Inspiring Excellence

Title of the Project: Movie Recommendation system

Group Members:

Reeyad Ahmed Ornate - 23141041

Farhan Bin Bahar - 20101519

Course Name: Artificial Intelligence

Course Code: CSE422

Semester: Spring 2023

Table of Contents

Introduction

This project analyzes the relationship among movie genre, star, director, released year, production company and based on that relationship, will provide recommendations for similar kinds of movies to watch. By utilizing a dataset containing these variables for a large number of movies, we can perform statistical analysis and machine learning techniques to analyze the impact of a movie's elements for the recommendation system. The analysis can also provide us with insights of which movies are similar to each other and in terms of what. For this, we will be utilizing KNN and Vectorization models on two publicly available dataset containing required features. After training these models, we will perform supervised learning tests and generate recommended movies based on the movie name taken as an input from the user.

Dataset description

- **Source**
  - Link:
    - Movie Industry
    - TMDB 5000 Movie Dataset
  - Reference:
    - Movie Industry: Kaggle Link
    - TMDB 5000 Movie Database: Kaggle Link
- **Dataset Description**
  - **Movie Industry:**
    - Features: 4
    - Classification problem: We are generating similar movies based on categorical data.In classification problem we predict discrete values and predicted names of recommended movies are considered as discrete values.
    - Data points: 5433
    - Categorical

- Correlation of the features(quantitative values were dropped) with label/class:

- **TMDB 5000 Movie Database:**
  - Features: 23
  - Classification problem: same as the previous dataset
  - Data points: 5000
  - Categorical features

Imbalanced Dataset

- Output is Unbalanced as the dataset has an uneven distribution of instances across columns which indicates that some classes are underrepresented while others are overrepresented.

Dataset pre-processing

- **Movie Industry:**

    - Faults:
        - Null Value: Null Value: The dataset has null values for rating, released, score, votes, writer, star, country, budget, gross, company and runtime. These rows with null values are not needed for statistical analysis and machine learning models.

    - Solutions:
        - Delete rows: Rows with null values in features and classes do not aid our machine learning model. Therefore, dropping the rows containing null values for the features and classes.
        - Delete columns: There are some unnecessary columns that we do not need to train our models. Therefore, dropping the columns that are not used for our model training such as rating, released, country, writer, votes, runtime, budget, gross.

    - Faults:
        - Categorical: Genre, director, star, company.

- Solutions:
    - Used one-hot encoding on genre, director, star and company as they are categorical values but machine learning models work with numerical values.

- **TMDB 5000 Movie Database:**
    - Faults:
        - Null Value: The dataset has null values for the columns titled “homepage”, “overview”, “release_date”, “runtime” and “tagline”. Not all of these columns with null values are not needed for statistical analysis and machine learning models. However, one of them, titled “homepage” has a significant impact on the whole dataset.
    - Solutions:
        - Delete Columns: There are some unnecessary columns that we do not need to train our models. Therefore, dropping the columns that are not used for our model training such as budget, original_language, original_title, popularity, production_companies, production_countries, release_date, revenue, runtime, spoken_languages, status, tagline, title, vote_average, vote_count, movie_id.
        - Fill Columns: In some cases, the significance of the data is so big that deleting that feature will result in very inaccurate output. For these cases, we can fill the null values with temporary readable data.

Feature scaling(as required)

- The features that we used have varying scales. Here, large scaled features can dominate small scale features which can lead to poor model training and poor performance. That is why we used standardization where features transform to have 0 mean and unit variance

and min-max scaling where features are scaled to specific ranges which is in our case 0 to 1(range).

Dataset splitting

- To train the models we used train_test_split function and assigned test_size = 0.3 and random_state = 1. Basically what it does is assigns 30% of the rows for testing and 70% for training. Also the random_state = 1 produces different set of random numbers which randomizes the test and training set everytime
    - Random: 1
    - Train set: 70%
    - Test set: 30%

Model training & testing

- **KNN:** KNN stands for K-Nearest Neighbor. This algorithm finds the K number of closest neighbors/ data points in the training set and assigns the class of the majority of the K neighbors to the new data point. As our recommendation system is a classification problem, we used the KNN model.
- **Vectorization:** We also used the Count vectorization and space vectorization/ Cosine similarity model in order to find the optimal solution to our problem.

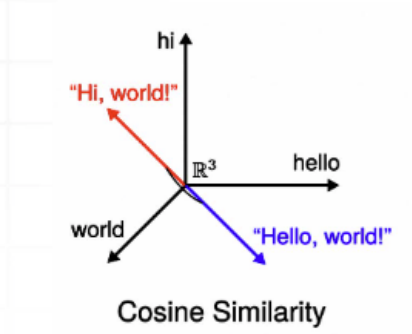
  Count Vectorization:

  Count Vectorization is a specific type of vectorization method in which a numerical value is assigned to every word so that they can be used in various numerical functions. For example, in our case, we used count vectorization to designate a numerical value to every word in the column termed as “all tags” of the “TMDB 5000 Movies” dataset. As can be seen below, count vectorization observes the number of times a word has been mentioned, and assigns a numerical value based on that.

Hero	World	Fear	Dominate	Virus	Hilarious
2	4	1	3	1	2

Space Vectorization:

On the other hand, the Cosine similarity method of space vectorization is a method of finding the similarity between two vectors. It can be used to find the similarity between the characteristics of two subjects. This is done so by drawing a 2D graph of the two vectors that are represented by the subjects. The angle θ works as the variable here. The smaller the value of theta is, the more similar the two subjects are. In our case, we used the cosine similarity model to find the similarity between two movies. And based on the similarity level, we recommend the top 5 most similar movies to the user.



Cosine Similarity

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

Model Selection

Based on the test results of our recommendation system, we selected KNN model as our go to model for this classification problem as it generated the best recommendations.

Conclusion

Based on our research and approach to the recommendation system, we concluded that the KNN model was our best choice in order to approach this problem as it is a classification problem. We faced some problems with training the models as it was required to be encoded and the train set had some missing variables which were present in the test set. Therefore, we were required to skip those data by ignoring the abnormalities. There is also the possibility that our approach to the project idea using the procured dataset was not the most optimal one which resulted in lower accuracy.