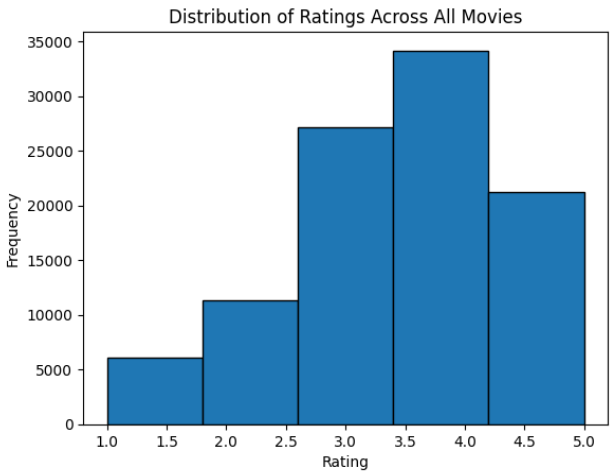
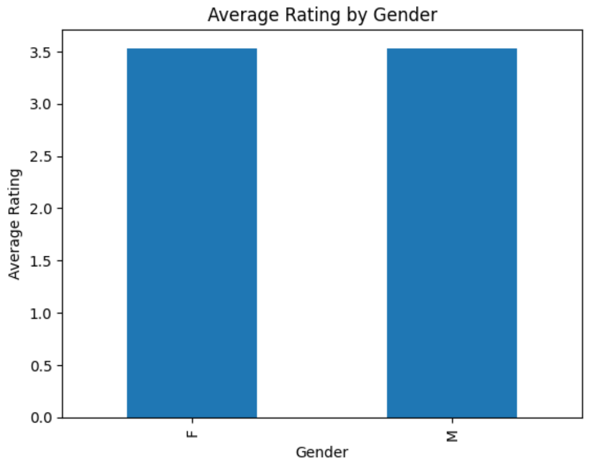
Introduction

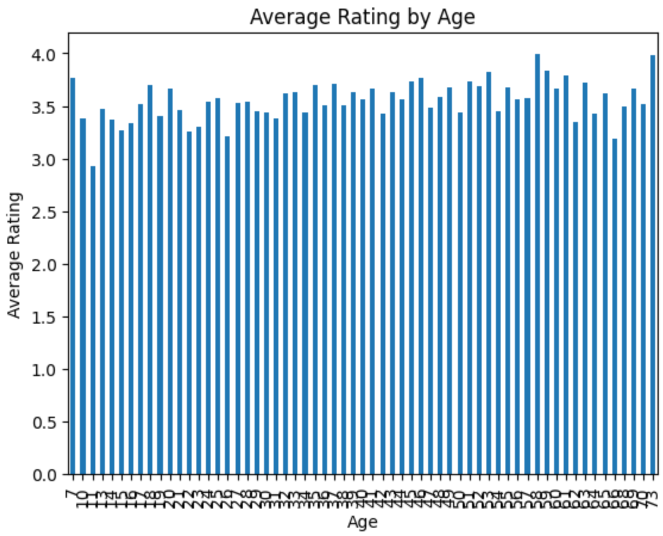
This report describes the development and implementation of a system designed to predict user ratings for movies. The system provides personalized predictions of how users might rate different movies.

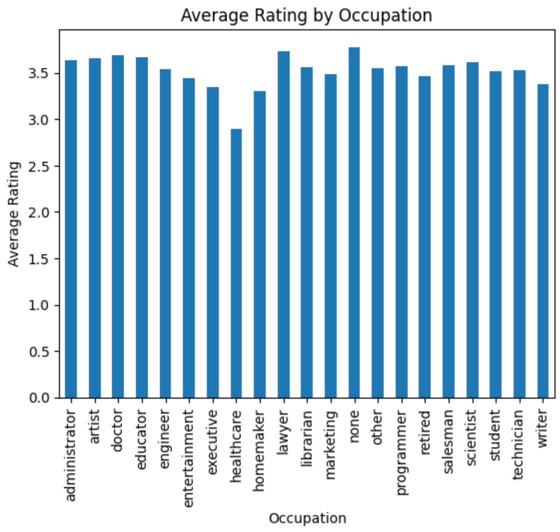
Data Analysis

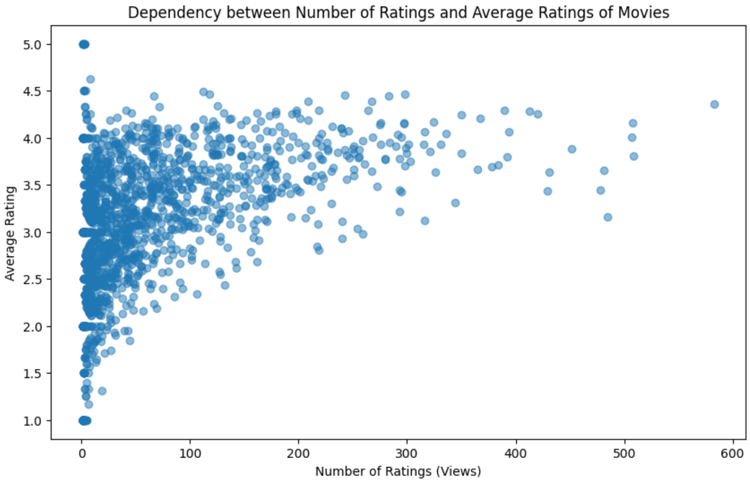
The analysis began with the acquisition of datasets containing user ratings, user demographics, and movie details. Preliminary examination showed a right-skewed distribution (picture 1) in user ratings, suggesting a tendency among users to watch and rate movies they are predisposed to like. The analysis also revealed consistent rating behaviors across user demographics such as gender (picture 2), age (picture 3), and occupation (picture 4), with no significant differences found. Also, there was found a relationship between the number of ratings a movie receives and its average rating (picture 5). It reveals a trend where movies with more ratings tend to have higher average ratings. This could indicate that more popular or widely watched movies receive better ratings, or that better-rated movies attract more viewers to rate them.











Model Implementation

The system for predicting user movie ratings was implemented using the Alternating Least Squares (ALS) algorithm. This model is a form of collaborative filtering that operates on the principle of matrix factorization, which decomposes the user-item interaction matrix into lower-dimensional user and item factors.

The choice of the Alternating Least Squares (ALS) algorithm was driven by several factors:

- **Fidelity to User Preferences**: ALS is adept at uncovering latent factors that determine user preferences, which is crucial for a recommendation system.
- **Implicit Data Handling**: Unlike some algorithms, ALS can work with implicit feedback data (such as views or purchases), not just explicit ratings.
- **Parallelization and Scalability**: ALS is designed to be parallelizable, which significantly reduces computation time and is therefore scalable to large datasets.
- **Ease of Use**: The ALS algorithm comes with a straightforward implementation in many machine learning libraries, which makes it accessible and reduces development time.

Model Advantages and Disadvantages

Advantages of ALS Model:

- **Simplicity**: Easier to implement and maintain than complex hybrid models.
- **Efficiency**: Less computationally intensive, leading to quicker training and lower operational costs.
- **Scalability**: Effectively handles large datasets, which is essential for growing user bases.
- **Proven Track Record**: Widely used and validated in industry-leading recommendation systems.

Disadvantages of ALS Model:

- **Content Ignorance**: Does not account for content-based features that could improve personalization.
- **Cold Start Issue**: Challenges with providing recommendations for new users or items without historical data.

Training Process

For the training process, a grid search was utilized to systematically work through multiple combinations of hyperparameters and determine the best configuration for the ALS model. This method ensures a thorough exploration of the parameter space and a more objective selection of parameter values. Cross-validation was employed as part of the hyperparameter tuning process to validate the model's performance on independent data subsets. This approach mitigates the risk of overfitting and ensures that the model's parameters are robust across different data samples. The following hyperparameters were determined to yield the best results based on the evaluation metrics:

- **Optimal Rank**: Set to 5, this parameter defines the number of latent factors used to represent users and movies. A rank of 5 was found to capture the complexity of the dataset adequately without overfitting.
- **Optimal regParam**: The regularization parameter was set to 0.1. This value helps to prevent overfitting by penalizing larger model coefficients, providing a balance between model complexity and generalization to new data.
- **Optimal maxIter**: The maximum number of iterations was set to 10. This number of iterations allowed the ALS algorithm to converge to a solution that minimized the prediction error.

These hyperparameter values were selected after a series of experiments that evaluated the model's performance on validation datasets. The iterative process of hyperparameter tuning was guided by the goal of minimizing the prediction error while avoiding overfitting to the training data.

Evaluation

The model was evaluated using various metrics, such as MSE (= 0.8766335848952713) and RMSE (= 0.9362871273788139), to assess the accuracy of the predictions. The evaluation also included examining the distribution of prediction errors and ensuring that the errors were normally distributed around zero, indicating no systematic bias in the predictions.

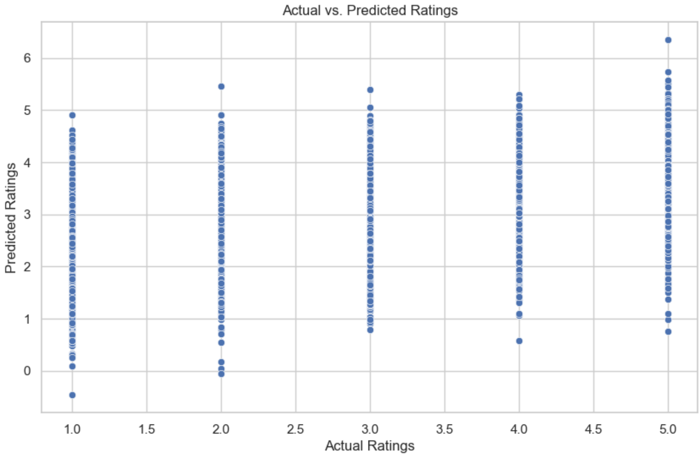
Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) were chosen as the evaluation metrics for the following reasons:

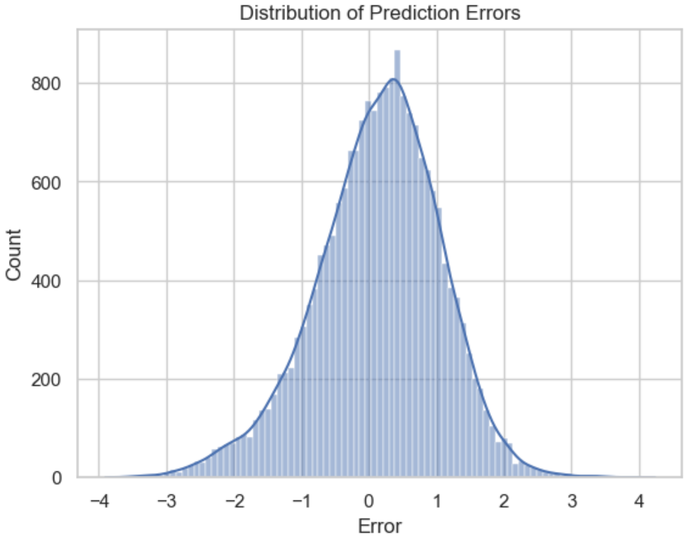
- **Sensitivity to Large Errors**: Both MSE and RMSE are sensitive to large errors, which is advantageous in a recommendation system where large deviations from actual ratings are particularly undesirable.

By presenting both MSE and RMSE, it becomes more straightforward to verify the consistency of the results.

Results

The results indicate that the model can predict user ratings with a degree of accuracy that is promising for the application in a live environment. The analysis of actual versus predicted ratings (picture 6) and the distribution of prediction errors (picture 7) support the model's efficacy. However, further testing and refinement may be necessary to improve performance and ensure scalability.


Actual vs. Predicted Ratings
Predicted Ratings
Actual Ratings


Distribution of Prediction Errors
Count
Error