أكاديمية سدايا
SDAIA Academy

SMS Spam or Ham project

NLP – Clustering

TABLE OF CONTENTS

01

- INTRODUCTION

02

- METHODS

03

- RESULT

04

- CONCLUSION

INTRODUCTION

We are a communications company that cares about fulfilling our customers' desires and listening to their requirements.

We received many complaints about the large number of unclear messages they had, which caused a lack of distinction between ham and spam messages.

Solution

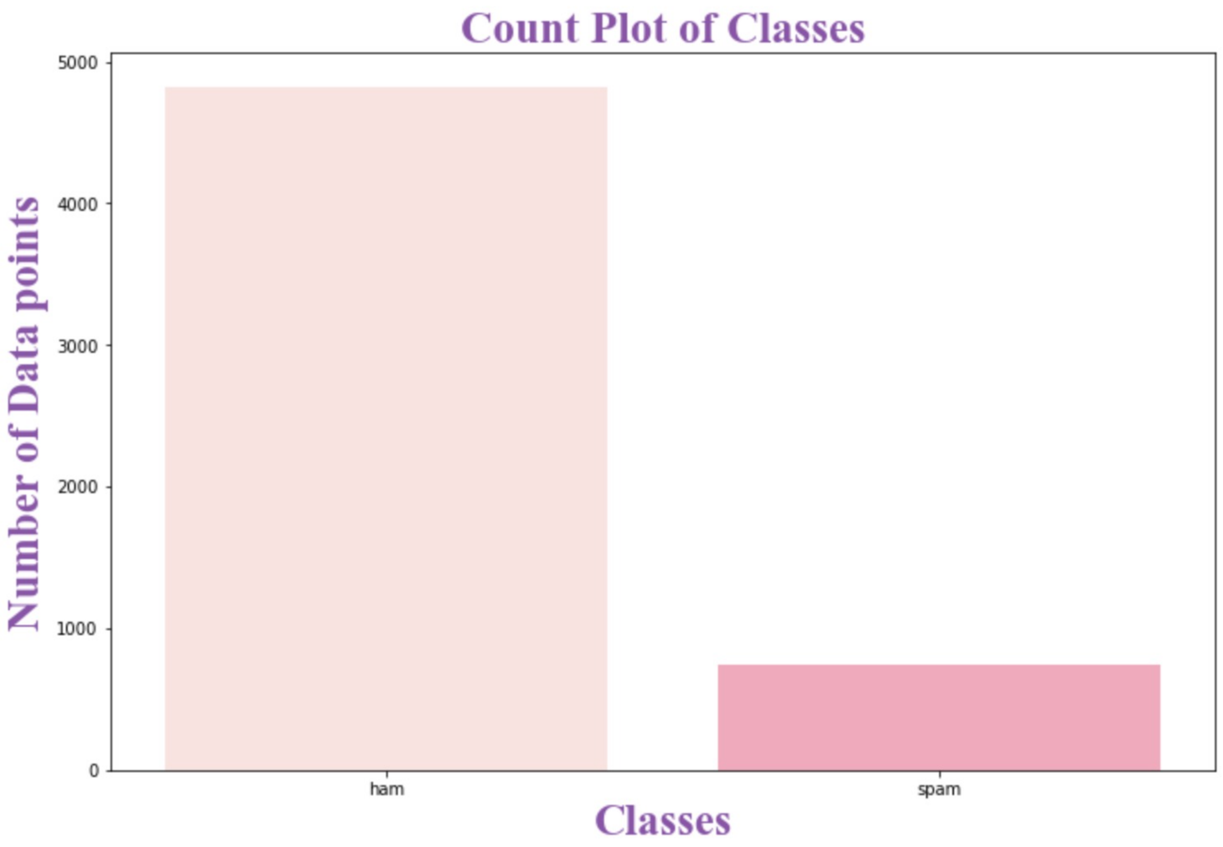
To solve this problem due to the desire of our customers, we used a set of data and artificial intelligence algorithms.

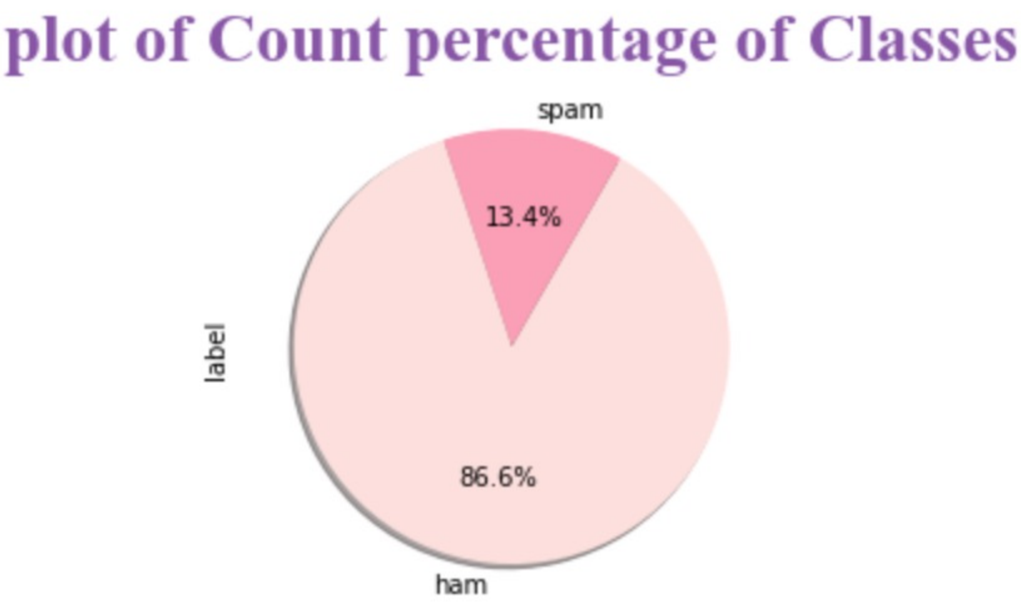
Tools:

1 Python Libraries

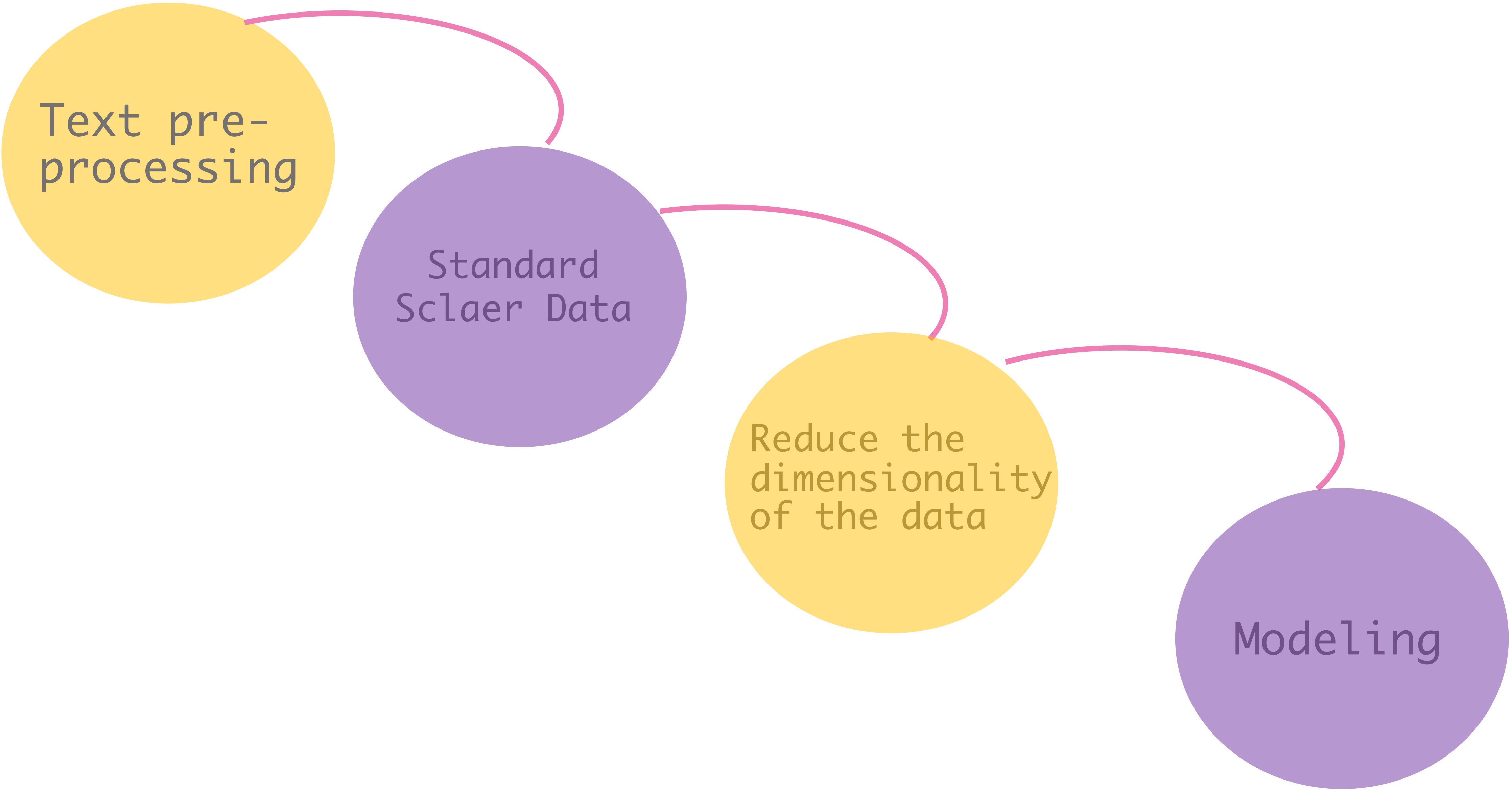
2 Jupyter Notebook

Data Exploration





Steps:



Text pre-processing

Standard Sclaer Data

Reduce the dimensionality of the data

Modeling

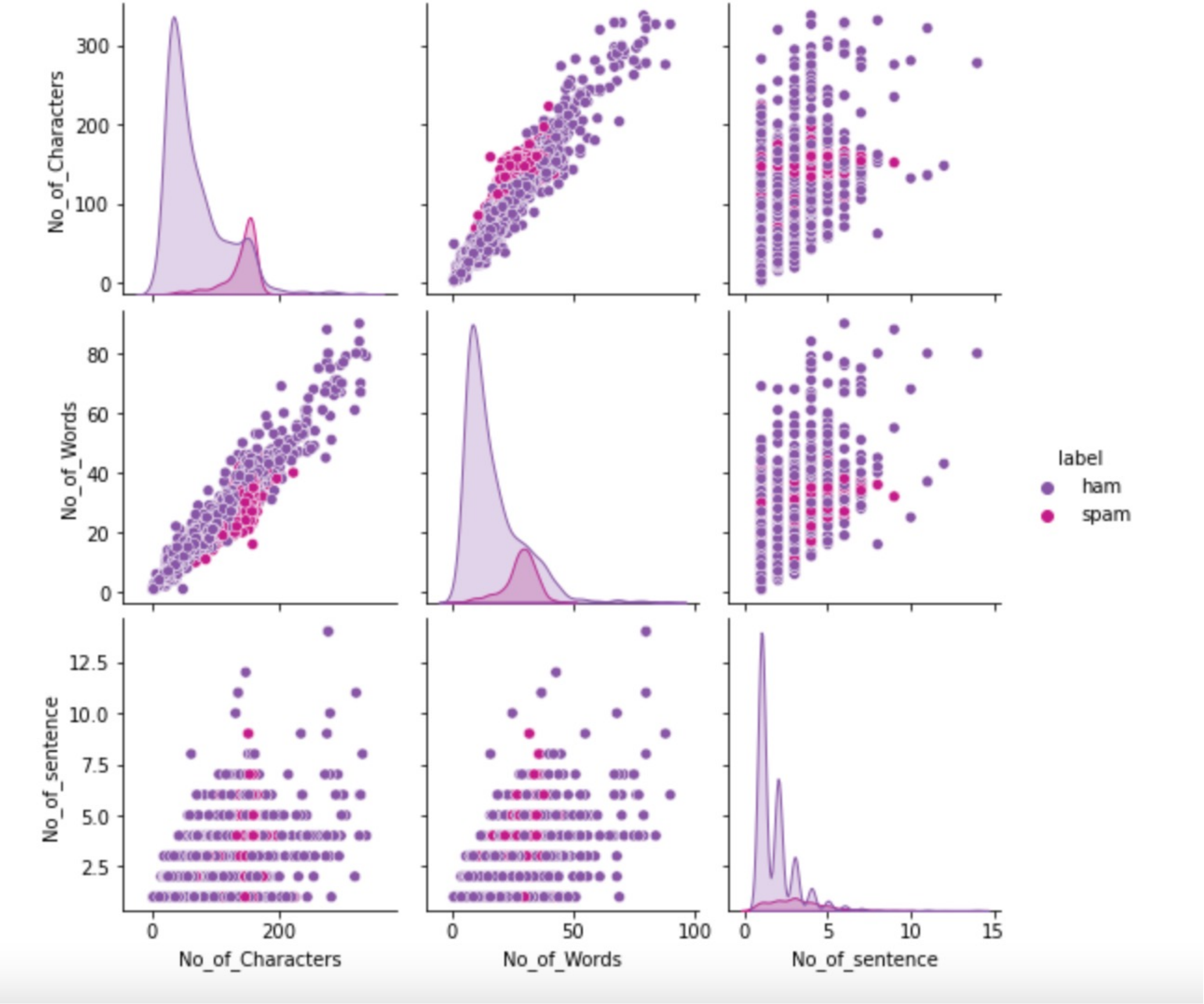
Feature Engineering:

Number of characters in the text message

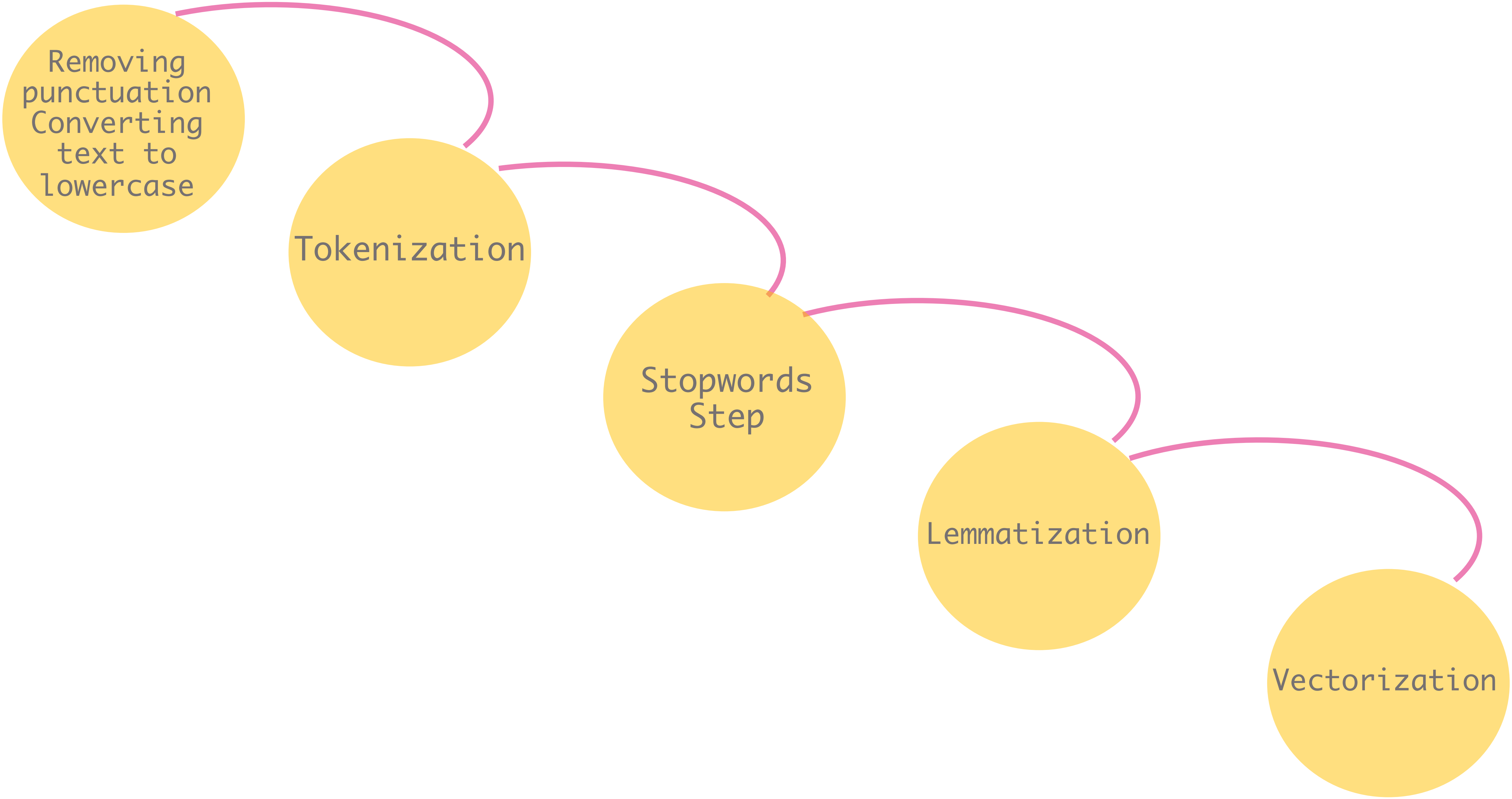
Number of words in the text message

Number of sentences in the text message

Feature Engineering:



NLP Steps



Removing punctuation Converting text to lowercase

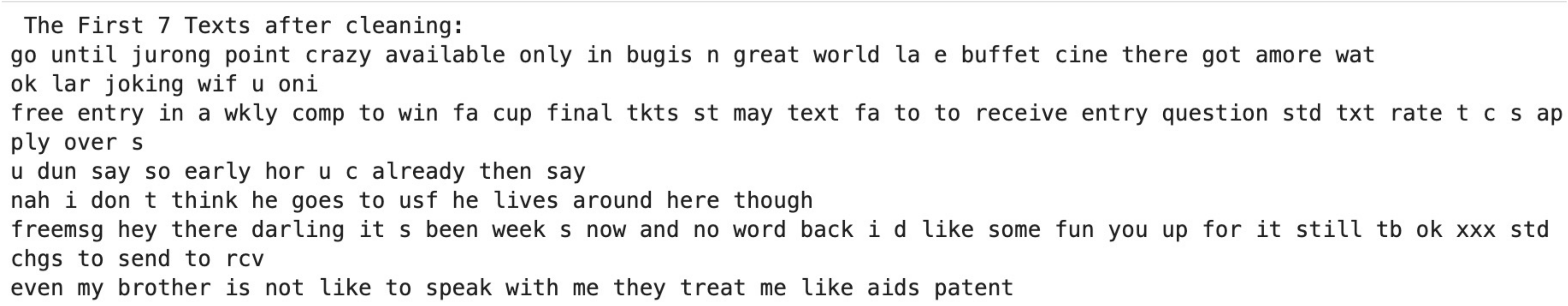
Tokenization

Stopwords Step

Lemmatization

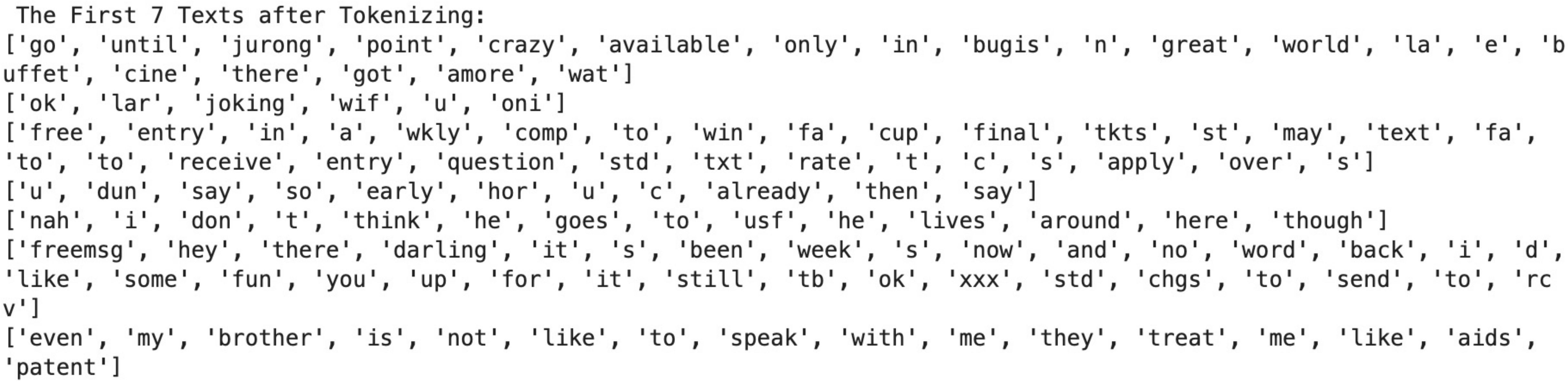
Vectorization

Removing punctuation Converting text to lowercase



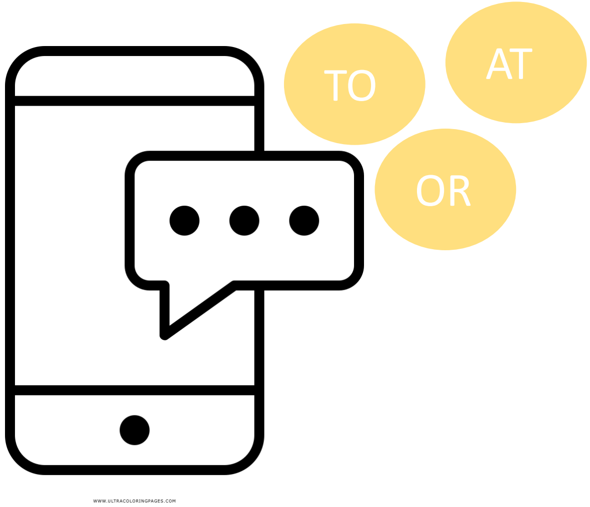
```
 The First 7 Texts after cleaning:
go until jurong point crazy available only in bugis n great world la e buffet cine there got amore wat
ok lar joking wif u oni
free entry in a wkly comp to win fa cup final tkts st may text fa to to receive entry question std txt rate t c s ap
ply over s
u dun say so early hor u c already then say
nah i don t think he goes to usf he lives around here though
freemsg hey there darling it s been week s now and no word back i d like some fun you up for it still tb ok xxx std
chgs to send to rcv
even my brother is not like to speak with me they treat me like aids patent
```

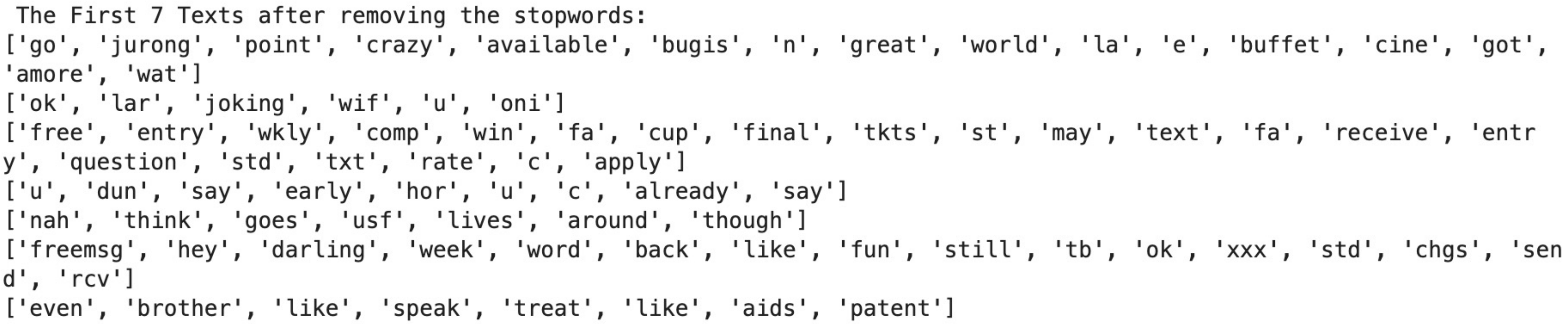
Tokenization



```
 The First 7 Texts after Tokenizing:
['go', 'until', 'jurong', 'point', 'crazy', 'available', 'only', 'in', 'bugis', 'n', 'great', 'world', 'la', 'e', 'b
uffet', 'cine', 'there', 'got', 'amore', 'wat']
['ok', 'lar', 'joking', 'wif', 'u', 'oni']
['free', 'entry', 'in', 'a', 'wkly', 'comp', 'to', 'win', 'fa', 'cup', 'final', 'tkts', 'st', 'may', 'text', 'fa',
'to', 'to', 'receive', 'entry', 'question', 'std', 'txt', 'rate', 't', 'c', 's', 'apply', 'over', 's']
['u', 'dun', 'say', 'so', 'early', 'hor', 'u', 'c', 'already', 'then', 'say']
['nah', 'i', 'don', 't', 'think', 'he', 'goes', 'to', 'usf', 'he', 'lives', 'around', 'here', 'though']
['freemsg', 'hey', 'there', 'darling', 'it', 's', 'been', 'week', 's', 'now', 'and', 'no', 'word', 'back', 'i', 'd',
'like', 'some', 'fun', 'you', 'up', 'for', 'it', 'still', 'tb', 'ok', 'xxx', 'std', 'chgs', 'to', 'send', 'to', 'rc
v']
['even', 'my', 'brother', 'is', 'not', 'like', 'to', 'speak', 'with', 'me', 'they', 'treat', 'me', 'like', 'aids',
'patent']
```

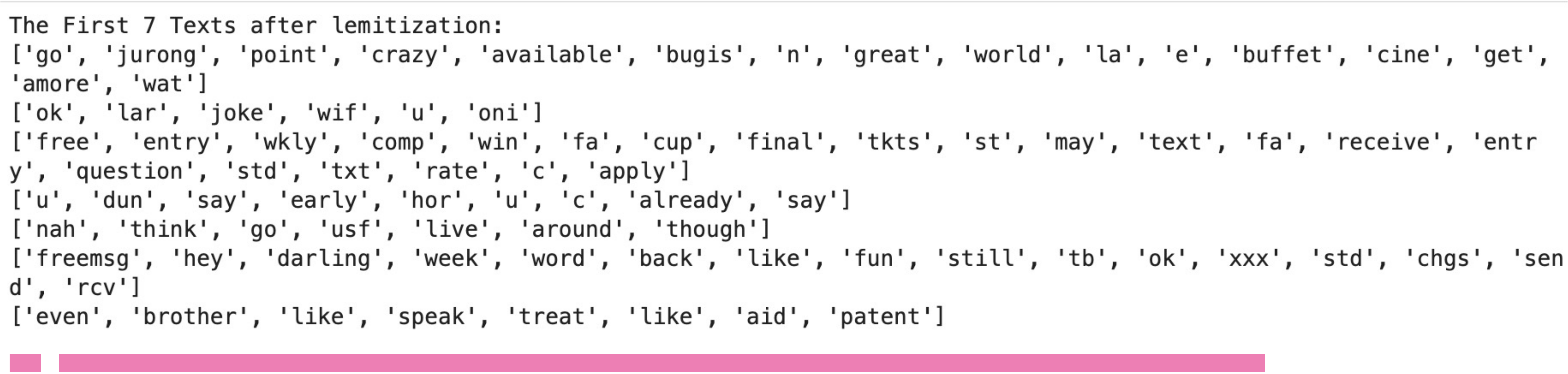
Stopwords Step





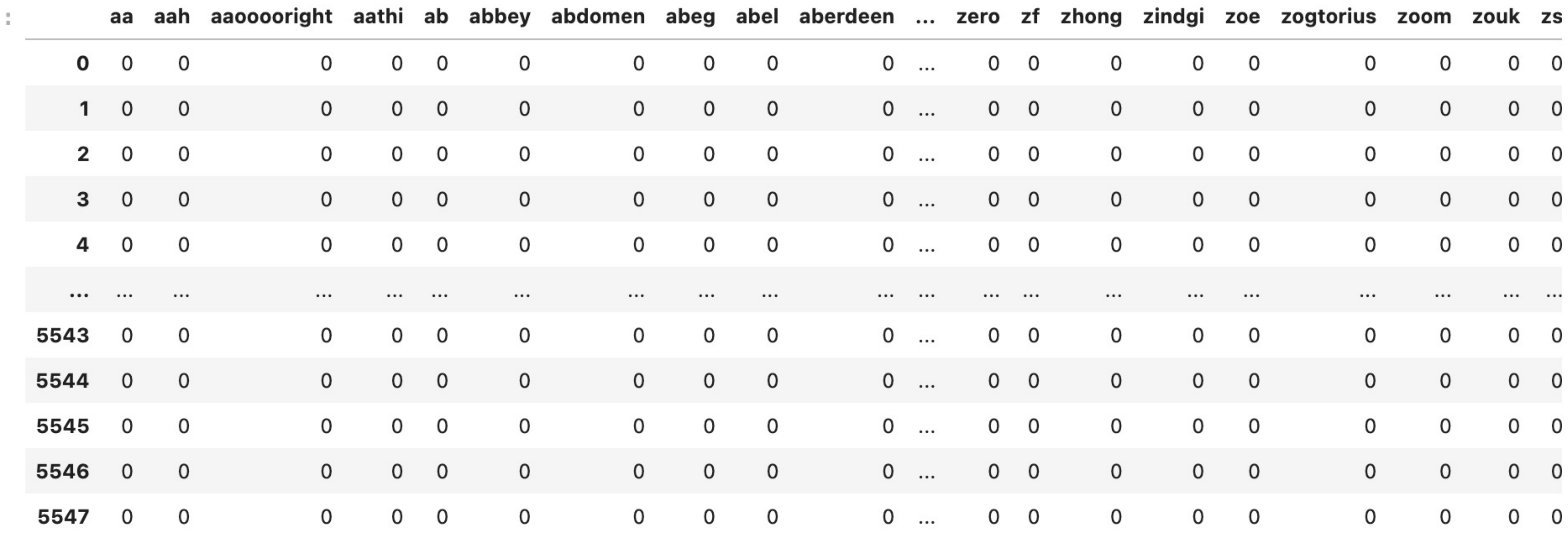
```
 The First 7 Texts after removing the stopwords:
['go', 'jurong', 'point', 'crazy', 'available', 'bugis', 'n', 'great', 'world', 'la', 'e', 'buffet', 'cine', 'got',
'amore', 'wat']
['ok', 'lar', 'joking', 'wif', 'u', 'oni']
['free', 'entry', 'wkly', 'comp', 'win', 'fa', 'cup', 'final', 'tkts', 'st', 'may', 'text', 'fa', 'receive', 'entr
y', 'question', 'std', 'txt', 'rate', 'c', 'apply']
['u', 'dun', 'say', 'early', 'hor', 'u', 'c', 'already', 'say']
['nah', 'think', 'goes', 'usf', 'lives', 'around', 'though']
['freemsg', 'hey', 'darling', 'week', 'word', 'back', 'like', 'fun', 'still', 'tb', 'ok', 'xxx', 'std', 'chgs', 'sen
d', 'rcv']
['even', 'brother', 'like', 'speak', 'treat', 'like', 'aids', 'patent']
```

Lemmatization

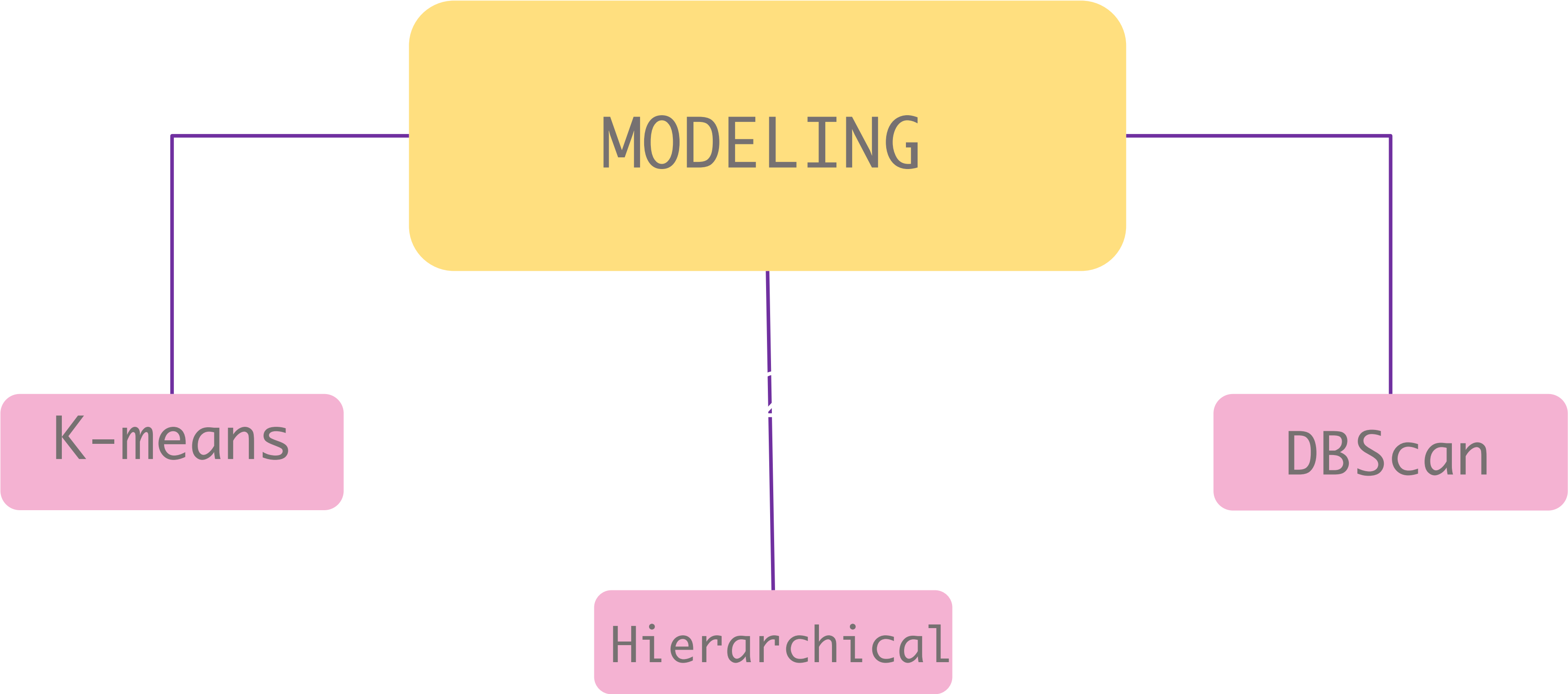


```
The First 7 Texts after lemitization:
['go', 'jurong', 'point', 'crazy', 'available', 'bugis', 'n', 'great', 'world', 'la', 'e', 'buffet', 'cine', 'get',
'amore', 'wat']
['ok', 'lar', 'joke', 'wif', 'u', 'oni']
['free', 'entry', 'wkly', 'comp', 'win', 'fa', 'cup', 'final', 'tkts', 'st', 'may', 'text', 'fa', 'receive', 'entr
y', 'question', 'std', 'txt', 'rate', 'c', 'apply']
['u', 'dun', 'say', 'early', 'hor', 'u', 'c', 'already', 'say']
['nah', 'think', 'go', 'usf', 'live', 'around', 'though']
['freemsg', 'hey', 'darling', 'week', 'word', 'back', 'like', 'fun', 'still', 'tb', 'ok', 'xxx', 'std', 'chgs', 'sen
d', 'rcv']
['even', 'brother', 'like', 'speak', 'treat', 'like', 'aid', 'patent']
```

Vectorization



	aa	aah	aaooooright	aathi	ab	abbey	abdomen	abeg	abel	aberdeen	...	zero	zf	zhong	zindgi	zoe	zogtorius	zoom	zouk	zs
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5543	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5544	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5545	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5546	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
5547	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0



MODELING

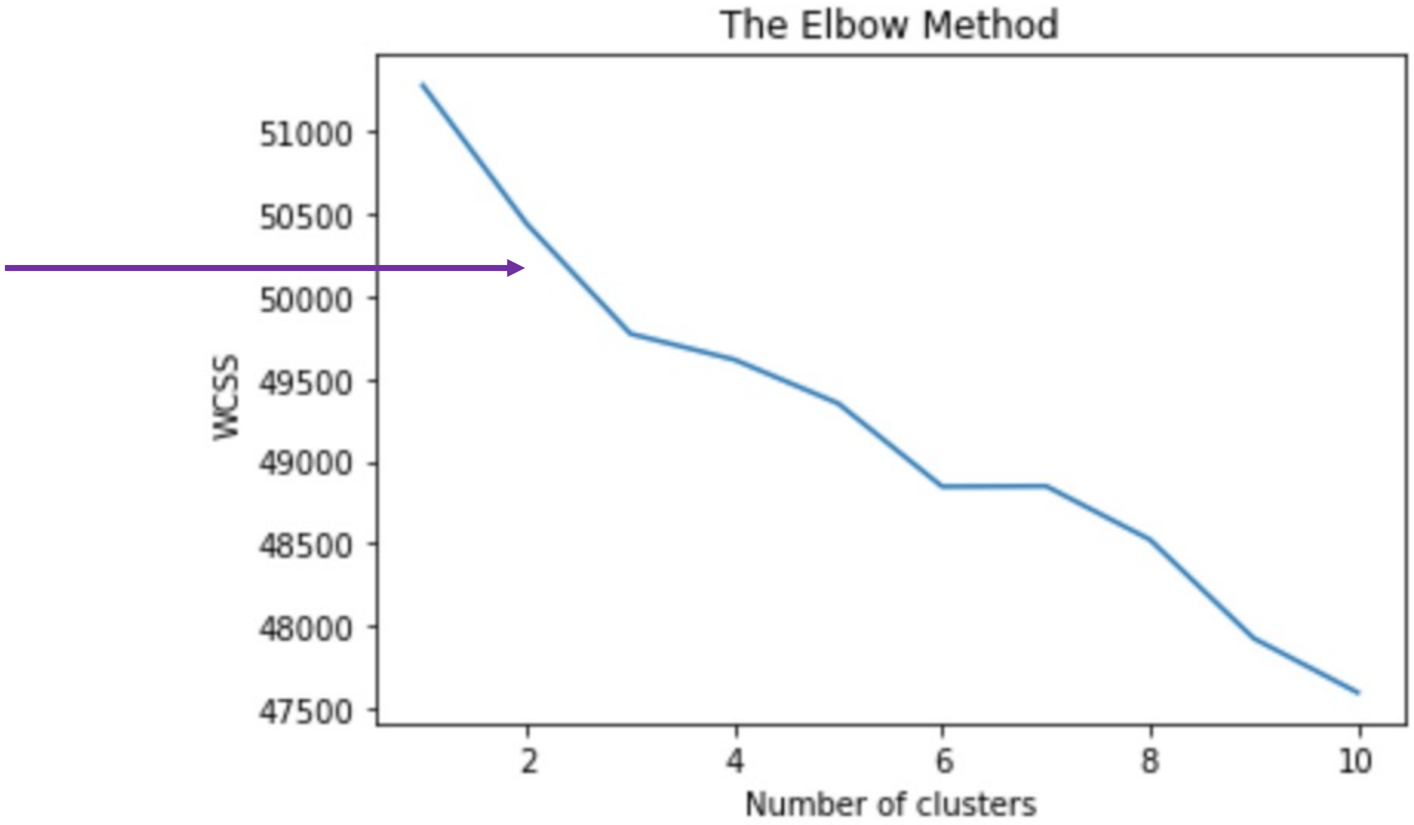
K-means

Hierarchical

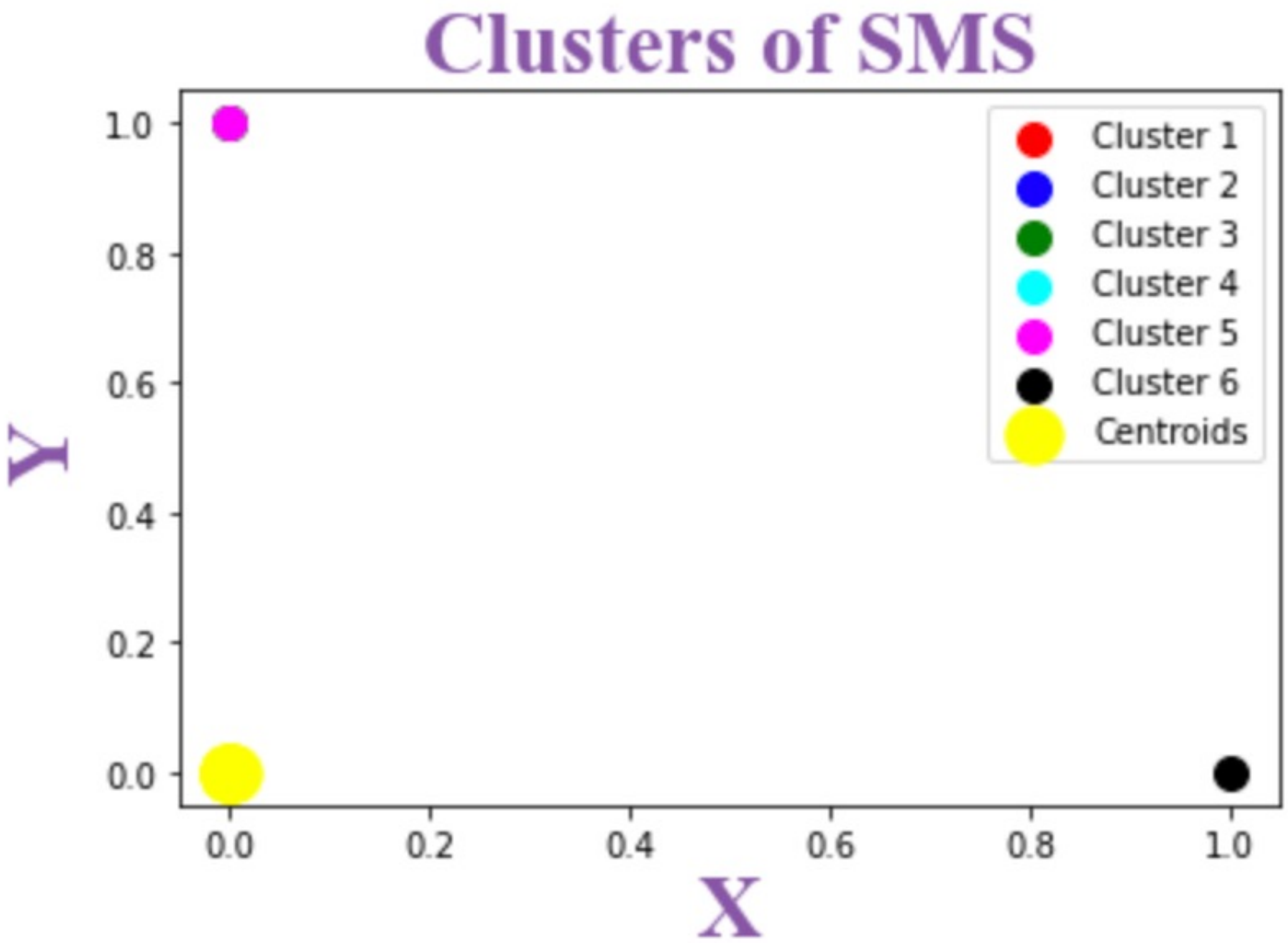
DBScan

Elbow

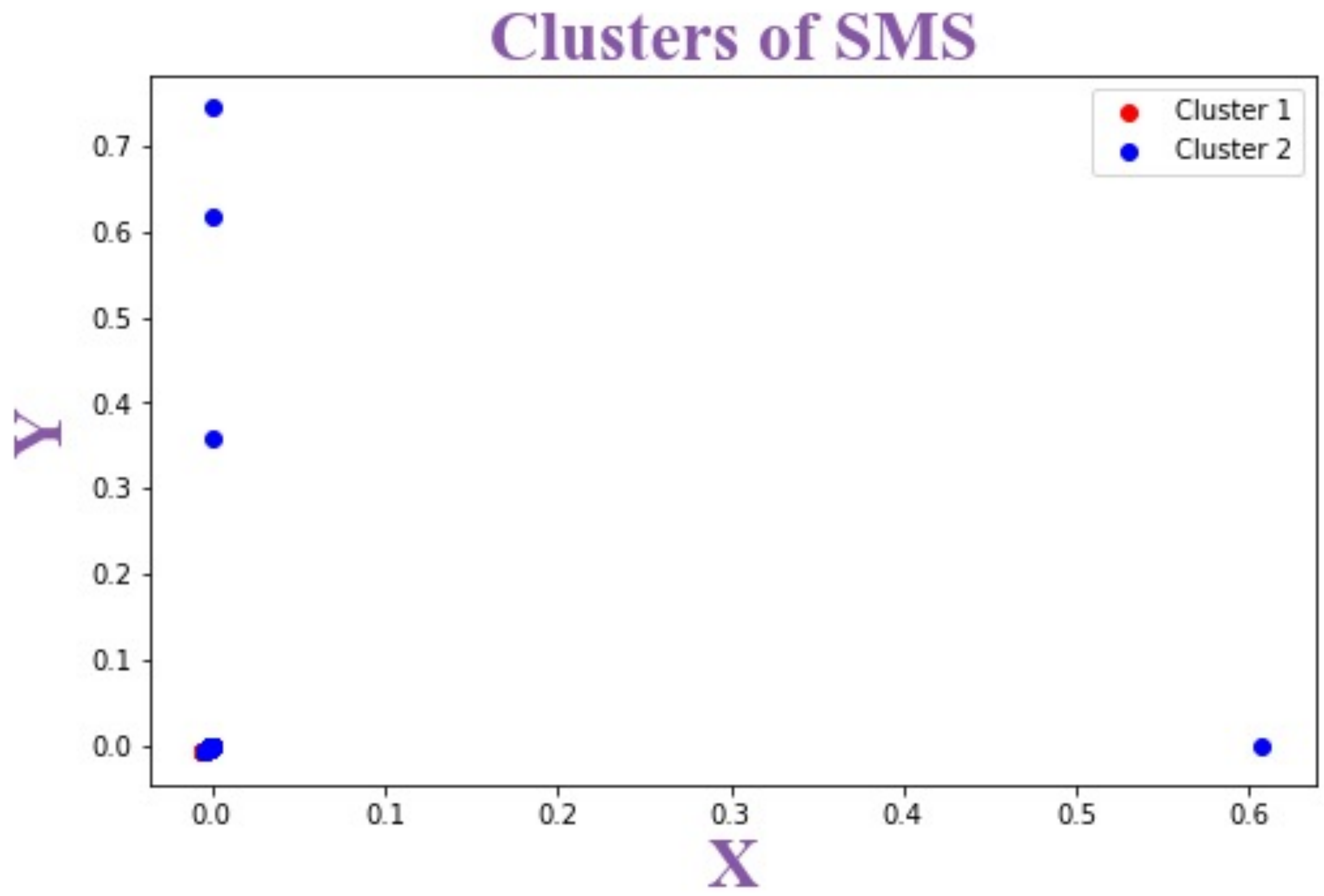
We use the Elbow Function To find the number of Cluster



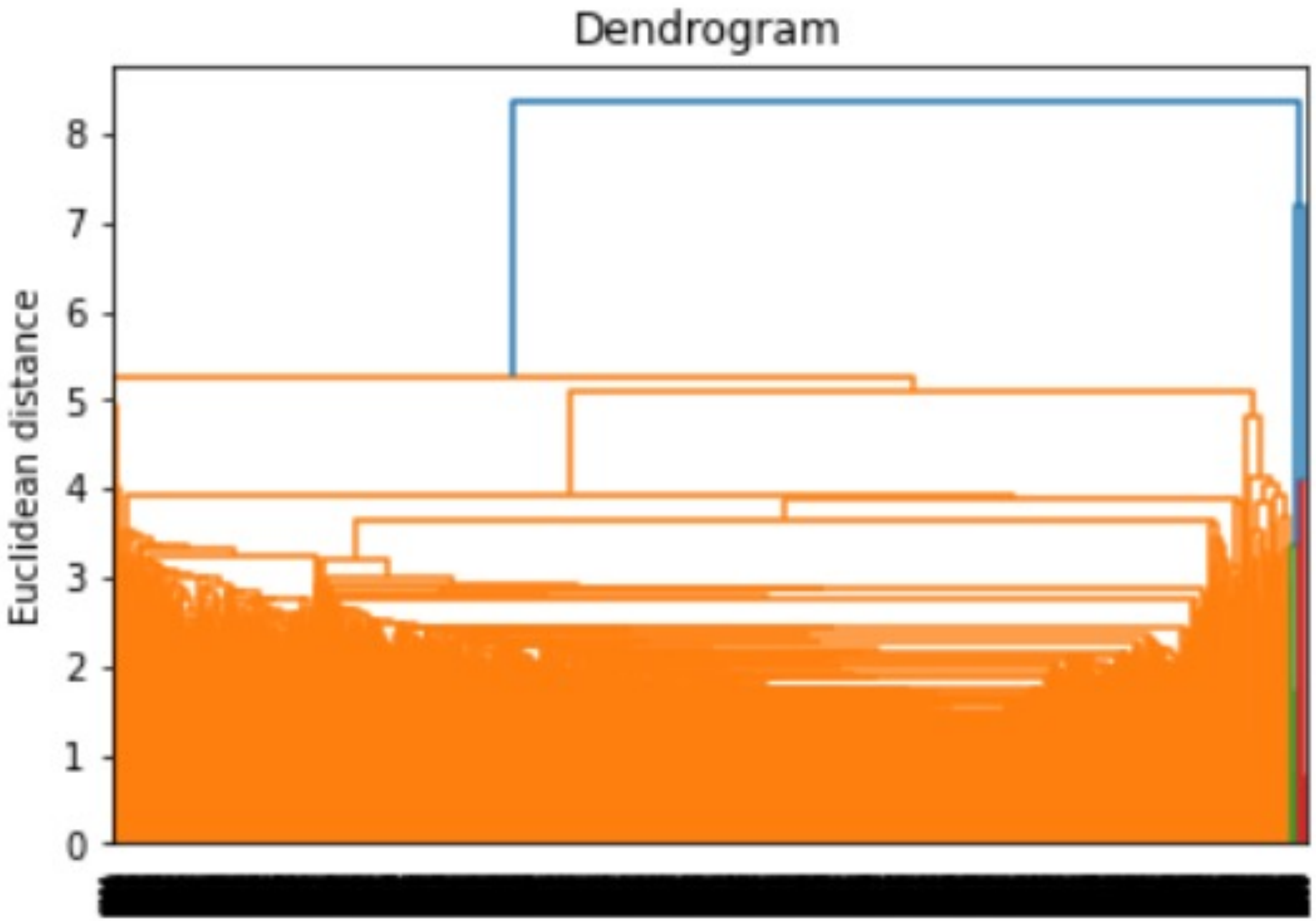
K-means Model



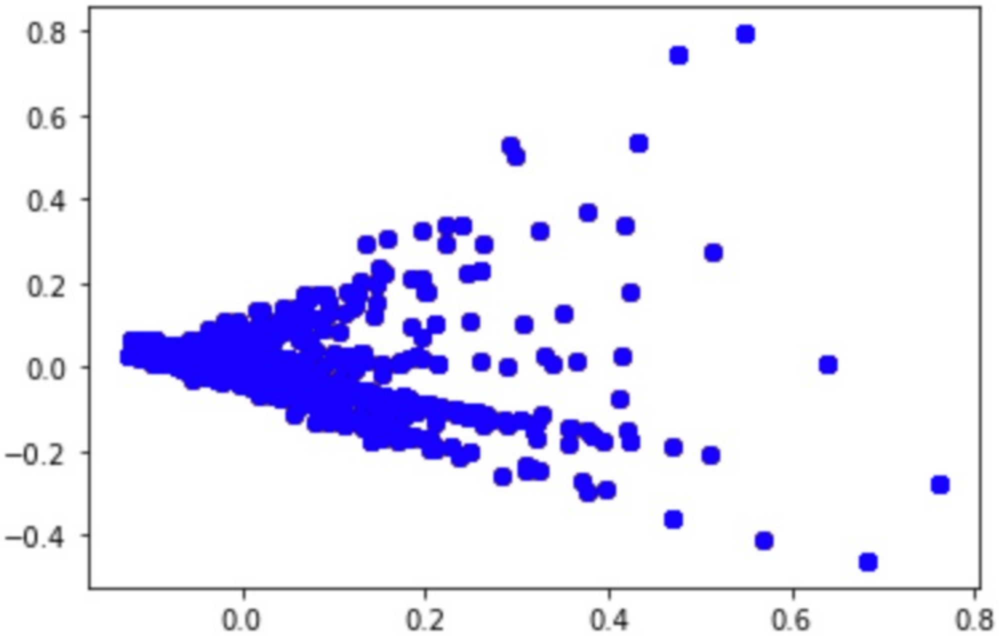
Hierarchical Model

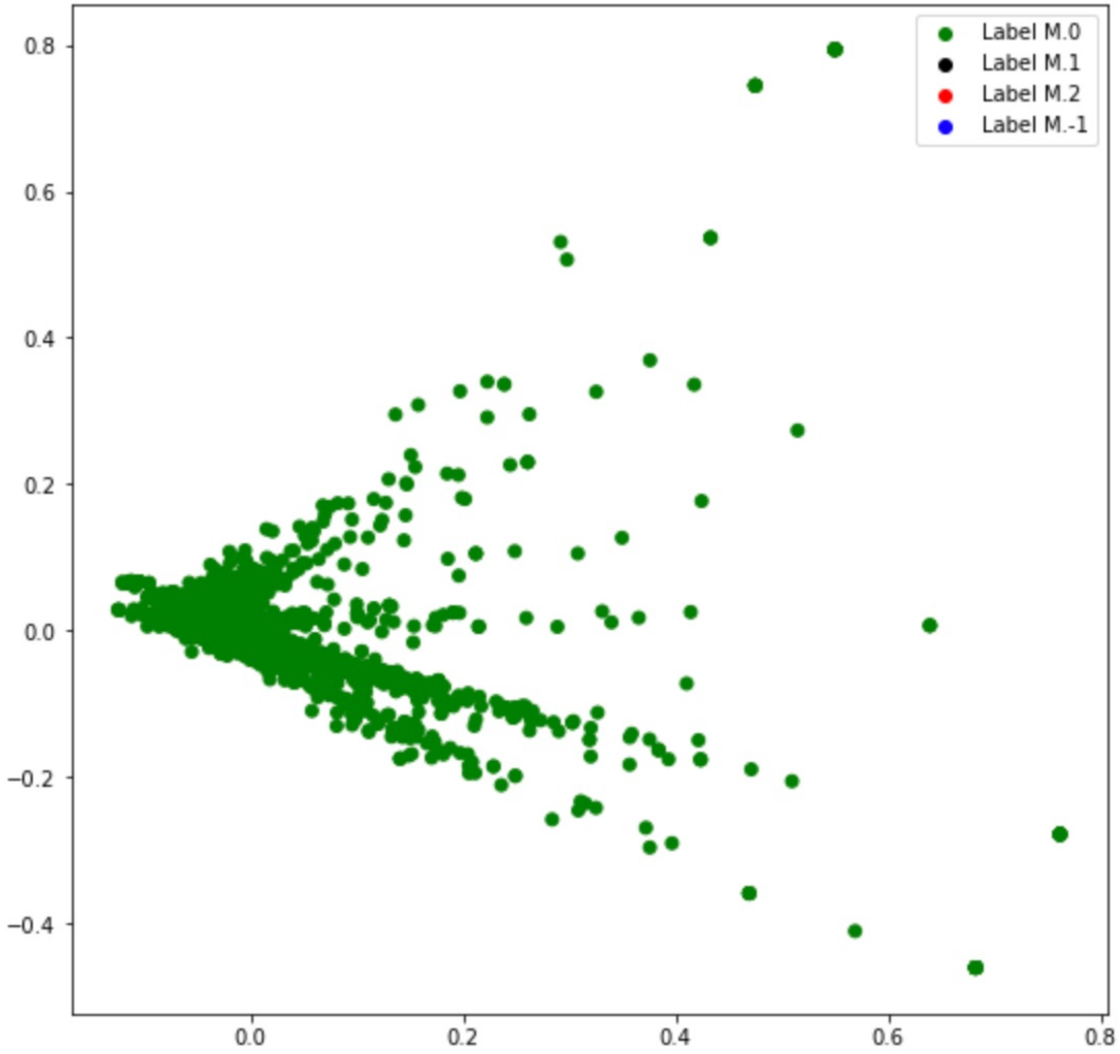


Hierarchical Model



DBScan Model





Conclusion

After analyzing the text and cleaning by NLP Steps, we found out that the text consists of two types (spam, ham SMS) according to the cluster models. so we will help our client to select the ham messages and ignore spam messages by ML tools.

Thank You

Any Question?

Team trainees

Nada Alqabbani
Shahad Almubki
Nada Alhamad
Sarah Alameer
Hala Almulhim

Instructor:

Mohammed Baddar