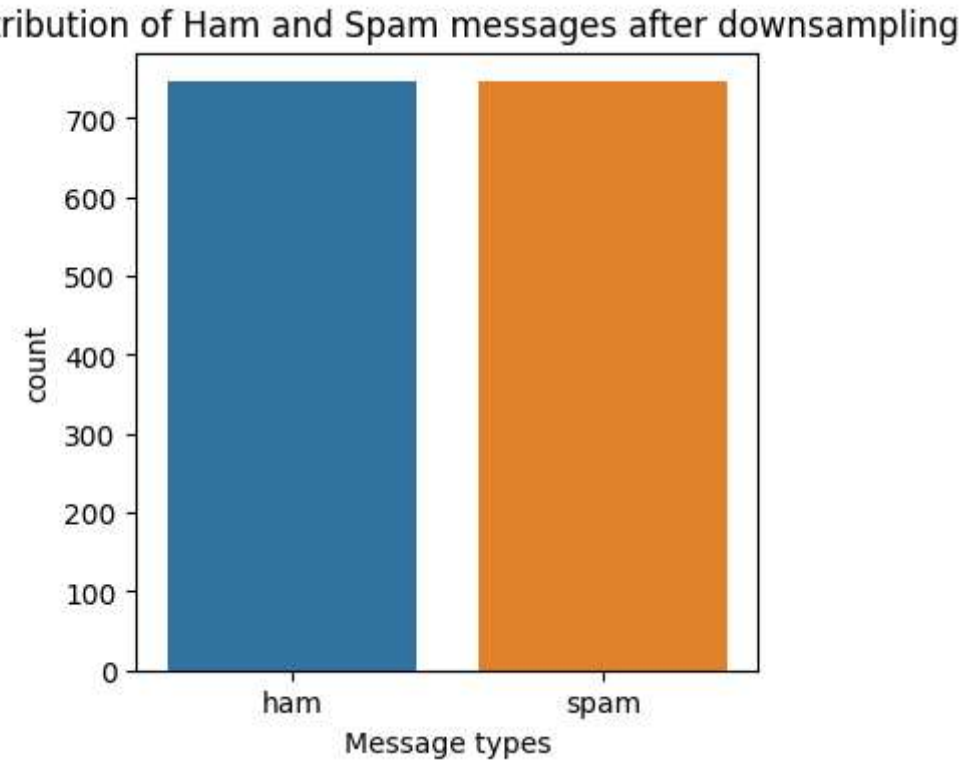
In [45]:
```
# Downsampling to balance the dataset
ham_msg = data[data.label == 'ham']
spam_msg = data[data.label == 'spam']
ham_msg = ham_msg.sample(n=len(spam_msg),
                         random_state=42)
```

In [46]:
```
balanced_data = pd.concat([ham_msg,spam_msg])
```

Visualizing the Balanced Dataset

In [47]:
```
# Plotting the counts of down sampled dataset

plt.figure(figsize=(4, 4))
sns.countplot(data = balanced_data, x='label')
plt.title('Distribution of Ham and Spam messages after downsampling')
plt.xlabel('Message types')
```

Out[47]:
```
Text(0.5, 0, 'Message types')
```



In [48]:
```
balanced_data['label']=balanced_data['label'].map({'ham':0,'spam':1})
```

In [49]:
```
X = balanced_data['cleaned_content']
y = balanced_data['label']

#train test split
train_msg, test_msg, train_labels, test_labels =train_test_split(X,y,test_size
```

Text Tokenization and Padding

- `Tokenizer` : Creates a tokenizer to convert text into sequences of integers. vocab_size limits the vocabulary to 500 words, and 'OOV' handles out-of-vocabulary words.
- `fit_on_texts` : Fits the tokenizer on the training data.
- `word_index` : Dictionary mapping words to their integer tokens.
- `texts_to_sequences` : Converts messages to sequences of integers.

- `pad_sequences` : Pads sequences so that all of them have the same length (max_len=50), which is needed for the model

In [50]:
```
vocab_size=500
oov_tok='<OOV>'
max_len=50
#preprocessing making tokens out of text
token=Tokenizer(num_words=vocab_size,oov_token=oov_tok)
token.fit_on_texts(train_msg)
padding_type='post'
truncate_type='post'
Trainning_seq=token.texts_to_sequences(train_msg)
Trainning_pad=pad_sequences(Trainning_seq,maxlen=50,padding=padding_type,trunc
Testing_seq=token.texts_to_sequences(test_msg)
Testing_pad=pad_sequences(Testing_seq,maxlen=50,padding=padding_type,truncatin
```

Model Building

- A simple deep learning model using:
  - Embedding layer: Converts integer tokens into dense vector embeddings of size 16.
  - GlobalAveragePooling1D: Averages the embeddings across the sequence.
  - Dense: A fully connected layer with 32 neurons and ReLU activation.
  - Dropout: Regularization technique to prevent overfitting (30% dropout).
  - Final Dense: Single output neuron with sigmoid activation to predict the probability of spam (binary classification).

In [51]:
```
#model
model=tf.keras.models.Sequential([tf.keras.layers.Embedding(vocab_size,16,inpu
                                  tf.keras.layers.GlobalAveragePooling1D(),
                                  tf.keras.layers.Dense(32,activation='relu'),
                                  tf.keras.layers.Dropout(0.3),
                                  tf.keras.layers.Dense(1,activation='sigmoid'
```

In [52]:
```
model.compile(loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),metric
```

In [53]:
```
epoch=30
early_stop = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=3)
history=model.fit(Trainning_pad, train_labels ,validation_data=(Testing_pad, t
```

```
Epoch 1/30
38/38 - 2s - loss: 0.6909 - accuracy: 0.4962 - val_loss: 0.6873 - val_accurac
y: 0.5318 - 2s/epoch - 49ms/step
Epoch 2/30
38/38 - 0s - loss: 0.6830 - accuracy: 0.6270 - val_loss: 0.6752 - val_accurac
y: 0.7090 - 179ms/epoch - 5ms/step
Epoch 3/30
38/38 - 0s - loss: 0.6637 - accuracy: 0.7519 - val_loss: 0.6453 - val_accurac
y: 0.8094 - 306ms/epoch - 8ms/step
Epoch 4/30
38/38 - 0s - loss: 0.6210 - accuracy: 0.8181 - val_loss: 0.5842 - val_accurac
y: 0.8796 - 150ms/epoch - 4ms/step
Epoch 5/30
38/38 - 0s - loss: 0.5489 - accuracy: 0.8617 - val_loss: 0.5005 - val_accurac
y: 0.8863 - 147ms/epoch - 4ms/step
Epoch 6/30
38/38 - 0s - loss: 0.4588 - accuracy: 0.8810 - val_loss: 0.4148 - val_accurac
y: 0.8829 - 277ms/epoch - 7ms/step
Epoch 7/30
38/38 - 0s - loss: 0.3757 - accuracy: 0.9003 - val_loss: 0.3512 - val_accurac
y: 0.8829 - 140ms/epoch - 4ms/step
Epoch 8/30
38/38 - 0s - loss: 0.3121 - accuracy: 0.9195 - val_loss: 0.3027 - val_accurac
y: 0.8863 - 273ms/epoch - 7ms/step
Epoch 9/30
38/38 - 0s - loss: 0.2663 - accuracy: 0.9254 - val_loss: 0.2736 - val_accurac
y: 0.8930 - 159ms/epoch - 4ms/step
Epoch 10/30
38/38 - 0s - loss: 0.2346 - accuracy: 0.9212 - val_loss: 0.2417 - val_accurac
y: 0.8963 - 177ms/epoch - 5ms/step
Epoch 11/30
38/38 - 0s - loss: 0.2040 - accuracy: 0.9321 - val_loss: 0.2208 - val_accurac
y: 0.9064 - 216ms/epoch - 6ms/step
Epoch 12/30
38/38 - 0s - loss: 0.1862 - accuracy: 0.9489 - val_loss: 0.2034 - val_accurac
y: 0.9130 - 165ms/epoch - 4ms/step
Epoch 13/30
38/38 - 0s - loss: 0.1710 - accuracy: 0.9464 - val_loss: 0.1920 - val_accurac
y: 0.9197 - 144ms/epoch - 4ms/step
Epoch 14/30
38/38 - 0s - loss: 0.1610 - accuracy: 0.9556 - val_loss: 0.1810 - val_accurac
y: 0.9398 - 155ms/epoch - 4ms/step
Epoch 15/30
38/38 - 0s - loss: 0.1433 - accuracy: 0.9589 - val_loss: 0.1699 - val_accurac
y: 0.9298 - 154ms/epoch - 4ms/step
Epoch 16/30
38/38 - 0s - loss: 0.1372 - accuracy: 0.9623 - val_loss: 0.1630 - val_accurac
y: 0.9331 - 138ms/epoch - 4ms/step
Epoch 17/30
38/38 - 0s - loss: 0.1252 - accuracy: 0.9598 - val_loss: 0.1568 - val_accurac
y: 0.9331 - 194ms/epoch - 5ms/step
Epoch 18/30
38/38 - 0s - loss: 0.1210 - accuracy: 0.9648 - val_loss: 0.1531 - val_accurac
y: 0.9298 - 135ms/epoch - 4ms/step
Epoch 19/30
38/38 - 0s - loss: 0.1122 - accuracy: 0.9631 - val_loss: 0.1485 - val_accurac
y: 0.9298 - 202ms/epoch - 5ms/step
```

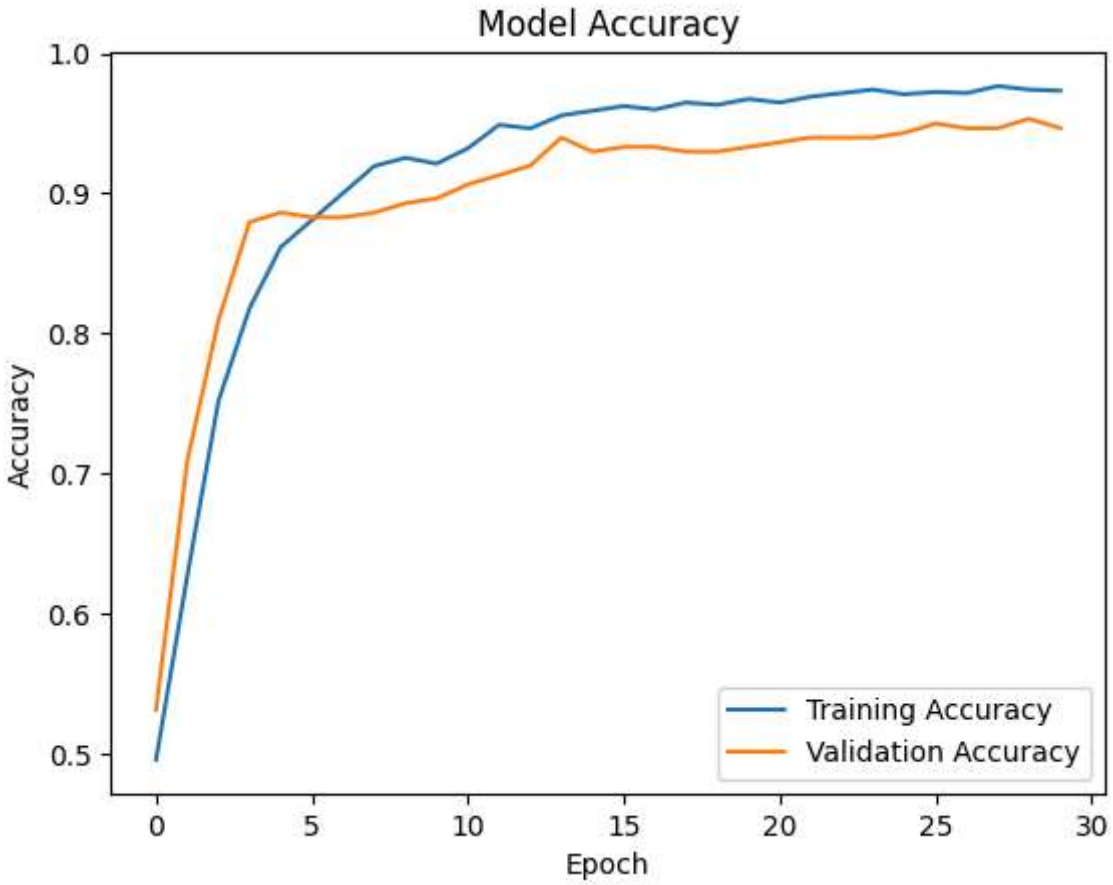
```
Epoch 20/30
38/38 - 0s - loss: 0.1089 - accuracy: 0.9673 - val_loss: 0.1445 - val_accurac
y: 0.9331 - 260ms/epoch - 7ms/step
Epoch 21/30
38/38 - 0s - loss: 0.1067 - accuracy: 0.9648 - val_loss: 0.1443 - val_accurac
y: 0.9365 - 187ms/epoch - 5ms/step
Epoch 22/30
38/38 - 0s - loss: 0.1025 - accuracy: 0.9690 - val_loss: 0.1412 - val_accurac
y: 0.9398 - 213ms/epoch - 6ms/step
Epoch 23/30
38/38 - 0s - loss: 0.0944 - accuracy: 0.9715 - val_loss: 0.1385 - val_accurac
y: 0.9398 - 154ms/epoch - 4ms/step
Epoch 24/30
38/38 - 0s - loss: 0.0950 - accuracy: 0.9740 - val_loss: 0.1374 - val_accurac
y: 0.9398 - 161ms/epoch - 4ms/step
Epoch 25/30
38/38 - 0s - loss: 0.0809 - accuracy: 0.9707 - val_loss: 0.1352 - val_accurac
y: 0.9431 - 163ms/epoch - 4ms/step
Epoch 26/30
38/38 - 0s - loss: 0.0860 - accuracy: 0.9723 - val_loss: 0.1364 - val_accurac
y: 0.9498 - 233ms/epoch - 6ms/step
Epoch 27/30
38/38 - 0s - loss: 0.0814 - accuracy: 0.9715 - val_loss: 0.1337 - val_accurac
y: 0.9465 - 163ms/epoch - 4ms/step
Epoch 28/30
38/38 - 0s - loss: 0.0774 - accuracy: 0.9765 - val_loss: 0.1338 - val_accurac
y: 0.9465 - 151ms/epoch - 4ms/step
Epoch 29/30
38/38 - 0s - loss: 0.0755 - accuracy: 0.9740 - val_loss: 0.1327 - val_accurac
y: 0.9532 - 292ms/epoch - 8ms/step
Epoch 30/30
38/38 - 0s - loss: 0.0739 - accuracy: 0.9732 - val_loss: 0.1351 - val_accurac
y: 0.9465 - 127ms/epoch - 3ms/step
```

Evaluating the Model

In [54]:
```
# Evaluate the model
test_loss, test_accuracy = model.evaluate(Testing_pad, test_labels)
print('Test Loss :',test_loss)
print('Test Accuracy :',test_accuracy)
```

```
10/10 [==============================] - 0s 4ms/step - loss: 0.1351 - accurac
y: 0.9465
Test Loss : 0.13512955605983734
Test Accuracy : 0.9464883208274841
```

In [55]:
```
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend()
plt.show()
```



Insights:

- **Early Improvement**: Both training and validation accuracies increase quickly in the first few epochs, indicating that the model learns well initially.
- **Generalization**: After about 7-10 epochs, the validation accuracy plateaus, suggesting that the model has learned most of the important features and isn't overfitting yet.
- **Overfitting Avoidance**: Since the training accuracy is higher than the validation accuracy after around epoch 10, the model may start slightly overfitting. However, the validation accuracy remains quite close to the training accuracy, which is a good sign.

Conclusion:

The model performs well, achieving high accuracy on both training and validation sets. However, training beyond **10-15** epochs does not yield significant improvement, and **early stopping** can be used to prevent overfitting.

Predicting Spam

In [56]:
```
predict_msg = ["Go until jurong point, crazy.. Available only in bugis n great
          "Ok lar... Joking wif u oni...",
          "Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005.
```

In [57]:
```
def predict_spam(predict_msg):
    new_seq = token.texts_to_sequences(predict_msg)
    padded = pad_sequences(new_seq, maxlen =50,
                      padding = padding_type,
                      truncating='post')
    return (model.predict(padded))
predict_spam(predict_msg)
```

```
1/1 [==============================] - 0s 131ms/step
```

Out[57]:
```
array([[0.00875654],
       [0.00329327],
       [0.99998045]], dtype=float32)
```

How to interpret the results for each message:

- First message (0.00875654):

This value is close to 0, meaning the model predicts a very low probability (0.87%) that the first message is spam. Therefore, this message is classified as ham (non-spam).

- Second message (0.00329327):

This value is even closer to 0, meaning the model predicts an extremely low probability (0.32%) that the second message is spam. Hence, this message is also classified as ham.

- Third message (0.99998045):

This value is extremely close to 1, meaning the model predicts a very high probability (99.99%) that the third message is spam. This message is classified as spam.

Summary:

- The first and second messages are classified as ham (non-spam).
- The third message is classified as spam.
- The predicted probabilities can be thresholded (typically, a threshold of 0.5 is used), meaning:
  - Probabilities ≥ 0.5 indicate spam.
  - Probabilities < 0.5 indicate ham.
- Thus, using a 0.5 threshold:
  - Messages 1 and 2 are ham.
  - Message 3 is spam.