Movie Recommendation System

Group name: By a time zone united
Name: Tehesuma Mensah Imoro
Email: mktehesuma24@gmail.com
Country: Ghana
College/Company:
Specialisation: Data Science
Link:
Github repo: https://github.com/Tehesuma007/Week-8-Deliverables

Problem Description

The video-on-demand streaming service is looking to develop a machine learning algorithm that can predict which movies a user will enjoy based on various factors such as genre, online ratings, and previous decisions. The primary objective is to create a system for movie recommendations.

Data Understanding

Dataset Overview

This dataset describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. It contains 33832162 ratings and 2328315 tag applications across 86537 movies. 330975 users created these data between January 09, 1995 and July 20, 2023. This dataset was generated on July 20, 2023.

Users were selected at random for inclusion. All selected users had rated at least 1 movie. No demographic information is included. An ID represents each user, and no other information is provided.

Data Description

The data is contained in the files genome-scores.csv, genome-tags.csv, links.csv, movies.csv, ratings.csv and tags.csv. The dataset files are written as comma-separated values files with a single header row. Columns that contain commas (,) are escaped using double-quotes ("). These files are encoded as UTF-8.

Ratings Data File Structure (ratings.csv):

userId, numeric - represents anonymised user IDs.
movieId, numeric - represents IDs used on the MovieLens website (e.g., id 1 corresponds to the URL https://movielens.org/movies/1).
rating, numeric - represents ratings made on a 5-star scale, with half-star increments (0.5 stars - 5.0 stars).
timestamp, numeric - represents seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970.

Tags Data File Structure (tags.csv):

userId, numeric - represents anonymised user IDs.
movieId, numeric - represents IDs used on the MovieLens website (e.g., id 1 corresponds to the URL https://movielens.org/movies/1).
tag, text - represents user-generated metadata about movies. Each tag is typically a single word or short phrase.
timestamp, numeric - represents seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970.

Movies Data File Structure (movies.csv):

movieId, numeric - represents IDs used on the MovieLens website (e.g., id 1 corresponds to the URL https://movielens.org/movies/1).
title, text - represents names of films and includes the year of release in parentheses.
genres, categorical - represents corresponding to a film genres in a pipe-separated list.

Links Data File Structure (links.csv):

movieId, numeric - represents IDs used on the MovieLens website (e.g., id 1 corresponds to the URL https://movielens.org/movies/1).
imdbId, numeric - represents an identifier for movies used by http://www.imdb.com. E.g., the movie Toy Story has the link http://www.imdb.com/title/tt0114709/.
tmdbId, numeric - represents an identifier for movies used by https://www.themoviedb.org. E.g., the movie Toy Story has the link https://www.themoviedb.org/movie/862.

Genome Scores Data Structure (genome-scores.csv):

movieId, numeric - represents IDs used on the MovieLens website (e.g., id 1 corresponds to the URL https://movielens.org/movies/1).
tagId, numeric - represents generated IDs for tags.

relevance, numeric - represents scores which show how strongly movies exhibit particular properties represented by tags.

Genome Tags Data Structure (genome-tags.csv):

tagId, numeric - represents generated IDs for tags.
tag, text - provides the tag descriptions for the tag IDs in the genome file.

Data Quality Assessment

Evaluating and validating our data

1. Missing Values (MVs):
   - Most of the datasets (genome-scores.csv, genome-tags.csv, movies.csv, and ratings.csv) have no missing values, indicating completeness in the majority of the data.
   - Tags.csv has 17 missing values in the tag column. While this is a small proportion of the total data, it's still important to handle these missing values appropriately.

2. Duplicate Rows:
   - There are no duplicate rows in any of the datasets, which ensures data integrity and avoids redundancy in the analysis.

3. Outliers:
   - Genome-scores.csv has over 1,000,000 outliers in the relevance column. Outliers in this context could indicate extreme scores that deviate significantly from the norm. Further investigation may be necessary to understand the nature of these outliers and determine whether they should be treated or excluded in the analysis.
   - Ratings.csv also has over 1,000,000 outliers in the rating column. These outliers may represent extreme ratings that are significantly higher or lower than the typical ratings in the dataset. Handling outliers in rating data is crucial for accurate analysis and modeling.

4. Unused Columns:
   - Links.csv has 126 missing values in the tmdbId column, but since this column is not needed for the analysis, it won't impact the overall quality of the analysis. This missing data can be safely ignored.

Overall, while the datasets are generally complete and free from duplicates, the presence of outliers in both the genome-scores and ratings datasets may require careful consideration and preprocessing before conducting further analysis or modeling. Additionally, handling missing values in the tags dataset is necessary to ensure robustness in the analysis of user-generated metadata.

Extent of Data Quality Problems and Potential Impact on Analysis:

1. Missing Values:
   - Missing values in the `tmdbId` column of `links.csv` (126 missing values) are not expected to impact the analysis since this column is not required for the current analysis.

- However, missing values in the `tag` column of `tags.csv` (17 missing values) might affect tag-based analyses or recommendation systems relying on user-generated metadata. It could potentially lead to incomplete or biased results in analyses involving tags.

2. Duplicates:
- The absence of duplicate rows in all datasets indicates good data integrity, reducing the risk of biased analyses or misleading results.

3. Outliers:
- With over 1,000,000 outliers in the `relevance` column of `genome-scores.csv` and `rating` column of `ratings.csv`, there could be a significant impact on analysis:
    - Outliers in `genome-scores.csv` might skew the measurement of how strongly movies exhibit particular properties represented by tags. This could lead to inaccurate assessments of movie characteristics or incorrect recommendations based on these scores.
    - Outliers in `ratings.csv` could distort the understanding of user preferences and affect the accuracy of recommendation systems. They might also influence statistical analyses and modeling efforts, potentially leading to biased results or unreliable predictions.

4. Tag Column Missing Values:
- Missing values in the `tag` column of `tags.csv` could affect analyses relying on user-generated metadata:
    - Recommendations based on tags may be incomplete or biased, as movies with missing tags may not be appropriately represented in the analysis.
    - Tag-based clustering or classification analyses might produce skewed or inaccurate results due to the absence of tags for certain movies.

Overall Impact:
- The quality issues identified in the dataset, particularly outliers and missing values, have the potential to significantly impact the accuracy, reliability, and interpretability of analyses and models.
- Failure to address these issues adequately could lead to biased recommendations, inaccurate insights, and unreliable predictions, undermining the effectiveness and trustworthiness of any derived systems or findings.
- Therefore, thorough data cleaning, outlier detection, and missing value imputation or handling strategies are essential to mitigate these issues and ensure the integrity of subsequent analyses.

Note: genome-scores.csv, genome-tags.csv, movies.csv and ratings.csv have no missing values(mvs). Links.csv has 126 mvs in the tmdbId column but we don't need this column(and file overall) in our case so it won't have an impact on the analysis. Tags.csv has 17 mvs in the tag column and probably we can simply remove these observations(??).

Data Preprocessing Requirements

After observing and analysing the datasets we will be using for this project, we will need to do some data cleaning and feature engineering. Below are some preprocessing tasks we will be performing:

• The timestamp values are not in their right format and will have to be converted to the appropriate datetime format. We will do this by importing the datetime library and using the fromtimestamp() function to convert our timestamp to a datetime object data type.

• We will be creating a new dataframe by appending, joining or merging our existing datasets. This will be our final dataframe and the data we are going to use to train our model.

• There is some categorical data in our dataframes. We will be converting them to numeric data using one_hot encoding or label_encoding. This is important because our training data requires numeric data.

• Missing values in relevant columns will be filled using the median value of the column it is located. This will be done by creating a function to replace all missing values with the median value of their respective columns. Also, we will be using the .dropna() to just drop rows with missing values in a specific column

• If any outliers that would have a negative impact on our work is detected, we will immediately handle them. We will handle them in the following ways after assessing their impact on our data and our project requirements:

a. We will create a function to return a dataframe with the outliers removed. Inside this function we will create a dataframe that replaces the outlier values with a NULL. Then we will use .dropna(), to drop the rows with NULL values.
b. Also, if we used the imputation method to fill our missing values, we can use this same method to handle outliers.

Exploratory Data Analysis (EDA) Plan

1. Data Overview:
• We will begin by loading the dataset and examining its structure i.e. the number of rows, columns, and data types.
• We will check for missing values and duplicates in each column of relevant dataframes and devise a strategy for handling them
• We will identify unique values for categorical variables to understand their range and distribution.

2. Univariate Analysis:
• For numeric data:
a. We will perform summary statistics: by checking the values of the mean, median, mode, range, variance, and standard deviation.
b. We will visualise distributions using histograms, box plots, and kernel density plots.
• For categorical data:
a. We will explore frequency distributions using bar plots.
b. We will calculate proportions and percentages for each category.

3. Bivariate Analysis:
- Here, we will explore relationships between pairs of variables:
• For numeric-numeric pairs:
a. We will calculate correlation coefficients (Pearson, Spearman) and we will perform visualisations using scatter plots.
b. We will use heatmaps to visualise correlation matrices.
• For numeric-categorical pairs:
a. We will compare the distribution of numeric variables across different categories using box plots or violin plots.
• For categorical-categorical pairs:
a. We will create contingency tables and perform chi-square tests for independence.
b. We will visualise using stacked bar plots or mosaic plots.

4. Data Visualization:
Here we are going to use interactive visualisations to explore complex relationships and patterns in the data:
• We are going to utilise libraries like matplotlib, seaborn for standard visualisations and plotly or bokeh for interactive plots.

5. Documentation and Reporting:
• We will document findings, insights, and decisions made during the EDA process.

6. Iterative Process:
• EDA is an iterative process, so we will revisit earlier steps as needed based on insights gained later in the analysis.
• We will continuously refine analysis techniques and explore alternative visualisations to gain a deeper understanding of the data.

Proposed EDA and Visualization Techniques

We will be utilising the following techniques and libraries for our EDA and Visualization for general and case-specific analysis:
• General Exploratory Data Analysis:
a. We will use the isna().sum() to check for missing values
b. We will create a function to find duplicates or utilise Microsoft Excel to check for duplicates
c. We will check for unique values using the nunique() function.

• Univariate Analysis:
a. We will conduct summary statistics using the describe() function.
b. We will use visualisations like boxplots, histograms, etc to check for the distribution of our data and to also check for outliers.
c. We will also create functions for detecting outliers and handling them
d. We will explore frequency distributions, calculate proportions and percentages using barplots, pie charts, etc

• Bivariate Analysis:
a. We will explore relationships between variables using scatter plots

b. We will create correlation matrices and use heatmaps to visualise them
c. We will compare the distribution of numeric variables across different categories using box plots or violin plots.
d. We will create pair plots to visualise relationships between two features.

• Data Visualization:
For our data visualisation, we are going to use the following libraries;
matplotlib, seaborn, plotly and bokeh to create visually appealing and interactive visualisations.

Data Understanding Report

Comprehensive Report on MovieLens Dataset

Introduction:
The MovieLens dataset provides comprehensive information on user ratings, tagging activity, and movie metadata, facilitating various analyses and recommendation system development. This report provides an overview of the dataset structure, contents, and potential applications.

Dataset Overview:
- Ratings Data (ratings.csv): Contains 33,832,162 ratings provided by 330,975 users for 86,537 movies. Ratings range from 0.5 to 5.0 stars.
- Tags Data (tags.csv): Consists of 2,328,315 tag applications by users for movies. Each tag typically represents user-generated metadata about movies.
- Movies Data (movies.csv): Provides information about 86,537 movies, including movie ID, title, release year, and genres represented in a pipe-separated list.
- Links Data (links.csv): Links movie IDs to IMDb and TMDb identifiers, facilitating external data linkage.
- Genome Scores Data (genome-scores.csv): Contains genome scores indicating how strongly movies exhibit particular properties represented by tags.
- Genome Tags Data (genome-tags.csv): Provides descriptions for the tag IDs used in the genome scores file.

Data Description:
- Users were randomly selected for inclusion, and each selected user has rated at least one movie.
- The dataset spans from January 09, 1995, to July 20, 2023.
- No demographic information about users is included.
- Data files are in CSV format with UTF-8 encoding.

Potential Applications:
1. Recommendation Systems: Ratings and tags data can be used to develop personalized movie recommendation systems based on user preferences and movie characteristics.
2. Content Analysis: Movie genres, tags, and genome scores are analyzed to understand trends and patterns in movie preferences.

3. Predictive Modeling: Predictive models are built to forecast user ratings for unseen movies based on historical rating patterns and movie features.
4. External Data Integration: Movie IDs can be linked to external databases such as IMDb and TMDb for enriching the dataset with additional metadata.
5. User Engagement Analysis: User tagging behavior and rating patterns are studied over time to understand user engagement dynamics.

Conclusion:
The MovieLens dataset offers a rich resource for exploring user preferences, movie characteristics, and recommendation system development. Its comprehensive nature and large-scale data make it suitable for various research and application domains within the field of recommender systems and movie analytics.

Data Documentation

Data Source:
The data is sourced from MovieLens, accessible at www.movielens.org/movies. MovieLens is a movie recommendation service that collects and provides movie ratings and metadata.

Data Collection Methods:
The datasets were obtained by downloading from the MovieLens website. The datasets include genome-scores.csv, genome-tags.csv, movies.csv, ratings.csv, tags.csv, and links.csv. These datasets contain information on user ratings, movie metadata, tags, and genome scores.

Data Quality Assessment:

1. Missing Values (MVs):
  - Most datasets (genome-scores.csv, genome-tags.csv, movies.csv, and ratings.csv) exhibit no missing values, indicating completeness in the majority of the data.
  - Tags.csv contains 17 missing values in the tag column. While this represents a small proportion of the total data, it's important to handle these missing values appropriately to ensure robustness in the analysis of user-generated metadata.

2. Duplicate Rows:
  - There are no duplicate rows in any of the datasets, ensuring data integrity and avoiding redundancy in analysis.

3. Outliers:
  - Genome-scores.csv has over 1,000,000 outliers in the relevance column, potentially indicating extreme scores deviating significantly from the norm. Further investigation may be necessary to understand these outliers' nature and determine appropriate preprocessing steps.
  - Ratings.csv also contains over 1,000,000 outliers in the rating column, which may represent extreme ratings significantly higher or lower than typical ratings. Handling these outliers is crucial for accurate analysis and modeling of user preferences.

4. Unused Columns:

- Links.csv has 126 missing values in the tmdbId column. However, since this column is not needed for the analysis, its missing data does not impact the overall quality of the analysis and can be safely ignored.

Overall Assessment:
While the datasets are generally complete and free from duplicates, the presence of outliers in both genome-scores and ratings datasets requires careful consideration and preprocessing before conducting further analysis or modeling. Additionally, handling missing values in the tags dataset is necessary to ensure robustness in the analysis of user-generated metadata.