	Class	Text	...	numClass	Count
0	ham	Go until jurong point, crazy.. Available only ...	...	0	111
1	ham	Ok lar... Joking wif u oni...	...	0	29
3	ham	U dun say so early hor... U c already then say...	...	0	49
4	ham	Nah I don't think he goes to usf, he lives aro...	...	0	61
6	ham	Even my brother is not like to speak with me. ...	...	0	77
...	...	...	...	...	...

```
print(ham)
```

	Class	Text	...	numClass	Count
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	...	1	155
5	spam	FreeMsg Hey there darling it's been 3 week's n...	...	1	148
8	spam	WINNER!! As a valued network customer you have...	...	1	158
9	spam	Had your mobile 11 months or more? U R entitle...	...	1	154
11	spam	SIX chances to win CASH! From 100 to 20,000 po...	...	1	136
...	...	...	...	...	...

```
print(spam)
```

```
vectorizer.fit_transform(data_read.Text)
```

```
(5570, 4161)  1
(5570, 903)   1
(5570, 1546)  1
(5571, 7756)  1
(5571, 5244)  1
(5571, 4225)  2
(5571, 7885)  1
(5571, 6505)  1
```

```
(0, 8030) 1
(0, 4350) 1
(0, 5920) 1
(0, 2327) 1
(0, 1303) 1
(0, 5537) 1
(0, 4087) 1
(0, 1751) 1
(0, 3634) 1
```

```
data_read.numClass
```

```
5567    1
5568    0
5569    0
5570    0
5571    0
```

```
0       0
1       0
2       1
3       0
4       0
```

```
Adaboost:

Accuracy in %:
98.08612440191388


F1 Score:
0.9130434782608695
```

```
KNN:

Accuracy in %:
94.91626794258373


F1 Score:
0.7745358090185676
```

```
SVM:

Accuracy in %:
98.20574162679426


F1 Score:
0.9308755760368664
```

```
SGD:

Accuracy in %:
98.44497607655502


F1 Score:
0.9304812834224598
```

```
tfidf_transformer = TfidfTransformer().fit(x)
dummy_transformed = tfidf_transformer.transform(x)
print(dummy_transformed)
```

```
(5570, 1546)  0.3402048888248921
(5570, 1438)  0.1429585509124154
(5570, 1084)  0.11225268140936365
(5570, 903)   0.3247623397615813
(5571, 7885)  0.42752913176432156
(5571, 7756)  0.14849350328973984
(5571, 6505)  0.5565029307246045
(5571, 5244)  0.39009002726386227
(5571, 4225)  0.5773238083586979
```

```
(0, 8489) 0.22080132794235655
(0, 8267) 0.18238655630689804
(0, 8030) 0.22998520738984352
(0, 7645) 0.15566431601878158
(0, 5920) 0.2553151503985779
(0, 5537) 0.15618023117358304
(0, 4476) 0.2757654045621182
(0, 4350) 0.3264252905795869
```

Now, lets check IDF for *you*, the most frequently repeated word in the message against *hey*, a least repeated word

```
you: 2.2548286210328206
hey: 4.907189916274442
```

As you can see, words with lower frequency are weighed higher than words with higher frequency in the dataset.

```
Multi-NB:

Accuracy in %:
98.08612440191388


F1 Score:
0.9285714285714285
```

```
Testing specific messages:


SMS1 = '[URGENT!] Your Mobile No 398174814449 was awarded a vacation'

SMS2 = 'Hello my friend, how are you?'

SMS1 is spam .. SMS2 is ham
```

```
DecisionTreeClassifier:

Accuracy in %:
96.88995215311004


F1 Score:
0.8864628820960699
```