

PENREC: A RECOMMENDER SYSTEM FOR PENSION SAVINGS

Onaopepo Adekunle¹, Chang Sun¹, Arno Riedl², Michel Dumontier¹

¹Department of Advanced Computing Sciences, Maastricht University, Netherlands

²Behavioral Economics Department, Maastricht University, Netherlands

ABSTRACT

Recommender systems are ubiquitous in various domains ranging from e-commerce to financial services to deliver personalized products and services at scale. This work designs and develops a proof of concept for personalized pension recommendations. PenRec, a novel pension savings recommender system, takes the combination of individuals' demographic data and risk preferences data as input. PenRec combines collaborative filtering and deep learning methods to provide individual savings recommendations based on individuals with similar characteristics as pension savings advice. To estimate the risk preferences of individuals, we implemented a survey, including incentivized economic experiments, among the Dutch working population (N = 4, 282); upon which we employed supervised learning techniques to estimate the risk preferences of the general population of the Netherlands based on register data from Statistics Netherlands. We present an analysis of the comparison between traditional collaborative filtering and deep learning for pension savings recommendations. Additionally, we investigate whether the inclusion of a behavioral measure as a feature, estimated risk preference in particular together with demographic data is important for pension savings recommendation. We find that traditional collaborative filtering tends to recommend average values of individual savings for specified categories of people such as self-employed, income source, and province with no deviations for such categories; while the deep learning-based recommender system can capture varying categories of people in the group of similar individuals and hence provides recommendations that track the real savings more closely relative to the traditional collaborative filtering method. In addition, we observe the inclusion of risk preference variable as a feature with which to compute individual similarity impacts the recommended savings marginally and encourages the addition of behavioral measures for a robust recommendation.

KEYWORDS

Recommender System, Pensions, Collaborative filtering, Deep Learning

1. INTRODUCTION

In terms of pensions, OECD countries, like the Netherlands, typically provide state pensions, encourage private pension savings, and have employer/employee pension contributions. In the case of employer/employee pension contributions, the self-employed are often not covered. They tend to save too little voluntarily and may have non-substantial business equity. Additionally, their vulnerability in the housing market makes them susceptible to inadequate pension [1]. This can result in a lower quality of life upon retirement. Moreover, the plethora of choices in individual pension funds and diverse pension scheme products can lead to information overload. A recommender system that provides advice is useful in this context to improve individual pension savings, especially among the self-employed. Recommender systems are designed to assist users in navigating an extensive array of options, and effectively select items that align with an individual's preferences or needs [2].

Recommender systems attempt to recommend the most suitable items (products or services) to particular users (individuals or businesses) by predicting a user's interest in an item based on related information about the users, items, and the interactions between items and users [3]. The core feature of a recommender system lies in its ability to "guess" a user's preferences and

interests by analyzing the behavior of this user and/or the behavior of other users to generate personalized recommendations [4]. Recommendation models are mainly categorized into collaborative filtering, content-based recommender system, and hybrid recommender system [5].

The rapid advancement of recommendation techniques has spurred the implementation of numerous real-world recommender systems. It was recently pointed out that application study is the primary focus of current recommender system research, particularly in the era of big data [3], [6]. The practical applications of recommender systems include recommending movies, music, television programs, books, documents, websites, conferences, tourism scenic spots, and learning materials, and involve the areas of e-commerce, e-learning, e-library, e-government, e-business, and financial services [7]. Despite the success of recommender systems in various research and development applications, there is still room for further research to implement them in domains that have yet to leverage the power of recommenders, especially in the pensions domain. The challenges include the long-time series characteristics of pension data with different modalities and uncertain outcomes relative to expected life expectancy after retirement. This motivates our work to develop and determine the most suitable models of recommender systems for the domain. To the best of our knowledge, no work has yet been dedicated to the recommendation of personalized pension savings using deep learning based on individuals with similar characteristics.

This chapter proposes a new deep learning-based hybrid recommender system for pension savings. We combine two types of collaborative filtering models that are a memory-based nearest neighbor model and a deep learning-based recommender system. As such, the first contribution of our study is that we propose a novel framework for the recommendation of pension savings using demographic data from the Dutch population. The second contribution is that we study the impact of including behavioral measures such as risk preference as a feature for recommendation. Finally, we study the differences between using traditional recommendation techniques to deep learning-based recommendation.

The remainder of the chapter is structured as follows. In Section 2, we discuss the state of the art and methodologies of recommender systems especially for financial services. Section 3 discusses the framework and implementation of the recommender system. In Section 4, we discuss the results and Section 5 covers the conclusion.

2. RELATED WORK

Recommender Systems is rooted in information retrieval [8], forecasting theories [9], user option modeling in business [10] and cognitive science [11]. In the mid-1990s, Recommender Systems appeared as an independent research domain [5]. An illustrative representation of the different kinds of recommender system is provided in Figure 1.

2.1. Content Based Recommenders

In content-based recommenders [12], recommendations are made based on the relationship between the description of the target outcome and the profile of the user's preferences and past experience [13]. A weighted vector list of item features and target descriptions is often generated to determine similar items to the ones that the user has preferred in the past. The weight represents the interest or likeness of the user toward each type of target. The weighted vector can be calculated from features of the items and user ratings of the outcome using simple techniques like average values of the target vector or more sophisticated techniques such as cluster analysis and Bayesian classifiers. Content-based filtering algorithms recommend items based on target similarity. An analysis of different item-based and item-item similarity-based recommendation algorithms is provided by Sarwar and co-authors [14]. Results were compared with baseline k-nearest neighbor model and the authors concluded that item-based techniques provide better performance and quality suggestions (suggestions that users vet as useful) than user-based techniques.

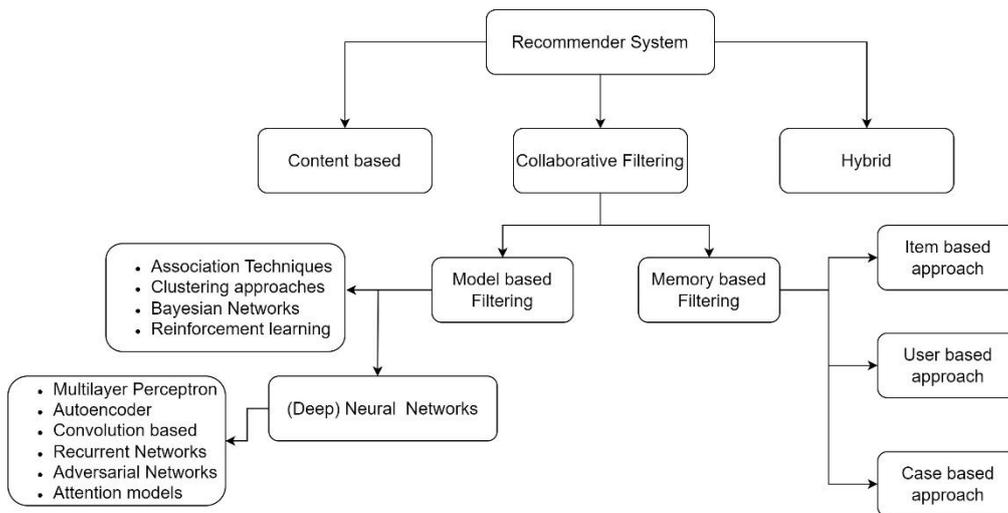


Figure 1: Classification of recommendation techniques.

2.2. Collaborative Filtering

Collaborative filtering provides recommendations based on the similar attributes of other people who share similar interests [15], [16]. Collaborative filtering techniques are based on a large amount of user information including their interests, preferences, activities, and behaviors to establish the information and tastes of a particular user. Prediction is made using the similarity with other users. Collaborative filtering for the task of recommending pension savings for example is based on the assumption that people with similar demographics and behavioral measures can afford to make similar savings for their pensions.

Collaborative filtering is divided into model-based and memory-based approaches. Memory-based collaborative filtering employs either all or a subset of user characteristics to compute a utility matrix. Upon this matrix, models such as the nearest neighbor and random forest are utilized for prediction. For instance, [17] applied a user-based nearest neighbor algorithm to recommend insurance products using demographic data. They claim that random forests provide lower prediction error and thus recommendation based on random forest is superior to other models. Memory-based collaborative filtering is further subdivided into user-based approach, item-based approach, and case-based approach. The user-based approach employs a cluster of similar users based on user preferences, while the item-based approach computes the average of all the target values of similar users to make a recommendation. Both methods have been implemented in [18] to study their performance in identifying the products a customer will buy next as a means to a recommendation of cross-selling offered by a financial services provider. They found out that user user-based approach performed better than the item-based approach. The case-based approach [19], [20] applies case-based reasoning (CBR) that solves the recommendation problem based on similar cases in the past. A case is defined in various ways, like product description, user preference, search criteria, and outcome of case. To find similar cases, similarity of attributes, collaborative patterns, or knowledge of the domain are usually applied. The work by [21] proposes a CBR application for recommending insurance policies. Some other works such as [22], [23] introduce case-based reasoning for portfolio recommendation. In their works, the authors propose a diversification technique for weighting candidate solutions in revised steps.

Model-based collaborative filtering employs machine learning techniques to develop a model for predicting the target variable. A model is learned from the user attributes and the model function is used to predict the target of new users. Such systems can make recommendations to users that were not present in the dataset, it does not require re-computation of similarities for every new user and is often more efficient for group recommendations.

Among collaborative filtering-based solutions, the majority of works apply item-based nearest-neighbor methods for recommending insurance riders [24], real estate [25], and venture capital [26]. There is also the preliminary concept of applying similarity-based recommendations for the stock market [27]. [28] apply matrix factorization for Bayesian personalized ranking in microfinance services. They propose a fairness-aware optimization with stochastic gradient descent. A significant contribution is published by [29], who propose five different collaborative filtering methods for the venture capital domain.

2.3. Hybrid Recommenders

Hybrid recommenders combine two or more approaches to capitalize on their respective strengths and mitigate their weaknesses. Hybrid methods can be more precise than conventional models. However, the efficient implementation is more complex because the process of combination adds more layers of complexity to the system. For example, a hybrid recommender focused on finding the most profitable stocks at the right time based on investor preference developed by [30]. They apply collaborative and content-based filtering at the algorithm level and social, economic, and semantical agents at the system level. Other examples employ the same combination for recommending insurance products [31], microfinancing [32], and stocks [33]. In recent years, we have witnessed the success of deep learning in solving many complex tasks in a wide range of applications such as speech and face recognition, natural language processing tasks, and recommendation problems [34]. Typically, Recurrent Neural Networks (RNNs) [35] Convolution Neural Networks (CNNs) [36], transformers [37] and Graph Neural Networks (GNNs) [38] are the dominant approaches in basic deep neural networks. There are seven main mechanisms of combinations used in recommender systems to build hybrids [39]: weighted [40], mixed [41], switching [42], feature combination, feature augmentation [42], [43], cascade [44] and meta-level [45]. Hybrid recommenders with deep learning have provided state-of-the-art results and the work presented here develops such an application for the pensions domain.

3. METHODOLOGY

3.1. Data

The data consists of demographic register data of the Dutch population from Statistics Netherlands and risk preference behavioral measures from a survey including incentivized economic tasks¹. The survey data were collected in a two-wave online survey in May and June of 2020 conducted in collaboration with Statistics Netherlands and research agency Flycatcher² Statistics Netherlands selected the stratified random sample, which allowed us to link the survey and experimental data with register data. Flycatcher programmed the online survey and experiments and collected the data. A total of 18,000 Dutch employees and 18,000 self-employed were randomly selected and invited through physical letters to participate in the online study. In total, 4,282 Dutch residents completed both waves and this represents the survey data. The survey data is then used to develop a machine learning model for predicting the risk preferences for the general Dutch population according to [46] as a feature together with 65 demographic and socio-economic variables. All features are categorized into financial data such as income, wealth, and debt, demographic data such as marriage status, occupation, and education, and behavioral measure which is the risk preference measure. The data are further divided into household data and individual data. The appendix provides a full list of features.

We have 22 categorical variables and 44 numerical variables and sampled 9,000,000 individuals randomly from the Dutch population data for the year 2020. We randomly select a test set of 100,000 from the subset of individuals with pension savings in a pension fund. Originally, there were a total of 79 features including the predicted risk preference and various features are similar to one another with different granularity. We apply a heuristic to keep the most granular variable out of similar ones and eliminate the rest. Thirteen features were eliminated including year, 3 occupation variables, indexes, 5 education level categories, comment on income relationship, and 2 employee sector variables. Out of the rest of the features, there is 55% missing data in the employee and employer pension contribution features. We replace the missing with zero in these features. All the rest have less than 1% missing and as such, any row with even one of the values missing here is removed. 99% of the data points remain after the full process.

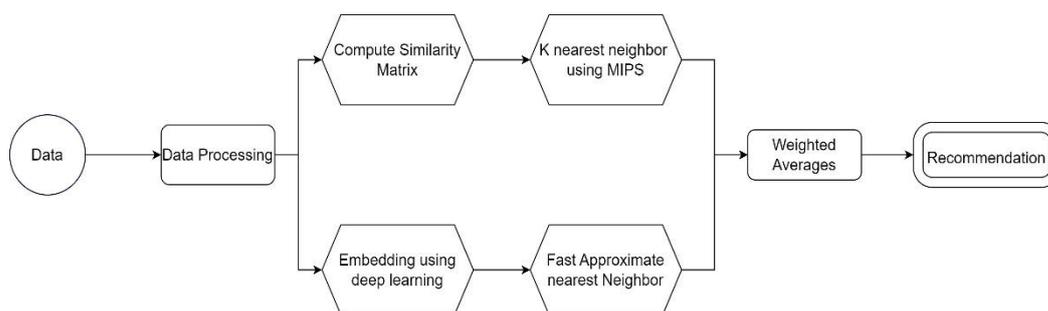
3.2. Proposed Framework

The proposed framework consists of a memory-based nearest neighbor model combined with a deep learning architecture illustrated in Figure 2. The memory-based nearest neighbor model employs a similarity matrix computed by transforming the categorical features to their dummy equivalent using one-hot encoding and numeric features to a twenty-bin quantile and normalizing to a range between 0 and 1. In other words, users' preferences are represented by the demographic variables¹, financial variables, and risk preference variables which are either quantiles or dummy values. Then, a dot product (since the similarity matrix is normalized, the dot product is equivalent to the cosine similarity) using maximum inner product search (MIPS) algorithm is used to compute similarity. Consider the similarity matrix where $X = \{X_i\}_{i=1,2,\dots,n}$ with n users and $x_i \in \mathbb{R}^d$ for each user in d -dimension of feature space. In the MIPS setup, given a query $q \in \mathbb{R}^d$, we would like to find a user $x \in X$ that has the highest inner product with q , i.e.

$$x_i^* = \arg \max_{x_i \in X} \langle q, x_i \rangle$$

This implies a full search with all other users which is computationally expensive with a big data user base. The recommendation is made by the standard K-nearest neighbor (KNN) algorithm:

- Step 1: Find similarity scores between a user and all other users.
- Step 2: Sort and choose the top K nearest neighbors of the user.
- Step 3: Take the mean as the predicted recommendation.



¹ A summary of data available from CBS Netherlands can be found at <https://www.cbs.nl/en-gb/our-services/customised-services-microdata/microdata-conducting-your-own-research/microdata-catalogue>

² complete overview of the material is available at <https://sites.google.com/view/preferences-beliefs-biases-nl/experimental-materials/materials-main-study>.

Figure 2: Framework architecture for pension savings recommenders system.

The proposed deep learning architecture consists of three hidden layers fully connected as illustrated in Figure 3. The proposed architecture uses embeddings which allow the deep learning algorithm to learn a richer representation for the users and the items jointly. This is similar to the natural language modeling approaches where each word is mapped to a vector space of a fixed size. This approach has been successfully employed for YouTube recommendations using deep neural networks [47], which are representations of each user x_i as vectors in N -dimensional space. Embeddings are lookup matrices that return an array of size D , where D is the cardinality of the embeddings. The input data to the neural network is the processed dataset where the categorical variables are mapped to integers called label encoding. The number of hidden layers, the number of nodes in each hidden layer, and the activation functions in each layer are all hyper-parameters that are to be specified a priori and tuned by the modeler. Each hidden layer computes a linear function followed by a batch normalization [48], a LeakyReLU function [49] with a dropout which acts as a method to control overfitting. The LeakyReLU function was applied because it shares the ease of computation of ReLU with the advantage that it does not suffer from the vanishing gradients problem. Vanishing gradients refer to the case where the gradients become zero or very close to zero preventing the network from learning. LeakyReLU provides a lower bound on the value of the computed gradients. The last layer is a fully connected linear layer embedded with dimension $k \leq d$ where d is the input feature space acting as a form of similarity matrix. The embedding is built using contrastive learning. Contrastive learning teaches the model to learn an embedding space in which similar users are close while dissimilar ones are far apart, i.e., users with similar attributes are pulled together, while more distinct users are pushed apart from each other. This yields a clustered space where the distance between similar users is small and the distance between dissimilar users is large. The recommendation is done by a fast approximate nearest neighbor [50] for efficiency reasons in combination with the KNN algorithm as described above. The appeal of this approach is that, in many cases, an approximate nearest neighbor is almost as good as the exact one with high efficiency.

Finally, both approaches are combined using the weighted average of the recommendations. The memory-based nearest neighbor method is termed as the traditional recommender while the deep learning method is the advanced recommender system.

3.3. Experiment

3.3.1. Baseline Model

The evaluation of the performance of the proposed model is achieved by comparing the recommendation to a reference. We adopt a baseline savings rate of 10% of gross income as the target variable which serves as a reference for recommended savings. This choice is motivated by lifecycle consumption-smoothing theory, which under standard assumptions implies optimal retirement saving rates in the range of 8–15% for median earners. Moreover, a 10% contribution rate aligns with commonly used policy and institutional benchmarks in international pension systems and is frequently employed in empirical and behavioral studies as a reference point [51]. Importantly, the baseline is not intended to represent an optimal rate for all individuals, but rather a neutral and behaviorally plausible anchor against which heterogeneity in savings behavior and alternative contribution scenarios can be assessed. Another baseline we used is the variable of

pension fund savings which is the amount saved by an employee through employer-employee contributions to pension savings or the private contributions of self-employed individuals to pension savings. As such, we selected 100,000 users as the test set from the set of users that have pension fund savings.

3.3.2. Computing Environment

All computations were performed on a shared cluster server with 500 GB RAM, and 16 cores of Intel Xeon Gold 6248R running Ubuntu 18.04.6 LTS. Given the large size of data from 9 million users, python libraries like Polars for data wrangling, Scalable Nearest Neighbors (ScaNN) for efficient vector similarity search at scale [52], Tensorflow similarity were leveraged to train neural network similarity model and search at scale.

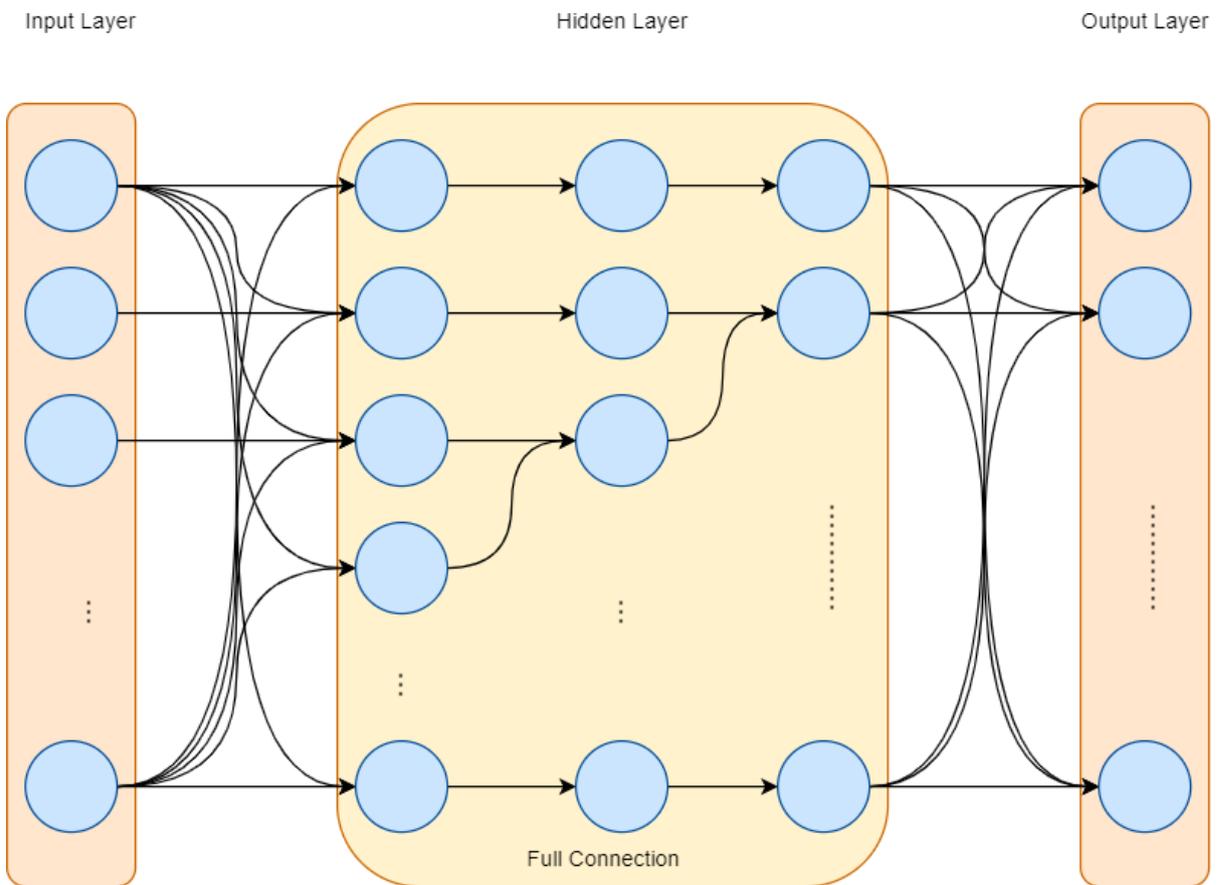


Figure 3: Fully connected Deep learning architecture for pension savings recommenders system.

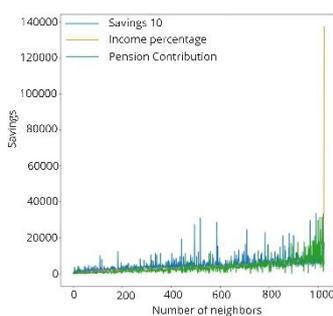
4. RESULTS AND DISCUSSION

This section illustrates the selection of hyperparameters over varying ranges including number of neighbors for nearest neighbor recommender and embedding size for the deep learning recommender, to glean the trends and insights about the recommendation over different configurations in terms of inclusion/exclusion of the risk preference variable. We applied 10-fold cross-validation with the goal of finding the model that provides recommendations closest to the baselines of 10% income savings and the pension fund savings variable for the prototype recommender system.

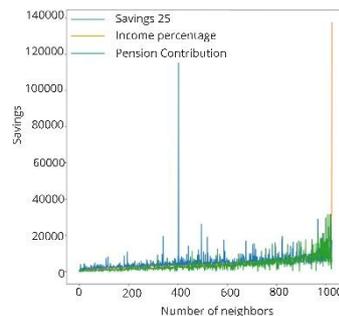
4.1. Nearest Neighbor Recommender

In the Nearest Neighbor recommender system, the only hyperparameter is the number of neighbors k for the KNN algorithm which was varied from 10 to 500. In Figure 4, the average difference between the recommended savings and the baselines are lower as k increases, and as a result, the blue plotted values becomes less prominent compared to the baselines less as k increases from 10 to 500. This means that the higher the number of neighbors selected, the more robust the recommendation to outliers for example, suppose we have 500 individuals in the dataset as we vary from 10 to 500 we go from few similar neighbors to all individuals as neighbors, including the outliers. However, at higher values of k , the model over-compensates for individuals with higher incomes at the expense of lower-income earners. Referring to the aforementioned example, if just two out of the 500 individuals earn over €1,000,000 per-year they will significantly inflate the average earning of the neighbors. Cross-validation on minimization of the difference between the recommended savings and the baselines yielded an optimal number of neighbors as being 25, hence we use this value for the final nearest neighbor recommender system model for both employed and self-employed individuals.

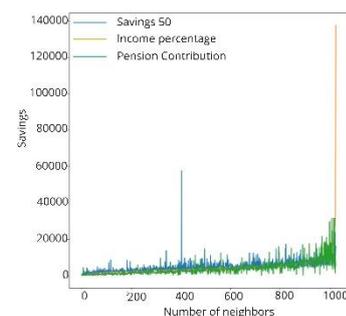
For the group of employees which constitutes the largest group, the recommended savings are mostly below or approximately the actual pension funds savings and 10% income saving baselines. Table 1 illustrates this at the aggregate level with an average difference of about €759 between the recommended savings of the nearest neighbor recommender and the 10% income savings. However, there is an average difference of €414 between the recommended savings greater than the pension savings among the test set of selected individuals. This implies that the recommender system encourages additional savings than the current employee contribution to their pension but less than 10% income savings. We believe this recommendation to be reasonable given that the recommendations are still below the 10% income savings and our data does not capture the employer contribution.



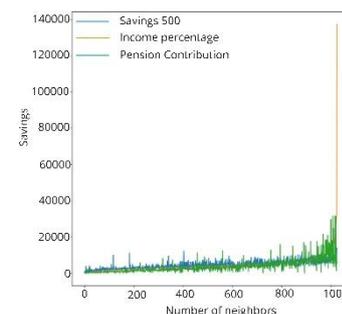
(a) $K = 10$



(b) $K = 25$



(c) $K = 50$



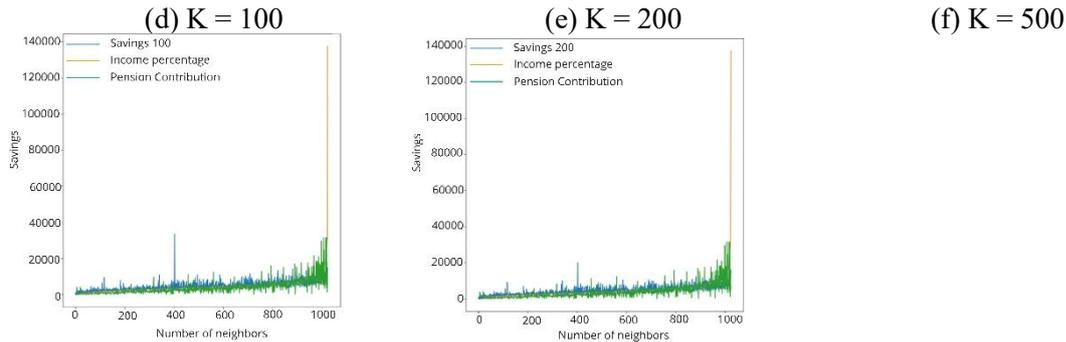


Figure 4: Number of Nearest Neighbors for the traditional recommender

While the self-employed tend to have lower recommended savings compared to their employed counterparts, there is a difference of about €550 on average between the recommended savings relative to the 10% income savings and €3,280 with regards to the pension savings. The average recommendation for this group is higher than both baselines. Particularly, the latter difference seems rather high and we hypothesize this could be due to but not limited to two main reasons. Firstly, it is observed and known from the literature that self-employed individuals vastly under-save for their pensions in the Netherlands [53], [54], [55] and in the world at large [56], [57]. The other potential reason is that individuals who are labeled self-employed have more extreme cases of individuals with lower incomes with for example a struggling small business and cases of very successful businesses with rather high incomes. These huge disparities can have a strong effect on nearest-neighbor algorithms with a mandated number of neighbors. For the Director shareholder, the recommended savings are on average lower than both baselines with an amount of about €5,473 relative to the 10% income savings and €4,663 with regards to the pension savings. The potential reasons for the high values here in comparison to the self-employed are that the second potential reason above also affects this group and some of their earnings are in the form of organizational assets. In the case of individuals that receive some sort of benefit shown in Table 1, the recommended savings generally lie below the baselines on average for individuals receiving social assistance benefits and they lie between the baselines for individuals that receive incapacitated benefits and those that receive unemployed benefit.

Hence, from Table 1 three groups have recommended savings in between the baselines which are the employees, receivers of incapacitated benefits, and unemployment benefits, while two groups have lower recommendations which are the director shareholders and receivers of social assistance benefits, and finally the self-employed group with higher recommendations with regards to the baseline. The general recommended savings tend to be slightly lower than the baselines. We conclude that the general recommendations are reasonable and mostly lie within the range of the baselines also observed that the nearest neighbors selected by the traditional recommender model mostly share the same education category, employment type, province, and income status.

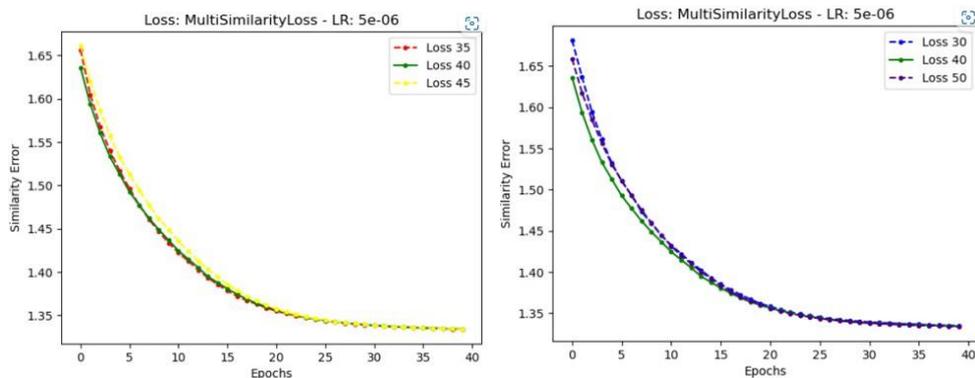
4.2. Deep Learning Recommender

The hyperparameters in a deep learning recommender system are the number of hidden layers, the number of nodes in each hidden layer, and the activation functions in each layer together with the embedding dimension, number of epochs, learning rate, and dropout rate. All hyperparameters have been evaluated with 10-fold cross-validation using 3 hidden layers with 128 nodes on the first layer, 64 nodes on the second layer, and 32 nodes on the third layer, all applying batch normalization and LeakyRELU activation function. This model is intuitively justified as a dimensionality reduction task that captures the nonlinear relationships in the data. In other to structure the output in the form of a similarity matrix that can be combined with the nearest neighbor model and to prevent biasing the model with the baseline savings like the model memorizing the target values, we excluded the 10% income savings and pension contributions as

target variables to minimize error. As such, the new target variable is the employee group, which is a categorical variable, and thus the multisimilarity loss function is applicable to the model. The number of epochs is 40, the learning rate is $5e-06$ and the dropout rate is 0.3.

Figure 5 illustrates the similarity loss plot for varying embedding dimension sizes. On the Y-axis is the similarity error plotted against the number of iterations (epochs) with the embedding sizes as legends. We observe that a lower embedding size generally implies a higher error value at the start and the convexity of the similarity error curve is slightly lower. The optimum embedding size seems to be 40 because in Figure 5(a), it has the lowest starting error every time and converges slightly faster at about 30 epochs, in Figure 5(b) it retains the lower starting error and is the most convex implying better convergence rate. Finally, in Figure 5(c), the convergence rate is quite similar to the embedding size of 60 which implies there is no more gain increasing the embedding size.

Upon inclusion of the deep learning recommender savings over the different employment groups illustrated in Table 1; we observe an average recommended savings difference of approximately e 100 relative to the average pension contribution compared to the average KNN recommended savings difference of at least e 400 for the employees group. It implies that the deep learning recommender makes recommendations closer to the actual pension contributions than the 10% income savings. However, the average deep learning recommendation is at least €2,000 greater than the current pension contribution on average. The average recommendation lies in between the baselines and captures the idea that the self-employed group need to save more than the current savings without overshooting the 10% income savings baseline.



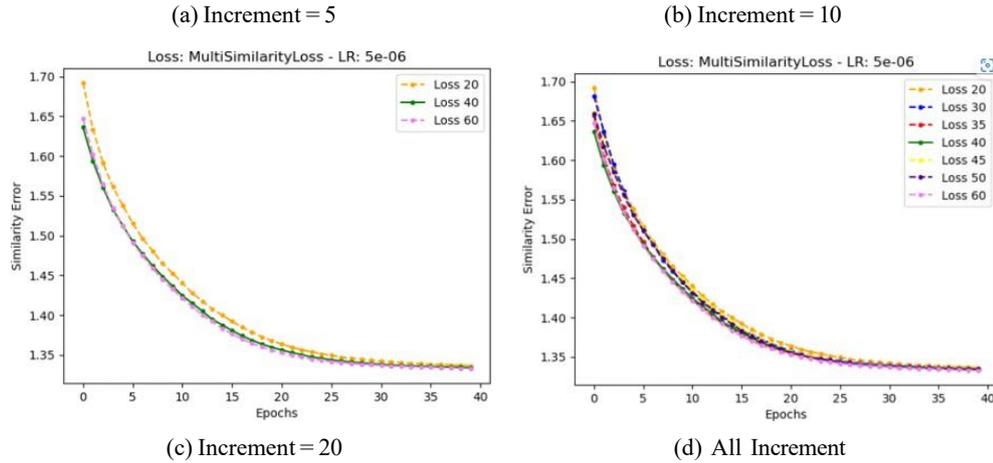


Figure 5: Similarity loss for different embedding sizes in the deep learning recommender

This is potentially due to the neural network overcoming the aforementioned shortcoming of the nearest neighbor algorithm. However, the average recommended savings remains significantly lower by at least €4,000 across both baselines for the director shareholder group, which is also true for the KNN recommended savings. We already made the case for the potential reasons for the observation above which still holds true for this case as well. In the case of individuals that receive some sort of benefit shown in Table 1, the deep learning recommended savings is leans towards the average pension contribution and generally lie between the baselines on average.

Table 1: Average individual savings per year for each recommender system

| | Average 10% Income savings per year (€) | Average pension contribution per year (€) | Average recommended savings per year (€) |
|---------------------------------------|--|--|---|
| (i) KNN Recommender | | | |
| Employee | 7,908 [+759] (+31) | 6,735 [-414] (-1,142) | 7,149 (7,877) |
| Self-Employed | 4,400 [-550] (-1,200) | 1,670 [-3,280] (-3,930) | 4,950 (5,600) |
| Director-Shareholder | 12,543 [+5,473] (+6,013) | 11,733 [+4,663] (+5,203) | 7,070 (6,530) |
| Social Assistance Benefit | 1,970 [+1,050] (+610) | 1,010 [+90] (-350) | 920 (1,360) |
| Incapacitated Benefit | 4,975 [+1,595] (+305) | 2,050 [-1,130] (-2,620) | 3,380 (4,670) |
| Unemployment Benefit | 3,767 [+347] (+217) | 2,833 [-587] (-717) | 3,420 (3,550) |
| (ii) Deep Learning Recommender | | | |
| Employee | 7,908 [+1,252] (+1,073) | 6,735 [+79] (-100) | 6,656 (6,835) |
| Self-Employed | 4,400 [+580] (+190) | 1,670 [-2,150] (-2,540) | 3,820 (4,210) |
| Director-Shareholder | 12,543 [+5,708] (+6,593) | 11,733 [+4,898] (+5,783) | 6,835 (5,950) |
| Social Assistance Benefit | 1,970 [+1,080] (+1,000) | 1,010 [+120] (+40) | 890 (970) |
| Incapacitated Benefit | 4,975 [+2,485] (+1,105) | 2,050 [-440] (-1,820) | 2,490 (3,870) |
| Unemployment Benefit | 3,767 [+1,037] (+103) | 2,833 [+103] (-153) | 2,730 (2,986) |

Notes: Table values were calculated with $K = 25$ for the KNN recommender and embedding size = 40 for the deep learning recommender for randomly selected 1000 individuals from the test set and categorized according to the employment status and rounded up to a whole number. In parentheses is the savings values and differences when the risk preference variable is excluded from the model and in brackets is the difference between the recommended savings and the baselines.

Over both models, we observe that the deep learning recommender has lower average values than the KNN recommender and in all groups but the director shareholder group the average values lie between the baselines. We also took a random sample of employee and self-employed and discovered that some of them had neighbors that did not match their employee group which is not the case for the nearest neighbor model. As such, the deep learning recommender tends to include

a diverse set of individuals in the list of nearest neighbors and is not limited to individuals with the same class of most of the categorical variables. This shows the deep learning recommender is more robust in terms of selection of similar individuals than the nearest neighbor recommender. However, the nearest neighbor recommender has an easier interpretation, less complexity, and less robustness given the strict categories of similar individuals based on education, employment type, province, and employee group as opposed to the deep learning recommender. Hence, for the combination of both models, a higher weight can be placed on each of the models depending on the desired properties of interpretability, complexity, and robustness.

4.3. Effect of Exclusion of Risk preference

In Table 1, we observe the influence of the exclusion of the risk preference variable in both models, which is the representative variable of the behavioral attributes. It affects the recommended savings significantly, with average difference of over e 100 over all employee groups. Figure 6 also shows that the blue plot which represents the recommendations for varying numbers of similar neighbors is generally more haphazard and wiggly than in Figure 4. This indicates the inclusion of the risk preference enhances the stability of recommendations provided by the model. However, the tendency for the nearest neighbors to have the same education, employment type, province, and income status remains the same and so robustness seems unaffected by the inclusion of risk preferences.

In the case of the deep learning model, the final similarity error is minimized to 1.38 as compared to 1.35 with the inclusion of risk preference over 40 epochs for embedding output of 40 dimensions. From Table 1, the deep learning recommended savings mostly have lower average differences across the table for the pension savings and the 10% income savings relative to the recommendation by the traditional recommender. In particular, the inclusion of risk preference generally skews the recommendation closer to the existing pension savings baseline but it is not always the case for the 10% income savings, it is only among director shareholders that excluding risk preference leads to higher average difference between the deep learning model with risk preference and the deep learning model without risk preference. Hence, we find that the effect of the risk preference is not consistently lower for the 10% income savings baseline, but the opposite is true for the actual pension savings. There is evidence to the statement that the deep learning model marginally makes use of the risk preference feature for the selected individuals of the test set. A clear statement of the importance of the risk preference is not obvious but this analysis supports the hypothesis that the inclusion of behavioral measures provides value for recommendations of the overall hybrid model.

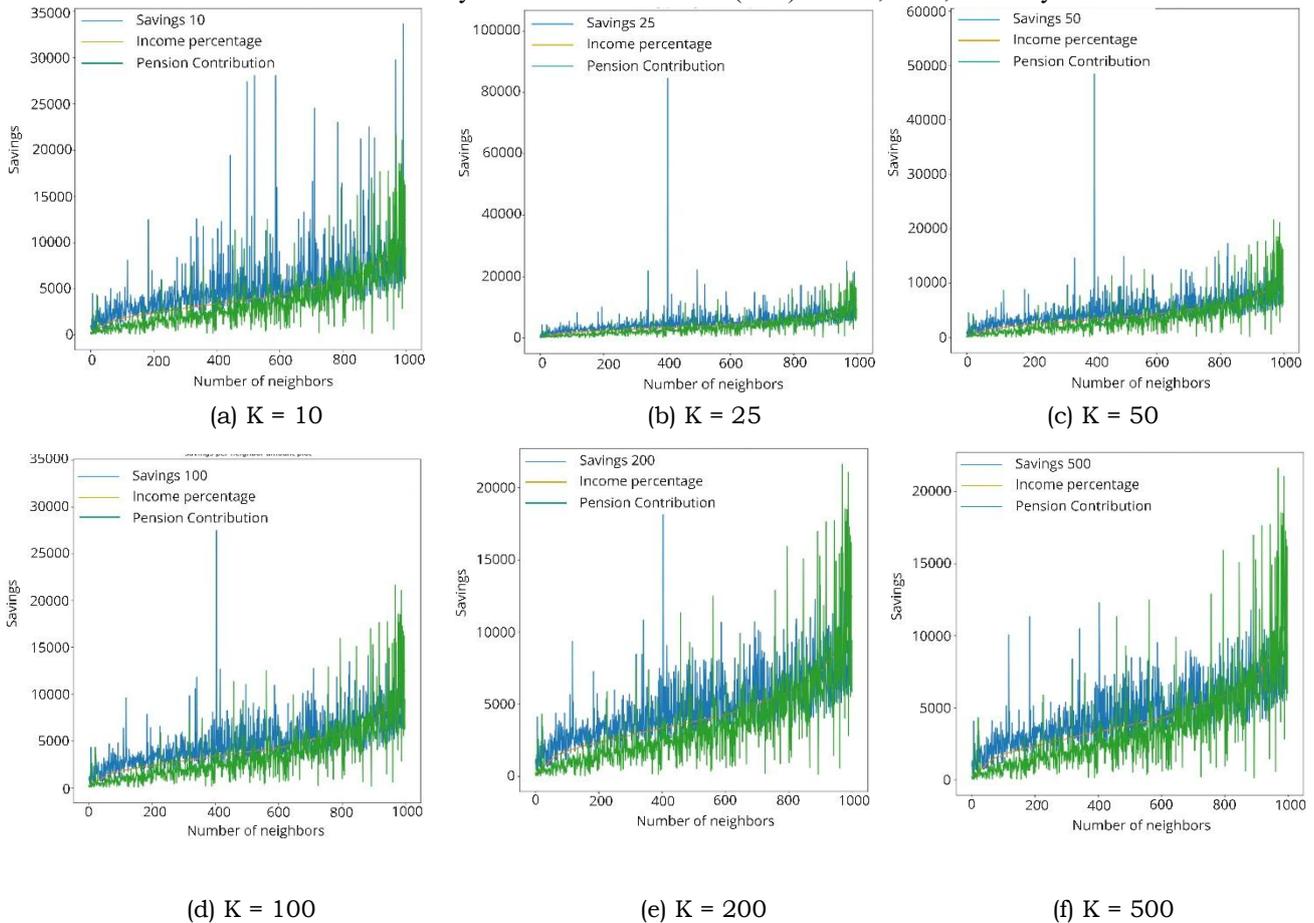


Figure 6: Number of Nearest Neighbors for traditional recommender excluding risk preference feature

Notes: Figures were plotted with the same 1000 individuals on the horizontal axis as in Table 1. The vertical axis represents the individual savings values in Euros. Each category has a different individual count since it is merely a subset of the individuals that match the test set. The blue line is the 10% income savings of the individuals and the orange line is the pension savings while the green line is the recommended savings amount excluding the risk preference feature is not used for recommendation.

4.4. Discussion

The result indicated that pension recommendations of the nearest neighbor recommender tend to provide recommendations similar to individuals with matching employment status and province in particular and other categorical variables like employment sector and education, hence the class balances of the categorical variables are essential for the models. On the other hand, the deep learning-based model tends to provide diverse recommendations in terms of classes of categorical variables like employment status and province. This opens up the avenue to investigate employed people similar to the self-employed in terms of demographic characteristics and risk preference. A rough look at 10 random individuals of self-employed that have neighbors in the deep learning model that are employees did not clearly indicate why the deep learning model made such an inference, in order to make additional hypotheses. Additionally, approximately 55% of the individuals in the study, with around 80% of them being male, have nonzero values for their pension savings, from which the test set is sampled. This gender imbalance could potentially bias the pension recommendations towards males in the overall population.

The methodological choice of evaluating with the selected baselines on a random sample of individuals discourages exact theoretical evaluation and so general facts that applies beyond this data are not stated. However, the results are empirically grounded in the performance relative to the pension savings benchmark and show that the impact of new features is significant to the recommended savings as shown in the result of the influence of risk preferences in Table 1 and Figure 6 as compared to the results of the Figure 4 with less stable individual recommendations. The traditional model is a memory-based, user-based recommender system uses the processed user attributes as a basis to find similar users as depicted in section 4.1. It also needs to compute the full similarity matrix of all individuals and thus for a new user that does not exist originally in the dataset, a computationally intensive process of calculating the new user's similarity with all other existing users is necessary. On the other hand, the deep learning recommender system learns a model function of user preferences and does not need additional re-computation for a new user. However, this is a black box model that does not specify what is important for the recommendation provided by the model.

The deep learning recommender has been modeled with the output being the embeddings with a dimension of 40 as shown in section 4.2 which is akin to a similarity matrix of the nearest neighbor recommender. This is a design choice to make the combination of the model results easier as both outputs have the same form, even though it is more common for the outputs of a deep learning based recommender system to be a target variable and the embedding to be a hidden layer of the neural network. The homogeneity of the model's output enables an easier combination of results that are already similar in terms of output and process.

Deep learning learns high-dimensional features from data in a process in the form of an end-to-end system, whereby feature engineering is automatically performed. By designing the model structure flexibly, deep learning can effectively handle various types of data such as text and images. This means that multi-modal data extension can be easily incorporated to improve the recommender system. It is also possible to reduce the training time of the model by incorporating a pre-trained model, thus subsequent re-initialization of the model for continuous integration and development is more efficient.

We observed in both models that the exclusion of the risk preference variable which is the representative variable of the behavioral attributes in the data affects the recommended savings significantly while the recommended savings still tend to lie in between the income savings baseline and pension fund savings. However, we hypothesize that the inclusion of multiple behavioral attributes will significantly impact the recommended savings, given the impact of the risk preference variable.

The evaluation of the recommender system is offline but customized for the domain of pensions as compared to general evaluation methods such as mean squared error, hit rate, diversity, and novelty. As we see, the proposed solution is flexible based on criteria such as robustness, interpretability, and complexity while recommending effective savings for retirement with regard to established baselines. The architecture is also straightforward and extensible to develop new models to add and improve the overall system.

Finally a potential deployment and use of the recommender system will fit in the role of helping pension funds advise customers on savings with good interpretability of why the recommendation works for them by skewing the model to the traditional recommender system and explaining to the customer that similar individuals are already saving that amount at least. However, we do not fully recommend exposing the system directly to the customer because in the exceptional cases where the value does not seem reasonable to the customer, they will potentially just assume the recommender system does not work and not see the nuances of the exceptional cases.

5. CONCLUSIONS

This work proposed a novel hybrid recommender system using KNN collaborative filtering combined with a deep learning approach. The proposed approach leverages a complex architecture with a deep neural network and uses embeddings to learn non-linear latent relationships of users with an embedding matrix (A similarity matrix), an output similar to the model of combination for homogeneity of the overall system. The input data consists of demographic variables, and financial variables obtained from administrative data at Statistics Netherlands together with individual risk preferences. To estimate the risk preferences of individuals, we implemented a survey, including incentivized economic experiments, among the Dutch working population (N = 4, 282); upon which we employed supervised machine learning techniques to estimate risk preferences of the general population of the Netherlands based on the aforementioned data from Statistics Netherlands.

The collaborative filtering recommender makes use of the input data where numerical features are quantized to bins, then normalized, and the categorical features are one-hot encoded to derive a similarity matrix upon which similar users can be extrapolated. The deep learning model uses the raw input data, being an end-to-end system to learn the embedding relationships of the user groups. We apply cross-validation to tune all the hyperparameters and implement a prototype. Evaluation is offline and customized for the pensions domain with baselines being 10% net income and the actual pension contribution. The combination of both models is achieved using weighted averages of both results and can be tuned based on criteria important to the modeler such as robustness, complexity, and interpretability.

We observe that both models generate recommendations that do not wildly differ from the baselines and are thus effective but the traditional collaborative filtering model is less robust and flexible while being more interpretable in comparison to the deep learning model. In conjunction with each other, we obtained the most optimized recommendations better than each individual model. The traditional model is biased to find similar users with the same class of most of the categorical features while the deep learning model yields more diverse similar users.

REFERENCES

- [1] M. Mastrogiacomo, "De pensioenpuzzel van zelfstandigen," Netspar Brief, vol. 7, 2016.
- [2] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in Proceedings of the 1st ACM conference on Electronic commerce, pp. 158–166, 1999.
- [3] J. Bobadilla, F. Ortega, A. Hernando, and A. Guti'erez, "Recommender systems survey," Knowledge-based systems, vol. 46, pp. 109–132, 2013.
- [4] P. Resnick and H. R. Varian, "Recommender systems," Communications of the ACM, vol. 40, no. 3, pp. 56–58, 1997.
- [5] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE transactions on knowledge and data engineering, vol. 17, no. 6, pp. 734–749, 2005.
- [6] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, "A literature review and classification of recommender systems research," Expert systems with applications, vol. 39, no. 11, pp. 10059–10072, 2012.
- [7] H. Ko, S. Lee, Y. Park, and A. Choi, "A survey of recommendation systems: recommendation models, techniques, and application fields," Electronics, vol. 11, no. 1, p. 141, 2022.

- International Journal on Cybernetics & Informatics (IJCI) Vol.15, No.2, February 2026
- [8] G. Salton, "Automatic text processing: The transformation, analysis, and retrieval of," Reading: Addison-Wesley, vol. 169, 1989.
 - [9] E. Pelik'an, "Principles of forecasting—a short overview," in International Conference on Current Trends in Theory and Practice of Computer Science, pp. 311–327, Springer, 1999.
 - [10] G. L. Lilien and A. G. Rao, "A model for allocating retail outlet building resources across market areas," *Operations Research*, vol. 24, no. 1, pp. 1–14, 1976.
 - [11] E. Rich, "User modeling via stereotypes," *Cognitive science*, vol. 3, no. 4, pp. 329–354, 1979.
 - [12] R. Van Meteren and M. Van Someren, "Using content-based filtering for recommendation," in Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop, vol. 30, pp. 47–56, Barcelona, 2000.
 - [13] M. Balabanovi'c and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.
 - [14] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the 10th international conference on World Wide Web, pp. 285–295, 2001.
 - [15] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.
 - [16] He X, Liao L, Zhang H, Nie L, Hu X, Chua TS. Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web 2017 Apr 3 (pp. 173-182).
 - [17] Y. Guo, Y. Zhou, X. Hu, and W. Cheng, "Research on recommendation of insurance products based on random forest," in 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), pp. 308–311, 2019.
 - [18] M. Bogaert, J. Lootens, D. Van den Poel, and M. Ballings, "Evaluating multi-label classifiers and recommender systems in the financial service sector," *European Journal of Operational Research*, vol. 279, no. 2, pp. 620–634, 2019.
 - [19] B. Smyth, "Case-based recommendation," in *The adaptive web: methods and strategies of web personalization*, pp. 342–376, Springer, 2007.
 - [20] F. Lorenzi and F. Ricci, "Case-based recommender systems: A unifying view," in IJCAI Workshop on Intelligent Techniques for Web Personalization, pp. 89–113, Springer, 2003.
 - [21] S. S. A. Rahman, A. A. Norman, and K. Soon, "Myins: a cbr e-commerce application for insurance policies," *Electronic Commerce Research*, vol. 5, no. 1, pp. 373–380, 2006.
 - [22] C. Musto, G. Semeraro, P. Lops, M. De Gemmis, and G. Lekkas, "Financial product recommendation through case-based reasoning and diversification techniques.," in RecSys Posters, 2014.
 - [23] C. Musto, G. Semeraro, P. Lops, M. De Gemmis, and G. Lekkas, "Personalized finance advisory through case-based recommender systems and diversification strategies," *Decision Support Systems*, vol. 77, pp. 100–111, 2015.
 - [24] L. Rokach, G. Shani, B. Shapira, E. Chapnik, and G. Siboni, "Recommending insurance riders," in Proceedings of the 28th annual ACM symposium on applied computing, pp. 253–260, 2013.
 - [25] L. Wang, X. Hu, J. Wei, and X. Cui, "A collaborative filtering based personalized top-k recommender system for housing," in Proceedings of the 2012 International Conference of Modern Computer Science and Applications, pp. 461–466, Springer, 2013.
 - [26] T. Stone, W. Zhang, and X. Zhao, "An empirical study of top-n recommendation for venture finance," in Proceedings of the 22nd ACM international conference on information & knowledge management, pp. 1865–1868, 2013.
 - [27] F. Sayyed, R. Argiddi, and S. Apte, "Generating recommendations for stock market using collaborative filtering," *Int. J. Comput. Eng. Sci*, vol. 3, pp. 46–49, 2013.

- International Journal on Cybernetics & Informatics (IJCI) Vol.15, No.2, February 2026
- [28] E. L. Lee, J.-K. Lou, W.-M. Chen, Y.-C. Chen, S.-D. Lin, Y.-S. Chiang, and K.-T. Chen, "Fairness-aware loan recommendation for microfinance services," in Proceedings of the 2014 international conference on social computing, pp. 1–4, 2014.
 - [29] X. Zhao, W. Zhang, and J. Wang, "Risk-hedged venture capital investment recommendation," in Proceedings of the 9th ACM Conference on Recommender Systems, pp. 75–82, 2015.
 - [30] M. Taghavi, K. Bakhtiyari, and E. Scavino, "Agent-based computational investing recommender system," in Proceedings of the 7th ACM conference on recommender systems, pp. 455–458, 2013.
 - [31] S. Mitra, N. Chaudhari, and B. Patwardhan, "Leveraging hybrid recommendation system in insurance domain," International Journal of Engineering and Computer Science, vol. 3, no. 10, pp. 8988–8992, 2014.
 - [32] J. Choo, D. Lee, B. Dilkina, H. Zha, and H. Park, "To gather together for a better world: Understanding and leveraging communities in micro-lending recommendation," in Proceedings of the 23rd international conference on World wide web, pp. 249–260, 2014.
 - [33] Y. Mao, M. Chen, B. Dorn, and Y. He, "A hybrid approach to similar stock recommender with taxonomy and behavior data," in 2022 IEEE International Conference on Big Data (Big Data), pp. 2313–2316, IEEE, 2022.
 - [34] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," ACM computing surveys (CSUR), vol. 52, no. 1, pp. 1–38, 2019.
 - [35] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in Proceedings of the tenth ACM international conference on web search and data mining, pp. 495–503, 2017.
 - [36] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, "A simple convolutional generative network for next item recommendation," in Proceedings of the twelfth ACM international conference on web search and data mining, pp. 582–590, 2019.
 - [37] Q. Zhang, C. Qin, Y. Zhang, F. Bao, C. Zhang, and P. Liu, "Transformer-based attention network for stock movement prediction," Expert Systems with Applications, vol. 202, p. 117239, 2022.
 - [38] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in Proceedings of the AAAI conference on artificial intelligence, vol. 33, pp. 346–353, 2019.
 - [39] R. Burke, "Hybrid web recommender systems," The adaptive web: methods and strategies of web personalization, pp. 377–408, 2007.
 - [40] B. Mobasher, X. Jin, and Y. Zhou, "Semantically enhanced collaborative filtering on the web," in European web mining forum, pp. 57–76, Springer, 2003.
 - [41] B. Smyth and P. Cotter, "A personalised tv listings service for the digital tv age," Knowledge-Based Systems, vol. 13, no. 2-3, pp. 53–59, 2000.
 - [42] D. Billsus and M. J. Pazzani, "User modeling for adaptive news access," User modeling and user-adapted interaction, vol. 10, pp. 147–180, 2000.
 - [43] D. O. Sullivan, B. Smyth, and D. Wilson, "Preserving recommender accuracy and diversity in sparse datasets," International Journal on Artificial Intelligence Tools, vol. 13, no. 01, pp. 219–235, 2004.
 - [44] R. Burke, "Hybrid recommender systems: Survey and experiments," User modeling and user-adapted interaction, vol. 12, pp. 331–370, 2002.
 - [45] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," Artificial intelligence review, vol. 13, pp. 393–408, 1999.

- International Journal on Cybernetics & Informatics (IJCI) Vol.15, No.2, February 2026
- [46] A. Onaopepo, R. Arno, and D. Michel, “Models towards risk behavior prediction and analysis: A netherlands case study,” Netspar Survey Paper, 2023.
- [47] P. Covington, J. Adams, and E. Sargin, “Deep neural networks for youtube recommendations,” in Proceedings of the 10th ACM conference on recommender systems, pp. 191–198, 2016.
- [48] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in International conference on machine learning, pp. 448–456, pmlr, 2015.
- [49] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” nature, vol. 521, no. 7553, pp. 436–444, 2015.
- [50] A. Andoni and P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions,” Communications of the ACM, vol. 51, no. 1, pp. 117–122, 2008.
- [51] G. S. Classon, The richest man in Babylon. Sristhi Publishers & Distributors, 2020.
- [52] R. Guo, P. Sun, E. Lindgren, Q. Geng, D. Simcha, F. Chern, and S. Kumar, “Accelerating large-scale inference with anisotropic vector quantization,” in International Conference on Machine Learning, 2020.
- [53] P. Bokern, J. Linde, A. Riedl, H. Schmeets, and P. Werner, “A comparison of pension-relevant preferences, traits, skills, and attitudes between the dutch self-employed and employees in the netherlands,” Network for Studies on Pensions, Aging and Retirement, no. 216, 2022.
- [54] M. Damman and H. van Solinge, “Senior self-employment-the case of the netherlands,” in Handbook of Research on Entrepreneurship and Aging, pp. 300–326, Edward Elgar Publishing, 2019.
- [55] M. Mastrogiacomo and R. J. Alessie, “Where are the retirement savings of self-employed? an analysis of ‘unconventional’ retirement accounts,” 2015.
- [56] J. Cribb, C. Emmerson, L. O’Brien, and D. Sturrock, “Private pensions for the self-employed: challenges and options for reform,” tech. rep., IFS Report, 2024.
- [57] R. Crawford and H. Karjalainen, Retirement saving of the self-employed. No. R181, IFS Report, 2020.

APPENDIX A ADDENDUM

A.1 *Administrative Data from Statistics Netherlands*

Table 2: Data dictionary of register data from statistics Netherlands with missing data percent-age.

| Data Description | Missing data(%) | Data type |
|---|------------------------|------------------|
| Sector (SBI) Employee | 0.000 | int32 |
| Sector (SBI) Self-Employed | 0.000 | int32 |
| Sector (SBI) Employee 38 Cat | 0.000 | int32 |
| Marital Status 4 Categories | 0.000 | category |
| Occupation (Social-Economic Category) | 0.005 | category |
| Contribution Pension Employee (2nd pillar) | 0.853 | float64 |
| Contribution Pension Employer (2nd pillar) | 0.853 | float64 |
| Contribution Private Insurance Old Age (3rd pillar) | 0.853 | float64 |
| Contribution Private Insurance Incapacitation | 0.853 | float64 |
| Income IND Personal Y/N | 0.005 | category |
| Income IND Personal Primary | 0.853 | float64 |
| Income IND Personal | 0.853 | float64 |
| Income IND Personal Before-Tax | 0.853 | float64 |
| Income IND Self-Employment | 0.853 | float64 |
| Contribution Reimbursement Health Insurance by Employer | 0.853 | category |
| Income IND Other Labour | 0.853 | float64 |
| Income IND in Kind | 0.853 | float64 |
| Income IND Director-Majorshareholder (DGA) | 0.853 | float64 |
| Income IND Civil Servant | 0.853 | float64 |
| Income IND employment | 0.853 | float64 |
| Contribution Incapacitation Benefit Employer | 0.853 | float64 |
| Contribution Unemployment Benefit Employer | 0.853 | float64 |

International Journal on Cybernetics & Informatics (IJCI) Vol.15, No.2, February 2026

| | | |
|---|--------|----------|
| HH Type | 0.006 | category |
| HH Position of Individual | 0.006 | category |
| HH Composition | 0.005 | category |
| Main Breadwinner | 0.000 | category |
| HH Size | 0.006 | float64 |
| HH Amount of Children | 0.006 | category |
| Wealth HH Total Excluding House | 0.853 | float64 |
| Wealth HH Total (1) | 0.853 | float64 |
| Wealth HH Possessions (1.1) | 0.853 | float64 |
| Wealth HH Financial Possessions (1.1.1) | 0.853 | float64 |
| Wealth HH Bank and Saving Balance (1.1.1.1) | 0.853 | float64 |
| Wealth HH Securities (1.1.1.2) | 0.853 | float64 |
| Wealth HH Real Estate (1.1.2) | 0.853 | float64 |
| Wealth HH House (1.1.2.1) | 0.853 | float64 |
| Wealth HH Other Real Estate (1.1.2.2) | 0.853 | float64 |
| Wealth HH Entrepreneurial Capacity (1.1.3) | 0.853 | float64 |
| Wealth HH Aanmerkelijk Belang (1.1.4) | 0.853 | float64 |
| Wealth HH Other Possessions (1.1.5) | 0.853 | float64 |
| Debt HH total (1.2) | 0.853 | float64 |
| Debt HH Mortgage (1.2.1) | 0.853 | float64 |
| Debt HH Study (1.2.2) | 0.853 | float64 |
| Debt HH Other (1.2.3) | 0.853 | float64 |
| HH Home Ownership | 0.005 | category |
| HH Position Towards Main Breadwinner | 0.005 | category |
| Homeowner if Breadwinner or Partner | 0.000 | category |
| Income HH Standardized Spendable | 0.853 | float64 |
| HH Main Income Source | 0.005 | category |
| HH Income Type | 0.005 | category |
| Income HH Primary | 0.853 | float64 |
| Education Level 3 Categories | 0.000 | category |
| Education Type Classification | 0.000 | category |
| Pension Fund | 0.000 | category |
| Pension Contribution Employer | 55.502 | float64 |
| Pension Contribution Employee | 55.502 | float64 |
| premie ww ten laste van werknemer | 0.853 | category |
| INPSBI2008V | 0.000 | int64 |
| Location Labour Market Region | 0.000 | int64 |
| Location COROP | 0.000 | int64 |
| Location Province | 0.000 | category |
| Location Urbanity Municipality | 0.000 | category |
| GEM2020 | 0.000 | int64 |
| g4 | 0.000 | category |
| Location Part of Country | 0.000 | category |
| Location Urbanity Neighbourhood | 0.000 | category |

Notes: Data description provides common English explanation of the variable type where HH is household and IND is individual. The missing data value are the direct percentages of values missing for each variable