Task 3 Final Report: Predicting User Ratings from Text Reviews

Israel Avendano Jr

June 5 2025

1. Task

This report addresses **Task 3**, which involves predicting the rating a user gives a product based on the term-frequency vector of their review of the product.

2. Lead

The primary contributor for this task is **Israel Avendano Jr**.

3. Methods

We began with **Linear Regression** as a baseline to establish a simple benchmark. To improve generalization, we tested **Ridge** and **Lasso Regression**, incorporating regularization and tuning hyperparameters.

Next, we applied a **Random Forest Regressor** to capture nonlinear relationships and feature interactions through ensemble learning.

Our best-performing model was a **Neural Network** using `MLPRegressor` with four hidden layers. It leveraged ReLU activations, early stopping, and tuned hyperparameters (learning rate and L2 regularization) to model complex user-item interactions effectively.

We also explored **ensembling** by combining outputs from multiple model types and averaging multiple MLPs. These approaches performed comparably to the best standalone MLP, offering no significant gains.

Model performance was measured using mean squared error (MSE) and R^2 score on a development set.

Submission Model Details

The final submission model is a **Multi-Layer Perceptron (MLP) Regressor** implemented using `scikit-learn`'s `MLPRegressor` class. This model was selected due to its superior performance in capturing complex nonlinear relationships between user-item interactions, as demonstrated by its low mean squared error and strong R^2 score on the development set.

Toolkit

- Python 3.11
- `scikit-learn` 1.6.1
- `scipy` 1.15.3
- `wandb` 0.19.11

Preprocessing

Input data was transformed into a sparse matrix using `scipy.sparse.coo_matrix`, with shape (num_users, num_items). Rows represent user IDs, columns represent item IDs, and values are explicit user ratings. Both training and development matrices were built with consistent dimensions using the maximum column index across both sets.

Hyperparameters

- `hidden_layer_sizes=(100, 100, 100, 100)`
- `solver='adam'`
- `alpha=0.001`
- `activation='relu'`
- `learning_rate_init=0.001`
- `early_stopping=True`

This model achieved the best generalization performance across all attempted approaches and is used for generating predictions on the test set.

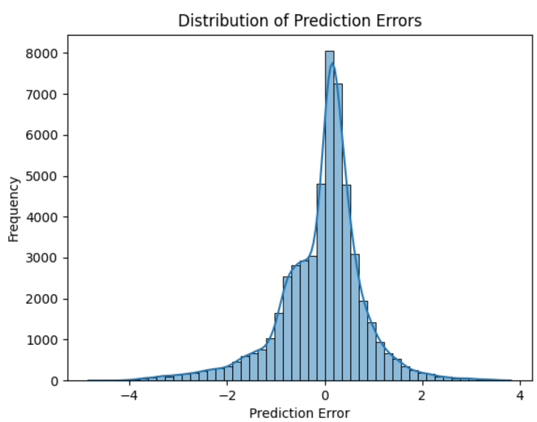

Figure 1: Prediction Errors for MLP Regressor.

Model	**MSE**	**R^2**
Linear	42.63	-24.28
Ridge	1.073	0.36
Lasso	1.61	0.04
RF	1.28	0.24
MLP	**0.77**	**0.54**

Table 1: Model performance on dev set.

The distribution of prediction errors for the MLPRegressor demonstrates a strong central peak near zero. Most deviations fall within ±1 rating point, which is acceptable given the 1–5 rating scale.

Overall, the plot confirms that the MLP generalizes well across the development set, producing both accurate and consistent predictions.

6. Distribution of Work

Israel Avendano Jr was the sole contributor for this task.