Stack Overflow Topic Modeling and Recommendation Systems

By: Crystal Han

Abstract

The goal of this project was to use the Stack Overflow data from the Google Cloud Big Query to analyze the text, group, and recommend based on content and user. This will be good for those who search questions and want to be recommended with specific topics. The data is analyzed using three tables: the user, the questions, and the answers. The table is joined by the user id to create a single dataframe.

Design

This data for this project is provided by Google Cloud Big Query. This data presents a table of users, questions, reputation, view counts, answers, and tags. We can use the questions to go through filtering, cleaning, and grouping to understand which topics go with which and recommend based on the topic of the question. The question's main words are then put into a list that we can compare or find when looking for topics most similar to the words.

Data

The dataset contains questions, answers, and user data. Some of these features required extensive cleaning using BeautifulSoup to remove html tags, regex to filter out code and special characters and html links, and Lemmatizer to identify certain words that we want and do not want. The data can then be used in CountVectorizer or TfidfVectorizer to create our document-term matrix. Then we can pass this through to a topic model such as LDA or NMF to find our topics. Afterwards we can use metrics, such as cosine similarity, to recommend topics or questions closest to the topics/word.

Algorithms

- Latent Dirichlet Allocation (LDA)
- Non-negative Matrix Factorization (NMF)
- CountVectorizer & TFIDF Vectorizer
- DBSCAN, K-means

Feature Engineering

1. Filtering for python related questions
2. Creating Word Clouds based on topics mentioned in questions mostly
3. Plotting a 3D matrix to visualize the LDA matrix in 3 dimensions
4. Cleaning, filtering text into words that relate to the topic
5. Using DBSCAN to find clusters

Models

Word Clouds, scatterplots, and histograms were used to observe the clusters and topics that were mentioned consistently. The word clouds allow me to deduce what type of topic the cluster is mentioning. I then used the K-means algorithm to visualize a scatter plot of the cluster centers and the data points. K was chosen to be at 30. I had attempted to use DBSCAN but failed to find any useful clusters or plot the data points correctly.

Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF) were used to model these topics. K-means was also used to find clusters using the TFIDF document-term matrix. Count Vectorizer and TFIDF Vectorizer were both used for LDA to compare the differences. NMF used the countvectorizer document-term matrix as the word 'Thank' was visible in most word clouds..

Tools

- NumPy and Pandas for data manipulation and cleaning
- Matplotlib, Seaborn, Plotly for plotting and visualizations
- SQL to retrieve data from Google Cloud Big Query
- BeautifulSoup to remove html tags
- Regex to remove html links, special characters and code

- Sklearn to vectorize and find clusters

Communication

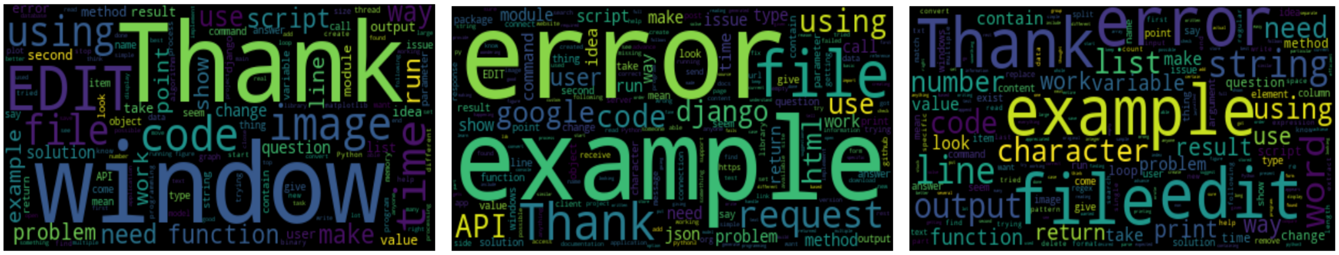

Figure 1 TFIDF Vectorizer Word Cloud using LDA

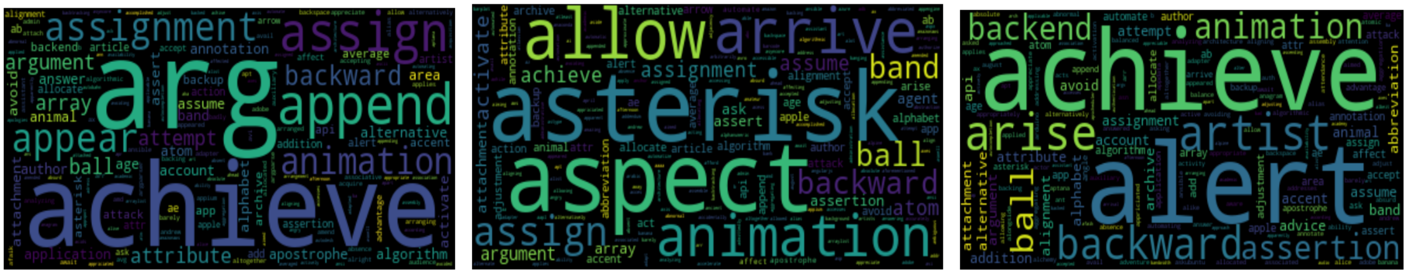

Figure 2 Count Vectorizer Word Cloud using LDA

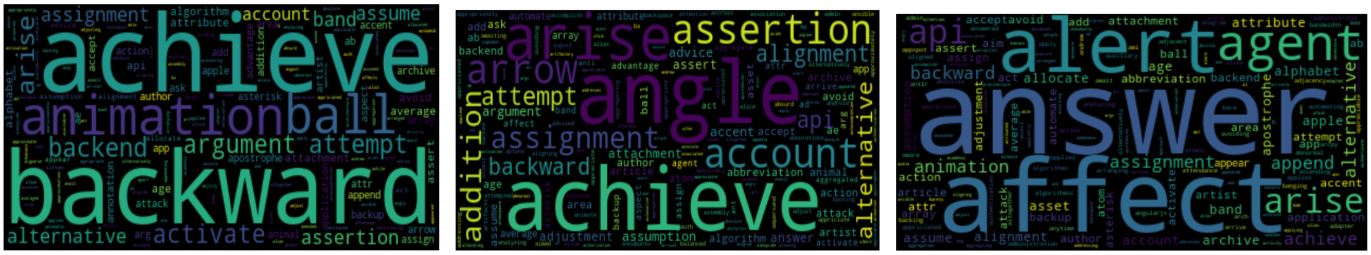

Figure 3 Count Vectorizer Word Cloud using NMF

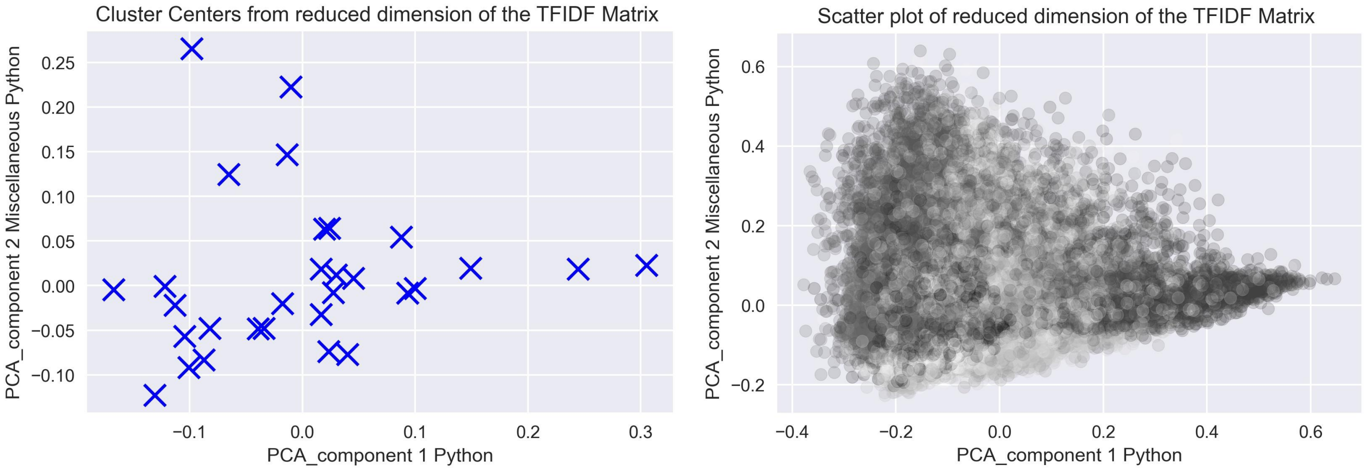

Figure 4 K-Means Clustering of Reduced Dimension of TFIDF Matrix using PCA

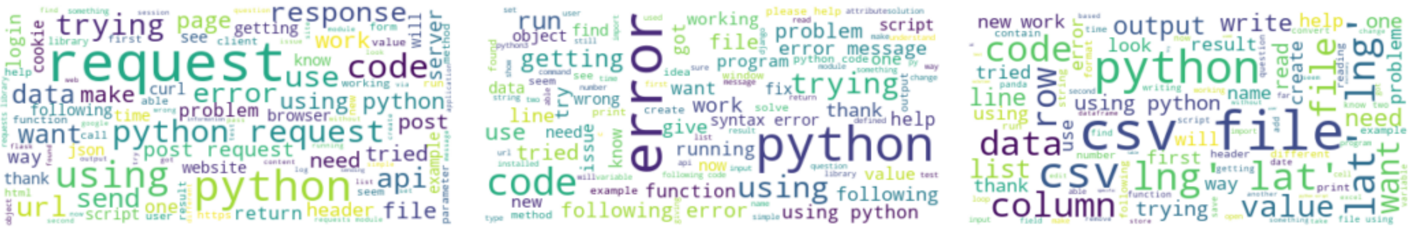

Figure 5 K-means Word Cloud of Topics

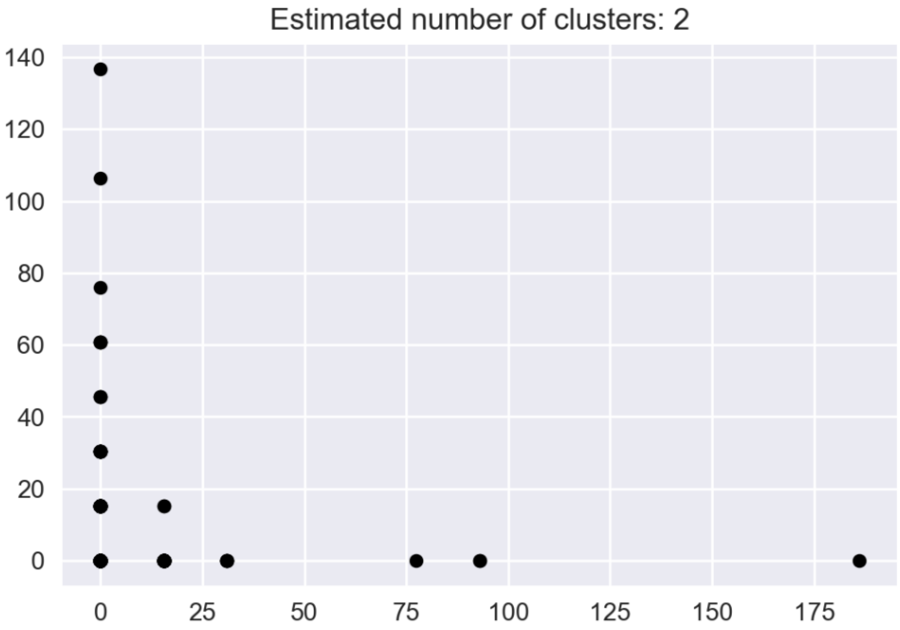

Figure 6. Failed DBSCSAN algorithm scatter plot