Matrix brigade team names:

- Lyna Bouikni
- Insaf Medjaouri
- Marius Roger

1 Introduction, context and aims:

In the last decade, recommendation systems have stood out as an essential tool in the quest of adapting content to users' preferences, thereby enhancing user experience and fostering platform engagement. Such systems are especially salient in the domain of movie streaming, where they serve as a bridge connecting users to cinematic experiences tailored to their tastes. However, the sparsity of movie rating data poses significant challenges, making the quest for an optimal recommendation algorithm a demanding endeavor.

To address this challenge, this work undertakes a systematic exploration. We initiate with a baseline approach that employs matrix factorization coupled with gradient descent. Subsequently, this model was augmented through an aggregation of predictions from 20 distinct matrix factorization models. Seeking to elevate the system's efficacy, we then delve into alternative methodologies, notably k-Nearest Neighbors (kNN) and Singular Value Decomposition (SVD). Beyond these, an ensemble method is incorporated, designed to synergistically leverage the strengths of individual techniques. The ambition of this study is twofold: to surpass the outcomes of the baseline approach and to navigate the challenges posed by sparse movie rating datasets.

2 Materials and methods:

The following sections provide a data description and analysis, followed by a comprehensive exposition of the employed methods. All the codes presented are available at: https://github.com/Master-IASD/assignment1-2023-matrix-brigade

2.1 Data exploration:

An important preliminary step is to conduct a comprehensive data exploration, aiming to understand the characteristics and structure of the dataset. In our case, we are working with three essential datasets: **ratings_train.npy**, **ratings_test.npy**, **names_genre.npy**. Throughout the data exploration process, we will harness the power of various data science tools and libraries, including **NumPy**, **Pandas**, **Matplotlib**, and **Seaborn**.

2.1.1 Analysis of the prevalence of each movie genre and of the training dataset:

To gain insights into the distribution of movie genres within the dataset, we visualized the prevalence of each genre among the movies (as showin in **Fig.**1). The figure delineates the heterogeneity of film genres in our dataset and primarily furnished an introductory comprehension of the data at hand.

Subsequent to our initial exploration, we delved into a comprehensive analysis of the training set, recognizing its pivotal role in the training process. This dataset encompasses 4980 movies and 610 users. As a preliminary phase of the analysis, we scrutinized the presence of missing data. **Fig**.2 illustrates the distribution of missing values across users.

Upon analysis of **Fig.**2, we observe missing values, indicative of many users refraining from rating a considerable number of movies. This observation highlights the dataset's inherent sparsity. A quantification of NaN values in the R matrix,

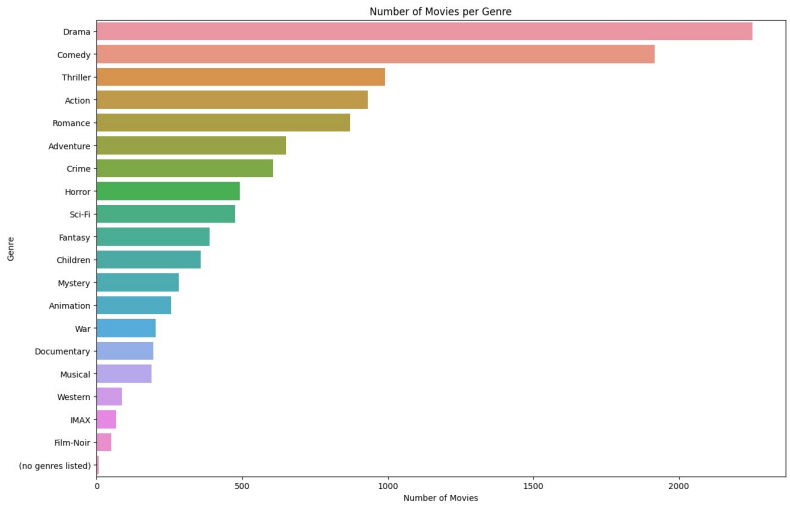

Figure 1: Exploratory Analysis of Genre Prevalence in Movies

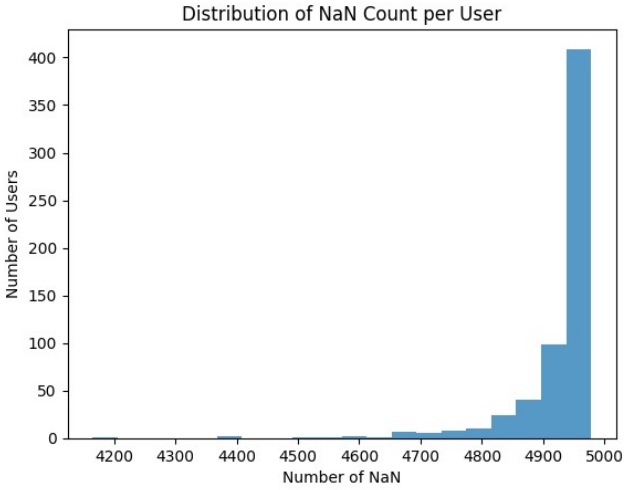

Figure 2: Distribution of NaN Count per User

relative to its total elements, reveals that approximately 98% of the data is missing. Given this pronounced sparsity, it is imperative to employ specialized methodologies to address missing values in the development of our recommendation system.

3 Employed methods:

3.1 Matrix Factorization with Gradient Descent

In this section, our goal is to identify matrices, I and U, such that their product approximates the matrix R. Initiation is achieved with random values for I and U. Gradient descent is systematically applied to diminish the disparity between R and the resultant of I and U. During these iterations, we leverage masking techniques to eschew superfluous computations, as shown below:

$$R_m = \text{np.ma.MaskedArray}(R, \texttt{nan\_mask\_R}) \tag{1}$$

and NaN entries are weighted as 0, rendering them as undefined:

$$S = R - I@U^T \tag{2}$$

$$S = \text{np.where}(R_m.\text{mask}, 0, S) \tag{3}$$

3.1.1 Matrix Factorization Model: Baseline

In the baseline implementation, a singular model approach is adopted to address the missing values. The gradient of the cost function with respect to the elements of matrices I and U was computed using:

$$\frac{\partial C}{\partial i_{iq}}(I,U) = 2 \sum_{j:(i,j)\in S} \left(r_{ij} - \sum_{s=1}^{k} i_{is} \cdot u_{js} \right) (-u_{jq}) + 2\lambda i_{iq} \tag{4}$$

$$\frac{\partial C}{\partial u_{jq}}(I,U) = 2 \sum_{i:(i,j)\in S} \left(r_{ij} - \sum_{s=1}^{k} i_{is} \cdot u_{js} \right) (-i_{iq}) + 2\mu u_{jq} \tag{5}$$

Where $S = \{i, j : r_{ij} \text{ is observed}\}$.

Regularization terms, characterized by λ and μ, are also introduced to prevent overfitting.

3.1.2 Matrix Factorization Ensembling: Enhanced Implementation

To enhance the simple Matrix factorization method, we employed an ensemble technique that aggregates predictions from 20 distinct models. The computation of $I@U^T$ is executed in parallel, bolstering result aggregation. In contrast to the averaging approach, which yields good results for matrix factorization ensembling, we found that voting is computationally expensive and less effective.

Hyperparameter Selection:

In determining the optimal hyperparameters for our matrix factorization model, our objective was twofold: ensuring rapid convergence and fostering robust generalization capabilities.

- **Latent Factor** K: We elected to set K at 5, a relatively modest value, to inhibit model over-complexity and consequently mitigate the risk of overfitting.
- **Learning Rate**: This was judiciously calibrated, especially in scenarios of augmented sample sizes, such as combined training and testing datasets.
- **Regularization**: To avert overfitting, regularization parameters were meticulously tailored. Given the disparities in the cardinality of coefficients within matrices I and U, we postulated that U necessitates approximately tenfold the regularization intensity of I.
- **Epochs and Parallel Models**: The determination of these parameters was strategically made, striking a balance between computational efficiency and desired performance.

It's paramount to underscore that the dimensions of the dataset and the computation duration considerably influenced these decisions. The aforementioned hyperparameter choices emanate from empirical assessments paired with a systematic approach to model tuning.

3.2 Employing the SVD and KNN Algorithms

Utilizing the Singular Value Decomposition (SVD) technique, we aimed to predict the ratings within our sparse matrix.

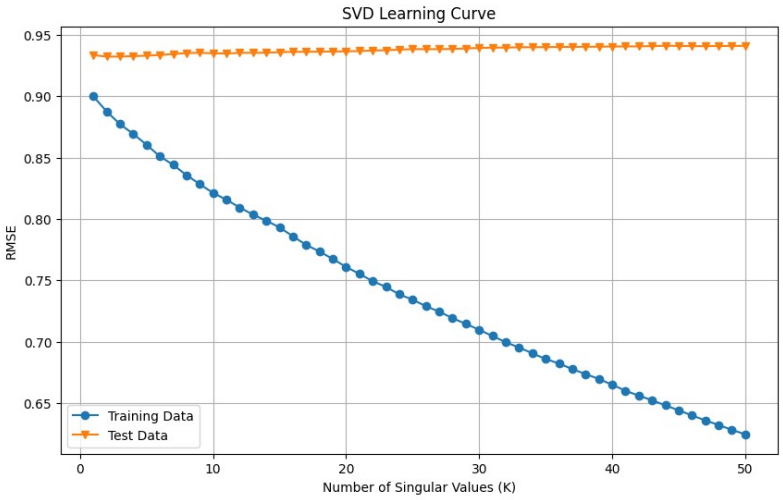

Figure 3: SVD Learning Curve: RMSE as a function of the Number of Singular Values for Training and Test Data.

Fig.3 showcases the SVD learning curve, illustrating the relationship between RMSE and the number of singular values (K). The training data's RMSE consistently declines with an increase in K, whereas the test data's RMSE stabilizes around 0.95. This discrepancy indicates the model's overfitting with the inclusion of more singular values. It is worth noting that Despite introducing bias terms or noise, overfitting remains a prevalent challenge.

In the implementation of KNN, the `nearestneighbours` library was utilized. A spectrum of k-values were examined, selecting the best based on performance metrics. Although various similarity measures, including cosine, Euclidean, and Minkowski, were explored, they presented analogous results. Thus, only cosine similarity was retained for further analyses.

3.3 Ensemble method

In the pursuit of refining the performance of our enhanced matrix factorization model, we employed an approach that leverages multiple models and aggregation schemes. The methodology proceeds as follows:

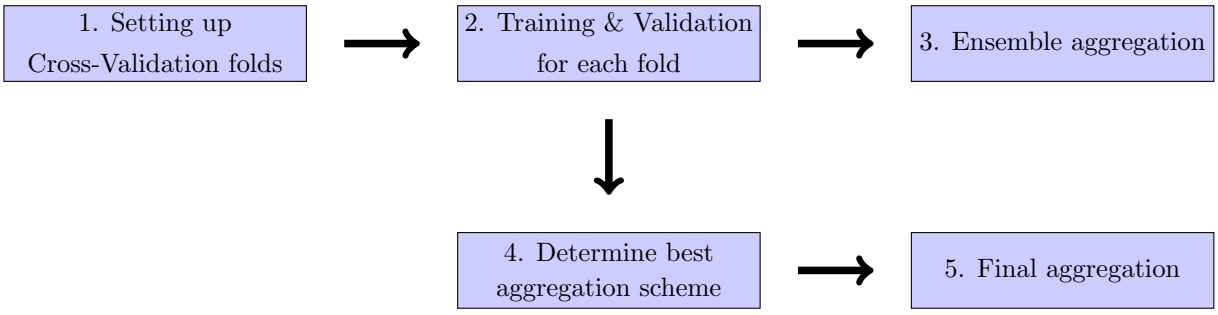

Figure 4: Flow diagram for the ensemble method with cross validation

Step 1: Fold-Wise Model Training and Prediction

1. For each fold in the cross-validation process:
    - Train the designated models (k-NN, SVD, Matrix Factorization) on the training subset of the fold.
    - Generate predictions for the validation subset of the fold.
    - Aggregate the predictions employing both 'average' and 'voting' strategies.

- Compute the RMSE for each aggregation scheme to discern the superior method for that specific fold.

Step 2: Identification of Optimal Aggregation Technique

2. Post completion of all the folds, ascertain the aggregation method, either 'average' or 'voting', that consistently registered the lowest RMSE across the various folds.

Step 3: Aggregation for Overall Prediction

3. Employing the optimal aggregation technique identified from the preceding step, generate predictions across the entire dataset.

4. Subsequently, aggregate these predictions adhering to the selected best scheme.

In our evaluation, the most optimized voting algorithm executed in approximately 17 seconds (largely invariant to the number of matrices predicted for aggregation) leveraging the `mode` function from Python's Standard Library's `statistics` module. However, compared to this method, averaging provides a more efficient and less resource-intensive approach for matrix factorization ensembling, often yielding superior results.

4 Results and Discussion:

Table 1: Evaluation Metrics for Different Approaches

	Baseline (MF, K=5, 3k epochs)	**Enhanced MF**	**Ensemble Method**
RMSE	0.969	0.89	0.992
Time	188.15	101.37	81.65
Accuracy (%)	24.4	24.89	27.73

Table 1 presents the evaluation metrics for the three used methods. The Enhanced MF emerges superior in RMSE performance , underscoring its predictive precision. While the Ensemble method is marginally less accurate in RMSE, but excels in computational efficiency. The suboptimal performance of the Ensemble method may be attributed to various factors. Model diversity within the ensemble could be insufficient, leading to overlapping errors. Moreover, the aggregation technique employed might not be ideal for the specific set of models, inadvertently introducing prediction biases.

5 Conclusion:

In addressing sparse movie rating datasets, our work aimed to surpass the baseline through a comprehensive exploration of recommendation methodologies, from matrix factorization to ensemble strategies. Unexpectedly, the ensemble method did not achieve the anticipated enhancement, alluding to possible shortcomings in model diversity or aggregation. These findings underscore the significance of meticulous model selection, necessitating rigorous empirical validation and fine-tuning. The insights gleaned from this work will undoubtedly act as a touchstone for future explorations and implementations in the realm of recommendation systems..