K Nearest Neighbors Recommender System

Michael Carlstrom
Case Western Reserve University
United States
rmc170@case.edu

1 INTRODUCTION

This paper discusses a recommender system for the MovieLens one-hundred thousand dataset and evaluates its performance. In this paper we will discuss K Nearest Neighbors a type of collaborative filtering by calculating the accuracy and root mean squared error.

2 METHODOLOGY

2.1 Data Preprocessing

First we started by applying normalization on the features for improved performance and accuracy. This is done by preventing features with large value ranges from overpowering features with small value ranges by giving them uniform value ranges from 0 to 1. To prevent over fitting we used 5 folds while training and evaluating K Nearest Neighbors.

2.1.1 *Normalization.*

For our data normalization we used Min-Max Scaling.

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

After using Min-Max Scaling we applied Z-score Normalization.

$$Z = \frac{x - \mu}{\sigma}$$

2.2 Evaluation Metrics

2.2.1 *Confusion Matrix.*

A confusion matrix is a table that summarize a classification models performance. Confusion matrices can be used to calculate all sorts of relevant metrics for classification including accuracy, precision, recall, F1 score, etc...

- TP (True Positives): Correct positive predictions
- TN (True Negatives): Correct negative predictions.
- FP (False Positives): Incorrect positive predictions.
- FP (False Negatives): Incorrect negative predictions.

Confusion Matrix	Positive	Negative
Positive	TP	FN
Negative	FP	TN

2.2.2 *Accuracy.*

Accuracy can be calculated from a confusion matrix with the following formula. We use scikit-learn's accuracy_score() implementation for calculating. Since the output of the used reccomender system is continuous we rounded the output to the nearest class for evaluation.

$$Acc = \frac{TP + TN}{TP + FP + FN + TN}$$

2.2.3 *Root Mean Squared Error.*

Root Mean Squared Error (RMSE) provides an estimation of model performance like accuracy but, is more useful for various classification models including reccomender systems. This is because the output of reccomender systems are continuous and RMSE can be calculated on these continuous output.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\widehat{y_i} - y_i)^2}{n}}$$

2.3 K Nearest Neighbors

K Nearest Neighbors is a supervised learning algorithm. Which uses distance to make predictions about grouping of data points.

The algorithm for creating K Nearest Neighbors is defined below.

(1) Load training and testing set
(2) Choose the value of K
(3) For each point in test data:
(4) find the Euclidean distance to all training data points
(5) store the Euclidean distances in a list and sort it
(6) choose the first k points
(7) assign a class to the test point based on the majority of classes present in the chosen points
(8) End

The objective function used K Nearest Neighbors was RMSE since that was what the model was being evaluated on.

The following is the equation used by the K Nearest Neighbors with Z Score implementation used.

$$\widehat{r}_{ui} = \mu_i + \sigma_i \frac{\sum_{j \in N_u^k(i)} sim(i,j) \times (r_{uj} - \mu_j)/\sigma_j}{\sum_{j \in N_u^k(i)} sim(i,j)}$$

3 EVALUATION AND PERFORMANCE

To assess the performance of the implemented K Nearest Neighbors we computed the mean and standard deviation of accuracy and RMSE across all the folds.

MovieLens Metrics

Metric	Mean	Standard Deviation
Accuracy	0.42028	0.0012603
RNSE	0.93672	0.0022658

4 CONCLUSION

For the MovieLens 100k dataset using hyper-parameter tuning we were able to marginally improve performance. We tuned across k sizes, minimum k, and whether to do item-based or user-based approaches. Taking the RMSE from the default implementation in scikit-surprise of 0.95 we improve performance by 0.02. Accuracy performance was not improved but, I imagine this is since rounding is necessary and that this whole integer rounding swallowed up the improvements shown in RMSE. Overall the results the accuracy measurement show why it is not a great measurement to conduct when a model is continuous. As shown from previous work in paper 1 accuracy and related metrics perform well when a model's classification is discrete.

5 ACKNOWLEDGMENT

I would like to thank Dr. Jing Ma for her exemplary teaching this semester. I would also like to thank all the teacher assistants to data mining for helping answer questions.

REFERENCES

(1) J. Ma, "CSDS 435 Data Mining Lectures" in https://canvas.case.edu/, 2025
(2) "scikit-learns Documentation" in https://scikit-learn.org, 2025
(3) "scikit-surprise Documentation" in https://surpriselib.com 2025