

Enhanced Freelance Matching: Integrated Data Analysis and Machine Learning Techniques

Ismail Sahnoun ¹ and Emna Ammar Elhadjamor ^{2,*}¹ Higher Institute of Applied Science and Technology of Sousse, Tunisia; e-mail : ismailsahnoun77@gmail.com² Laboratory-ENSI, Manouba University, Tunisia; e-mail : emna.ammar.tn@gmail.com

* Corresponding Author : Emna Ammar Elhadjamor

Abstract: The objective of this research is to devise a personalized recommendation system for a freelancing platform to optimize the freelancer project matching process. This enhancement is intended to improve user experience and increase the success rate of projects. The system will recommend projects to freelancers based on their skills and preferences by employing data analysis and machine learning methodologies. The research methodology adheres to the Cross Industry Standard Process for Data Mining (CRISP-DM), incorporating six stages: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. The proposed project employs a hybrid recommendation strategy, integrating Content-Based Filtering through KNearest Neighbors (K-NN) and Cosine Similarity, Collaborative Filtering via Singular Value Decomposition (SVD), and recommendations derived from Word2vec. Evaluation metrics such as precision, recall, F1 score, MAP, and MRR are utilized to assess model performance. The results, including precision scores of 0.80 for KNN and 0.728 for SVD, recall scores of 0.60 for KNN and 0.623 for SVD, and F1 scores of 0.69 for KNN and 0.671 for SVD, as well as a MAP of 0.75 and MRR of 0.80 for Word2vec, demonstrate the efficacy of the hybrid recommendation system in delivering accurate and varied project suggestions to freelancers, with a weighted average ensemble learning model emerging as the most effective solution.

Keywords: CRISP-DM methodology; Data analysis; Hybrid recommendation; Machine learning; Recommender systems.

1. Introduction

Recommendation systems (RS) are a subset of information filtering systems designed to predict user preferences for various topics. These systems simulate the social practice of seeking recommendations from others while further enhancing the process[1]. Initially employed as digital bookshelves in academic research[2], RS gained broader commercial application by developing the first 'collaborative filtering' engine, Tapestry, in the early 1990s by Goldberg et al. (1992). Despite its name suggesting a collaborative interaction between recommenders and requesters, this was later refuted due to the anonymity of the involved parties [3]. RS technology has significantly advanced and is now widely utilized to personalize learning experiences according to individual learner goals[4]–[6].

Recommender systems (RSs) have become indispensable in the digital world, offering personalized suggestions on various online platforms such as social media, e-commerce, and travel. These automated recommendations profoundly impact user decisions and activity on these platforms. By analyzing user data, RSs understand individual preferences, benefiting both the user and stakeholders involved in the recommendation process[7]. Nevertheless, the pervasive use of RSs has led to ethical concerns, particularly where transparency and explainability are at stake. Challenges in addressing these issues stem from proprietary limitations and regulatory gaps. With the ascension of the gig economy and freelancing platforms, there emerges a necessity for efficient systems to align freelancers with appropriate projects. Freelancers often expend considerable time in job searches, which is a substantially inefficient process.

Received: February, 14th 2024Revised: April, 24th 2024Accepted: May, 2nd 2024Published: May, 9th 2024

Copyright: © 2024 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Developing recommendation systems aims to reduce this time investment and increase overall efficiency for freelancers[8]. This project aims to construct a recommender system for a freelancing website to enhance the user experience and foster successful partnerships. Utilizing data analysis and machine learning techniques, the system will propose personalized project recommendations that match freelancers' skills and inclinations. Such a tailored approach is intended to simplify the process of identifying suitable job opportunities for freelancers and assist businesses in finding the most compatible candidates for their projects.

Our approach includes the phases of data collection, preprocessing, and exploration of collaborative filtering and content-based filtering algorithms to render personalized recommendations. The assessment of the system's performance will be facilitated by evaluation metrics such as precision, recall, accuracy, and user feedback. Overcoming obstacles like data sparsity and achieving scalability are essential for elevating system performance. The primary objective of this project is to augment the user experience on a freelancing platform for both freelancers and businesses. This enhancement aims to be achieved by developing an advanced, personalized recommender system designed to optimize the matching process between freelancers and relevant projects. The ultimate goal is to increase user engagement and satisfaction, thereby reinforcing the platform's competitive stance within the gig economy market.

Two prominent techniques of recommendation systems are content-based filtering and collaborative filtering, each with its advantages and challenges. Content-based filtering is attribute-based, while collaborative filtering is based on user behavior and preferences. Word2vec and its variants like Doc2Vec are being used more in these systems to boost their performance, notably when handling data sparsity and the cold start problem. Content-based collaborative filtering integrating word vectors, for example, Doc2Vec, could make collaborative filtering more accurate and help recall recommendations by capturing semantics, grammar, and order of words [9]. The use of a hybrid approach, that is, combining content-based and collaborative filtering with word embeddings like Word2Vec, could outperform the traditional models; such a method would more likely perform better, especially when cold-starting, by utilizing semantic vectors to represent item features[10].

Graph-based methods can surpass pure collaborative or content-based approaches by using user-content links inferred from user ratings and content descriptions; they leverage graph techniques for collaborative filtering [11]. The hybrid recommendation algorithms, which combine collaborative filtering with Word2Vec, may substantially improve accuracy and efficiency, mainly on large-scale datasets, and make recommendations based on user feedback [12].

The rationale for selecting specific methodologies for the recommender system stems from a strategic analysis of their strengths and limitations and the synergistic potential inherent in their integration. The methodologies in question include Content-Based Filtering, Collaborative Filtering, and deep learning techniques, specifically Word2vec models. Each of these methods offers unique advantages in the context of recommendation systems for freelancing platforms.

Content-based filtering is predicated on analyzing freelancers' skills, historical projects, and expressed preferences, enabling the algorithm to suggest projects that align with their proven expertise and interests. This method is chosen for its capacity to deliver highly personalized recommendations tailored to the individual profiles of users. Collaborative Filtering leverages user interaction and historical data to identify projects that might appeal to a freelancer based on similarities with other users' preferences and behaviors. This approach is valuable for capturing and utilizing communal insights and trends in user preferences, thereby enriching the recommendation process. Word2vec-based Suggestions utilize advancements in deep learning to comprehend the semantic relationships within project descriptions and freelancer profiles. This method enhances the accuracy of recommendations by understanding the context and nuanced meanings of terms used, offering a significant improvement in user experience.

The integration of these methodologies is necessitated by the objective of constructing a recommender system that is accurate and diverse in its suggestions. While powerful, each method exhibits specific limitations that can be mitigated through a hybrid approach. For instance, content-based filtering may narrow recommendations (the so-called "filter bubble" effect), collaborative filtering may not perform well for new users or projects due to the cold

start problem, and Word2vec models require extensive computational resources and data for effective training.

The hypothesis underpinning the integration of these methods posits that a hybrid recommender system can harness the strengths of each to offset their individual weaknesses. This would culminate in a system that provides precise, personalized, and varied project recommendations. Such a system would not only cater to the explicit preferences of users but also introduce them to a broader array of potentially interesting projects, thereby enhancing user satisfaction and engagement.

The structure of this paper is as follows: Literature Review the foundational concepts leveraged in our work such as business processes and KPIs, as well as delineating the primary research challenges and synthesizing recent related works founded on analogous concepts. Section 3 expounds upon the research methodology employed to attain the pre-defined goals. Section 4 details our proposed approach. Section 5 elaborates on applying this approach to the emergency care case study. The concluding section provides a succinct synopsis of our findings. This paper contributes to understanding user experience enhancement in the gig economy and seeks to open new vistas for recommender systems dedicated to freelancers and businesses.

2. Literature Review

This literature review aims to impart a broad comprehension of recommender systems' role in the freelancing domain and establish the context for our study.

2.1. Recommendation Techniques

There are three chief recommendation approaches: Collaborative Filtering (CF), Content-based Recommendation (CB), and Hybrid Recommendation Systems, as underscored by [9]. Content-based recommendation (CB) is predicated on identifying characteristics similar to those in a user's past preferences to inform future recommendations. A CB RS architecture is depicted in Figure 1 and encompasses three primary components: a Content Analyzer, a Profile Learner (Generator), and a Filtering Component, as expounded by [10]. The Content Analyzer collates item content from various sources and leverages feature extraction techniques to construct item representations. The Profile Learner then applies machine learning methods to generalize user data, creating user profiles that reflect historical preferences. The Filtering Component subsequently aligns user profiles with potential recommendations.

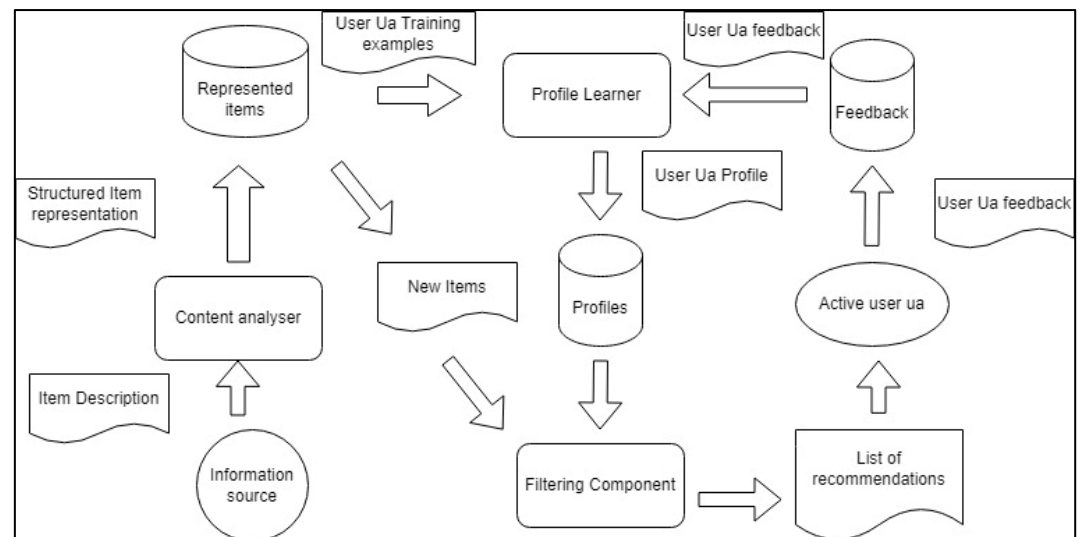


Figure 1. High level architecture of content-based recommender [10]

Collaborative Filtering (CF) : recommendations are based on user behavior or ratings of the recommended items. This method suggests articles that users with similar preferences prefer, exploring a range of content options[11]–[13]. Using learner profiles containing information such as age, country, previous learning activities and educational background, recommender systems (RS) can recognize learners with similar learning preferences and recommend

suitable materials[14]. The CF algorithm either generates predicted ratings or suggests a list of the N best items, as shown in Figure 2.

Hybrid Recommendation Systems (RS) represent a synergy of Content-Based (CB) and Collaborative Filtering (CF) methodologies. Such systems harness attributes of both techniques by integrating individual predictions, infusion of content insights into collaborative models, and weighted averaging of CB and CF recommendations. Additionally, hybrid RS may derive their final suggestions from a conjoint rankings analysis. Knowledge-Based Recommendation Systems are increasingly utilized in e-learning, drawing on Semantics- or Knowledge-Based frameworks that include Context-Based and Ontology-Based approaches. These systems employ ontologies to structure content knowledge and information regarding stakeholders within the recommendation process, enabling them to align suitable learning resources with relational knowledge.

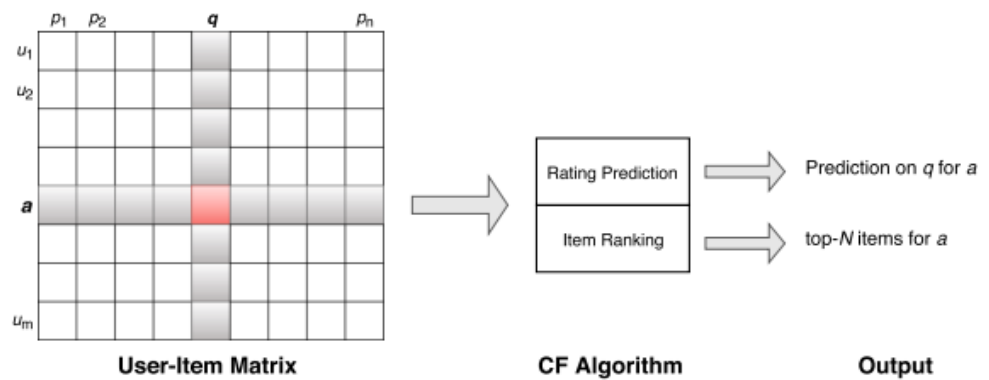


Figure 2. CF-based recommendation system. Source[15]

All RSs function within these established frameworks, laying the foundation for addressing particular recommendation tasks. A comprehensive system taps into datasets, machine learning (ML) algorithms, and evaluation methodologies and yields outcomes which will be discussed in depth in the following sections.

2.2 Related Works

In today’s digital era, characterized by information surfeit, Recommender Systems (RSs) have become integral to delivering personalized services and enhancing decision-making. Within multi-faceted digital environments, RSs mitigate information overload and enrich user interactions.

Various experiments have explored combining CB and CF strategies to mitigate the cold start problem by factoring in learners’ preferences, backgrounds, interests, and memory capacities. For example, Benhamdi et al. [16] conducted evaluations of multidimensional similarity, integrating learners’ prior knowledge, interests, time allocation for tests, and memory capacities employing correlation matrices and predictive ratings. Incorporating memory capacity has distinctly refined the accuracy of recommendations, pinpointing appropriate learning materials—an observation echoed by Ronghuai et al.[17]. Nonetheless, challenges such as data scarcity and algorithm scalability remain if new learning resources go unassessed. Hossain et al.[8] deployed diverse techniques such as artificial neural networks, decision trees, logistic regression, and support vector machines (SVM) to foresee student challenges in upcoming digital design course sessions. This approach parallels studies by Herlocker et al.[6], Vahdat et al[18], and Donzellini and Ponta[19]. Complementarily, Li et al. [20] employed these methods to curate a selection of educational resources tailored to user interests a methodology in concurrence with the research by Zhang et al. [21], Chen et al. [22], and Huang et al.[23].

Karga and Satratzemi [24] utilized a similarity matrix coupled with a ‘tag and ratings’ framework to blend CF and CB methods for improving the learning design process. Clements et al. [25] and Bennett et al. also encourage this methodology[26]. Despite being more resource-intensive, the hybrid approach is deemed superior, offering a broader range of benefits, including greater accuracy in limited-ratings scenarios.

In the rapidly growing freelancing domain, RSs provide substantial utility in pairing freelancers with projects that align with their skills and preferences. RS applications within the

freelancing sector are gaining momentum as they streamline job searches for freelancers and elevate project results for businesses. Various studies have looked at different RS algorithms and techniques to deliver project recommendations tailored to freelancers.

For instance, Abhinav et al. [27] introduced CrowdAdvisor, a multidimensional evaluation framework that assesses freelancers based on present and past job performances. The algorithms developed for this framework evaluate several dimensions and have outperformed baseline algorithms. Difallah et al. [28] showcased a personalized Crowdsourcing platform for human intelligence task assignments, capitalizing on social network influence. Their system matches workers with tasks by analyzing the worker's social media profiles to better align crowdsourcing platform task requirements with individuals' preferences. Their experiments revealed that a proactive assignment methodology was more effective than traditional, reactive ones.

The systematic review of existing literature underscores the critical advancements and applications of Recommender Systems (RSs) across diverse domains, notably within educational contexts and the increasingly significant freelancing sector. Despite these advancements, a discernible research gap is evident, particularly concerning the adaptation and optimization of RSs for freelancing platforms, which are characterized by their highly dynamic nature and the varied needs of their users. This gap is pronounced in several key areas that warrant further scholarly attention.

Firstly, exploring RS applications within the freelancing domain remains markedly limited. Most extant research focuses on deploying RSs in e-commerce and educational settings, leaving a notable void in studies dedicated to understanding and addressing the unique challenges presented by freelancing platforms. With their rapidly changing project landscapes and diverse participant requirements, such platforms offer a fertile ground for RS innovation that has yet to be fully explored.

Secondly, scalability and data scarcity are recurrent themes within the literature, particularly concerning introducing new resources or projects and managing a highly varied user base. The strategies to effectively navigate these challenges, crucial for the dynamic freelancing environment where new opportunities and participants continuously emerge, are insufficiently documented and explored.

Thirdly, while the literature introduces multidimensional evaluation frameworks, such as those proposed by Abhinav et al. [27], for assessing freelancers, incorporating these frameworks into RSs specifically designed for freelancing platforms is inadequately investigated. The potential for such frameworks to significantly improve the accuracy of recommendations and overall user satisfaction in freelancing contexts remains largely untapped.

Furthermore, the influence of social media and the potential for personalization in RSs, as exemplified by the work of Difallah et al. [28], highlight the effectiveness of leveraging digital footprints for task assignments in crowdsourcing platforms. However, the literature has not thoroughly examined the extension of these methodologies to enhance freelancer project recommendations through a comprehensive analysis of their social media profiles and online behaviors.

Lastly, the resource intensiveness of hybrid RSs, despite their acknowledged superior performance, presents practical concerns regarding their implementation in freelancing platforms, where efficiency and rapid response times are paramount. The literature lacks a focused investigation into optimization techniques that could mitigate computational demands without diminishing the quality of the recommendation output.

3. Methodology

For this project, we embraced the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology, which is esteemed for its structured approach in data science endeavors. The ultimate aim is to fortify user engagement and satisfaction, consolidating the platform's position in the competitive gig economy market. Content-based filtering stands as a prevalent technique in recommender systems (RS) for freelancing platforms. By analyzing freelancers' skills, previous projects, and preferences, content-based filtering algorithms are poised to propose projects that align with their expertise and interests. Additionally, collaborative filtering methods leverage user interactions and historical data to identify projects that a specific freelancer might find appealing based on similarities with other freelancers. Hybrid recommender systems, which integrate both content-based and collaborative filtering, have

demonstrated efficacy in rendering more precise and diverse project recommendations. These hybrid models leverage each algorithm's strengths and address their limitations, thereby presenting freelancers with a broader selection of suitable project opportunities. Furthermore, advancements in deep learning-based RSs, such as Word2vec models, have facilitated a better understanding of semantic connections in project descriptions and freelancer profiles. This advancement has led to heightened accuracy in recommendations and an enhanced user experience.

3.1. Data Understanding

Understanding the structure and significance of database tables is crucial for future data analysis despite the absence of real-time data. Key tables examined include Projects, Desired Talents, and Skills, providing essential information for guiding subsequent data generation methods. The Projects Table contains details about different projects, such as project ID, description, employer type, and status. The Desired Talents Table specifies the desired talents for specific projects, including attributes like talent title, average daily rate, and experience level. The Talents Table holds crucial details about registered freelancers, including talent description, type, and work time. Skills Tables encompass various datasets capturing the skills possessed by talents and the desired skills for specific talent profiles. Understanding these database tables lays the groundwork for data generation and subsequent project steps.

3.2. Data Preparation

At this stage, the data is processed and prepared for further analysis. For Content-Based Filtering, text data is transformed into a numeric format using techniques such as TF-IDF or embedding for analysis with K-NN or Cosine Similarity. For Collaborative Filtering using SVD, user-item interaction data is structured into a matrix to reveal hidden patterns.

Data preparation is a critical step in the data analysis pipeline, involving the cleaning and transforming raw data to ensure suitability for processing and analysis. Various techniques are employed for effective data preparation, including cleaning, handling missing values, feature engineering, and data normalization.

Data cleaning encompasses handling missing values, removing duplicates, and correcting inconsistent data formats to ensure reliable and consistent data. The datasets generated are free of null values and duplicates, enhancing data reliability. Categorical variables, such as skills, are encoded using the Get Dummies method to facilitate machine learning algorithms' compatibility and improve similarity measurement based on skills.

Feature selection involves identifying and removing unnecessary variables from the datasets to streamline data for the recommender system. Additionally, feature engineering is performed to create new features like skill lists for projects and freelancers, along with an "is favorite" column indicating the project favored by freelancers. These enhancements contribute to improved performance and accuracy of the recommender system.

The data analysis phase plays a crucial role in extracting valuable insights from the provided data. Formulating pertinent questions related to each dataset, appropriate tools are utilized to conduct a comprehensive exploration. Leveraging the datasets and employing analytical techniques, meaningful patterns and essential knowledge are uncovered to guide future modeling processes. Key analyses include examining the number of desired talent profiles per project, distribution of desired talent over projects, number of projects per talent type and experience level, proportion of projects by desired talent type, work time distribution, favorites projects distribution, and number of skills per talent.

3.3 Modeling

Various modeling techniques, including content-based filtering, collaborative filtering, Word2vec, and hybrid recommendation systems, are explored and evaluated to develop a robust recommendation system.

3.3.1. Content-Based Filtering with K-NN and Cosine Similarity

Content-based filtering is a recommendation technique that leverages the features of items and user preferences to generate personalized recommendations. It represents items and users through their attributes and past interactions, respectively. By computing the similarity between item attributes and user preferences, content-based filtering recommends items

that align with the user's interests. This approach is particularly valuable when rich information is available about the items, enabling more accurate and tailored recommendations.

One technique utilized within content-based filtering is Cosine Similarity. Cosine similarity is a widely used metric for measuring the similarity between two vectors. It calculates the cosine of the angle between the vectors, resulting in a value between -1 and 1. This metric is particularly suitable for content-based filtering as it is less sensitive to vector magnitudes and focuses on their orientation.

Another technique explored within content-based filtering is K-Nearest Neighbors (K-NN). K-NN is a simple yet powerful machine-learning algorithm that works in conjunction with cosine similarity. It represents freelancers and projects as vectors in a high-dimensional space, with each dimension corresponding to a skill. K-NN identifies the projects that are most similar to a freelancer's skill set by measuring the cosine similarity between freelancer and project skill vectors. These top K projects are then recommended to the freelancer, ensuring that the recommendations align with their skills and preferences.

While content-based filtering offers personalized recommendations based on item features and user preferences, it has certain limitations. It primarily relies on the user's past interactions and preferences, which can lead to a filter bubble where users are only exposed to similar items, limiting the discovery of new and diverse projects. It may also face challenges with new users with limited interaction history, resulting in the cold-start problem. Furthermore, content-based filtering does not consider the preferences of other users, which could provide valuable insights into project popularity and quality.

A hybrid approach that combines content-based filtering with other recommendation techniques was considered to address these limitations and develop a more comprehensive recommender system.

3.3.2. Collaborative Filtering with SVD

Collaborative filtering is another recommendation technique that utilizes users' past behavior and preferences to generate personalized recommendations. It assumes that users who have interacted similarly with items in the past will have similar preferences in the future. Collaborative filtering can be divided into two main approaches: user-based and item-based.

In the user-based collaborative filtering approach, similar users with comparable preferences are identified, and recommendations are generated based on the items preferred by those similar users. On the other hand, item-based collaborative filtering identifies similar items based on user interactions and recommends items similar to those with which the user has previously interacted.

One modeling technique employed within collaborative filtering is Singular Value Decomposition (SVD). SVD is a matrix factorization technique that decomposes a given matrix into three matrices: a left singular matrix, a diagonal matrix of singular values, and a right singular matrix. By applying SVD to the matrix representing interactions between freelancers and projects, latent features can be captured, enabling accurate recommendations based on similar users or items.

Additionally, a binary labels matrix representation was considered within collaborative filtering. This matrix represents the interactions between freelancers and projects using binary values, where 1 indicates that a freelancer likes a project and 0 indicates otherwise. By applying SVD to this binary labels matrix, latent features explaining the observed interactions can be captured, and recommendations can be generated based on similar users or items.

3.3.3. Word2vec-Based Recommendations

Natural Language Processing (NLP) techniques were explored to leverage project descriptions and provide more contextually relevant recommendations. Specifically, Word2vec, a widely used NLP technique, was utilized in the project.

Word2vec employs neural networks to generate vector representations, known as word embeddings, for words in a continuous vector space. These embeddings capture semantic relationships between words and enable meaningful comparisons and computations. In the project context, Word2vec was utilized to recommend projects to freelancers based on project descriptions. By converting textual project descriptions into continuous vector representations, the similarity between project descriptions and freelancer interests or past projects can be measured using cosine similarity. The projects with the highest similarity scores can then be recommended to the freelancers, ensuring that the recommendations align with their interests and preferences.

Word2vec enhances the recommendation system by considering the meaning and context of project descriptions, resulting in more accurate and personalized project recommendations for freelancers.

3.3.4. Hybrid Recommendation

A hybrid approach combining multiple recommendation techniques was explored to enhance the recommendation system's accuracy and diversity. Hybrid recommendation systems integrate various methods to overcome individual limitations and deliver more accurate and diverse recommendations.

Ensemble learning, a technique amalgamating multiple models to enhance overall performance and accuracy, was employed in the final hybrid recommendation system. Adopting a weighted average ensemble learning approach, outputs from different models were combined using assigned weights. These weights, determined based on each model's performance on historical data and evaluation metrics such as precision, recall, F1 score, MAP, and MRR, ensured an optimal combination. The resultant recommendations, derived from the weighted average formula, provided a final recommendation score for each project, with projects receiving the highest scores recommended to freelancers.

A comprehensive set of recommendations was generated by amalgamating predictions from KNN, SVD, and Word2vec models with appropriate weights (as illustrated in Figure 3). The weighted average ensemble learning model demonstrated superior performance compared to individual models across various metrics, including precision, recall, F1 score, MAP, and MRR. Leveraging the strengths of each model while mitigating their limitations, this approach yielded highly accurate and diverse project recommendations, thereby enhancing freelancers' overall platform experience.

Therefore, the weighted average ensemble learning model was selected as the final recommender system. Its superior performance and provision of well-rounded recommendations tailored to freelancers' preferences solidify its efficacy in optimizing user engagement and satisfaction on the platform.

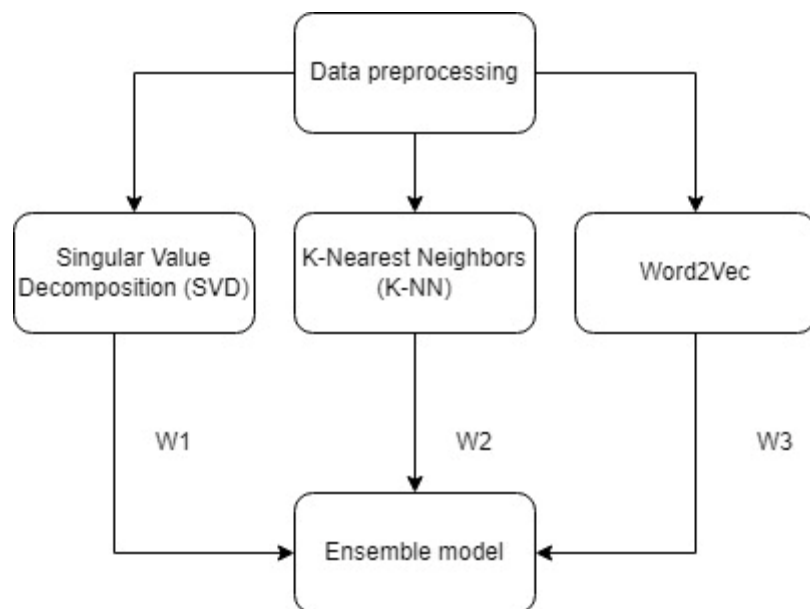


Figure 3. Weighted Average Ensemble Learning Approach

3.4. Train-Test Split

The train-test split method involves partitioning a dataset into separate subsets to enable model training and evaluation. By segmenting the dataset into training and testing subsets, the performance of models can be effectively assessed. This study employed the train-test split approach to accommodate various models, allocating training and testing sets based on a predetermined test size. This division allows for developing and testing alternative modeling algorithms and parameter settings. Subsequently, based on an initial assessment of the model's performance, final tuning of the model settings is conducted to optimize its performance.

4. Results and Discussion

Evaluation metrics such as precision, recall, F1 score, MAP, and MRR were utilized to assess model performance. A weighted average ensemble learning model emerged as the most effective solution, offering accurate and diverse recommendations to freelancers on the platform. Table 1 presents the evaluation of the KNN and SVD models, while Table 2 provides the evaluation of the Word2vec model.

Table 1. KNN and SVD evaluation.

Metrics	KNN	SVD
Precision	0.80	0.728
Recall	0.60	0.623
F1 score	0.69	0.671

For the KNN model:

- Precision: The precision is 0.80, indicating that 80% of the recommended projects are relevant.
- Recall: The recall is 0.60, indicating that the model identified and recommended 60% of the relevant projects.
- F1 score: The F1 score is 0.69, offering a combined measure of precision and recall, reflecting the model's overall performance.

For the SVD model:

- Precision: The precision is 0.728, indicating that the freelancers liked 72.8% of the recommended projects in the test set.
- Recall: The recall is 0.623, suggesting that the model captured 62.3% of the projects that the freelancers in the test set liked.
- F1 score: The F1 score is 0.671, representing the harmonic mean of precision and recall, providing an overall measure of the model's performance.

Table 2. Word2vec evaluation.

Metrics	Word2vec
MAP	0.75
MRR	0.80

Regarding the Word2vec model evaluation:

- Mean Average Precision (MAP): The MAP value is 0.75, indicating the average precision across all queries. A higher MAP suggests better performance.
- Mean Reciprocal Rank (MRR): The MRR value is 0.80, representing the average reciprocal rank of the relevant results. A higher MRR indicates better performance.

These evaluation results showcase the performance of each model in matching freelancers with suitable projects. Higher precision, recall, F1 score, MAP, and MRR values indicate the effectiveness of the models in generating relevant recommendations. The weighted average ensemble learning model emerged as the most effective solution after evaluating the models and comparing their performance across the defined metrics.

The deployment and implementation phase seamlessly integrates the recommender system model into the production environment for real-time recommendations. Key tasks include establishing a FastAPI model endpoint, containerizing with Docker, storing in GitHub for version control, deploying on the Render Cloud Web Service, and validating with tools like Postman, ensuring readiness for the Freelancing platform.

5. Conclusions

The paper provides a detailed examination of creating and assessing a hybrid recommendation system designed for freelancers. It underscores the crucial function of recommenda-

tion systems in the freelance marketplace by proficiently connecting freelancers with appropriate projects. This study generated a realistic dataset by emulating user behavior and integrating actual market trends. Leveraging this dataset, we devised three distinct approaches: Content-Based Filtering utilizing Cosine Similarity, Collaborative Filtering employing Singular Value Decomposition (SVD), and a Word2vec technique. Each method exhibited specific strengths that allowed for capturing user-item interactions and understanding semantic connections within project descriptions. The investigation then delved into the realm of hybrid recommendation systems, ingeniously combining the strengths of the three approaches via ensemble learning techniques. By applying a weighted average method, we were able to utilize the predictive power of each approach while compensating for their respective limitations, resulting in a sophisticated and adaptable recommendation system that provides personalized project suggestions to freelancers. In the evaluation phase, we thoroughly examined our models' performance and accuracy, confirming our method's validity and offering valuable perspectives for further improvement. Deploying recommendation systems (RSs) within the freelancing sphere poses various challenges. Managing data scarcity, resolving the cold start issue for newcomers, and ensuring scalability are critical for a proficient RS. Subsequent research in this field should aim to devise innovative algorithms to surmount these obstacles and elevate the precision and efficiency of recommendation systems. Furthermore, incorporating additional data sources, such as user evaluations and sentiment analyses, could significantly refine the quality of the project recommendations.

Author Contributions: Conceptualization: Ismail Sahnoun and Emna Ammar Elhadjamor; methodology: Ismail Sahnoun and Emna Ammar Elhadjamor.; software: Ismail Sahnoun; validation: Ismail Sahnoun and Emna Ammar Elhadjamor; formal analysis: Ismail Sahnoun and Emna Ammar Elhadjamor; resources: Ismail Sahnoun; data curation: Ismail Sahnoun; writing: Ismail Sahnoun and Emna Ammar Elhadjamor; writing—review and editing: Emna Ammar Elhadjamor; supervision: Emna Ammar Elhadjamor

Funding: This research received no external funding

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

References

- [1] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–58, Mar. 1997, doi: 10.1145/245108.245121.
- [2] V. Gupta and S. R. Pandey, "Recommender systems for digital libraries: A review of concepts and concerns," *Libr. Philos. Pract.*, 2019.
- [3] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992, doi: 10.1145/138859.138867.
- [4] A. Klačnja-Milićević, M. Ivanović, B. Vesin, and Z. Budimac, "Enhancing e-learning systems with personalized recommendation based on collaborative tagging techniques," *Appl. Intell.*, vol. 48, no. 6, pp. 1519–1535, Jun. 2018, doi: 10.1007/s10489-017-1051-8.
- [5] L. Balby Marinho *et al.*, *Recommender Systems for Social Tagging Systems*. Boston, MA: Springer US, 2012. doi: 10.1007/978-1-4614-1894-8.
- [6] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, Jan. 2004, doi: 10.1145/963770.963772.
- [7] M. Fabbri, "Social influence for societal interest: a pro-ethical framework for improving human decision making through multi-stakeholder recommender systems," *AI Soc.*, vol. 38, no. 2, pp. 995–1002, Apr. 2023, doi: 10.1007/s00146-022-01467-2.
- [8] M. S. Hossain and M. Shamsul Arefin, "Development of an Intelligent Job Recommender System for Freelancers using Client's Feedback Classification and Association Rule Mining Techniques," *J. Softw.*, pp. 312–339, Jul. 2019, doi: 10.17706/jsw.14.7.312-339.
- [9] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005, doi: 10.1109/TKDE.2005.99.
- [10] F. Narducci, C. Musto, M. Polignano, M. de Gemmis, P. Lops, and G. Semeraro, "A Recommender System for Connecting Patients to the Right Doctors in the HealthNet Social Network," in *Proceedings of the 24th International Conference on World Wide Web*, May 2015, pp. 81–82. doi: 10.1145/2740908.2742748.
- [11] J. Han, J. Jo, H. Ji, and H. Lim, "A collaborative recommender system for learning courses considering the relevance of a learner's learning skills," *Cluster Comput.*, vol. 19, no. 4, pp. 2273–2284, Dec. 2016, doi: 10.1007/s10586-016-0670-x.

- [12] J.-K. Kim, J.-S. Lee, D.-K. Park, Y.-S. Lim, Y.-H. Lee, and E.-Y. Jung, "Adaptive mining prediction model for content recommendation to coronary heart disease patients," *Cluster Comput.*, vol. 17, no. 3, pp. 881–891, Sep. 2014, doi: 10.1007/s10586-013-0308-1.
- [13] S. Smys and G. J. Bala, "Performance analysis of virtual clusters in personal communication networks," *Cluster Comput.*, vol. 15, no. 3, pp. 211–222, Sep. 2012, doi: 10.1007/s10586-012-0209-8.
- [14] R. Yera and L. Martínez, "A recommendation approach for programming online judges supported by data preprocessing techniques," *Appl. Intell.*, vol. 47, no. 2, pp. 277–290, Sep. 2017, doi: 10.1007/s10489-016-0892-x.
- [15] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, "A review on deep learning for recommender systems: challenges and remedies," *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 1–37, Jun. 2019, doi: 10.1007/s10462-018-9654-y.
- [16] S. Benhamdi, A. Babouri, and R. Chiky, "Personalized recommender system for e-Learning environment," *Educ. Inf. Technol.*, vol. 22, no. 4, pp. 1455–1477, Jul. 2017, doi: 10.1007/s10639-016-9504-y.
- [17] H. Ronghuai, J. Yang, and H. Yongbin, "From Digital to Smart: The Evolution and Trends of Learning Environment," *Open Educ. Res.*, 2012, [Online]. Available: <https://api.semanticscholar.org/CorpusID:111567973>
- [18] M. Vahdat, L. Oneto, D. Anguita, M. Funk, and M. Rauterberg, "A Learning Analytics Approach to Correlate the Academic Achievements of Students with Interaction Data from an Educational Simulator," in *Design for Teaching and Learning in a Networked World*, 2015, pp. 352–366. doi: 10.1007/978-3-319-24258-3_26.
- [19] G. Donzellini and D. Ponta, "A Simulation Environment for e-Learning in Digital Design," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3078–3085, Dec. 2007, doi: 10.1109/TIE.2007.907011.
- [20] H. Li, H. Li, S. Zhang, Z. Zhong, and J. Cheng, "Intelligent learning system based on personalized recommendation technology," *Neural Comput. Appl.*, vol. 31, no. 9, pp. 4455–4462, Sep. 2019, doi: 10.1007/s00521-018-3510-5.
- [21] Y. Zhang, D. Lo, X. Xia, G. Scanniello, T.-D. B. Le, and J. Sun, "Fusing multi-abstraction vector space models for concern localization," *Empir. Softw. Eng.*, vol. 23, no. 4, pp. 2279–2322, Aug. 2018, doi: 10.1007/s10664-017-9585-2.
- [22] H. Chen, Z. Li, and W. Hu, "An improved collaborative recommendation algorithm based on optimized user similarity," *J. Supercomput.*, vol. 72, no. 7, pp. 2565–2578, Jul. 2016, doi: 10.1007/s11227-015-1518-5.
- [23] J. Huang and Z. Huang, "Personalized recommendation algorithm for e-commerce based on user behavior," *J. Phys. Conf. Ser.*, vol. 62, no. 1, 2015.
- [24] S. Karga and M. Satratzemi, "A hybrid recommender system integrated into LAMS for learning designers," *Educ. Inf. Technol.*, vol. 23, no. 3, pp. 1297–1329, May 2018, doi: 10.1007/s10639-017-9668-0.
- [25] K. Clements, J. Pawlowski, and N. Manouselis, "Open educational resources repositories literature review – Towards a comprehensive quality approaches framework," *Comput. Human Behav.*, vol. 51, pp. 1098–1106, Oct. 2015, doi: 10.1016/j.chb.2015.03.026.
- [26] S. Bennett, S. Agostinho, and L. Lockyer, "Technology tools to support learning design: Implications derived from an investigation of university teachers' design practices," *Comput. Educ.*, vol. 81, pp. 211–220, Feb. 2015, doi: 10.1016/j.compedu.2014.10.016.
- [27] K. Abhinav, A. Dubey, S. Jain, G. Virdi, A. Kass, and M. Mehta, "CrowdAdvisor: A Framework for Freelancer Assessment in Online Marketplace," in *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*, May 2017, pp. 93–102. doi: 10.1109/ICSE-SEIP.2017.23.
- [28] D. E. Difallah, G. Demartini, and P. Cudré-Mauroux, "Pick-a-crowd: tell me what you like, and i'll tell you what to do," in *Proceedings of the 22nd international conference on World Wide Web*, May 2013, pp. 367–374. doi: 10.1145/2488388.2488421.