Unsupervised Trait-Clustering for Adobe Audience Manager

Data Science Institute
COLUMBIA UNIVERSITY

Lisa Kim, Hongyue Lan, Wyatt Ford, Midhun Gundapuneni, Mohamed Maskani Filali
Industry Mentors: Charles Menguy, Handong Zhao
Faculty Mentor: Eleni Drinea

Data Science Capstone Project with Adobe

Trait-Clustering for Unary Dataset

Adobe Audience Manager allows marketers to create audience segments based on fine-grained user trait data.

Our goals:

- Develop a trait-clustering algorithm
- Make a trait recommendation system
- Build an interactive tool to visualize the results

Description of the data

Adobe provided us with a **fully anonymized dataset** that has over **137 million rows** and **5,196 columns**.

Each **row** represents a unique **uuids**, and each **column** represents a **particular trait** (binary input).

The sparsity rate of data is 0.0018, indicating that the data is **extremely sparse**.

Figure 1: Histogram of traits appearance in the data

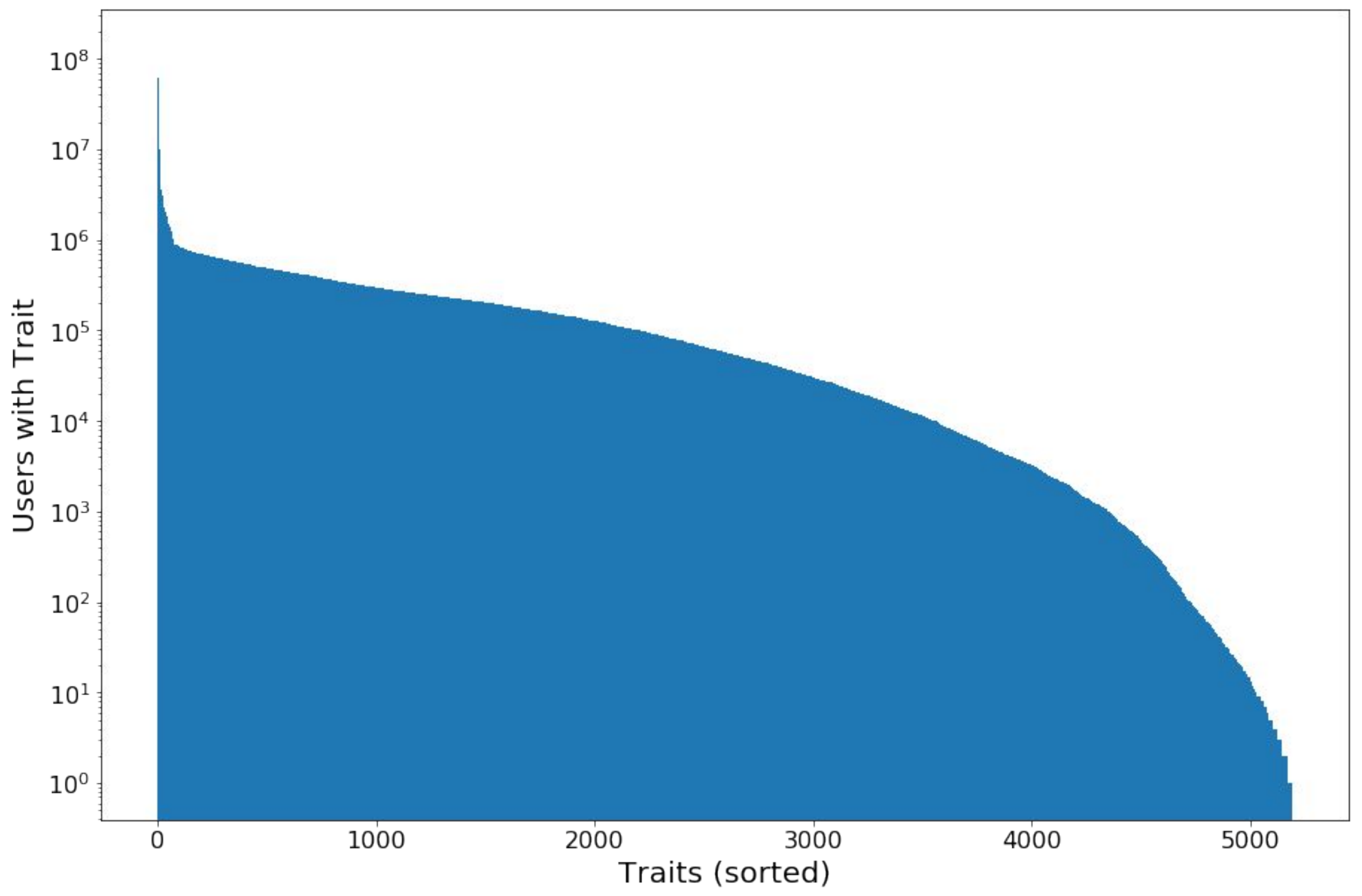
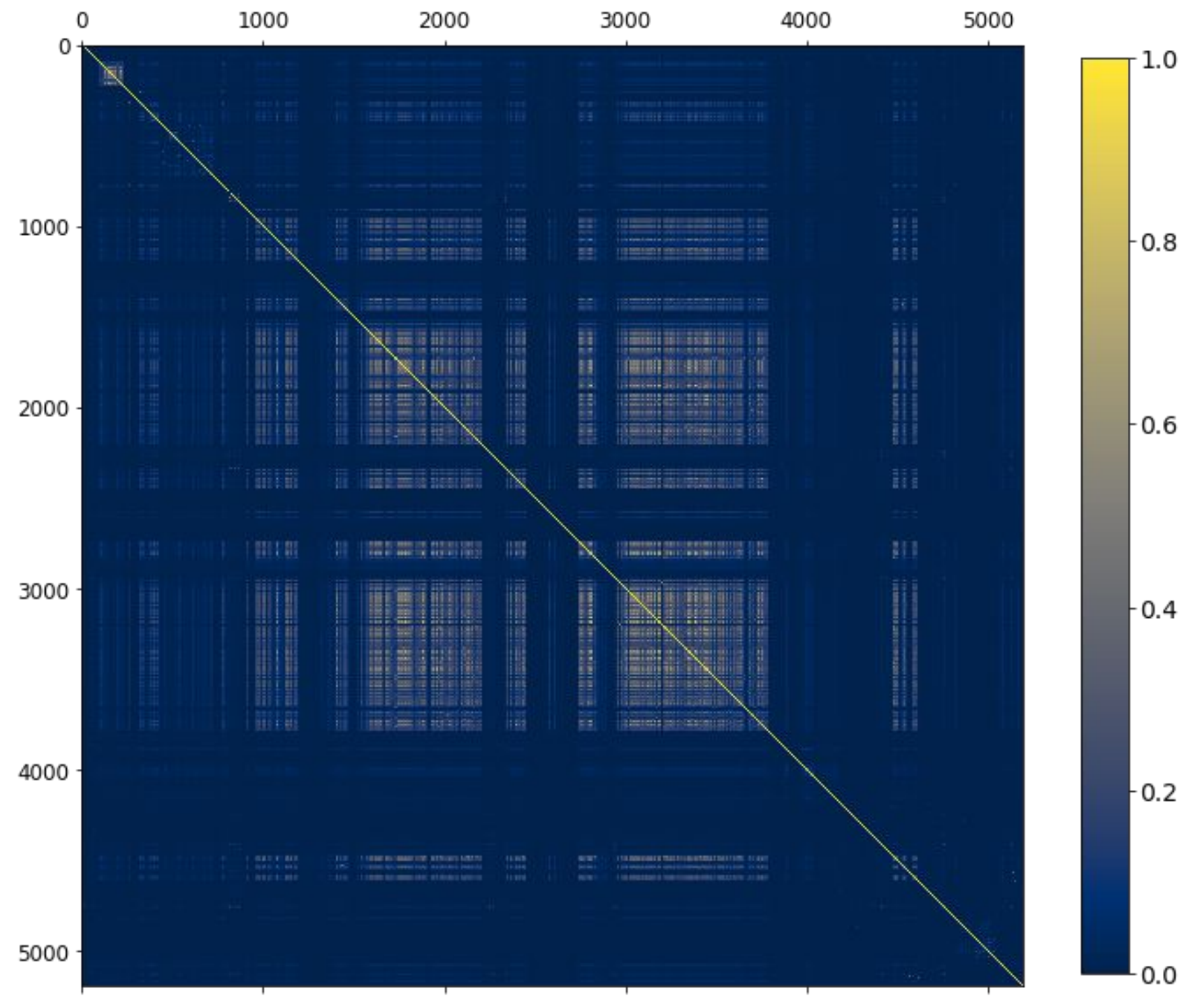


Figure 2: Jaccard Similarity Matrix between traits



Methodology

Applying NLP Models to Unary Dataset

- Our dataset has properties which are similar to text datasets
- Appealing properties of NLP models:
  - Remarkable success in practice
  - Heavy ongoing research field
  - Lend themselves to top-N recommendations

Term Frequency-Inverse Document Frequency

- Traits as words, users as documents

$$tf-idf(d,t) = tf(t) * idf(d,t)$$

$$idf(d,t) = log\frac{n}{df(d,t)}$$

Item2Vec

Barkan & Noenigstein(2016)[1] applied Word2Vec to item-based collaborative filtering.

$$\begin{aligned}\text{minimize } J &= -\log P(w_c | w_{c-m}, \ldots, w_{c-1}, w_{c+1}, \ldots, w_{c+m}) \\ &= -\log P(u_c | \hat{v}) \\ &= -\log \frac{\exp(u_c^T \hat{v})}{\sum_{j=1}^{|V|} \exp(u_j^T \hat{v})} \\ &= -u_c^T \hat{v} + \log \sum_{j=1}^{|V|} \exp(u_j^T \hat{v})\end{aligned}$$

Where u_i is the embedding of the output vector w_i' and $\hat{v}$ is the average of the embeddings of the $2m$ context words

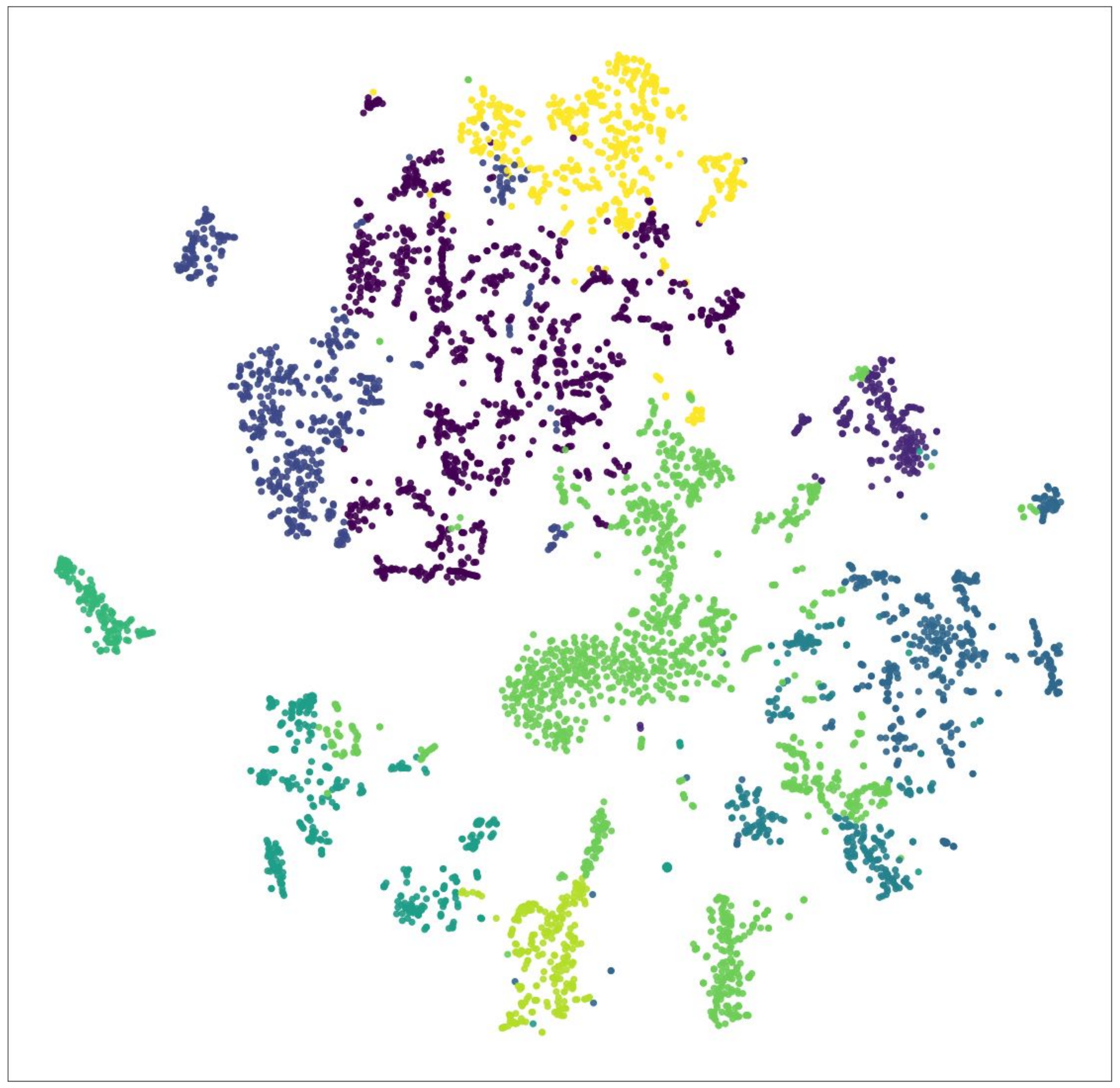
Item2Vec Algorithm

1. Build dataset containing 5 random shuffles of each row.
2. Run word2Vec on this dataset
   - vector size: 100
   - min count: 1
   - window: 8
   - negative sampling: 16
   - iterations: 20

Results and Evaluation

Figure 3: TSNE Embedding of Item2Vec Model with Agg. Clustering (n=10)

Silhouette score: 0.121 Cluster Stability: 0.597



Conclusion & Future Work

Both the models described allow the traits to be treated as a set, in NLP terms a bag of words. Hence, The results of these models, particularly the Item2Vec model, suggests that NLP can be applied with success to non-textual datasets. There has been some work on parallelization of word2vec models[2], in future we will look more closely into such models and architectures.

Acknowledgments

We would like to acknowledge our faculty mentor **Eleni Drinea** and industry mentors **Charles Menguy** and **Handong Zhao** for their support, patience, and guidance.

Reference(s)

[1] Oren Barkan & Noam Koenigstein, *Item2vec: Neural item embedding for collaborative filtering*. CoRR, abs/1603.04259, 2016.

[2] Ji S, Satish N, Li S, Dubey P. *Parallelizing word2vec in shared and distributed memory.* arXiv preprint arXiv:1604.04661. 2016.