Predicting Movie Ratings on Netflix Using K-Nearest Neighbours Regression

Shyra A, Author

Abstract—This paper examines the application of the K-Nearest Neighbors (KNN) regression algorithm to predict movie ratings in the Netflix Prize dataset. The dataset includes user ratings along with movie metadata. Through preprocessing, feature engineering, and model optimization via grid search, the study evaluates how the choice of hyperparameters affects prediction accuracy. The results demonstrate the ability of KNN to predict ratings and highlight key performance metrics.

I. INTRODUCTION

Netflix, a leading streaming platform, has vast amounts of user-generated ratings for movies and TV shows. Predicting these ratings can improve recommendations and user engagement. This paper uses the Netflix Prize dataset to build a predictive model using the K-Nearest Neighbors (KNN) regression algorithm. We apply data preprocessing, feature engineering, and perform hyperparameter tuning via grid search to improve the model's accuracy. The paper also discusses the results of model evaluation using various regression metrics.

II. METHODOLOGY

A. Dataset

The data for this study is sourced from the Netflix Prize competition, which consists of movie ratings by users. The key data files used in this paper include:

- combined_data_1.txt, combined_data_2.txt, combined_data_3.txt, combined_data_4.txt: These files contain user ratings, movie IDs, rating values, and the timestamp of each rating.
- movie_titles.csv: This file contains metadata about movies, including movie IDs, release years, and titles.

B. Loading and Merging the Data

The dataset is loaded into a pandas DataFrame, and the following steps are carried out:

1. Parsing Data:
    - The ratings data is processed by reading each line and extracting movie IDs, user IDs, ratings, and the date of the rating.
    - The date column is converted into a datetime type to enable easy extraction of features like the year, month, and day.
2. Combining Data:
    - The movie metadata from movie_titles.csv is merged with the ratings dataset based on the movie_id. Missing values are removed to ensure clean data for model training.

C. Feature Engineering

Several new features are created:

1. Categorical and Numerical Features:
    - LabelEncoder is used to convert user_id and movie_id into categorical variables.
    - The date column is converted into a datetime type to enable easy extraction of features like the year, month, and day.
2. Combining Data:

- The movie metadata from movie_titles.csv is merged with the ratings dataset based on the movie_id. Missing values are removed to ensure clean data for model training.

Exploratory data analysis (EDA) is conducted to understand the data's structure and properties:

1. Categorical and Numerical Features:
    - The columns in the dataset are analyzed to identify which are numerical and which are categorical.
    - The frequency of ratings by users and movies is calculated.
2. Selection of Active Users and Movies:
    - The top 5000 users (based on the number of ratings) and the top 1000 most rated movies are selected to focus on high-quality data.
    - Ratings for these users and movies are merged to create a refined dataset.

III. MODEL BUILDING

A. Data Splitting

The dataset is split into training and testing sets using train_test_split, with 80% of the data for training and 20% for testing.

B. Training the KNN Model

The K-Nearest Neighbors algorithm is used for the regression task. The model is initialized with n_neighbors = 5 (default). Predictions are based on the 5 nearest neighbors in the feature space.

C. Hyperparameter Optimization via Grid Search

Hyperparameter tuning is performed using GridSearchCV, testing the following parameters:

- n_neighbors: Various values (5, 9, 17).
- weights: Different weighting strategies ('uniform', 'distance').
- p: The distance metric (1 for Manhattan, 2 for Euclidean distance).

A 5-fold cross-validation approach is used to find the optimal hyperparameters. The grid search yielded the following best parameters:

- Best Hyperparameters: n_neighbors = 17, p = 1, weights = 'uniform'
- Best Cross-Validation Score (negative mean squared error): -1.0715653058012764

D. Model Evaluation

After training the model with the optimal hyperparameters, we evaluate the performance on the test set. The performance metrics are:

- Mean Squared Error (MSE): 1.0695405387987844
- Mean Absolute Error (MAE): 0.8312078484520644
- R^2 Score: 0.06207947475839404

These metrics suggest that while the KNN model can predict Netflix movie ratings, there is room for improvement, especially in explaining variance (low R^2 score).

IV. RESULTS AND DISCUSSION

A. Best Model Hyperparameters

The grid search results indicate the following best hyperparameters:

- n_neighbors: 17
- p: 1 (Manhattan distance)
- weights: 'uniform'

These hyperparameters resulted in the best cross-validation score and model performance, but the performance on the test set still left room for improvement.

B. Evaluation Metrics

The final evaluation metrics for the KNN model on the test set are:

- MSE: 1.0695

- MAE: 0.8312
- R^2: 0.0621

While the model has a relatively low R^2 score, indicating limited ability to explain the variance in the ratings, the MAE and MSE suggest that it can still make reasonably accurate predictions.

V. CONCLUSION

This study demonstrated the use of K-Nearest Neighbors (KNN) regression to predict movie ratings using the Netflix Prize dataset. By optimizing the model's hyperparameters via grid search, the best-performing model was identified. Although the R^2 score suggests that the model could benefit from further refinement, the achieved performance indicates that KNN is a useful method for movie rating prediction. Future work could explore incorporating additional features, such as movie genres or demographic information, to improve model accuracy.

VI. REFERENCES

[1] Netflix, "Netflix Prize Dataset," 2017. [Online]. Available: https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data

[2] Harrison, Onel. "Machine Learning Basics with the K-Nearest Neighbors Algorithm" 2018. [Online]. Available: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

[3] Patil, Prasad. "What is Exploratory Data Analysis?" 2018. [Online]. Available: https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15