CS 386: PROJECT 3 REPORT **Shivam Pandey** (160010003)
October 29, 2018

1 Introduction and Overview

At the heart of this project is performing various document modeling techniques such as LSA and TFIDF using python modules like sklearn, NLTK, scipy to model an E-Commerce customer review dataset for the purpose of Information Retrieval(IR). The dataset provided consists of the information regarding Women's E-Commerce Clothing Reviews and Ratings. Thereafter, a research question is formulated which entails "Classification of the customer reviews as *Recommended* or *Not Recommended*" by training over the labelled review dataset using Machine Learning models such as Logistic Regression(LGR) and Support Vector Machine(SVM).

2 Methods

Information Retrieval
Two document modeling methods are incorporated for IR viz. *TF-IDF* and *LSA on TF(or TFIDF) matrix*. Initially, a particular clothing-ID was chosen to work upon. Commencement of the IR entails pre-processing the corpus which encompasses *punctuation removal(using Regex), lower casing, stemming, lemmatization* and *stop-words removal(using NLTK)*. A vocabulary is built out of the pre-processed corpus. Using scikit-learn, a TF(Term Frequency) matrix is built whose rows represent the documents and columns represent the vocabulary terms(as features), and a TFIDF matrix is also built. Some key-words are taken as a query vector, whose cosine is taken with each of the document's tfidf vector(each row of TFIDF matrix) followed by sorting, to obtain a document ranking. For LSA, SVD(Singular Value Decomposition) is performed on TF(or TFIDF) matrix and each of the document is represented in some lower dimension 'K'(we have taken K=2), cosine of each of which is taken with query vector followed by sorting to obtain a document ranking.

Research Question : Customer Review Classification
Machine learning techniques such as Logistic Regression and Support Vector Machine was applied to the entire corpus for classification of reviews as "recommended" and "not recommended". The inception of this approach entails assigning some mathematical entity to each document which was obtained by dimensionality reduction using SVD, to map each review text with a K-dimensional vector(here, K=2). Hereafter, the labelled dataset was trained using LGR and SVM models.

3 Analyses of Results

Comparision of TFIDF and LSA methods
TFIDF method's running time is 0.032775 secs, while LSA(on TF-matrix) method's running time is 0.02542 secs. We discern that LSA method runs faster than the TFIDF method. We diagnose that LSA will take lesser space than TFIDF method. The reason being, in TFIDF method we need to deal with 'D x V' matrix (D = no. of documents, V = no. of words). Elseways, in LSA method, we need to store only 'D x K' matrix, (K = reduced dimention), and the Eigen matrix 'V x K'. Between LSA and TFIDF, the former prevails in terms of computational aspects such as space-time complexity. It is subjective to decide which method outputs better ranking. Upon scrutinizing the documents ranking, we discern that LSA entails context understanding given the key words, i.e., it not only just ranks the corpus with given keywords, it also considers documents which mean the same in context involving some different words. In case of TFIDF, we diagnose that it outputs first those documents with the exact keywords given by the user. By this analysis, we conclude that LSA works better overall than TFIDF approach.

Inference from LGR and SVM applied for Review Classification
Upon applying LGR and SVM to the labelled dataset, we diagnose that both the classification methods end up displaying approximately same score(accuracy).

- Accuracy of the Logistic Regression Model : 81.544 %
- Accuracy of the SVM Model using Linear Kernel : 81.557 %
- Accuracy of the SVM Model using Gaussian Kernel : 81.557 %

By Training LGR and SVM on the reduced dimensional vectors given by LSA , we got around 81.5% accuracy in both cases. Upon scrutinizing we diagnose that, both models are always predicting as "Recommended" and Recommended-IND = 1 was in 80% of data. Hence, we infer that LSA might be learning some features which cannot be used to differentiate between documents based on the label as recommended and not recommended. We need more rigorous experimentation to solve this problem.

References

[1] Alex Thomo : *Latent Semantic Analysis* (Tutorial)