Domain Background

“What movie should I watch this evening?” Have you ever had to answer this question at least once when you come home from work? As for me:yes, and more than once. From Netflix to Hulu, the need to build robust movie recommendation systems is extremely important given the huge demand for personalized content of modern consumers.

An example of recommendation system is such as this:

- User A watches Game of Thrones and Breaking Bad.
- User B does search on Game of Thrones, then the system suggests Breaking Bad from data collected about user A.

Recommendation systems are used not only for movies, but on multiple other products and services like Amazon (Books, Items), Pandora/Spotify (Music), Google (News, Search), YouTube (Videos) etc.

I decided to build a full django website based on the idea of the movie recommendation systems. The website itself is not included in the scope of the project, but the movie recommendation system itself will be the intended scope in the capstone .You can read more about movie recommendation systems through this link https://medium.com/@james_aka_yale/the-4-recommendation-engines-that-can-predict-your-movie-tastes-bbec857b8223.

Problem Statement

The problem which I would like to find a solution for here in the capstone is to help users find the best suitable movie that match their taste in movies.

The solution for the problem is to build clusters that could help recommending movies for the users that exist in each cluster.

Datasets and Inputs

“The dataset(s) and/or input(s) to be used in the project are thoroughly described. Information such as how the dataset or input is (was) obtained, and the characteristics of the dataset or input, should be included. It should be clear how the dataset(s) or input(s) will be used in the project and whether their use is appropriate given the context of the problem.

The dataset which I will be using throughout the project is the movielens dataset which is presented here: https://grouplens.org/datasets/movielens/. I will use the Full Movielens dataset which is a dataset of 27,000,000 ratings and 1,100,000 tag applications applied to 58,000 movies by 280,000 users. Includes tag genome data with 14 million relevance scores across 1,100 tags. Last updated 9/2018.

Solution Statement

The solution which I will be using in the capstone project will use the existing Movielens dataset and it's ratings to create clusters of users based on their ratings to different types of movies.

The input: Should be new user ratings for different types of movies.

The processing: The model should then add the user to a predefined cluster.

The output: Should be suggestion to watch new movies.

Benchmark Model

The benchmark model which I will be looking for in the project is the K-means clustering project which was already implemented in the nano-degree on a smaller dataset. The K-means clustering shows great results in successfully build clusters of users based on their ratings to different types of movies. Instead, I will use DBSCAN to build the cluster that I would like to use in my capstone project.

Evaluation Metrics

I will be using DBSCAN clustering algorithm. The DBSCAN proved a good performance in clustering data that are graphically related/Close to each other which will be the case in the MovieLens dataset when plotting users/ratings.

For comparing my capstone's model with the benchmark models, I will use Adjusted Rand index which proves a good performance in comparing clustering models.

Project Design

I will start by parsing the Full Movielense dataset. Then build a model using the DBSCAN algorithm after deciding the best settings for the algorithm.

The current Full dataset is not very ready for such a project, so I will do extra steps in preparing the data and adjust the ratings/user to be ready for the clustering process. Then, I will start building the clusters using the adjusted K-means model.

Finally, I will use the clusters to classify the new users against the different generated clusters and after clustering, I will use the knowledge in the same cluster to make recommendations.