DATA6100 Final Project

Sun Zhengxiao & Zhou Huiming

December 2024

1 Executive Summary

For our final project, we chose the Movie Dataset task. In this task, the dataset consists of user ratings for various movies on a 0-5 star scale. Our goal is to build a model that can predict a user's rating for a specific movie based on existing information, such as the user's ratings for other movies and their associated genres. The following summary outlines the steps we took to complete this project.

2 Data Pre-Processing

After importing all the required libraries and datasets, we performed data wrangling to clean and prepare the data for modeling. First, we combined the training and testing datasets and added a column to label the data source for each row. We observed that the genres in the dataset were separated by the — symbol. Therefore, we split the genre column and applied One-Hot Encoding to represent each genre, as there were only 19 unique genres. Finally, we split the combined dataset back into the training and testing sets.

We initially planned to use forward and backward stepwise selection for feature engineering. However, after testing both methods, the selected features did not significantly improve model performance. As a result, we decided to exclude feature engineering from the project.

3 Modelling

We split the training dataset into 80% training and 20% validation sets. We then trained three models: **Logistic Regression**, **Random Forest**, and a **Neural Network**.

3.1 Logistic Regression

Since the target variable in this project is a continuous rating (e.g., 0.0, 0.5, 1.0, 1.5, etc.), Logistic Regression, which is designed for discrete outcomes, posed

a challenge. To address this, we multiplied each rating by 10 to convert them into integers. After making predictions, we divided the results by 10 to obtain the final ratings. However, this approach yielded poor performance compared to the other two models, and we ultimately decided to abandon the Logistic Regression model.

3.2 Random Forest

To perform an outstanding Random Forest, we firstly use grid search to find the best tuning parameters. Due to limitations of the computer, we first figure out the best combo of number of iterations and number of variables, and then use the optimized `n_estimators` and `max_features` to find the suitable `max_depth` and `min_sample_split`. The best parameters are 500 iterations, 60% features, maximum 10 levels decisions and minimum 5 sample splits.

We then train the model with the chosen parameters on the training datasets with `random_state = 42` for reproduction. Using the model for prediction provides us the test set RMSE = 93.

3.3 Neural Network

We first define `init_params`, `forward`, loss function and `update` with reference to the code on the course GitHub. For the loss function, we simply use the MSE, and for the forward function, we use `tanh` as the activation function. While updating, we use gradient descent and an L2-penalty optimizer to prevent overfitting. Additionally, we apply gradient clipping to ensure that the gradients do not explode, which can lead to unstable training. Gradient clipping helps maintain the stability of the training process by capping the gradients to a specified range.

After defining the main functions, we tried adding hidden layers and increasing the number of epochs for better performance. We also added early stopping to prevent overfitting when there is no improvement in the validation loss. Furthermore, we standardized the data for better performance, which finally provided us with a test RMSE = 96.