Founder Matching: Optimizing Team Formation with Machine Learning

INDENG 242 – Machine Learning and Data Analytics
Project Report

Idris Houir Alami
Youssef Miled
Ryan Michael Chekkouri

December 18, 2025

Abstract

Forming effective startup teams is a critical yet largely unstructured challenge, often relying on ad hoc networking or heuristic-based matching. This report presents a data-driven approach to cofounder matching that leverages machine learning to optimize team formation based on both similarity and complementarity of founder profiles. Using a synthetically generated yet realistic dataset of founder attributes, we develop a collaborative filtering model that learns latent founder archetypes through matrix factorization, enabling personalized pairwise compatibility recommendations. To address the limitation of pairwise matching when forming larger teams, we extend the methodology with a clustering-based approach designed to promote complementary skill composition within multi-founder teams. The strong generalization performance and interpretability of user archetypes highlights the model's potential as a scalable decision-support tool for founder matching platforms while also identifying key limitations related to outcome validation and team-level optimization.

Video Presentation: https://www.youtube.com/watch?v=Bpox4HFx8yk

GitHub Repository: https://github.com/idrisalami/ML$_2$42A

Contents

1 Introduction

1.1 Project Description and Motivation

Building a successful startup is a high-risk endeavor, and industry data consistently identifies the "team" as a primary determinant of success or failure. The process of finding a suitable co-founder, however, remains largely unstructured, relying on serendipitous networking or inefficient manual filtering. While platforms like YC Match (a "Tinder for founders") exist, they often rely on simple heuristic filters rather than intelligent, data-driven compatibility scoring.

In this project, we aim to conceptualize and build a **Co-Founder Matching System**. Our use-case is a platform that takes a founder's profile—including their skills, personality traits, and startup ideas—and recommends the most compatible partners.

The motivation is to move towards a more nuanced approach that considers:

- **Complementary Skills**: Matching technical founders with business-oriented founders.
- **Behavioral Compatibility**: Aligning risk tolerance, communication styles, and leadership tendencies.
- **Shared Vision**: Matching founders interested in similar problem spaces or industries.

This project applies machine learning techniques to automate and optimize this matching process, aiming to reduce the friction in team formation.

2 Data and Learning Problem

2.1 Dataset Description and Generation

For this project, we required a rich dataset of potential founders. Since public datasets of detailed co-founder profiles are scarce, we constructed a synthetic yet realistic dataset based on real-world social data.

We utilized a seed dataset of social profiles (`Tinder_Data_v3_Clean_Edition.csv`). To transform this into a professional context, we built a data generation pipeline using a Large Language Model (LLM). For each of the approximately 1,200 profiles, we fed the existing biographical data (age, gender, bio) into the LLM with a specific prompt to hallucinate a plausible "Founder Persona."

The resulting `founders_dataset.csv` includes:

- **Professional Attributes**: `preferred_role` (CEO, CTO, COO, etc.), `industry`, `is_technical`, `years_of_experience`, and `education_level`.
- **Skills**: Lists of `tech_stack`, `strengths`, and `weaknesses`.
- **Startup Concept**: `idea_title`, `idea_description`, and `problem_space`.
- **Behavioral Scores**: Numerical scores (scale 1-5 or 0-1) for traits such as `risk_tolerance`, `leadership`, `collaboration_openness`, and `communication_intensity`.

2.2 Data Exploration and Relationships

We expect to see natural correlations in the data that our models can leverage. For instance, "technical" founders (CTOs) likely have high proficiency in specific tech stacks (Python, React) but might self-report lower scores in "sales" or "public speaking" (weaknesses). Conversely, CEOs might index higher on "leadership" and "vision" scores.

We explicitly engineered the generation process to create a balanced distribution of roles (weighted towards CEOs and CTOs) and diverse industries (Fintech, Healthtech, Consumer Social) to simulate a realistic founder pool.

2.3 Challenges with the Data

A significant challenge in this dataset lies in the unstructured text features: `idea_title`, `idea_description`, and `problem_space`. These fields contain full sentences and semantic nuance that are critical for matching founders with a shared vision.

Standard preprocessing techniques like Label Encoding or One-Hot Encoding are unsuitable for these text-heavy features. To fully utilize them, Natural Language Processing (NLP) techniques such as text embeddings (e.g., TF-IDF or BERT) would be required to calculate semantic similarity. While we acknowledge this necessity, extensive NLP implementation is beyond the current scope of this project. For our core models, we primarily rely on the structured categorical data and numerical behavioral scores, treating the text fields as metadata for future expansion.

3 Methodology

To address the matching problem, we propose comparing two distinct machine learning approaches. This comparison allows us to evaluate different paradigms of "team formation":

3.1 Collaborative Filtering (Pair-wise Matching)

This section describes the methodology used to construct a collaborative filtering system for cofounder matching. The system transforms each founder's attributes into numerical representations, computes similarity and complementarity relationships, constructs a combined compatibility matrix, and learns latent archetypes through matrix factorization.

3.1.1 Data Preprocessing and Feature Engineering

Each founder's attributes must first be converted into numerical feature vectors. Features are classified into two categories: (1) similarity features, where alignment increases compatibility (industry, communication style scores, responsiveness, risk tolerance, execution speed, collaboration openness, semantic embeddings from idea text fields), and (2) complementarity features, where differences strengthen a founding team (roles, tech_stack, strengths, weaknesses, preferred roles, experience, education level).

Multi-label variables are encoded as multi-hot vectors. Single-label categories are one-hot encoded. Numerical attributes are standardized. Free-text fields are embedded using a sentence-transformer model to obtain dense semantic representations.

All encoded features are concatenated into a unified matrix X. For similarity computations, each feature vector is L2-normalized. Based on these feature groups, we construct our own "true" compatibility values R_{ij} prior to learning the approximation $\hat{R}_{ij}$ via matrix factorization.

3.1.2 Similarity and Complementarity Computation

Similarity Matrix Similarity is computed across features where alignment matters. Using cosine similarity $\cos(x, y) = (x \cdot y)/(\|x\|\|y\|)$ applied to X_{sim}, we compute the similarity matrix S_{ij}, reflecting alignment in mission, communication style, and idea space.

Complementarity Matrix Complementarity is computed across features where difference is beneficial. For multi-hot vectors, complementarity uses Jaccard distance: $C_{ij} = 1 - J(A, B)$ where $J(A, B) = |A \cap B|/|A \cup B|$. Jaccard ignores co-absence and reflects functional non-overlap. Numerical complementarity uses standardized distances.

Combined Compatibility Matrix Similarity and complementarity are combined with equal weight:

$$R_{ij} = 0.5\, S_{ij} + 0.5\, C_{ij}.$$

3.1.3 Matrix Factorization

Matrix factorization decomposes R into two matrices P and Q, where each founder is represented as a "user" vector $P[i]$ and an "item" vector $Q[j]$. Predicted compatibility is:

$$\hat{R}_{ij} = \mu + b_u[i] + b_i[j] + P[i] \cdot Q[j].$$

Here, μ is the global average, $b_u[i]$ measures how founder i tends to evaluate others, $b_i[j]$ measures how founder j tends to be evaluated, and $P[i] \cdot Q[j]$ models latent interactions.

We minimize:

$$\sum_{i,j} (R_{ij} - \hat{R}_{ij})^2 + \lambda(\|P\|^2 + \|Q\|^2 + \|b_u\|^2 + \|b_i\|^2),$$

using Stochastic Gradient Descent (SGD). Regularization λ prevents overfitting.

3.1.4 Model Training, Hyperparameter Tuning, and Recommendation Output

The collaborative filtering model is trained using stochastic gradient descent to minimize the regularized squared error objective. Two key hyperparameters govern model expressiveness and generalization: the number of latent archetypes A and the regularization coefficient λ. These parameters are optimized using grid search with cross-validation, selecting the configuration that minimizes validation RMSE and achieves a clear bias–variance tradeoff.

After training, the model computes predicted compatibility scores $\hat{R}_{ij}$ for all pairs of founders. For each founder, these scores are ranked to produce a personalized list of recommended co-founders, balancing similarity in vision and behavior with complementarity in skills and roles.

3.2 Clustering (Team Formation)

Complementing the pair-wise approach, we use Clustering algorithms (K-Means) to partition the founder pool into groups. This method simulates the formation of co-founding teams (size 2+) by maximizing intra-cluster diversity (ensuring a mix of roles like CEO + CTO) while maintaining compatibility in vision and industry interest.

3.2.1 Data preprocessing for clustering

Each observation in this dataset corresponded to one particular combination of feature types: categorical variables, numerical metrics, and multi-label attributes (we decided to drop NLP variables because it is outside of the scope of this class). We encoded multi-label variables and categorical values. We also did numerical standardization.

The resulting feature matrix combined all the transformed features into 368 dimensions, each founder possessed. In order to handle high dimensionality and potential noise, Principal Component Analysis was performed to reduce the features down to 29 components, retaining 90% of the variance. This significantly improves the efficiency of the clustering process.

3.2.2 Complementarity-based clustering methodology

Conventional similarity-based clustering clusters founders with similar attributes, which is not ideal for team formation with complementary skills. We developed a complementarity-based clustering solution that clusters founders into complementary skill, role, or attribute clusters.

A custom complementarity scoring formula was used to evaluate the affinity between founders for each of the four important factors. Role difference was given 35% weight as a binary metric of differences in favored roles. Role diversity was given 30% weight as Jaccard diversity of secondary roles. Tech stack diversity was given 20% weight as Jaccard diversity of technical skills. Strengths-weaknesses overlap was given 15% weight as a metric of how well one founder's strengths overlap with another's weaknesses.

This complementarity matrix is used for clustering for grouping founders who would make balanced teams as opposed to homogeneous founders.

K-Means clustering was then performed on the reduced data space using the principal components. Using the elbow method, k=6 clusters were found to strike the right balance between intra-cluster difference and inter-clusters.

4 Results and Discussion

4.0.1 Results and Discussion (Collaborative Filtering)

The matrix factorization model learned 24 latent archetypes that capture distinct and interpretable patterns in founder characteristics. By correlating each archetype's latent weights with the original features, we identify meaningful founder profiles. These results demonstrate that the learned latent factors align with intuitive archetypes. Detailed archetype interpretations are provided in the Appendix through correlation heatmaps.

The model performance is evaluated using RMSE. The best-performing configuration achieved a training RMSE of 0.0789, a validation RMSE of 0.0796, and a test RMSE of 0.0790. The

small gap between validation and test errors indicates strong generalization and minimal overfitting. This behavior reflects an appropriate bias vs variance tradeoff achieved through tuning the number of archetypes and the regularization coefficient λ.

Overall, the collaborative filtering approach scales efficiently and generates personalized recommendations for all founders in the dataset. The latent archetypes uncover non-obvious structure in founder profiles that can inform strategic matching decisions. Moreover, the explicit combination of similarity (shared industry, communication style, and risk tolerance) and complementarity (roles, skills, and experience) reflects the practical requirements of successful cofounder teams.

On the other hand, the primary constraint is the absence of ground-truth outcome data linking recommendations to actual cofounder success, which limits the interpretation of RMSE as a proxy for real-world effectiveness. Additionally, the collaborative filtering framework operates at the pairwise level, as similarity and complementarity are defined between two founders at a time. Founders aiming to form larger teams of four or five individuals are therefore not addressed. In such cases, we need an approach beyond pairwise matching. This where the clustering-based approach comes into play to support multi-founder team formation.

4.0.2 Cluster analysis and complementary pair evaluation

Overall, the cluster profiles reveal a founder population dominated by hybrid CEO-CTO types who combine deep technical fluency with product vision and leadership, but who vary in scale, maturity, and style of execution. Across all clusters, creativity, user empathy, and scrappiness consistently emerge as core strengths, suggesting a common bias toward vision-first, customer-centric building. Python, React, Node.js, and AWS form a common technical backbone, with lighter-weight tools such as Figma, Webflow, and Firebase appearing more in smaller or earlier-stage clusters.

The key differences are in experience and focus: clusters with heavier CTO representation-for example, Clusters 2 and 5, display greater technical depth and rapid learning, struggling more with the business fundamentals, hiring, and delegation, while more CEO-heavy clusters, like Clusters 1 and 3, exhibit the reverse, bringing stronger leadership and ownership with risks of perfectionism, distraction, and burnout. Early-stage or scrappier profiles, such as Cluster 0, bring creativity and speed but may struggle with focus and prioritization. Perhaps most consistently across nearly all groups, struggles to delegate and intolerance of slow processes mark systemic weaknesses and a core transition challenge from individual contributor to scalable leader.

A Appendix

A.1 Model Selection and Archetype Analysis

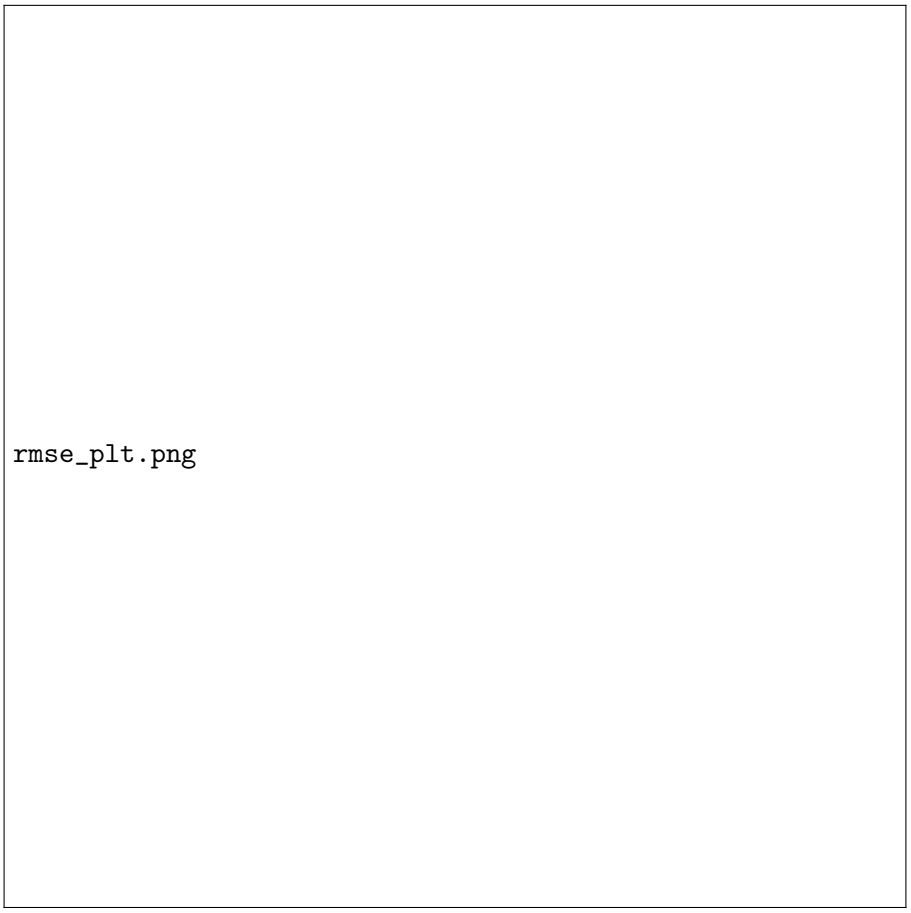

Figure 1: Plot of RMSE to the number of archetypes tradeoff

A.2 Archetype Interpretation

The matrix factorization model identifies 24 latent founder archetypes, each capturing distinct and interpretable patterns in founder characteristics. Archetypes are interpreted by analyzing correlations between their latent weights and the original behavioral, technical, and industry features. Below, we describe the most salient archetypes based on their strongest associations.

Archetype 1: Risk-Taking Innovators Archetype 1 primarily captures risk-taking tendencies. High weights on this archetype results in elevated risk tolerance (0.228) and interest in Climate/Greentech (0.240) and Biotech (0.204). This archetype is also associated with higher responsiveness (0.187) and familiarity with TensorFlow (0.125).

Archetype 2: Generalist Collaborators Archetype 2 represents generalist founders with moderate responsiveness (0.123) and openness to collaboration (0.096), but no strong industry or technical specialization. Its relatively uniform associations across multiple industries suggest broad applicability rather than domain depth.

Figure 2: Archetype weight by industry heatmap

Archetype 3: Technical Specialists Archetype 3 captures technically sophisticated founders, it captures proficiency in TensorFlow (0.195) and Solidity (0.112). An interest in AI/Deeptech (0.277) and Consumer/Social (0.222) sectors is also observed. This archetype also exhibits moderate risk tolerance (0.133) and secondary associations with Fintech, Climate/Greentech, and Web3/Crypto.

Archetype 4: Data-Driven Fintech Builders Archetype 4 is strongly aligned with Fintech founders (0.292) with some overlap with AI/Deeptech (0.180). These founders display high responsiveness (0.172) and strong analytical tool familiarity including TensorFlow (0.109), Webflow (0.083), and Tableau (0.077). This archetype reflects data-driven, execution-oriented founders in financial technology.

techstack.png

Figure 3: Archetype weight by tech stack heatmap

Archetype 5: Product-Focused Leaders Archetype 5 corresponds to product-focused founders, particularly those with CPO roles (role: 0.070, preferred role: 0.067). It is most prominent in Fintech (0.254) and Healthtech (0.207), with additional presence in SaaS and AI/Deeptech. This archetype is associated with TensorFlow usage (0.073) and characteristic weaknesses such as perfectionism (0.072), prioritization challenges (0.063), and weaker sales capabilities (0.059), consistent with product-centric leadership profiles.

Overall, these archetypes capture meaningful latent dimensions that integrate behavioral traits, technical skills, industry focus, and functional roles. Their interpretability is critical for explaining founder-matching recommendations and for fostering trust in the system's outputs.

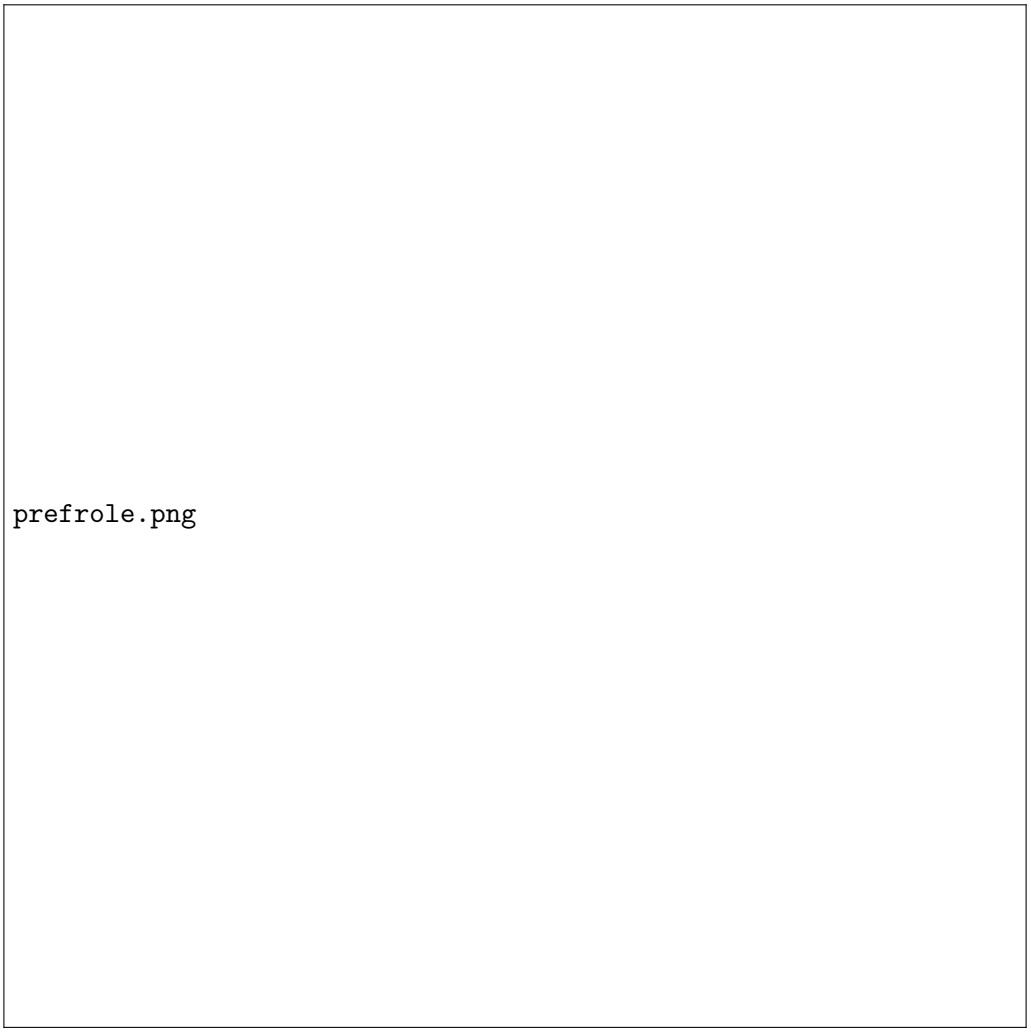

Figure 4: Archetype weight by preferred role heatmap