MovieLens Project

Andreea Ariadna Szilagyi

March 9, 2020

PROJECT OVERVIEW:

Creating a movie recommendation system using the MovieLens dataset.

The MovieLens data to be used, a database with over 20M ratings for over 27K movies by more than 138K users, was generated by the GroupLens research lab.

The MovieLens data will be downloaded and the code will be run as provided, in order to generate the datasets.

The goal of the project is to train a machine learning algorithm on the datasets by using the inputs in one subset to predict movie ratings in the validation set.

RMSE (Root Mean Square Error) will be used to evaluate how close the predictions are to the true values, because RMSE is a standard way to measure the error of a model in predicting quantitative data.

METHOD:

The model below uses regularization to estimate the movie and the user effects.

It computes regularized estimates in order to eliminate large errors that can increase the RMSE.

The outcomes have different sets of predictors, and the model uses the following:

average_rating = the averate rating in the train dataset

movie_effect = the movie-specific bias

user_effect = the user-specific bias

lambda is a tuning parameter and the prediction model uses cross-validation to choose it.

The movie rating predictions are compared to the true ratings in the validation set using RMSE.

CODE:

```r
RMSE <- function(true_rating, predicted_rating) {
  sqrt(mean((true_rating - predicted_rating) ^ 2))
}

lambdas <- seq(0, 10, 0.25)
rmses <- sapply(lambdas, function(lambda){
  average_rating <- mean(edx$rating)

  movie_effect <- edx %>%
    group_by(movieId) %>%
    summarize(movie_effect = sum(rating - average_rating) / (n() + lambda))

  user_effect <- edx %>%
    left_join(movie_effect, by = "movieId") %>%
    group_by(userId) %>%
    summarize(user_effect = sum(rating - movie_effect - average_rating) / (n() + lambda))

  predicted_rating <- validation %>%
    left_join(movie_effect, by = "movieId") %>%
    left_join(user_effect, by = "userId") %>%
    mutate(predicted = average_rating + movie_effect + user_effect) %>%
    .$predicted

  return(RMSE(predicted_rating, validation$rating))
})

RMSE <- min(rmses)
RMSE
```

```
## [1] 0.864817
```

RESULTS:

The obtained value of the RMSE is 0.864817.

As per the definition:

"Regularization permits us to penalize large estimates that are formed using small sample sizes."

"The general idea behind regularization is to constrain the total variability of the effect sizes."

With the above model that uses regularization, we get to shrink deviations from the average towards 0, thus obtaining a lower value of the RMSE, meaning a lower typical error we make when predicting a movie rating, and therefore a higher model performance.

CONCLUSION:

The goal was to find an algorithm that produces predictors for an outcome that minimize the mean squared error.

For each set of algorithm parameters being considered, the MSE was estimated, then the parameters were chosen with the smallest value.

Cross-validation provided this estimate.

Root mean squared error (RMSE) was used to quantify the results, to see how well the ratings were predicted.

In terms of RMSE, the lower the better, therefore the 0.864817 obtained value depicts a very good model.