June 2020 **Braude College of Engineering**

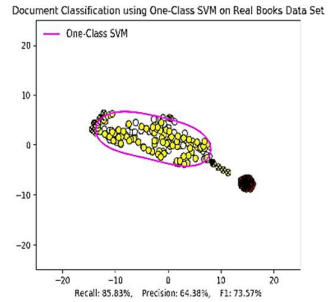
Data Mining and Machine Learning (61761) – Final Project (Python)

Subject: One-Class SVMs for Document Classification

Group Members (#2): **Raz Malka** **(#########)**
Shoham Yamin **(#########)**
Raz Itzhak Afriat **(#########)**

Repository Link: **https://github.com/RazMalka/SVM-DC**

Brief:



Implementation of One-class SVM that uses binary, frequency, tf-idf and hadamard representations for document classification. Based upon a research paper by Larry M. Manevitz and Malik Yousef, 'One-Class SVMs for Document Classification'.

It includes a graphical user interface with choice of representation, kernel, data cache, outlier detection and SVM view control.

Study Process:

-> Reading the major article and some of its bibliography references
-> Further delving into the subject of One-Class SVM and its uniqueness
-> Understanding the meaning and definition of each representation and kernel
-> Realizing the importance of embedding high-dimensional data in a low-dimensional space of two dimensions for visualization.

Project Flow:

-> Analyzing research papers and summarizing essential information
-> Definition of main goals and schedule planning
-> Gathering documents in the form of books as a dataset
-> Application of key principles from the research paper into code
-> Planning and design of a graphical user interface and control over parameters
-> Repetitive process of research and remodeling of each component until success
-> Continuous use of version control to track changes and optimize parallel work

Obtained Results:

Figure 1 – Table of One-class SVM alongside various representations

Kernel / *Representation*	*Linear*			*Radial (RBF)*		
	F_1	R	P	F_1	R	P
Binary	0.638	0.744	0.558	0.773	0.902	0.676
Frequency	0.550	0.641	0.481	0.737	0.860	0.645
TF-IDF	0.673	0.785	0.589	0.703	0.820	0.615
Hadamard	0.564	0.658	0.493	0.725	0.846	0.634
∗ Value of each cell is the average of 100 samples of identical nature – m = 15						

Conclusions:

One-Class SVM presented different results in accordance with different settings of dataset, representation and kernel. The radial kernel (RBF) generally gave significantly better results versus linear kernel on most datasets, aside from noticeably inconsistent datasets with high variance (samples are very far apart).

A field where linear kernel performs especially poorly, as expected, is on linearly inseparable data, with insignificant keyword differences (e.g. keywords from a pair of a drama book and a cooking book will be more easily distinguishable than two books of the same category) – radial kernel performed better there, detecting more subtle differences.