



Dept. of Computer Science and Software Engineering

COMP 6721 : Applied Artificial Intelligence

Winter 2020

Project 2

Submitted To: **Dr. René Witte**

Submitted By:
Team **FL-G06**

Name	Student ID	Email
Mehrnaz Keshmirpour	40063320	keshmirpour@gmail.com
Naghmeh Shafiee Roudbari	40129324	naghmehshafiee@gmail.com
Jasmine Kaur	40103309	jasmine14concordia@gmail.com

Analysis

- Accuracy** = (number of instances correctly classified / Total number of instances) * 100
 = $(734/800)*100$
 = **91.75%**

	Instance in class Spam	Instance in class Ham
Model identified as spam	339	5
Model identified as ham	61	395

- Precision_(spam)** = number of instances that are in class Spam & model labeled as spam / total number of instances model labeled as Spam = $339 / (339+5) = \mathbf{0.985}$
Precision_(ham) = number of instances that are in class Ham & model labeled as ham / total number of instances model labeled as Ham = $395 / (395+61) = \mathbf{0.866}$
- Recall_(spam)** = number of instances that are in class Spam & model labeled as spam / all instances in class Spam = $339 / (339+61) = \mathbf{0.848}$
Recall_(ham) = number of instances that are in class Ham & model labeled as ham / all instances in class Ham = $395 / (395+5) = \mathbf{0.988}$
- F-measure** (a weighted combination of precision & recall)
 $F = (\beta^2 + 1) * PR / (\beta^2 P + R)$; $\beta = 1$ since, precision and recall have same importance
 $F_{(spam)} = (2 * 0.985 * 0.848) / (0.985 + 0.848) = 1.67056 / 1.833 = \mathbf{0.911}$
 $F_{(ham)} = (2 * 0.866 * 0.988) / (0.866 + 0.988) = 1.711216 / 1.854 = \mathbf{0.923}$

		Precision	Recall	F1-measure
SPAM class		0.985	0.845	0.911
HAM class		0.864	0.987	0.923

- Confusion Matrix / Contingency Table**

correct class (that should have been assigned)	classes assigned by the learner		
	Ham	Spam	Total
Ham	395	5	400
Spam	61	339	400

This shows test dataset consist of equal distribution of spam and ham files. However, the model identifies 5 Ham files and 61 Spam files incorrectly i.e. it marked them to their opposite classes.

References

[1] Word tokenization using python regular expressions:

(<https://stackoverflow.com/questions/6202549/word-tokenization-using-python-regular-expressions>)

[2] Find encoding source

(<https://stackoverflow.com/questions/31019854/typeerror-cant-use-a-string-pattern-on-a-bytes-like-object-in-re-findall>)

[3] Word count source

<https://towardsdatascience.com/very-simple-python-script-for-extracting-most-common-words-from-a-story-1e3570d0b9d0>