SPAM FILTR

Michaela Šromová

Úkolem bylo klasifikovat emaily, zda jsou spam či nikoliv. K dispozici jsem měla dva datasety emailů s jejich klasifikací.

V mém spam filtru jsem použila klasifikační metodu kNN (k nearest neighbours), kde se porovnává k nejpodobnějších emailů z trénovací množiny s dotazovaným emailem. Pokud je nadpoloviční většina spam, daný email se vyhodnotí jako spam, jinak se vyhodnotí jako ham.

Pro měření podobnosti jsem využila kosinovou podobnost (cosine similarity), která porovnává kosinus úhlu mezi dvěma vektory, nikoliv jejich vzdálenost.

Co se týče předzpracování textu, těla emailů jsem rozdělila na slova, z nichž jsem si vytvořila slovník bez tzv. stopwords (spojky, zájmena apod.). Také jsem se snažila odstranit koncovky (playing -> play).

Pro každý email jsem vytvořila vektor slov za pomoci tf-idf (term-frequency, inverse document frequency), což pomáhá určit důležitost daného slova v emailu.

Jelikož by vektory měli tisíce položek, použila jsem PCA (principal component analysis), která snižuje dimenzi na stanovený počet (pro který to stále dává smysl).

Při trénování jsem balancovala trénovací množinu, aby počet spamů a hamů byl stejný. Jelikož metoda kNN je citlivá na nevyváženost trénovacího datasetu.

Trénovala jsem filtr na prvním datasetu a testovala na tom druhém a naopak.

Všechny výše zmíněné metody jsem zvolila proto, že nejsou tolik složité na implementaci.

Výsledky z mého trénování a testování:

Trénováno na množině	Testováno na množině	Míra kvality
Dataset 1	Dataset 1	0.756
Dataset 1	Dataset 2	0.717
Dataset 2	Dataset 2	0.776
Dataset 2	Dataset 1	0.743

S dosaženými výsledky jsem spokojená, filtr by ale mohl být rychlejší.

Seznam použitých zdrojů:

https://stackoverflow.com/questions/17874360/python-how-to-parse-the-body-from-a-raw-email-given-that-raw-email-does-not
https://en.wikipedia.org/wiki/Tf%E2%80%93idf
https://stackoverflow.com/questions/18424228/cosine-similarity-between-2-number-lists
https://scikit-learn.org/1.5/modules/feature_extraction.html#text-feature-extraction
https://en.wikipedia.org/wiki/Principal_component_analysis
https://pythonspot.com/nltk-stop-words/
https://blog.georgeshakan.com/singular-value-decomposition-and-pca/