Received: 23 February 2024 | Revised: 9 April 2024 | Accepted: 14 May 2024 | Published online: 6 June 2024

RESEARCH ARTICLE

Journal of Comprehensive Business Administration Research
2024, Vol. 00(00) 1–11
DOI: 10.47852/bonviewJCBAR42022677

BON VIEW PUBLISHING

A Machine Learning-Based Suitable Startup Recommendation Scheme and Investor Assistance Mobile Application

Asfak Asif[1] and Mahfuzulhoq Chowdhury[1,*]

[1]*Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Bangladesh*

Abstract: Investing in startups is a complex decision-making process that requires identifying suitable companies and matching them with compatible investors. The existing works did not present any intelligent startup suitability prediction for funding and investor assistance mobile applications by taking into account growth rate, annual profit, rating, and reputation factors. This paper initiates a machine learning-based recommendation system for predicting suitable startups and matching them with appropriate investors. The proposed system leverages a comprehensive dataset of startup and investor characteristics, including market segment, geographical region, and city. In comparison with singular value decomposition and principal component analysis schemes, our results show that Term Frequency-Inverse Document Frequency scheme is selected for startup recommendation due to its high recall and precision value. Our investor assistance mobile application offers features like login, apply for funds, best funder recommendation, best entrepreneur selection, and user rating features. Our application evaluation results indicated that more than 60 percent of users are satisfied with the feasibility of the proposed mobile application.

Keywords: machine learning, recommendation system, startup funding, investor selection, similarity computation

1. Introduction

A startup is a young and innovative company that is founded by entrepreneurs to generate a unique service, business model, and product. Startups can be typically described by their pursuit of high-growth potential, scalability, and disruption of existing markets or industries [1]. Startups are vital drivers of innovation and economic growth, propelling forward new ideas, products, and technologies. However, for these ventures to flourish and reach their full potential, securing funding and finding suitable investors is of paramount importance. The process of startup funding and investor matchmaking, though critical, can often be a daunting task characterized by numerous challenges and uncertainties [2]. Investments in startups refer to the process of providing financial resources, typically in the form of capital (i.e., during early production or high-growth initiatives). Startup investments are made with the expectation of generating a significant return on investment in the future. Investing in startups can offer various opportunities and benefits, but it also carries inherent risks. There are several kinds of startup investments such as equity investments, debt investments, convertible notes, crowdfunding, and venture capital (VC) funding.

Existing works on startup funding and investor matchmaking [1, 3–10] have typically relied on subjective decision-making processes and personal networks. Such approaches are prone to biases, limited information, and lack of objectivity, leading to inefficiencies and suboptimal outcomes. To address these challenges and augment the decision-making process, this work centers on the development of a machine learning (ML)-based recommendation system for startup selection and investor assistance by taking into account multiple factors (e.g., profit, market demand, and growth rate). The existing works did not investigate Term Frequency-Inverse Document Frequency technique (TF-IDF), singular value decomposition technique (SVD), and principal component analysis (PCA) scheme-based startup prediction for the investors by taking multiple factors like growth rate, profit, or market demand. To overcome the existing limitations, the primary goal of this paper is to implement a robust recommendation system that leverages ML techniques to predict suitable startups for funding and match them with interested investors. By harnessing the power of data-driven algorithms, this system aims to enhance the efficiency, effectiveness, and objectivity of the startup funding and investor selection processes. The foundation of the recommendation system lies in the analysis of critical startup and investor characteristics. Factors such as market segment, geographical region, company size, industry expertise, and financial metrics are essential in determining the compatibility between startups and investors. By incorporating

*Corresponding author: Mahfuzulhoq Chowdhury, Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Bangladesh. Email: mahfuz@cuet.ac.bd

these dimensions, the recommendation system can provide suitable recommendations that optimize the match between startups (with fund-seeking nature) and investors looking for promising opportunities.

The main contributions of this paper are detailed as follows:

(i) To develop a startup recommendation system, a comprehensive dataset is collected in this paper by encompassing startup and investor information.
(ii) Through the use of ML techniques (i.e., TF-IDF, SVD, and PCA), the developed system generates personalized recommendations that align startups with the most suitable investors.
(iii) This paper also develops a mobile application featuring investor suggestions for entrepreneurs (based on reputation, yearly income, and company network), entrepreneur selection (based on market price and rating), add review and rating, apply for funds, login, and signup options.
(iv) This paper also provides application evaluation results based on users' feedback and review results.
(v) This paper selects a suitable model for startup recommendation for investors by comparing TF-IDF, SVD, and PCA schemes.

The literature review of the existing research papers is presented in Section 2. Section 3 visualizes the proposed ML-based startup selection framework. Section 4 mentions the developed mobile application features. Sections 5 and 6 deliver the evaluation of results and summary of this article, respectively. Section 2 presents a literature review of existing research papers. Section 3 describes the proposed ML-based startup selection framework. Section 4 describes the features of the developed mobile application. Sections 5 and 6 present the evaluation results and summary of this article, respectively.

2. Literature Review

In this section, this article will deliver the related literature works associated with the investor selection, startup assistance, investor behavior, and startup selection. In Xu et al. [2], the authors selected suitable investors for emerging new startups. To recommend investors, they used approximately thirty-four thousand investment data from the Itjuzi website. In Narode et al. [3], the authors built a recommendation system by collecting clients' feedback from the respective company. Then they extracted feedback by using text mining. They used a naive Bayes classifier for feedback classification as positive type, negative type, and neutral type. Venture capitalists require market data and company information for investment decisions or startup selection. To do so, the researchers can use different forms of data collection methods such as interview or questionnaire-based approaches [4]. Several studies showed that only the collection of financial data regarding a startup is not sufficient for venture capitalists [11]. The reason behind this is the new startup does not have extraordinary market data or financial information [5]. Thus, to make a feasible startup selection decision, venture capitalists need to look into other factors too such as market growth, profit, or user feedback. In Liu et al. [6], the authors used matrix factorization to generate investment recommendations for investors. In Färber [7], the authors used a Crunchbase-based dataset for company selection for the investors. This dataset contains information regarding companies, people, investors, and investment information. In Kim et al. [12], the authors discussed that multiple investments from the same investor (at the same company) could not improve the prediction accuracy regarding investment recommendations by the VCs. In Zhao et al. [8], the authors discussed different investment risks for the investors. They also used five risk assessment-based startup selection and probability-based matrix factorization models for information filtering. In Alexy et al. [13], the authors discussed the impact of the social capital of VCs on the funding of startup firms. In Luef et al. [1], the authors found that for potential investors, deciding which business to invest in can be challenging, when there is little publicly available information. In Bosch [14], the authors discussed the issue of biases during the startup scouting process. They also developed a content-based, knowledge-based, and collaborative filtering-based hybrid recommender system for the startup scouting process. In Liang et al. [15], the authors tried to find out the impact of social network data on the funding behavior of both investors and companies. In Raju [16], the authors developed a recommendation system for the food startup by taking into account the automatic retention of good restaurants, users, and customer feedback. In Tikka [17], the authors discussed several factors that may impact the early-stage funding decision for Finnish-based startups.

In Arroyo et al. [18], the authors indicated that a multi-class ML classifier could be helpful for the success rate increment of an investor regarding low-risk startup company selection. In Eloranta [19], the authors discussed the impact of investment on the employment and profitability of Finnish startups. But the limitation of their research is they did not use any ML algorithms to predict the equity funding types or the hypothesis. In Ross et al. [20], the authors utilized a ML-based prediction model to determine the startup's outcome (i.e., exit prediction through an IPO) by monitoring the investment, acquisition, and failure. The authors in Zhong [9] tried to predict VC's investment by taking into account social connection information between venture capitalist firms and startup members. In Bai and Zhao [10], the authors used linear and tree-based ML models for startup investment decision prediction. But the limitation of their work is they only used one VC firm data. In Ang et al. [21], the authors utilized an ML-based prediction model for the funding decision of startups, success decision, and post-money valuation by taking into account region and funding amount.

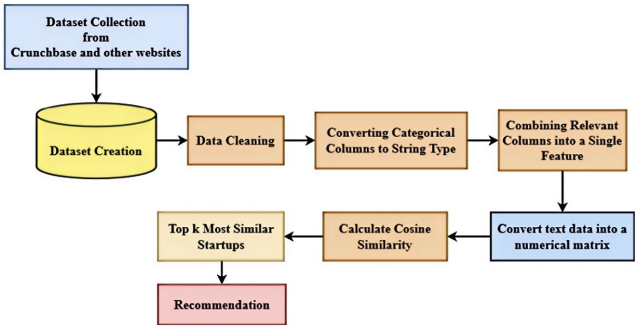
The authors of Yadab et al. [22] looked into how deep learning technology affected the agricultural startup. The authors of Rajendran et al. [23] talked about the use of AI in developing startup strategies. The writers of Abuzaid et al. [24] talked about the challenges and facilitators of startups in the future economy. The authors of Vukmirović et al. [25] talked on the contribution of IT education to the development of Croatian startups.

Differing from existing works, this paper develops a ML-based suitable startup prediction scheme by taking into account region, city, growth rate, profit, and rating of startups. This paper also presents a mobile application for both entrepreneur and investor assistance by taking into account best entrepreneur selection, fund applies feature, startup selection, add rating, and review features.

3. Proposed Framework

The methodology of the proposed startup recommendation system is depicted in Figure 1. The first step in building the recommendation system is dataset collection. In this case, the data are collected from the Crunchbase website [7], a widely recognized platform for startup and investor information. We have collected different investor and startup company information from Crunchbase website and developed a dataset by

Figure 1
Proposed startup recommendation system



inserting their specific information in both rows and columns (see Figures 2, 3, and 4). With a wide range of variables, including company name, category, market, region, yearly profit amount (USD), and growth rate (%), the dataset encompasses key aspects that are crucial in understanding the startup ecosystem. The main variables are Company Name, Company Category, Company Market, Company Region, Investor Name, Investor Market, Investor Region, First Funding date, Last Funding Date, Funded Year, Yearly Profit Amount (USD), and Growth Rate (%). This dataset serves as the foundation for subsequent analysis and recommendation generation. Collecting data from Crunchbase website ensures that the recommendation system is built upon reliable and up-to-date information from a trusted source in the startup ecosystem. The dataset contains approximately 1,00,000 entries, each row and column representing the variables of company details also with investor details. A glimpse of the dataset column is shown in Figure 2. For proper understanding, we have provided a sample of dataset in Figures 3 and 4.

3.1. Data loading and preprocessing

In the data loading and preprocessing step, we perform data loading and preprocessing to prepare the dataset for building the startup recommendation system. The dataset is loaded from a CSV file. The CSV file contains information about various startups, such as their company name, category list, market, region, yearly profit amount in USD, and growth rate in percentage. Once the dataset is loaded, the code selects only the relevant columns that are essential for building the recommendation system. The relevant columns list includes "company name," "category list of the company," "market of the company,", "company region," "Yearly Profit amount (in USD)," and "growth rate in percentage." The columns "company name," "company category list," "company market," and "company region" are crucial for generating TF-IDF representations for startups, and the "Yearly Profit amount (in USD)" and "growth rate (in Percentage)" columns might be used for additional analysis or filtering of startups. Next step is handling missing values.

After selecting the relevant columns, the code drops any rows with missing values (NaN) in any of the selected columns. The rationale behind dropping rows with missing values is to ensure the data used for the recommendation system are complete and valid. Missing data might affect the accuracy of the recommendation process.

The next step involves converting certain categorical columns (company category list, company market, and company region) to string type. This conversion is necessary because the "TF-IDF vectorizer" expects text input and operates on string data to generate the TF-IDF representation. Converting categorical columns to strings allows them to be combined into a single textual feature that can be used for the recommendation process.

After converting the categorical columns to strings, the code combines these relevant columns ("company category list," "company market," "company region") into a new feature named combined features. The combined features feature is created by concatenating the values of the three categorical columns for each startup. For example, if a startup's "company category list" is "Health, Technology" and its "company market" is "Healthcare" and "company region" is "United States," the resulting combined features will be "Health, Technology Healthcare United States." This combined features feature represents the textual characteristics of each startup and will be used as input to the TF-IDF vectorizer. After completing Step 1, the dataset is preprocessed, and the combined features column contains textual

Figure 2
A glimpse of dataset columns

(a)

Company name	Company category list	Company market
Test company	Banking security	Content discovery
Andrew burnett dot com	Internet services SEO	Internet
ABO data	Enterprise software	Enterprise software
ABO data	Enterprise software	Enterprise software
Ikro	Video messaging	Healthcare services

(b)

Investor name	Investor category list	Investor market
Jamessss Bondddd	Semiconductor video hardware	Co working
Ekaf	Chemicals energy trading	Business development
Antonio Murronni	Technology finance venture	Design
Filippo Murronni	Payments P2P money transfer	Automotive
CRP companhia de participacoes	Finance consumers venture	News

Figure 3
Sample of the dataset's first part

(a)

Company name	Company category list	Company market	Company country code
Test company	Banking security	Content discovery	ARE
Andrew burnett dot com	Internet services SEO	Internet	GBR
ABO data	Enterprise software	Enterprise software	USA
ABO data	Enterprise software	Enterprise software	USA
Ikro	Video messaging	Healthcare services	BRA

(b)

Company state code	Company region	Company city	Investor name
NC	Dubai	Dubai	Jamessss Bondddd
IN	Edinburgh	Edinburgh	Ekaf
TX	TX (other)	Italy	Antonio Murronni
TX	TX (other)	Italy	Filippo Murronni
LA	BRA (Other)	Canoas	CRP companhia de participacoes

Figure 4
Sample of the dataset's second part

(a)

Investor category list	Investor market	Investor country code	Investor state code
Semiconductor video hardware	Co working	TWN	SK
Chemicals energy trading	Business development	GBR	ID
Technology finance venture	Design	MUS	AL
Payments P2P money transfer	Automotive	CAN	DC
Finance consumers venture	News	SVK	NM

(b)

Investor region	Investor city	Funding round per-code	Funding round type
Vejie	Bucharest	c308k7	seed
Philadelphia	Tel Aviv	14fe2	seed
New Orleans	Netanya	809e3	seed
Ontario	Nijehaske	46c3	seed
Seminole	Spring house	5bdf6	venture

representations of each startup. These textual representations will serve as the input for the TF-IDF representation, which will then be used to find similar startups and make personalized recommendations to investors based on their preferences.

3.2. TF-IDF representation

Our next step is TF-IDF representation. TF-IDF is a useful presentation technique that can convert text data into a numerical matrix. It is designed to capture the importance of each term (word) in a document (text) relative to a collection of documents (corpus). The TF-IDF representation helps to emphasize the significance of rare terms in a document while reducing the importance of frequently raised (common) terms in many documents. The TF-IDF representation of a document in a collection is calculated in two steps. First, we measured the term frequency. Term frequency measures the frequency of a term (word) within a document. The formula for calculating the term frequency of a single term is given as $TF(t,d)$ = Total time number with term t appearing in document d/Total number of terms in document d.

The result is a positive value, usually in the range of [0,1], indicating the relative importance of the term within the document. A higher value suggests that the term appears frequently in the document. Inverse document frequency can measure the informativeness of a term across the full document corpus. It penalizes common terms that appear in many documents and emphasizes rare terms that are more informative. The formula for inverse document frequency of a term t is given by: $IDF(t)$ = log(Total number of documents in corpus/ Number of documents containing term t). The result is a non-negative value. The IDF is typically higher for rare terms and lower for common terms. The TF-IDF score of a term t within document d is measured by multiplying the $TF(t,d)$ and $IDF(t)$. The TF-IDF score can show the importance of the term in the specific document and across the entire corpus. It is higher when the term appears frequently in the document but infrequently across other documents in the corpus. Each document in the collection is represented as a vector, where each dimension of the vector corresponds to a unique term from the entire corpus. The value in each dimension is the TF-IDF score of the corresponding term in the document. The resulting vectors form a TF-IDF matrix, in which rows can represent documents and columns can represent terms. In the context of the code, the TF-IDF representation is created using the TF-IDF vectorizer from sci-kit-learn. It converts the textual data in the combined-feature column (which represents each startup's characteristics) into a TF-IDF matrix. This matrix will serve as the numerical representation of the startups, allowing the system to measure similarity and make personalized recommendations based on investor preferences.

3.3. Dimensionality reduction

The dimensionality reduction technique can be used to retain the most significant and relevant information while reducing the features or dimensions. PCA technique is employed to reduce the dimensionality of the TF-IDF matrix used for startup recommendation. Let us delve into the details of PCA. The first step in PCA is to compute the important covariance matrix from the original dataset. The covariance matrix measures how different features are related to each other. It indicates the direction and strength of the linear relationships between features. Next, PCA calculates the eigenvectors and eigenvalues (i.e., from the covariance matrix). Eigenvectors represent the principal components of the data, which are new axes that point in the directions of maximum variance. Eigenvalues correspond to the amount of variance captured by each principal component. The eigenvalues are arranged in descending order, representing the amount of variance captured by each corresponding eigenvector. The first principal component (eigenvector with the largest eigenvalue) explains the most variance in the data, followed by the second principal component, and so on. In PCA, the number of principal components retained (k) is a hyperparameter that needs to be determined. It defines how many dimensions to keep in the reduced feature space. The choice of k depends on the trade-off between computational efficiency and the amount of variance that needs to be preserved. A common approach is to select a value of k that captures a significant portion of the total variance, such as 95% or 99%. After determining the number of principal components (k), the original data are projected onto the new reduced feature space. The projection involves transforming the data points from the original high-dimensional space into a lower-dimensional space spanned by the selected principal components. The reduced feature space contains the k principal components that capture the most important patterns and variations in the data. Each data point in the reduced space is represented as a linear combination of the k principal components.

SVD is another feature number reduction technique with relevant information preservation. It is a powerful linear algebra technique that decomposes a matrix into three separate matrices, allowing us to retain the most important patterns and variations in the data. In the context of the provided code, SVD is employed to reduce the dimensionality of the TF-IDF matrix used for startup recommendation. Let's delve into the details of SVD.

Given an m × n matrix X (i.e., m = samples number, n = features number), SVD decomposes X into three separate matrices: U, $\sum(m \times n)$, and V^T. U (m × m) contains the left singular vectors. $\sum(m \times n)$ is a diagonal matrix with singular values within the diagonal place. Singular values represent the importance of each dimension in the reduced feature space. V^T (n × n) contains the right singular vectors, which represent the column-wise relationships between features in a lower-dimensional space. The singular values in $\sum(m \times n)$ are arranged in descending order. This implies that the first singular value corresponds to the most important dimension, the second singular value corresponds to the second most important dimension, and so on. Similar to PCA, selecting the number of components (k) in SVD is a crucial step. It determines the dimensionality of the reduced feature space. A common approach is to choose k by taking into account the cumulative sum of the squared singular values. By retaining the first k singular values that contribute to a significant portion of the total variance, we can effectively reduce the dimensionality while preserving most of the information. After determining the number of components (k), the original data are projected onto the new reduced feature space. The projection calculation involves the dot product value of the three metrics: U, $\sum(m \times n)$, and V^Tmatrices, and then keeping only the first k columns of U, the first k rows and columns of $\sum(m \times n)$, and the first k rows of V^T. The result consists of the lower-dimensional value representation from the original data. The reduced feature space contains the k most important dimensions (represented by the first k columns of U) that capture the most significant patterns and variations in the data. Each data point in the reduced space is represented as a linear combination of these k dimensions.

3.4. Data splitting

In this step, data can be divided into the training and testing set. This process is crucial for building and evaluating the recommendation system. Data splitting is a standard practice in ML and recommender systems to ensure that the model is evaluated on unseen data (the testing set) that was not used during the model training (the training set). The main purpose of data splitting is to measure how well the recommendation system generalizes to new data and to avoid overfitting, in which the model can perform better on the training data but worse on the unknown data. The data splitting is done using the train-test split function from sci-kit-learn, which randomly shuffles the dataset and splits it into two portions based on a specified ratio. In this case, the code splits the dataset into 80% for training and 20% for testing, as specified by test size = 0.2 in the function. The training dataset would be used for the recommendation model build-up process, where the testing dataset would be used for the performance evaluation of different models (i.e., to make accurate and personalized recommendations for investors based on unseen data). The train-test-split function uses randomization to split the data, and the random-state parameter ensures reproducibility. Setting a fixed random seed allows the data split to be the same each time the code is executed, ensuring consistent evaluation results during testing.

3.5. Recommendation process

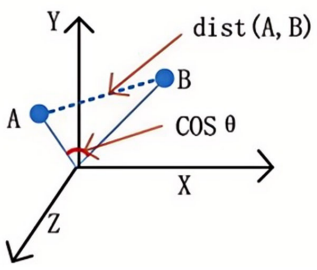
In Step 5, the code performs the recommendation process for each investor in the testing set based on their preferences and the characteristics of startups in the training set. Let us break down the recommendation process into sub-steps. The code uses a loop to iterate over each row in the testing set (test data), where each row corresponds to an investor's preferences. For each investor, their preferences are extracted from the combined features column, which represents the textual characteristics of startups. The TF-IDF vectorizer, which was previously fit on the training data, is used to transform the investor's preferences into a TF-IDF representation. The TF-IDF representation represents the textual features of the investor's preferences in the same vector space as the startup characteristics.

3.6. Calculate cosine similarity

Similarity calculation is a critical step in the recommendation system to identify the degree of similarity between investor preferences and startups. In this case, cosine similarity is employed as a similarity metric. Cosine similarity has several benefits over other similarity metrics. Cosine similarity is calculated between the TF-IDF representation of the investor's preferences and the TF-IDF matrix obtained from the training set (TF-IDF-matrix). Cosine similarity value can measure the two vectors' cosine angle in the TF-IDF vector space, representing the investor preferences and the startups' features. It represents the similarity between the investor's preferences and each startup in the training set. The TF-IDF matrix, obtained during the feature extraction step, serves as the basis for the calculation. The cosine similarity score can range from −1 to 1 (i.e., a value closer to one means higher similarity). The cosine similarity calculation process is shown in Figure 5. To calculate the cosine similarity, the investor preferences, represented as a vector in the TF-IDF space, are compared with each startup's TF-IDF vector. The calculation involves two vectors dot product value generation. It is then divided by the product of their magnitudes. This process generates a similarity score for each startup, indicating how closely it matches the investor's specified preferences. By employing cosine similarity, the recommendation system quantifies the similarity between investor preferences and startups, allowing for an objective measure of compatibility. This similarity calculation enables the system to identify the startups that align most closely with the investor's specified criteria, assisting in the selection of potential investment opportunities that are in line with the investor's preferences.

Figure 5
Cosine similarity calculation procedure demonstration

$$cosine\_similarity(x,y) = \frac{x \cdot y}{||x|| \cdot ||y||} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$



3.7. Get the top *K* most similar startups

The code identifies the top k (here, $k = 5$) startups with the highest cosine similarity scores as the most similar to the investor's preferences. These top k startups are selected as the recommendations for the investor. To evaluate the recommendation quality, the code compares the recommended startups with the ground truth (actual) startups that the investor is interested in. A set of ground truth company names for each investor's preferences is created from the testing set. The code calculates precision@k and recall@k to measure the accuracy and completeness of the recommendations. Precision@k measures the proportion of relevant startups (among the recommended top k startups). Recall@k measures the proportion of relevant startups found among all relevant startups. The recommended startups for each investor are stored in a list (recommended startups). The precision@k and recall@k values for each investor are also stored in separate lists (precision-pr-curve and recall-pr-curve, respectively). To calculate mean average precision (i.e.,MAP) and normalized discounted cumulative Gain metrics (i.e., NDCG), the ground truth and recommended startup lists are converted to binary format. The MultiLabel Binarizer is used to convert the ground truth and recommended startup lists to binary matrices (true-labels-bin and predicted-scores-bin). After completing Step 5, the recommendation process has been performed for each investor in the testing set. The system has generated personalized startup suggestions or recommendations by assessing the textual similarity between investor preferences and startup characteristics. The evaluation metrics (precision, recall, MAP, and NDCG) are computed to assess the performance of the recommendation system in terms of accuracy and relevance.

3.8. Recommendation output

In this step, the code processes the results of the recommendation process and generates personalized startup recommendations for each investor in the testing set. Let us delve into the details of this step. For each investor's preferences in the testing set, the code calculates the cosine similarity scores between the investor's TF-IDF representation and the TF-IDF representations of all startups in the training set. The top k startups with the highest cosine similarity scores are selected as the most similar to the investor's preferences. These top k startups are considered personalized recommendations for each investor. After obtaining the top k recommended startups, the code evaluates the quality of the startup recommendations using the well-known precision@k metrics and recall@k metrics. Precision@k offers the relevant startup's proportion value among the top k recommended startups. A relevant startup is one that the investor is interested in (present in the ground truth set). Recall@k offers the relevant startup proportions found among all relevant startups. It helps to assess the completeness of the recommendations. To calculate MAP and NDCG metrics, the code converts the ground truth and recommended startup lists into binary format. The ground truth company names for each investor's preferences are collected from the testing set and converted into a binary format using the MultiLabel Binarizer. The same process is applied to the top k recommended startups to create binary matrices (true-labels-bin and predicted-scores-bin) required for calculating MAP and NDCG. For each investor in the testing set, the code provides a list of personalized startup recommendations based on the TF-IDF similarity scores. The recommendations are sorted in

descending order of relevance (measured by cosine similarity), with the most relevant startup appearing first in the list. The recommendation output serves as a set of personalized suggestions for each investor, showcasing startups that match their preferences and characteristics. Apart from generating personalized recommendations, the code examines several evaluation metrics to compute the recommendation systems' quality. The precision@k and recall@k values are calculated for each investor and stored in lists (precision-pr-curve and recall-pr-curve). The MAP and NDCG metrics are also computed to measure the overall recommendation systems performance.

3.9. Evaluation of recommendation system

To examine the recommendations systems quality, we used Average Precision@K, Average Recall@K, MAP, and NDCG. We showed a comparison in these scores between not using any dimensionality reduction and using SVD or PCA for dimensionality reduction. For evaluation, the top *k* (here, 5) most similar startups are selected based on the cosine similarity scores. AP@K score measures the average precision value of the top *K* items in the list. The formula to calculate Average Precision@K is shown in Figure 4. Precision@i is the precision at the Ith position (within the ranked list). Relii is an indicator function that takes a value of one if the item is relevant at position *i* and zero otherwise. *K* is the cutoff position in the ranked list. The average precision takes into account both the precision and the number of relevant items retrieved within the top K positions. Higher values of AP@K indicate better rankings.

Figure 6 shows the comparison of the AP@K score between not using any dimensionality reduction (TF-IDF) and using SVD or PCA for dimensionality reduction. From Figure 4, it can be seen that the TF-IDF scheme offers a better AP@K value than the SVD and PCA schemes. Average Recall at *K* (AR@K) measures the average recall of the top *K* items in the list. The formula to calculate Average Recall@K is depicted in Figure 7, and Recall@i is the recall at ith position in the ranked list. Reli is an indicator function that takes a value of one if the item at *i*th position is relevant and zero otherwise. *K* is the cutoff position in the ranked list. Figure 7 shows the comparison of the Recall@K score between not using any dimensionality reduction (TF-IDF) and using SVD or PCA for dimensionality reduction. From Figure 5, it can be noticed that TF-IDF offers a better average Recall@K than both SVD and PCA schemes. The area under the precision-recall curve (AUC-PR) score summarizes the classifier's ability to balance precision and recall across different threshold values. Higher values of AUC-PR indicate better classifier performance. Figure 8 provides the AUC-PR score comparison for TF-IDF, SVD, and PCA schemes. MAP is a metric used to evaluate the effectiveness of information retrieval systems that return a ranked list of items. The formula to calculate MAP is shown in Figure 9. MAP considers the precision at different recall levels for each query and provides a single score that indicates the overall performance of the system. Higher values of MAP indicate better retrieval performance. Figure 9 indicates that the TF-IDF scheme provides a better MAP than both SVD and PCA schemes. NDCG examines the usefulness of the ranked list by providing higher scores to relevant items appearing higher in the list. The discounted cumulative gain (DCG) is computed by summing up the relevance scores of items at each position, with diminishing returns for items further down the list. The formula to calculate DCG@K is shown in Figure 10, where reli is the relevance score of the item at ith position (in the ranked list). *K* is the cutoff position in the ranked list. The ideal DCG (IDCG) is calculated by sorting the relevant items in decreasing order of relevance and computing the DCG@K with the same cutoff position *K*. NDCG is then obtained by normalizing the DCG by the IDCG. Higher values of NDCG indicate better-ranked lists that prioritize relevant items more effectively. Figure 10 shows that the TF-IDF scheme offers a better NDCG@K score than both SVD and PCA schemes.

Figure 6
Average precision score comparison

$$AP@K = \frac{\sum_{i=1}^{K}(Precision@i \times rel_i)}{\min(K, \text{number of relevant items})}$$

Algorithms	AP@K Score
TF-IDF (Term Frequency-Inverse Document Frequency)	0.1842
SVD (Singular Value Decomposition)	0.1433
PCA (Principal Component Analysis)	0.1457

Figure 7
Average recall score comparison

$$AR@K = \frac{\sum_{i=1}^{K}(Recall@i \times rel_i)}{\text{number of relevant items}}$$

Algorithms	$AR@K$ Score
TF-IDF (Term Frequency-Inverse Document Frequency)	0.9211
SVD (Singular Value Decomposition)	0.7166
PCA (Principal Component Analysis)	0.7286

4. Mobile App Development

The proposed investor and entrepreneur assistance mobile application offers several features such as login and signup feature,

Figure 8
AUC-PR score comparison

AUC-PR Score Comparison of Different Techniques

Algorithms	AUC-PR Score
TF-IDF (Term Frequency-Inverse Document Frequency)	0.1
SVD (Singular Value Decomposition)	0.1
PCA (Principal Component Analysis)	0.1

Figure 9
MAP score comparison

$$MAP = \frac{\sum_{q=1}^{Q}(AP_q)}{Q}$$

AP_q is the average precision for query q.
Q is the total number of queries

Algorithms	MAP Score
TF-IDF (Term Frequency-Inverse Document Frequency)	0.68395
SVD (Singular Value Decomposition)	0.4890
PCA (Principal Component Analysis)	0.4890

Figure 10
NDCG@K score comparison

$$DCG@K = \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

$$NDCG@K = \frac{DCG@K}{IDCG@K}$$

Algorithms	NDCG@K Score
TF-IDF (Term Frequency-Inverse Document Frequency)	0.8332
SVD (Singular Value Decomposition)	0.6972
PCA (Principal Component Analysis)	0.7011

user role selection, user profile, funding opportunity application, funder selection for startups, entrepreneur selection feature, and add review and rating. Figure 11(a) provides the login screen, in which users can sign in using their email and password information. Figure 11(b) offers a registration screen, in which the user needs to provide a name, email, password, and confirm password information for account creation. Users can only log in after account creation through the registration process. Figure 12(a) shows the user role selection option that allows users to choose their role upon registration or through their profile settings. This selection enables users to specify whether they want to use the application as an investor or in a different capacity. The application includes a profile page, where investors can provide and manage their profile information. User can access their profile page after login.

The profile page in Figure 12(b) allows users to input and update their preferences, such as company market, company region, investment criteria, and other relevant details. This profile page offers several features such as selecting entrepreneur for funding, selecting training, select funder (for entrepreneur), apply for funding feature, add rating, and review feature. Figure 13 shows some funding opportunity options for the investor. Users can select their respective companies and offer funding. Figure 14(a) offers the startup recommendation feature for

Figure 11
Login and signup page of the mobile app

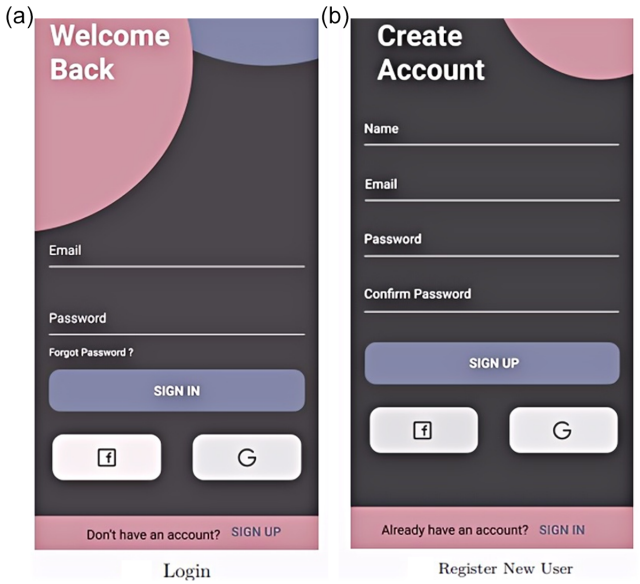
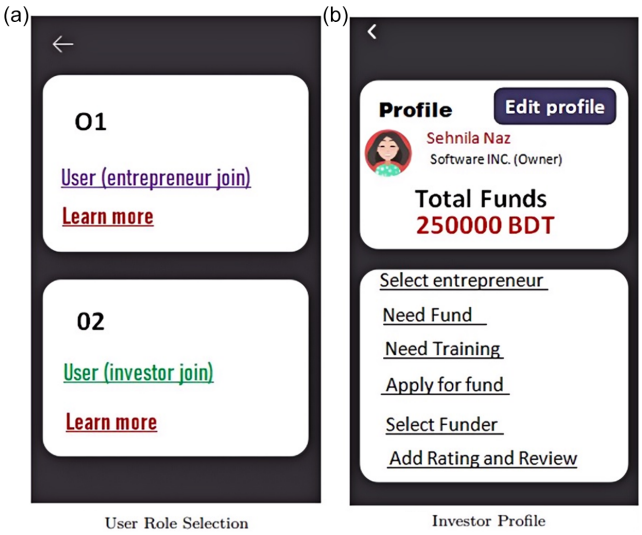


Figure 12
User role selection and investor profile



investment (for the investors) based on their reputation, company net worth, and yearly income value. Figure 14(b) shows the details of a startup company and funding options (for investors). Figure 15 depicts the entrepreneur selection feature for the investor based on market price and rating options. The investor can see the entrepreneur page and give reviews/ratings. Figure 16 visualizes the add review and rating feature. By using the feature, any user can provide feedback regarding the entrepreneur/investor and submit a rating point.

5. Evaluation Results

This section will present the application evaluation results. This mobile application was created using Flutter which is a front-end platform and Google Firebase software which is a backend platform. We conducted three separate surveys to validate our findings. The data are collected from users through an online and offline interview process. The first experiment uses interview

Figure 13
Recommendation and application for the fund

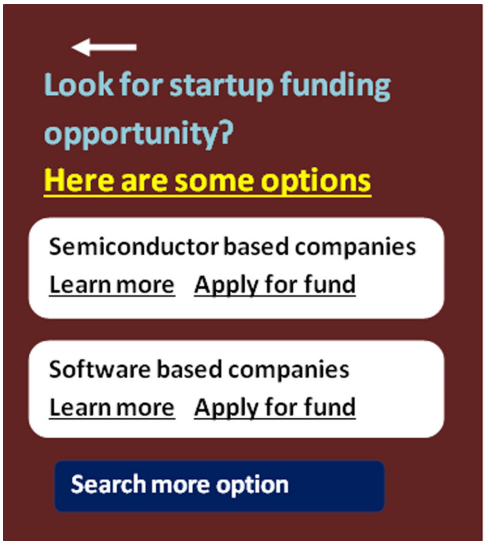


Figure 14
Funder recommendation and company profile

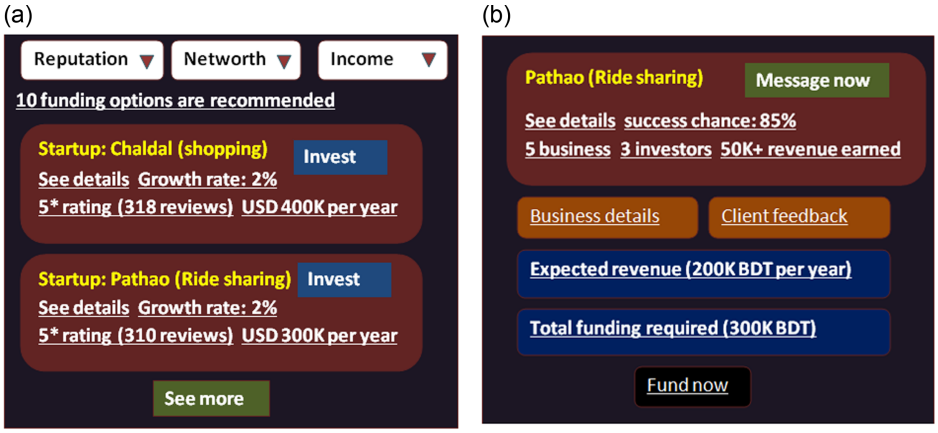


Figure 15
Entrepreneur suggestion for funding

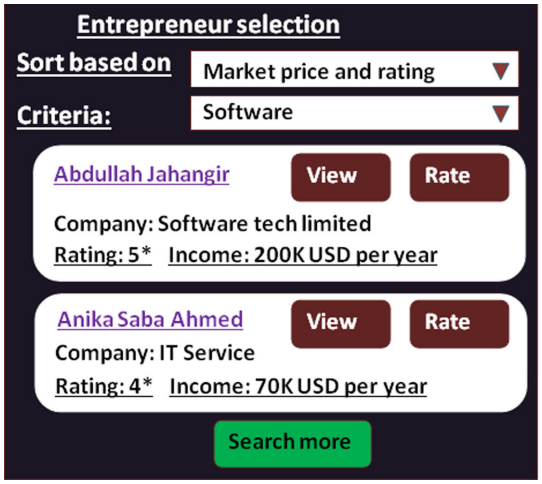
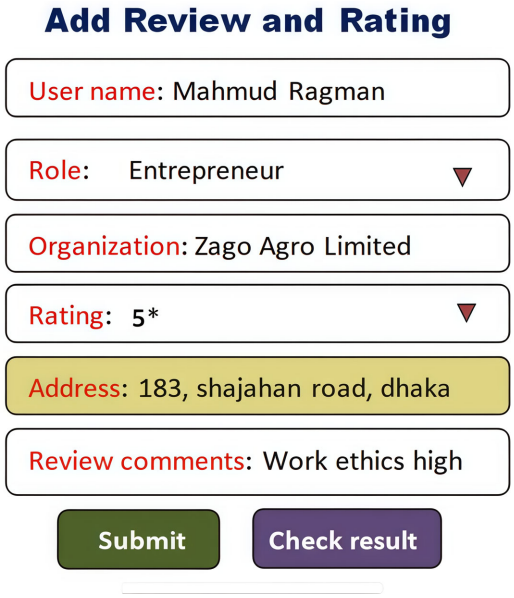


Figure 16
Review and rating



comments from 44 users. The second and third performance evaluations are conducted by collecting comments from 200 users. The users/reviewers' age range is between 20 and 50. The users/reviewers' professions include teachers, students, startup owners, IT specialists, and investors.

Figure 17 shows the application evaluation results. For this analysis, we have collected rating points regarding our proposed application from 44 users. The reviewers examined the trained model's performance, app design and navigation, ease of use, and overall performance of the mobile application. This feedback and rating system facilitates knowledge sharing, collaboration, and continuous improvement within the startup community. From Figure 17, it can be seen that the major portion of the reviewers/users selected a good comment regarding our proposed mobile application nature. On average, the number of average, worse, and no comment secured second, third, and fourth positions, respectively. Figure 18 investigates the proposed system's effectiveness by collecting responses from 200 users. The examining issues include app usefulness (time and cost), market readiness, user demand satisfaction, app response, and support. We discovered that for various effectiveness checking issues, users with

Figure 17
App evaluation by users

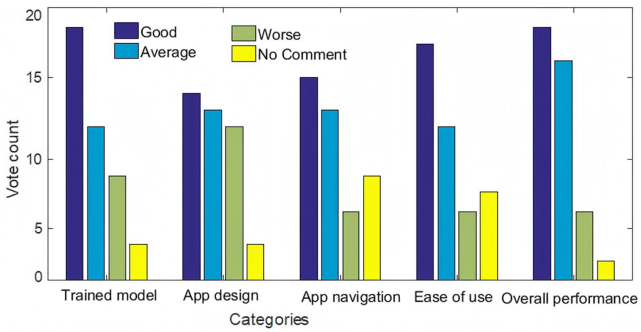


Figure 18
System effectiveness analysis

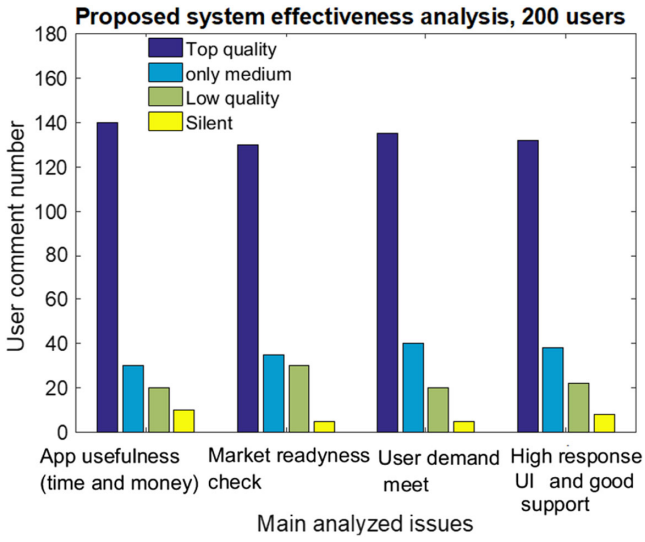
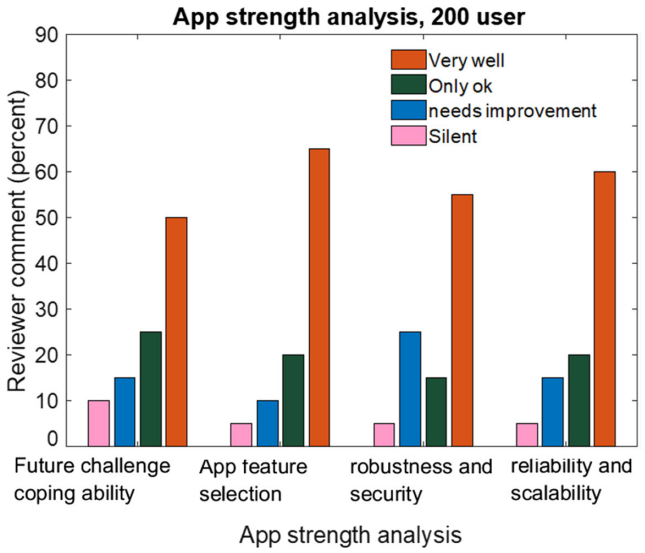


Figure 19
App strength analysis



top quality remarks, only medium quality, low quality, and silent remarks, ranked first (65%–70%), second (15%–20%), third (10%–20%), and fourth (2%–5%), respectively. Figure 19 examines application strength analysis. For this experiment, we used survey responses from 200 users. The app's strengths include whether or not it meets future challenges, developing appropriate app features, app robustness and security, and app reliability and scalability. For these strength factors, users ranked first (50%–60%), second (15%–25%), third (10%–25%), and fourth (5%–10%), respectively. Based on this app strength and effectiveness analysis, it is clear that our application receives appropriate feedback from app reviewers. At the moment, we have verified that over 20000 users can use our application without issue. The system configuration for our proposed application includes an Intel Core i7 processor-based server with 16 GB DDR4 memory and a 512 GB SSD. However, in the future, to deal with large numbers of users, we will try to incorporate high processing power server resources and cloud-based security resources into our system deployments.

6. Conclusion

This paper presented a ML-based recommendation system for predicting suitable startups for funding and matching them with investors. The proposed scheme recommends startups by evaluating company category, market, region, yearly profit, and growth rate. The proposed scheme involved data collection from reliable sources, data preprocessing, feature extraction, similarity calculations, and personalized recommendation generation. The system leveraged ML techniques, specifically TF-IDF and cosine similarity, to provide accurate and relevant recommendations. The results showed that the TF-IDF-based approach (without any dimensionality reduction) offers better average precision, average recall, MAP, and NDCG value than the SVD or PCA (with dimensionality reduction). Thus, TF-IDF scheme is selected for startup recommendation due to its high recall and precision value. This work also provides an investor and entrepreneur assistance mobile application with a multi-factor-based entrepreneur selection feature, login/signup feature, add review and rating feature, startup selection, funding application, and user profile feature. The application evaluation results indicated that more than 60 percent of users are satisfied with the proposed application features. Moreover, more than 70 percent of reviewers are satisfied with the application's scalability, features, security, and robustness issues. To achieve higher accuracy and relevant recommendations information, future work can focus on refining the recommendation algorithm by incorporating more advanced ML techniques (e.g., transfer learning, deep learning with attention). Expanding the dataset by incorporating data from various sources, such as social media, news articles, or industry reports, can provide a more comprehensive understanding of startups and investors. Stakeholders can use our proposed system for suitable startup and investor selection. The mobile application can add some additional features such as security enhancement features, quantum computing, ML, and federated learning-based customer security and data privacy enhancement, collaboration and networking features, additional factor-based investor and entrepreneur recommendations, real-time monitoring, and tracking feature, among others. Other research issues that can be investigated in the future include different geographical locations and different business-based investor selection using artificial intelligence and deep learning, fake investor identification, fake startup determination by using ML, investor risk analysis, customer survey-based startup selection taking future market gains and customer demands into account, and so on.

7. Recommendation

The findings hinted that TF-IDF scheme is most suitable for investor and startup selection.

Acknowledgement

The authors are grateful to CUET, CSE department for research facilities.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The dataset used in this paper is available on the Crunchbase website.

Author Contribution Statement

Asfak Asif: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Mahfuzulhoq Chowdhury:** Conceptualization, Methodology, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration.

References

[1] Luef, J., Ohrfandl, C., Sacharidis, D., & Werthner, H. (2020). A recommender system for investing in early-stage enterprises. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 1453–1460. https://doi.org/10.1145/3341105.3375767

[2] Xu, S., Zhang, Q., Lü, L., & Mariani, M. S. (2020). Recommending investors for new startups by integrating network diffusion and investors' domain preference. *Information Sciences*, *515*, 103–115. https://doi.org/10.1016/j.ins.2019.11.045

[3] Narode, S., Pitrubhakta, H., Sonawane, A., Nikam, V., & Kadam, S. (2018). Recommendation system for startups. *International Journal of Engineering and Techniques*, *4*(2), 471–475.

[4] Davila, A., & Foster, G. (2005). Management accounting systems adoption decisions: Evidence and performance implications from early-stage/startup companies. *The Accounting Review*, *80*(4), 1039–1068. https://doi.org/10.2308/accr.2005.80.4.1039

[5] Miloud, T., Aspelund, A., & Cabrol, M. (2012). Startup valuation by venture capitalists: An empirical study. *Venture Capital*, *14*(2–3), 151–174. https://doi.org/10.1080/13691066.2012.667907

[6] Liu, X., & Wangperawong, A. (2018). A collaborative approach to angel and venture capital investment recommendations. *arXiv Preprint:1807.09967*. https://doi.org/10.48550/arXiv.1807.09967

[7] Färber, M. (2019). Linked crunchbase: A linked data API and RDF data set about innovative companies. *arXiv Preprint:1907.08671*. https://doi.org/10.48550/arXiv.1907.08671

[8] Zhao, X., Zhang, W., & Wang, J. (2015). Risk-hedged venture capital investment recommendation. In *Proceedings of the 9th ACM Conference on Recommender System*, 75–82. https://doi.org/10.1145/2792838.2800181

[9] Zhong, H. (2019). *Venture capital investment: From rule of thumb to data science*. Doctoral Dissertation, Rutgers University. https://doi.org/doi:10.7282/t3-t0xh-zy74

[10] Bai, S., & Zhao, Y. (2021). Startup investment decision support: Application of venture capital scorecards using machine learning approaches. *Systems*, *9*(3), 55. https://doi.org/10.3390/systems9030055

[11] Silva, J. (2004). Venture capitalists' decision-making in small equity markets: A case study using participant observation. *Venture Capital*, *6*(2–3), 125–145. https://doi.org/10.1080/13691060410001675974

[12] Kim, J. H., & Wagman, L. (2016). Early-stage entrepreneurial financing: A signaling perspective. *Journal of Banking and Finance*, *67*, 12–22. https://doi.org/10.1016/j.jbankfin.2016.03.004

[13] Alexy, O. T., Block, J. H., Sandner, P., & Ter Wal, A. L. (2012). Social capital of venture capitalists and start-up funding. *Small Business Economics*, *39*(4), 835–851. https://doi.org/10.1007/s11187-011-9337-4

[14] Bosch, K. T. D. (2022). *A hybrid recommender-system for startup scouting*. Master's Thesis, University of Twente.

[15] Liang, Y. E., & Yuan, S. T. D. (2016). Predicting investor funding behavior using crunchbase social network features. *Internet Research*, *26*(1), 74–100. https://doi.org/10.1108/IntR-09-2014-0231

[16] Raju, M. N. (2020). Recommender system for food startup. *IOP Conference Series: Materials Science and Engineering*, *993*, 012054. https://doi.org/10.1088/1757-899X/993/1/012054

[17] Tikka, J. A. (2022). *Effects of early-stage funding decisions to the performance of Finnish Born Global start-ups*. Master's Thesis, University of VAASA.

[18] Arroyo, J., Corea, F., Jimenez-Diaz, G., & Recio-Garcia, J. A. (2019). Assessment of machine learning performance for decision support in venture capital investments. *IEEE Access*, *7*(1), 124233–124243. https://doi.org/10.1109/ACCESS.2019.2938659

[19] Eloranta, O. (2018). *Investment criteria of Finnish venture capitalists*. Master's Thesis, LUT University of Technology.

[20] Ross, G., Das, S., Sciro, D., & Raza, H. (2021). CapitalVX: A machine learning model for startup selection and exit prediction. *The Journal of Finance and Data Science*, *7*, 94–114. https://doi.org/10.1016/j.jfds.2021.04.001

[21] Ang, Y. Q., Chia, A., & Saghafian, S. (2022). Using machine learning to demystify startups' funding, post-money valuation, and success. In V. Babich, J. Birge & G. Hilary (Eds.), *Innovative technology at the interface of finance and operations* (pp. 271–296). Springer.

[22] Yadav, A., & Sharma, N. (2024). Deep learning's diverse impact on sustainability of agricultural tech startup. In *IEEE International Conference on Computing, Power and Communication Technologies*, 610–615.

[23] Rajendran, V., Tiwari, A., & Tripathi, V. (2024). Application of AI in determining the strategies for the startups. *International Conference on Healthcare Innovations, Software and Engineering Technologies*, 374–376.

[24] Abuzaid, A., & Alsbou, M. (2024). AI and entrepreneurship: Enablers, obstacles, and startups' role in shaping the future economy. In *International Conference on Knowledge Engineering and Communication Systems*, 1–6.

[25] Vukmirović, S., & Fabić, M. (2024). The influence of IT education on the performance and IT usage of startups in Croatia. In *47th MIPRO ICT and Electronics Convention*, 526–531.

How to Cite: Asif, A., & Chowdhury, M. (2024). A Machine Learning-Based Suitable Startup Recommendation Scheme and Investor Assistance Mobile Application. *Journal of Comprehensive Business Administration Research*. https://doi.org/10.47852/bonviewJCBAR42022677