Founder Matching: Optimizing Team Formation with Machine Learning

INDENG 242 - Machine Learning and Data Analytics

Idris Houir Alami Youssef Miled Ryan Michael Chekkouri

University of California, Berkeley

Motivation

- Team quality is a key determinant of startup success
- Founder matching is often informal and heuristic-based
- Existing platforms lack data-driven compatibility modeling

Goal

Build a scalable, interpretable ML system for cofounder matching that balances:

- **similarity** (vision, communication style)
- **complementarity** (skills, roles, experience)

Dataset

- Synthetic but realistic dataset of $\sim$ 1,200 founders
- Generated using LLM-based persona construction

Feature types:

- **Categorical:** industry, preferred role
- **Multi-label:** tech stack, strengths, weaknesses, roles
- **Numerical:** risk tolerance, collaboration openness, experience
- **Text (idea title/description):** used for similarity analysis

Learning Problem

- Founders act as both "users" and "items"
- **Objective:** predict compatibility score R_{ij} between founders i and j

$$R_{ij} = \alpha S_{ij} + (1 - \alpha) C_{ij}$$

- $\alpha = 0.5$
- S_{ij}: similarity score
- C_{ij}: complementarity score

Feature Engineering

Similarity Features

- Industry
- Behavioral scores (risk, communication, responsiveness)
- Idea semantics

Complementarity Features

- Roles and preferred roles
- Tech stack
- Strengths and weaknesses
- Experience and education

Similarity Computation

Similarity is computed using cosine similarity:

$$cos(x, y) = \frac{x \cdot y}{||x||||y||}$$

- Applied to L2-normalized similarity feature matrix
- Captures alignment in vision, style, and industry

Complementarity Computation

- **Multi-label features use Jaccard distance:**
  - Ignores co-absence
  - Emphasizes functional diversity

$$C_{ij} = 1 - \frac{|A \cap B|}{|A \cup B|}$$

- Numerical features use standardized distances

Collaborative Filtering Model

We apply matrix factorization:

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

- $P, Q \in \mathbb{R}^{n \times A}$: latent archetypes
- A: number of archetypes

Optimization

Objective function:

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

- Optimized using SGD
- Grid search over A and λ

Results: Model Performance

Best configuration:

- Archetypes $A = 24$
- Regularization $\lambda = 0.02$

RMSE:

- Train: 0.0789
- Validation: 0.0796
- Test: 0.0790
- Small validation-test gap $\rightarrow$ good generalization

RMSE vs Number of Archetypes

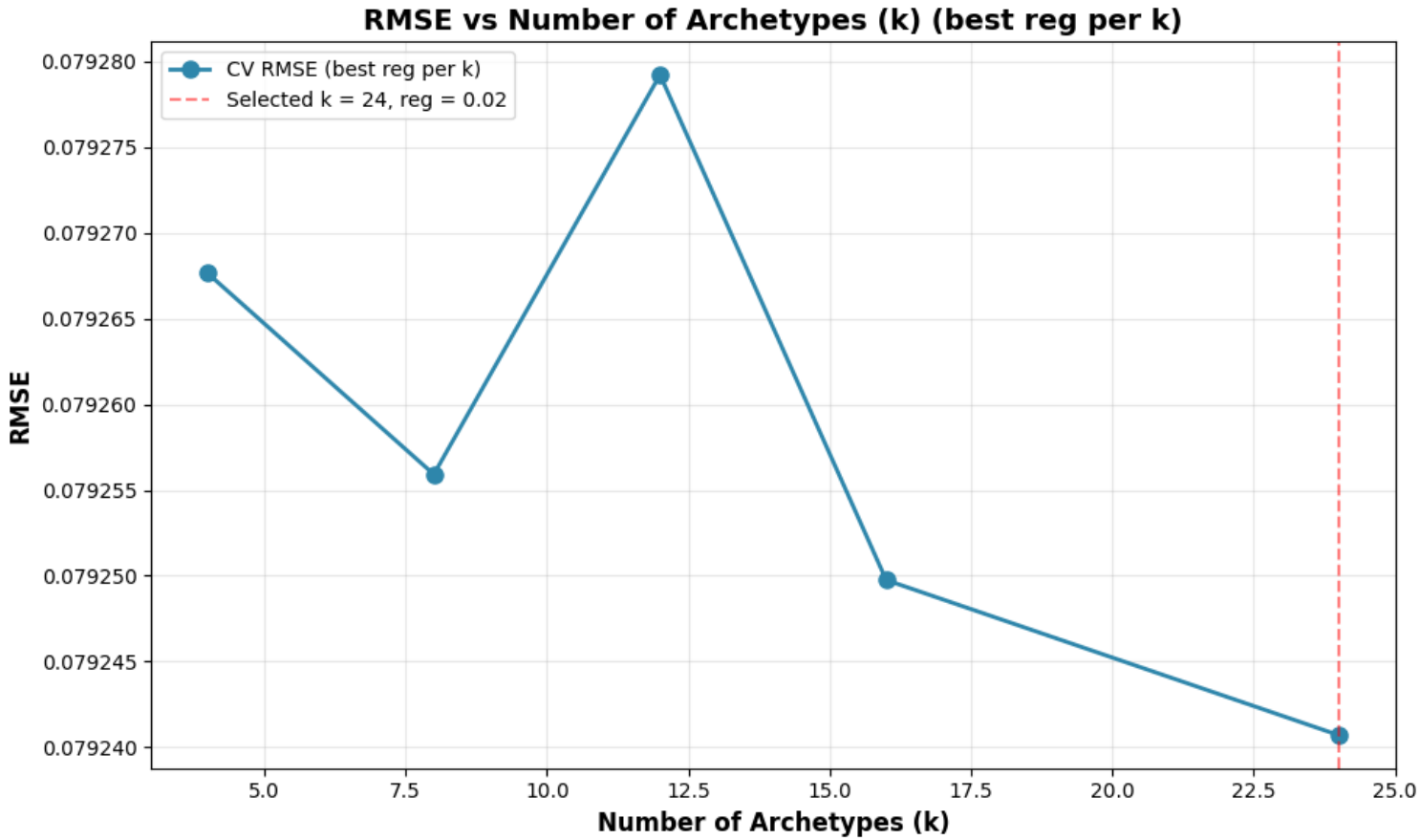

Figure: RMSE as a function of the number of latent archetypes

Latent Archetype Interpretation

- 24 interpretable founder archetypes learned
- Capture patterns across industry, skills, behavior, and roles

Examples:

- Risk-tolerant Climate/AI founders
- Product-focused CPO profiles
- Technical CTO-heavy archetypes

See heatmaps in appendix for details

Limitations

- No ground-truth outcome data (no success labels)
- RMSE does not directly measure recommendation quality
- Pairwise matching only

Motivation for clustering:

- Team formation beyond pairs (4–5 founders)

Clustering for Team Formation

- Simulates co-founding team formation (size 2+)
- Maximizes intra-cluster diversity (roles: CEO, CTO, etc.)
- Maintains compatibility in vision and industry

Goal

- Form balanced, multi-founder teams
- Maximize role and skill diversity

Data Preprocessing for Clustering

- **Feature types:** categorical, numerical, multi-label
- **Dropped NLP variables** (outside scope)
- **Initial dimensions:** 368 features per founder

Dimensionality Reduction via PCA:

- Retained 90% of variance
- Improves efficiency and reduces noise

Complementarity Scoring Formula

$$C_{ij} = 0.35 \cdot RD + 0.30 \cdot RDiv + 0.20 \cdot TD + 0.15 \cdot SW$$

- **Role Difference (35%):** Binary metric of different preferred roles
- **Role Diversity (30%):** Jaccard diversity of secondary roles
- **Tech Stack Diversity (20%):** Jaccard diversity of technical skills
- **Strengths-Weaknesses Overlap (15%):** How strengths cover weaknesses

Creates balanced teams rather than homogeneous clusters

Clustering Results & Analysis

K-Means with $k = 6$ clusters (elbow method)

Key Findings:

- Founder population dominated by **hybrid CEO-CTO types**
- **Common strengths:** creativity, user empathy, scrappiness
- **Tech backbone:** Python, React, Node.js, AWS
- **CTO-heavy clusters** (2, 5): deeper technical depth, weaker business fundamentals
- **CEO-heavy clusters** (1, 3): strong leadership, risks of perfectionism
- **Early-stage clusters** (0): creative & fast, but struggle with focus
- **Systemic weakness:** struggles to delegate, intolerance of slow processes

Conclusion

- Collaborative filtering provides scalable, interpretable pairwise matching
- Latent archetypes uncover meaningful founder structure
- Clustering extends approach to team-level formation

Future Work

- Outcome-based validation
- Human-in-the-loop feedback
- Multi-objective optimization