**Kamil Peza & Yohannes Berhane: Linear SVM Sentiment Analysis Using Python and Scikit-Learn**

Text mining is the process of using artificial intelligence and data analysis to obtain impactful and meaningful data from text. Text mining has wide-spread applications in society, ranging from risk management software for businesses to cyber-crime prevention to digital media advertising and in general, providing more information to whoever wants to experiment with their data and understand patterns.

In this project, we are utilizing python with various libraries to create an accurate sentiment analysis classifier for movie reviews. The training and test data set for this classifier is a long CSV file that has two column attributes (review content, positive/negative). It is already preprocessed. After loading in the dataset, we have to tokenize and vectorize it. Here we are using **TF-IDF**. **“Term Frequency and Inverse Document Frequency”** is word frequency scores that try to highlight words that are more interesting, e.g. frequent in a document but not across documents. Then in order to create the SVM, we used sklearn imports and created a linear kernel classifier.

After we run our test data using this SVM classifier, we can get a classification report using sklearn-metrics imports. The training data set here was 2000 movie reviews. The tested data set for positive classifications had

* positive: {'precision': 0.9191919191919192, 'recall': 0.91, 'f1-score': 0.9145728643216081, 'support': 100}

while the negative classification’s had similar stats as well:

* negative: {'precision': 0.9108910891089109, 'recall': 0.92, 'f1-score': 0.9154228855721394, 'support': 100}

In order to test the accuracy of the classifier, I randomly selected reviews from highly rated movies and poorly rated movies and checked if the classifier classified the sentiment polarity correctly. The classifier correctly predicts most reviews especially if there are blatant keywords there that TF-IDF easily recognized.