# Startup Success Forecasting Framework: A Multi-Agent Framework for Startup Success Prediction

**Xisen Wang**[1]* xisen.wang@keble.ox.ac.uk
**Fuat Alican**[2] fuat@vela.partners
**Yigit Ihlamur**[2] yigit@vela.partners

[1]University of Oxford, Oxford, UK [2]Vela Partners, San Francisco, USA

## Abstract

Predicting startup success is highly uncertain and has long relied on VC intuition. While large language models (LLMs) promise new capabilities, our experiments reveal a fundamental flaw: when applied directly, LLMs systematically *over-predict*, yielding poor precision under severe class imbalance. Hence, we introduce the **Startup Success Forecasting Framework (SSFF)**, the first multi-agent architecture for early-stage venture evaluation. By combining LLM-enhanced prediction (random forests, founder–idea fit), multi-agent analysis (segmentation, evaluation), and retrieval-augmented market intelligence, SSFF delivers structured, interpretable assessments. On a realistic dataset (10% success rate), SSFF reduces optimism bias achieving a nearly 6× performance improvement over GPT baselines. More broadly, SSFF serves as a generalizable template for integrating LLMs with conventional machine learning in high-stakes, imbalanced decision-making tasks.

## 1 Introduction

Evaluating startups at inception is a notoriously difficult task, traditionally dependent on the judgment of seasoned venture capital (VC) analysts. The inherent dynamism of startups, coupled with the unpredictability of market reception, makes it challenging to identify ventures poised for long-term success [1, 2].

Recent advances in large language models (LLMs) and multi-agent systems raise the possibility of automating aspects of such reasoning. Yet the predictive reliability of LLMs in this setting has scarcely been tested. With known limitations such as hallucinations and bias, it remains unclear whether LLMs can provide trustworthy assessments of startup potential.

In this work, we uncover a fundamental failure mode: when applied directly to startup forecasting, vanilla LLMs display a systematic *over-prediction bias*, consistently overestimating the likelihood of success. This phenomenon is not only a barrier to practical deployment in venture contexts, but also reveals an important gap in our understanding of how LLMs behave under extreme class imbalance and noisy founder narratives.

To address this limitation, we introduce the **Startup Success Forecasting Framework (SSFF)**, a multi-agent architecture designed not as a monolithic predictor, but as a modular and interpretable system. SSFF integrates three blocks that collectively mitigate optimism bias: (i) a *prediction block* that fuses LLM-derived features with machine learning (ML) models such as random forests and neural networks, (ii) an *analysis block* that incorporates founder segmentation and idea–fit scoring

*Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Generative AI for Finance.

to expose the underlying drivers of outcomes, and (iii) an *external knowledge block* that retrieves real-time market intelligence to ground evaluations beyond founder self-descriptions.

Rather than positioning SSFF as a benchmark competitor, we frame it as a *generalizable template* for integrating LLMs with conventional ML in high-stakes, imbalanced decision-making. Startup forecasting serves as an illustrative case study: vanilla LLMs exhibit a clear over-prediction bias, while SSFF's modular design yields measurable gains in both accuracy and interpretability.

**Contributions.** Our work makes three contributions: (1) we identify and quantify a systematic over-prediction bias in LLM-based startup forecasting, exposing a broader limitation of LLMs under severe class imbalance; (2) we propose **SSFF**, the first multi-agent framework explicitly designed to mitigate this failure mode through modular prediction, analysis, and external knowledge integration; and (3) we demonstrate SSFF's effectiveness on a realistic startup dataset, showing improved performance over LLM-only baselines as well as interpretability features (e.g., founder segmentation, idea–fit scoring) that support decision transparency.

## 2 Methodology

### 2.1 Baseline: LLMs as Startup Predictors

**Dataset Setup** We begin by evaluating the predictive reliability of vanilla LLMs using a curated dataset of early-stage ventures. The primary pool contains **9,745 startups** (41.4% classified as successful). Founders in the dataset average **30.8 years of professional experience**, with 81.3% of companies including at least one prior startup founder. The sector coverage is broad, with Software, Health Care, and IT being the most frequent, and the top geographies represented are the USA, the UK, and India. Additional descriptive statistics and exploratory data analysis are provided in the Supplement.

Following prior work (e.g., FounderGPT [3]), we define a startup as **successful** if it either raised $\geq \$500M$, or was acquired for or reached an IPO valuation of $\geq \$500M$; all others are labeled unsuccessful. Founder profiles were collected from verified LinkedIn pages, restricted to historical data available *up to the time of founding or first VC involvement*, ensuring no leakage of post-outcome information. These records include educational backgrounds, work histories, and leadership roles. We further enrich the dataset with Crunchbase data, which offers additional professional and educational context for founders (subject to licensing restrictions).

**Findings** We evaluate GPT-4o-mini on a stratified test set of 1,000 unseen startups, constructed to reflect a realistic skewed distribution of outcomes (≈90% failures, 10% successes; i.e., a 1:10 success ratio). Parallel evaluations with GPT-4o and o3-mini are also under exploration. The results reveal a pronounced **over-prediction bias**: the model attains strong recall but low precision, consistently overestimating the likelihood of success. In effect, the LLM "believes" almost every startup will succeed, yielding high false positive rates.

Table 1: Performance of baseline LLM methods across different models and datasets.

| Model | Dataset | Records | Acc. | Prec. | F1 | Pred. Bias | Support (Succ./Fail.) |
|---|---|---|---|---|---|---|---|
| GPT-4o | 30% Success | 991 | 30.8% | 30.2% | 46.4% | 122.88 | 297 / 694 |
| GPT-4o | 20% Success | 993 | 21.1% | 20.2% | 33.6% | 75.38 | 198 / 794 |
| GPT-4o$^\dagger$ | 10% Success | 992 | 16.3% | 10.7% | 19.3% | 44.65 | 99 / 893 |
| o3-mini | 10% Success | 992 | 12.8% | 10.3% | 18.6% | 34.43 | 99 / 893 |
| GPT-4o-mini | 10% Success | 992 | 10.8% | 10.1% | 18.3% | 123.00 | 99 / 893 |

### 2.2 Startup Success Forecasting Framework

#### 2.2.1 Analyst Block

The Analyst Block of the SSFF serves as a core mechanism for evaluating startup ecosystems. It employs a divide-and-conquer strategy within a multi-agent framework, where startup data is decomposed into targeted dimensions and distributed across specialized agents. The VC-Scout Agent initiates this process by categorizing each startup along 18 dimensions—ranging from growth rate

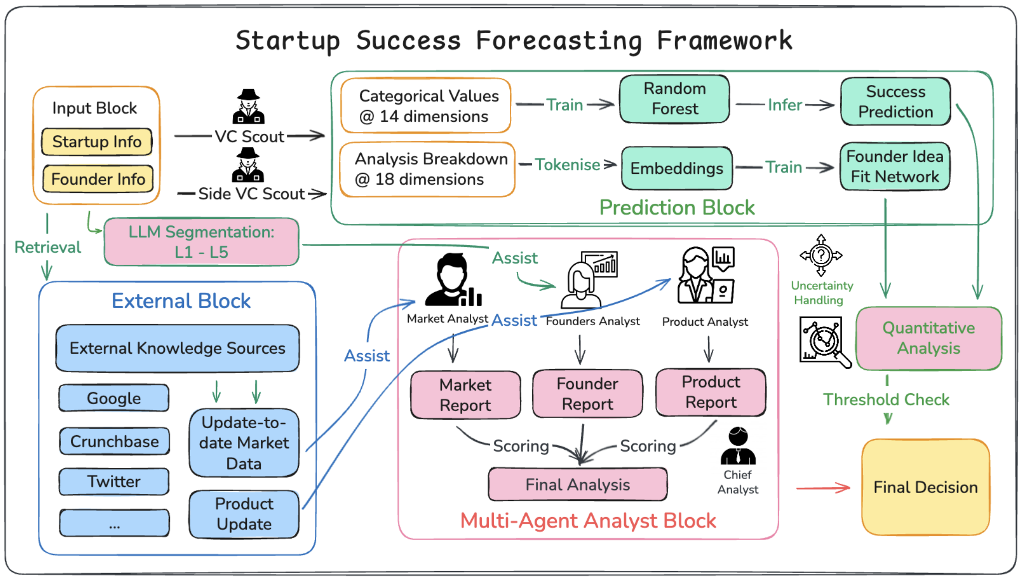


Figure 1: **Startup Success Forecasting Framework (SSFF).** The system integrates structured startup and founder inputs, external market intelligence, predictive models, and multi-agent analyst reports. Categorical and textual features feed a Random Forest and a Founder–Idea Fit Network, whose outputs are combined with analyst evaluations through quantitative checks and uncertainty handling to yield the final success decision.

and market size to patents and regulatory approvals. Each dimension is then assigned to a dedicated agent acting as a domain expert, enabling focused and independent analysis of its respective aspect.

The main contributors include the **Market Agent**, which examines strategic positioning and growth potential; the **Product Agent**, which evaluates innovation, scalability, and user adoption; and the **Founder Agent**, which reviews the team's expertise and long-term vision (shared more in the Appendix). Once individual assessments are complete, an Integration Agent synthesizes the results into a cohesive evaluation, simulating the collaborative effort of a venture capital team. The outcome is a well-rounded analysis that balances depth and breadth across all key factors (Figure 2 in the Appendix).

### 2.2.2 Prediction Block

While the Analyst Block focuses on descriptive evaluation, the Prediction Block complements it by transforming historical and extracted data into predictive signals of startup success. It consists of two components: (1) an **LLM-Enhanced Random Forest (LLM-RF)** and (2) a **Founder–Idea Fit Network (FIF-Network)**. Data preparation, label definitions, and technical details are provided in the Appendix.

**LLM-Enhanced Random Forest (LLM-RF)** We augment Random Forests with LLM-derived categorical encodings, using GPT-4o to extract startup features across 14 dimensions (e.g., industry growth, market size, development pace, product–market fit). This enhancement overcomes Random Forest's limitations with categorical data while retaining interpretability. The model achieves **77% accuracy** on a balanced dataset (vs. 68% with GPT-3.5 and fewer samples), and under a realistic skewed distribution (1:4 success ratio) it maintains **80% accuracy** with an **F1 score of 54.6**.

**Founder–Idea Fit Network (FIF-Network)** Founder–idea alignment is captured through the **Founder–Idea Fit Score (FIFS)**, defined in the Appendix, normalized to $[-1, 1]$ to quantify the compatibility between founders' expertise and their ventures' core ideas. Founders and startups are embedded with *text-embedding-3-large*, and cosine similarity is used as a proxy for alignment. While linear models showed weak explanatory power ($r = 0.173$, $R^2 = 0.03$), a lightweight neural network captured nonlinear patterns, reducing the mean squared error from **0.718 to 0.041** with stable validation loss.

Together, LLM-RF and FIF-Network provide complementary predictive insights—structured categorical analysis and founder–idea alignment—forming the core of the automated prediction block in SSFF.

### 2.3 SSFF Integration with External Knowledge Block

The External Knowledge Block enriches SSFF through a Retrieval-Augmented Generation (RAG) framework that integrates web-scraping APIs to gather and synthesize real-time market insights from news, reports, and trends data.

SSFF integrates the three core blocks: the Analyst Block processes founder/startup data across 18 dimensions; dual predictive paths via LLM-RF and FIF-Network; and the External Knowledge Block providing real-time market intelligence. A Chief Analyst agent consolidates these insights into comprehensive reports with quantitative scores and actionable recommendations ("Invest"/"Hold"), delivering enhanced accuracy and interpretability beyond standalone LLMs.

## 3 Evaluations & Results

### 3.1 Setup

Given the systematic over-prediction bias observed in baseline LLMs, we evaluate all methods on a realistic 10% success rate test set (1,000 startups, 100 successful) to simulate real-world VC decision difficulty. We benchmark against vanilla GPT-4o-mini, Chain-of-Thought prompting, R.A.I.S.E. (2024), and FounderGPT (2023), covering zero-shot, reasoning-enhanced, agentic, and founder-centric approaches, with SSFF-REG (Pro) integrating additional quantitative signals (check the Appendix).

### 3.2 Results

Table 2: Performance comparison on 1,000 unseen startups (10% success rate).

| Model | Acc | Prec ($P_1$) | Rec ($R_1$) | $F1_0$ | $F1_1$ | Wtd F1 |
|---|---|---|---|---|---|---|
| Vanilla GPT-4o-mini | 10.8 | 10.1 | **100.0** | 0.9 | 18.3 | 3.4 |
| CoT Prompt | 11.8 | 10.0 | **100.0** | 2.0 | 18.5 | 5.4 |
| R.A.I.S.E. (2024) | 11.9 | 10.1 | 99.0 | 2.2 | 18.4 | 5.7 |
| FounderGPT (2023) | 90.0 | 0.0 | 0.0 | **100.0** | 0.0 | 85.2 |
| SSFF-REG Basic | 28.2 | 10.1 | 79.4 | 22.5 | 17.9 | 34.3 |
| SSFF-REG (Pro) | **59.7** | **10.8** | 42.3 | **61.6** | **17.2** | **67.8** |

Baseline methods exhibit extreme prediction imbalance: vanilla LLMs and reasoning-enhanced approaches overwhelmingly predict success (99-100% recall, 10% precision), while FounderGPT collapses to predicting universal failure. In stark contrast, SSFF-REG Pro achieves 59.7% accuracy and 67.8% weighted-F1—nearly 6× improvement over vanilla GPT—while maintaining realistic prediction balance (42.3% recall). The quantitative integration in SSFF-REG Pro proves to be crucial: removing LLM-RF and FIF-Network components (Basic variant) drops performance to 28.2% accuracy, confirming that structured multi-agent reasoning with traditional ML significantly outperforms pure LLM approaches for startup evaluation under realistic class imbalance.

## 4 Conclusion and Future Work

This work demonstrates that conventional LLMs are prone to systematic over-prediction in startup assessments, yet structured multi-agent reasoning can substantially correct this distortion. The proposed SSFF framework—combining RF on LLM-derived features, founder–idea fit modeling, and external market signals—achieves state-of-the-art performance on a large, imbalanced test set, performing nearly 6× stronger than GPT baselines. These results demonstrate that integrating interpretability, quantitative structure, and real-time knowledge not only yields higher accuracy, but also provides more transparent and trustworthy decision support for venture capital. Future work will extend to larger datasets, additional LLM families, and human-in-the-loop evaluations.

## References


[1] Thomas Åstebro and Samir Elhedhli. The effectiveness of simple decision heuristics: Forecasting commercial success for early-stage ventures. *Management Science*, 52(3):395–409, March 2006. doi: 10.1287/mnsc.1050.0468. URL `https://doi.org/10.1287/mnsc.1050.0468`.

[2] Aswath Damodaran. Valuing young, start-up and growth companies: Estimation issues and valuation challenges. *SSRN Electronic Journal*, June 2009. URL `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1418687`.

[3] Sichao Xiong and Yigit Ihlamur. Founder-gpt: Self-play to evaluate the founder-idea fit. *arXiv preprint*, December 2023. URL `https://arxiv.org/abs/2312.12037`.

[4] Will Gornall and Ilya A. Strebulaev. Squaring venture capital valuations with reality. *SSRN Electronic Journal*, 2018. doi: 10.2139/ssrn.2955455. URL `https://doi.org/10.2139/ssrn.2955455`.

[5] Francesco Corea et al. Hacking the venture industry: An early-stage startups investment framework for data-driven investors. *Machine Learning with Applications*, 5:100062, September 2021. doi: 10.1016/j.mlwa.2021.100062. URL `https://doi.org/10.1016/j.mlwa.2021.100062`.

[6] Ekin Ozince and Yigit Ihlamur. Automating venture capital: Founder assessment using llm-powered segmentation, feature engineering and automated labeling techniques. *arXiv preprint*, July 2024. URL `https://arxiv.org/abs/2407.04885`.

[7] J. Preuveneers, J. Ternasky, F. Alican, and Y. Ihlamur. Reasoning-based ai for startup evaluation (r.a.i.s.e.): A memory-augmented, multi-step decision framework. *arXiv preprint*, 2025. URL `https://arxiv.org/abs/2504.12090`.

[8] Emily Gavrilenko et al. Improving startup success with text analysis. *arXiv preprint*, December 2023. URL `https://arxiv.org/abs/2312.06236`.

[9] Mark Potanin et al. Startup success prediction and vc portfolio simulation using crunchbase data. *arXiv preprint*, September 2023. URL `https://arxiv.org/abs/2309.15552`.

[10] Abdurahman Maarouf et al. A fused large language model for predicting startup success. *arXiv preprint*, September 2024. URL `https://arxiv.org/abs/2409.03668`.

[11] Alec Radford et al. Language models are unsupervised multitask learners, 2018. OpenAI Blog.

[12] OpenAI. Gpt-4 technical report. *arXiv preprint*, March 2023. URL `https://arxiv.org/abs/2303.08774`.

[13] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. Language models are few-shot learners. *arXiv preprint*, May 2020. URL `https://arxiv.org/abs/2005.14165`.

[14] Hugo Touvron et al. Llama: Open and efficient foundation language models. *arXiv preprint*, February 2023. URL `https://arxiv.org/abs/2302.13971`.

[15] Yutao Zhu et al. Large language models for information retrieval: A survey. *arXiv preprint*, 2024. URL `https://arxiv.org/abs/2308.07107`.

[16] Zane Durante et al. Agent ai: Surveying the horizons of multimodal interaction. *arXiv preprint*, January 2024. URL `https://arxiv.org/abs/2401.03568`.

[17] Murray Shanahan et al. Role-play with large language models. *arXiv preprint*, May 2023. URL `https://arxiv.org/abs/2305.16367`.

[18] Jason Wei et al. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint*, October 2022. URL `https://arxiv.org/abs/2201.11903`.

[19] Shunyu Yao et al. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint*, May 2023. URL `https://arxiv.org/abs/2305.10601`.

[20] Krishna Srinivasan et al. Quill: Query intent with large language models using retrieval augmentation and multi-stage distillation. *arXiv preprint*, October 2022. URL `https://arxiv.org/abs/2210.15718`.

# A Additional Related Work

## A.1 Startup Evaluation Pipeline

Evaluating a startup requires a deep understanding of the startup's market and technology. Startups typically have no history, minimal revenue, and low survival rates [2]. The high-risk, high-return nature of venture capital leads to few investors achieving impressive results [4, 5]. Traditionally, this process has relied on venture capitalists' intuition, which is inefficient and biased [1, 5]. Recent trends favor data-driven approaches, using advanced analytics and ML to quantify success factors [5].

In recent years, there has been more research around the venture capital and startup ecosystem to improve the performance of startup investing. Xiong (2023) introduced the Founder GPT framework, which uses LLMs to evaluate the "founder-idea" fit, showing that personalized evaluation of a founder's idea alongside her background is crucial to predict startup success [3]. In addition, Ozince (2024) showed that LLMs can be effective in segmenting and labeling founders to improve the quality of ML methods to predict the success of startups [6]. Preuveneers et al. (2025) leveraged chain-of-thought reasoning and structured distillation to enable more stable, interpretable decision-making in startup evaluation [7]. Meanwhile, Gavrilenko (2023) and Maarouf (2024) demonstrated how free-form text description of startups can be predictive of future success, while Potanin (2023) provided a high-performance model that can deliver 14× return on capital [8, 9, 10].

## A.2 LLM Agent

The emergence of LLMs has revolutionized natural language understanding, showing transformative potential across various research fields [11, 12, 13, 14, 15]. As foundational platforms, LLMs have expanded the concept of AI agents, defined as systems that perceive and respond to environmental data, producing meaningful actions [16]. Role-Play support systems, for example, enable AI agents to become increasingly human-like in diverse scenarios [17]. These agents can autonomously analyze data and support decision-making, identifying complex patterns beyond human capabilities. Despite these advancements, a widely adopted AI agent specifically for startup analysis has yet to emerge.

## A.3 Prompting Techniques

The effectiveness of LLMs in various applications depends heavily on prompt engineering, which involves crafting queries to guide the model for specific outputs. Well-designed prompts enhance LLM performance, making this skill crucial for startup evaluation. Techniques like Chain-of-Thought, Tree-of-Thought, Few-shot Learning, and Retrieval Augmented Generation improve AI's accuracy and data retrieval [18, 19, 20, 15]. This paper employs these techniques in a divide-and-conquer framework to improve AI's utility and reliability in problem-solving, strategic planning, information retrieval, and decision-making.

# B Baseline Experiment Methodology

## B.1 Extracted Baseline Prompts

### B.1.1 Vanilla LLM Baseline

```
You are an experienced venture capitalist analyzing a startup. Based on
the provided information, give a comprehensive analysis and give your
investment recommendation. Looking at the potential, will you invest?

Your analysis should include:
```

```
1. Market analysis
2. Product/technology evaluation
3. Founder/team assessment
4. Overall score (1-10)
5. Investment recommendation (must be 'Successful' or 'Unsuccessful')

Criteria for future success: Startups that raised more than $500M,
acquired for more than $500M or had an initial public offering
with a valuation exceeding over $500M valuation are defined as success.
Startups that raised between $100K and $4M but did not achieve
significant success afterwards are considered as failed.
```

#### B.1.2 Chain-of-Thought (CoT) Baseline

```
You are an experienced venture capitalist analyzing a startup. Based on
the provided information, please think step-by-step to give a
comprehensive analysis and predict if the startup will be successful or
not.

Your step-by-step thinking process should lead to the following final
outputs:
1. Market analysis (Consider TAM, SAM, SOM, competition, market trends,
and barriers to entry)
2. Product/technology evaluation (Consider innovation, scalability,
   defensibility, and product-market fit)
3. Founder/team assessment (Consider experience, execution ability, team
   completeness, and advisor quality)
4. Overall score (1-10, derived from your step-by-step analysis)
5. Investment recommendation (must be 'Successful' or 'Unsuccessful',
based on your overall analysis and score)
```

#### B.1.3 FounderGPT Baseline

```
Simulate THREE seasoned venture-capital analysts collaborating. They will
brainstorm step-by-step, critique each other, back-track on flaws, and
converge on the most logical assessment.

STEP 1 (Brainstorm Features): Brainstorm 4-6 bullet "Successful
Founder/Idea Features" by comparing the input to reference cases.
STEP 2 (Rate Features): For EACH feature identified, each expert rates
"likelihood of success contribution" (0-1) and debates until consensus.
STEP 3 (Determine Scores): Based on discussion, determine
founder_score_eta_f and startup_score_eta_s (both 0-1).
```

#### B.1.4 R.A.I.S.E. Baseline

```
Apply the given decision policy to predict success:
- IF founder_has_relevant_deep_domain_expertise THEN
likelihood_of_success = HIGH.
- IF founder_has_prior_successful_exit_in_same_sector THEN
  likelihood_of_success = HIGH.
- IF product_addresses_clear_underserved_market_need THEN
  likelihood_of_success = HIGH.
- IF founder_lacks_technical_skills_for_tech_product AND
no_technical_cofounder THEN likelihood_of_success = LOW.
- IF market_is_highly_saturated_with_strong_incumbents AND
  product_is_not_differentiated THEN likelihood_of_success = LOW.
```

### B.2 Baseline Experimental Procedure

The baseline methodology employs direct LLM prompting without additional structural components. Each experiment processes startup descriptions through a single-agent framework where the LLM acts as a simulated venture capitalist. The procedure begins by loading curated datasets with varying success rates (10%-50%) and processes each startup record individually using the specified model (GPT-4o, GPT-4o-mini, or o3-mini). For each startup, the system extracts the integrated startup description, applies the method-specific prompt (vanilla, CoT, FounderGPT collaborative simulation, or R.A.I.S.E. rule-based reasoning), and requests a structured JSON response containing analysis text, numerical scores, and binary investment recommendations ("Successful"/"Unsuccessful"). All methods use identical success criteria ($500M+ funding/acquisition/IPO threshold) and process records sequentially with 1-second delays to prevent API rate limiting. Results are saved incrementally to JSON files with automatic resume functionality, enabling robust experimentation across thousands of startup evaluations. This standardized pipeline ensures fair comparison by isolating prompt design as the primary differentiating factor between baseline approaches.

**Key procedural elements:**

- **Standardized input**: Same startup descriptions and success criteria across all methods
- **Controlled environment**: Identical API parameters, rate limiting, and error handling
- **Structured output**: Consistent JSON schema with scores and binary predictions
- **Incremental saving**: Robust data collection with resume capability
- **Reproducible evaluation**: Timestamped outputs and systematic file organization

## C SSFF Framework Variants: Basic vs. Pro

### C.1 SSFF-REG Basic Configuration

SSFF-REG Basic represents a simplified multi-agent approach that integrates LLM-based analysis across three specialized domains without quantitative enhancement components. The Basic variant processes startup information through the Market Agent, Product Agent, and Founder Agent to generate structured assessments of market viability, product potential, and founder competency respectively. These individual analyses are then consolidated by the Integration Agent using only qualitative reasoning, following the `integrated_analysis_basic` pathway that synthesizes market, product, and founder insights into an overall investment recommendation. The Basic configuration omits quantitative signals such as founder segmentation levels (L1-L5), founder-idea fit scores, and random forest predictions, relying instead on pure LLM reasoning to evaluate startup potential. This approach demonstrates the performance of structured multi-agent analysis without traditional ML augmentation.

### C.2 SSFF-REG Pro Configuration

SSFF-REG Pro extends the Basic framework by incorporating quantitative decision-support components that significantly enhance predictive accuracy. In addition to the three-agent qualitative analysis pipeline, the Pro variant integrates: (1) **Founder Segmentation** that classifies founders into competency levels (L1-L5) based on experience, track record, and domain expertise, where L5 founders show 3.79× higher success likelihood than L1 counterparts; (2) **Founder-Idea Fit Networks** that compute alignment scores between founder backgrounds and startup concepts using cosine similarity on embedded representations; and (3) **LLM-Enhanced Random Forest** predictions trained on categorical features extracted from the multi-agent analyses. The Integration Agent processes these quantitative signals through the `integrated_analysis_pro` pathway, which explicitly incorporates founder segmentation levels, idea-fit scores (-1 to +1 range), and RF predictions (~65% accuracy) into the final investment decision. This hybrid approach tempers LLM optimism bias with statistical evidence, resulting in more balanced and reliable predictions under realistic class imbalance conditions.

### C.3 Performance Impact of Configuration Differences

The quantitative enhancement in SSFF-REG Pro produces substantial performance improvements: accuracy increases from 28.2% (Basic) to 59.7% (Pro), while maintaining more realistic prediction ratios compared to baseline LLM approaches that suffer from extreme over-prediction bias. This demonstrates that structured integration of traditional ML components with multi-agent LLM reasoning creates more reliable decision-support systems for high-stakes financial applications.

## D Founder Segmentation Details

### D.1 Data Preparation

A comprehensive dataset of over 2,000 entries was curated from verified LinkedIn profiles, capturing only the historical data available up to each startup's founding or VC involvement to ensure an authentic reflection of founders' early backgrounds. Detailed records include educational credentials, work experiences, and key roles, and are further enriched with startup composition data that tracks changes across life stages—providing essential context on team dynamics and early-stage structures. Startups are classified as "successful" or "unsuccessful" based on market valuations (with those exceeding $500M deemed successful) and augmented with supplementary insights from Crunchbase, a paid data source offering deeper information on founders' professional and educational histories (subject to licensing restrictions). For our segmentation analysis, 1,000 entries from each label were randomly sampled, forming a robust foundation for evaluating founder segmentation.

We initially curated a dataset comprising founders' LinkedIn profiles, associated with startups classified as either successful or unsuccessful. This classification was based on the companies' market valuations, with successful ones having valuations over $500M. The dataset was enriched with detailed profiles, including education and work backgrounds, extracted in JSON format from LinkedIn URLs. This preparation phase was crucial for ensuring a robust foundation for our segmentation analysis. Both labels contained more than 2,000 entries of founders and their respective companies. After processing, 1,000 entries of founders, sampled randomly from each label, were used for segmentation.

### D.2 Segmentation Process

Founders were segmented into five levels—from Level 1 (least qualified) to Level 5 (most qualified)—using GPT-4 to evaluate their LinkedIn profiles. The segmentation criteria, informed by years of VC expertise, focus on key indicators such as leadership roles, business achievements, and educational background. Tailored prompts were iteratively refined to ensure accurate and nuanced categorization. Additional details—including the prompt, criteria, and sample segmentation results—are provided throughout the Appendix.

### D.3 Segmentation Results

The segmentation results demonstrate a strong correlation between founder levels and startup success rates. Level 5 (L5) founders, characterized by their proven track record in building significant businesses or holding executive roles at leading technology companies, were substantially more likely to lead their startups to success. Specifically, Level 2 (L2) founders were 1.12× more likely to succeed than Level 1 (L1) founders, while L5 founders were an impressive 3.79× more likely to succeed than their L1 counterparts.

These findings highlight the significant impact of a founder's background on startup success and suggest that incorporating this nuanced segmentation into evaluation frameworks can enhance predictive accuracy. Moreover, exploring even more granular segmentation may further refine these predictions.

Table 3: Success and failure rates by founder level, showcasing the predictive power of founder segmentation on startup success.

| Level | Success | Failure | Success Rate |
|---|---|---|---|
| L1 | 24 | 75 | 24.24% |
| L2 | 83 | 223 | 27.12% |
| L3 | 287 | 445 | 39.21% |
| L4 | 514 | 249 | 67.37% |
| L5 | 93 | 8 | 92.08% |

### D.4 Founders Segmentation Supplements

#### D.4.1 Segmentation Criteria

The segmentation of founders into levels L1 through L5 is achieved through a supervised, hybrid approach that leverages both LLMs and manual expert review, drawing on years of venture capital expertise.

- **Level 1 (L1):** Founders with negligible experience—typically recent graduates, dropouts, or individuals largely disconnected from tech circles.
- **Level 2 (L2):** Founders with limited experience (usually fewer than 10 years) who have worked at reputable companies (e.g., Google or McKinsey) or are accelerator graduates.
- **Level 3 (L3):** First–time entrepreneurs with 10—15 years of technical and management experience, often including individuals with advanced degrees (e.g., PhDs) or backgrounds from top-tier institutions.
- **Level 4 (L4):** Repeat entrepreneurs who have successfully exited previous ventures via small to medium-sized exits, or professionals with high-level executive experience at notable technology companies.
- **Level 5 (L5):** Elite founders who have built companies achieving substantial milestones—such as $100M+ ARR, or IPOs or acquisitions exceeding $500M valuation—considered the most successful and influential in the ecosystem.

#### D.4.2 Founders Segmentation Prompt

This section contains the complete prompt template used to categorize founders into segmentation levels (L1—L5). The prompt instructs the LLM to evaluate a founder's background (including education, work experience, leadership roles, and achievements) and assign a segmentation level accordingly.

**Example Prompt:**

```
You are an analyst. Your task is to output one of the options: [L1, L2,
    L3, L4, L5]. Do not output anything else.

Think step by step and consider the following criteria:
- L5: Entrepreneur who has built a $100M+ ARR business, taken a company
    public, or achieved an exit exceeding $500M valuation.
- L4: Entrepreneur with a small to medium-size exit or who has held a
    high-level executive role at a notable technology company.
- L3: First time entrepreneur with 10 to 15 years of technical and
    management experience, often holding advanced degrees or coming from
    top-tier institutions.
- L2: Entrepreneur with a few years of experience or an accelerator
    graduate.
- L1: Entrepreneur with negligible experience (e.g., recent graduate or
    dropout) but with potential.

Based on the founder's LinkedIn profile information below, determine the
    appropriate segmentation level:
{founder_info}
```

Additional prompt versions and iterative refinements are also documented in our version-controlled repository.

# E Prediction Block Details

A prediction block is designed to learn from past data and predict the likelihood of success as numerical metrics to support the reliability of an automated startup evaluation pipeline. The prediction block is separated into two parts: (1) LLM-Based Random Forest Model and (2) Founder-Idea Fit Network.

## E.1 LLM-Based Random Forest Model Design

The conventional Random Forest algorithm, celebrated for its effectiveness and explainability, often faces challenges with categorical variables due to its inherent design constraints. To overcome these limitations, we introduce an LLM-based Random Forest model. This novel approach utilizes LLMs, particularly GPT-4o, for the extraction of features, providing the model with the flexibility to handle a broad spectrum of categorical variables.

In our framework, startup and founder information is processed through an LLM to categorize data across 14 dimensions, including industry growth, market size, development pace, and product-market fit, among others. This method allows for a nuanced understanding of startup dynamics, which is critical for accurate prediction. The sample data analyzed by the LLM is structured as follows:

```
{
  "startup_analysis_responses": {
    "industry_growth": "Yes",
    "market_size": "Large",
    "development_pace": "Faster",
    ...
    "timing": "Just Right"
  }
}
```

The mathematical formulation of the algorithm is summarized as a pseudo-algorithm below.

This approach to leveraging LLMs for feature extraction and encoding provides a flexible and robust framework for startup success prediction, making full use of categorical variables without the constraints of traditional models.

---

**Algorithm 1** LLM-enhanced Random Forest Model

---

**Require:** Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$
**Ensure:** Trained Random Forest model $\mathcal{F}_{\text{RF}}$

1: Define features $\{f_1, f_2, \ldots, f_{14}\}$ with corresponding outcomes
2: Use LLM to categorize data $A$ into $\{c_1, c_2, \ldots, c_N\}$.
3: Encode categorical features into numerical values.
4: Split dataset $\mathcal{D}$ into training set $\mathcal{D}_{\text{train}}$ and testing set $\mathcal{D}_{\text{test}}$.

$$\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N_{\text{train}}}, \quad \mathcal{D}_{\text{test}} = \{(x_i, y_i)\}_{i=N_{\text{train}}+1}^{N}$$

where $N_{\text{train}} + N_{\text{test}} = N$.

5: Train Random Forest model $\mathcal{F}_{\text{RF}}$ on $\mathbf{X}_{\text{train}}$ and $\mathbf{Y}_{\text{train}}$.

$$\Theta = \arg\min_{\Theta} \mathcal{L}(\mathcal{F}_{\text{RF}}(\mathbf{X}_{\text{train}}; \Theta), \mathbf{Y}_{\text{train}})$$

6: Predict $\mathbf{Y}_{\text{test}}$ using trained model $\mathcal{F}_{\text{RF}}$.

$$\hat{\mathbf{Y}}_{\text{test}} = \mathcal{F}_{\text{RF}}(\mathbf{X}_{\text{test}}; \Theta)$$

7: Calculate accuracy of predictions.

$$\text{Accuracy} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \nVdash(\hat{y}_i = y_i)$$

where $\nVdash(\cdot)$ is the indicator function.

---

**Note:** The full prompt and methodology, including question design and LLM interactions, are shared throughout the paper.

**LLM-Based Categorical Data Extraction** A cornerstone of our LLM-based Random Forest model is the extraction of categorical data from startup and founder information. This process is guided by a CoT prompting technique, where the LLM is presented with a series of questions designed to elicit specific insights into various aspects of a startup's potential for success. These questions cover a wide range of topics for comprehensiveness.

Some illustrative questions used in this process are as follows:

1. "Is the startup operating in an industry experiencing growth? [Yes/No/N/A]"
2. "Is the target market size for the startup's product/service considered large? [Small/Medium/Large/N/A]"
3. "Does the startup demonstrate a fast pace of development compared to competitors? [Slower/Same/Faster/N/A]"

The corresponding encoding is presented in the table below.

Table 4: Adjusted Category Mappings with "Mismatch" Included

| **Category** | **Mappings** |
|---|---|
| Industry Growth | No, N/A, Yes, Mismatch |
| Market Size | Small, Medium, Large, N/A, Mismatch |
| Development Pace | Slower, Same, Faster, N/A, Mismatch |
| ...... | |

This structured approach to querying provides a rich dataset from which we can extract categorical variables with high relevance to startup success. The responses to these questions are then encoded into numerical values, forming the basis for training our LLM-based Random Forest model.

**Model Performance Evaluation**

Our LLM-based Random Forest model was trained on a random & balanced subset of 1,400 instances—comprising equal numbers of successful and unsuccessful startups—drawn from our curated dataset. For feature extraction and model guidance, we employed GPT-4o. Table 5 summarizes the classification performance of our model on the test set.

Table 5: Classification report for the model.

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.79 | 0.72 | 0.75 | 137 |
| 1 | 0.75 | 0.81 | 0.78 | 142 |
| Accuracy | | | 0.77 | 279 |
| Macro avg | 0.77 | 0.77 | 0.77 | 279 |
| Weighted avg | 0.77 | 0.77 | 0.77 | 279 |

Notably, training with only 200 data points using GPT-3.5 yielded an average accuracy of 68%, whereas scaling up to 1,400 data points with GPT-4o resulted in a performance improvement of approximately 10 percentage points. Moreover, when evaluated under a realistic, skewed distribution with a 1:4 ratio of successful to unsuccessful startups, the model achieved an accuracy of 80.00% and an F1 score of 54.55%. These results highlight not only the robustness of our approach but also the critical impact of training data size and balance on predictive accuracy, underscoring the promise of integrating advanced AI techniques with traditional ML models for startup evaluation.

### E.2 Founder-Idea Fit Network

SSFF incorporates a novel network to assess founder-idea fit, which is a crucial aspect of startup success prediction. The Founder-Idea Fit Model is designed to quantitatively assess the alignment between founders' expertise and characteristics and their startup's core idea and market positioning.

**Measuring Founder-Idea Fit** Our analysis highlights a strong correlation between founder segmentation levels and startup outcomes. For instance, Level 5 founders are over 3× as likely to succeed as Level 1 founders. However, exceptions exist, with some Level 5 founders failing and some Level 1 founders succeeding. To capture these nuances, we introduce the Founder-Idea Fit Score (FIFS), a metric that quantifies the compatibility between a founder's experience and the viability of their startup idea.

We define the preliminary FIFS as:

$$FIFS(F, O) = (6 - F) \times O - F \times (1 - O)$$

where F represents the founder's level (ranging from 1 to 5) and O denotes the outcome (1 for success, 0 for failure). To facilitate comparison across startups, the FIFS is normalized to a range of [-1, 1] using the formula:

$$\text{Normalized FIFS} = \frac{FIFS}{5}$$

On this scale, a score of 1 indicates the optimal founder-idea fit—achieved when a Level 1 founder succeeds—while a score of -1 reflects the poorest fit, as observed when a Level 5 founder fails. This normalized metric enables straightforward comparisons, highlighting startups with the strongest or weakest alignment between founder capabilities and their business concepts.

**Preprocessing: Embedding & Cosine Similarity** The first step in the Founder-Idea Fit Model is to generate dense vector representations for both startup descriptions and founder backgrounds. We employ OpenAI's *text-embedding-3-large* model to transform textual data into 100-dimensional embeddings, capturing the semantic essence of each description. These embeddings encapsulate details such as a startup's mission, technology, and market, as well as a founder's education and employment history. We then compute the cosine similarity between each founder's embedding and the corresponding startup's embedding, using this metric as a proxy for the semantic alignment—or "fit"—between the founder's background and the startup's concept.

**Statistical Analysis and Further Model Considerations** Our statistical analysis revealed only a modest linear association between cosine similarity and the Founder-Idea Fit Score (FIFS), with a Pearson correlation coefficient of 0.173 and an R-squared value of 0.030 from an Ordinary Least Squares (OLS) regression. These findings indicate that cosine similarity accounts for merely 3% of the variability in FIFS. Although statistically significant, the low predictive power, positive autocorrelation, and deviations from normality suggest that a linear model is inadequate. As a result, we developed a neural network to better capture the nonlinear relationships among the features, thereby improving predictive accuracy.

**Model Architecture and Performance Analysis** Our neural network model uses the embeddings and their cosine similarity as input features to predict the Founder-Idea Fit Score. The model features a straightforward architecture with an input dense layer of 128 neurons followed by a second dense layer of 64 neurons, both activated by ReLU and regularized with dropout rates of 20% and 30%, respectively. Trained on a dataset of founder-startup pairs, the network achieved a dramatic reduction in mean squared error—from 0.7182 during initial epochs to 0.0407 at convergence—and a validation loss of 0.0386. These results underscore the robustness and scalability for early-stage founder-idea fit evaluation.

### E.3 Prediction Block Prompts

#### E.3.1 VC Scout Agent

```
As an analyst specializing in startup evaluation, categorize the given
    startup based on the following criteria.
Provide a categorical response for each of the following questions based
    on the startup information provided.
Use ONLY the specified categorical responses for each field. Do not use
    any other responses.

1. Industry Growth: [Yes/No/N/A]
2. Market Size: [Small/Medium/Large/N/A]
3. Development Pace: [Slower/Same/Faster/N/A]
```

```
4. Market Adaptability: [Not Adaptable/Somewhat Adaptable/Very Adaptable/
    N/A]
5. Execution Capabilities: [Poor/Average/Excellent/N/A]
6. Funding Amount: [Below Average/Average/Above Average/N/A]
7. Valuation Change: [Decreased/Remained Stable/Increased/N/A]
8. Investor Backing: [Unknown/Recognized/Highly Regarded/N/A]
9. Reviews and Testimonials: [Negative/Mixed/Positive/N/A]
10. Product-Market Fit: [Weak/Moderate/Strong/N/A]
11. Sentiment Analysis: [Negative/Neutral/Positive/N/A]
12. Innovation Mentions: [Rarely/Sometimes/Often/N/A]
13. Cutting-Edge Technology: [No/Mentioned/Emphasized/N/A]
14. Timing: [Too Early/Just Right/Too Late/N/A]

Provide your analysis in a JSON format that matches the
    StartupCategorization schema.
If you cannot determine a category based on the given information, use 'N
    /A'.
Do not include any explanations or additional text outside of the JSON
    structure.

Startup Information:{startup_info}
```

#### E.3.2 LLM-based RF Model

Mapping is attached here.

```
"industry_growth": ["No", "N/A", "Yes"],
"market_size": ["Small", "Medium", "Large", "N/A"],
"development_pace": ["Slower", "Same", "Faster", "N/A"],
"market_adaptability": ["Not Adaptable", "Somewhat Adaptable", "Very
    Adaptable", "N/A"],
"execution_capabilities": ["Poor", "Average", "Excellent", "N/A"],
"funding_amount": ["Below Average", "Average", "Above Average", "N/A"],
"valuation_change": ["Decreased", "Remained Stable", "Increased", "N/A"],
"investor_backing": ["Unknown", "Recognized", "Highly Regarded", "N/A"],
"reviews_testimonials": ["Negative", "Mixed", "Positive", "N/A"],
"product_market_fit": ["Weak", "Moderate", "Strong", "N/A"],
"sentiment_analysis": ["Negative", "Neutral", "Positive", "N/A"],
"innovation_mentions": ["Rarely", "Sometimes", "Often", "N/A"],
"cutting_edge_technology": ["No", "Mentioned", "Emphasized", "N/A"],
"timing": ["Too Early", "Just Right", "Too Late", "N/A"]
```

| | | Predicted Class | |
|---|---|---|---|
| | | 0 | 1 |
| Actual Class | 0 | 99 | 38 |
| | 1 | 27 | 115 |

Table 6: Confusion matrix for LLM-based random forest model.

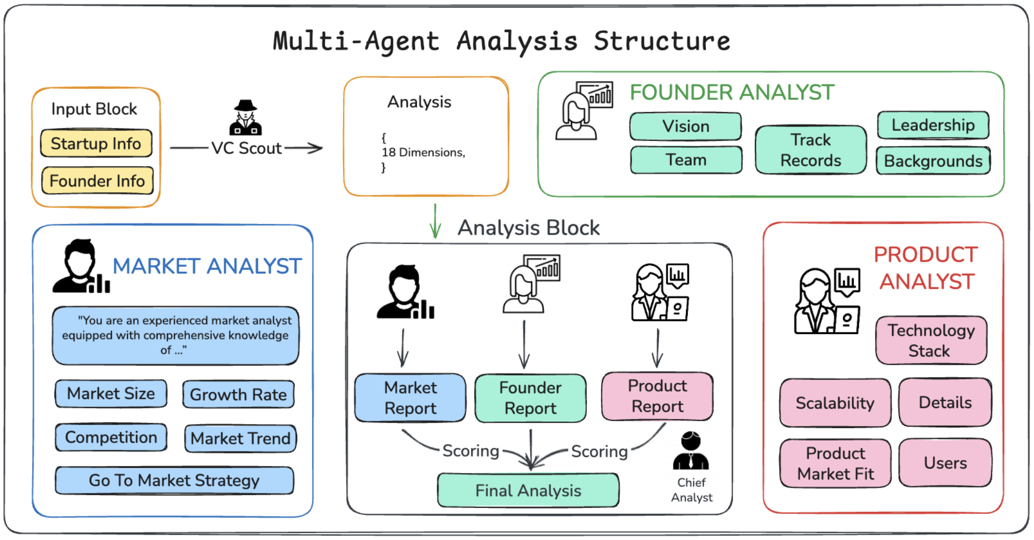


Figure 2: **Agent Analysis Structure.** The SSFF framework organizes startup evaluation into a structured multi-agent pipeline. The *VC Scout* gathers startup and founder information as input, which is analyzed across 18 dimensions by three specialized agents: the **Market Analyst** (market size, growth rate, competition, go-to-market strategy), the **Product Analyst** (scalability, technology stack, product–market fit, user feedback), and the **Founder Analyst** (leadership, vision, team, and track record). Their formulated and scored analyses are synthesized by the **Chief Analyst** to produce the final assessment output.

# F Analyst Block Details

As illustrated in Figure 2, the SSFF agent workflow begins with the *VC Scout*, which collects structured information about both the startup and its founders. This information is passed into the *Analysis Block*, where three domain-specialized agents conduct parallel evaluations: the Market Analyst examines macroeconomic indicators and competitive dynamics; the Product Analyst inspects scalability, technology, and product–market fit; and the Founder Analyst assesses leadership, experience, and team composition. Each analyst formulates sub-scores and intermediate insights, which are then aggregated and normalized by the *Chief Analyst* to yield a unified, interpretable assessment. This modular architecture ensures coverage across 18 analytical dimensions while maintaining interpretability and traceability in decision reasoning.

## F.1 Prompts

### Product Agent

```
You are a professional product analyst in a VC firm evaluating a
    potential investment opportunity.

Company Information:
{startup_info}

Product Information:
{product_info}

Product Research Report:
{external_knowledge}

Based on this comprehensive product research and the initial data, please
     provide:
1.Technical Innovation Analysis:
-How innovative is the technology?
```

```
-Is it feasible to implement?
-What are the technical risks?

2.Feature Set Evaluation:
-How complete is the product's feature set?
-How does it compare to competitors?
-What are the key differentiators?

3.Implementation Assessment:
-What are the main technical challenges?
-How realistic is the development timeline?
-What resources are required?

4.Market Readiness:
-Is the product ready for its target market?
-What further development is needed?
-How strong is the product-market fit?

Please reference specific data points from the product research report in
     your analysis, and conclude with:
-Product potential score (1-10)
-Innovation score (1-10)
-Market fit score (1-10)
```

**Market Agent**

```
You are a professional agent in a VC firm to analyze a company. Your task
     is to analyze the company here. Context: {startup_info}

Your focus is on the market side. What is the market? Is the market big
    enough? Is now the good timing? Will there be a good product-market-
    fit?

Specifically here are some relevant market information: {market_info}.

Your intern has researched more around the following topic for you as
    context {keywords}.

The research result: {external_knowledge}

Provide a comprehensive analysis including market size, growth rate,
    competition, and key trends. Analyze step by step to formulate your
    comprehensive analysis to answer the questions proposed above.

Also conclude with a market viability score from 1 to 10.
```

**Founders Agent**

```
As a highly qualified analyst specializing in startup founder assessment,
     evaluate the founding team based on the provided information.
         Consider the founders' educational background, industry
             experience, leadership capabilities, and their ability to
             align and execute on the company's vision.
         Provide a competency score, key strengths, and potential
             challenges. Please write in great detail.
```

A representative input snippet from the *VC Scout* analysis (founder subset) is shown below. This structured JSON-like format encapsulates key founder attributes—including educational background, prior achievements, leadership experience, and vision alignment—which are subsequently parsed and analyzed by the corresponding specialized agents.

```
startup_info = {
    "founder_backgrounds": "MBA from Stanford; 5 years at Google as
        Product Manager",
```

```
    "track_records": "Launched two successful Google products, one
        surpassing 1M users",
    "leadership_skills": "Managed a cross-functional team of 10 engineers
         and designers",
    "vision_alignment": "Demonstrates strong passion for applying AI to
        healthcare",
    "description": "AI-powered health monitoring wearable device"
}
```

### F.2 Integration Agent

We have two prompts attached here, one for integration decision and another for quantitative decision (to separate out reasoning from sub-reports).

**Integration Analysis Prompt**

```
Imagine you are the chief analyst at a venture capital firm, tasked with
    integrating the analyses of multiple specialized teams to provide a
    comprehensive investment insight. Your output should be structured
    with detailed scores and justifications:

As the chief analyst, you should stay critical of the company and listen
    carefully to what your colleagues say. You are also assisted by
    statistical models trained by your firm. You should not be over
    confident (or over-critical) for a firm and should rely on your
    strength of reasoning.
Many startups present themselves with good words but the truth is that
    few will be successful. It is your task to find those that have the
    potential to be successful and give your recommendations.

Example 1:
Market Viability: 8.23/10 - The market is on the cusp of a regulatory
    shift that could open up new demand channels, supported by consumer
    trends favoring sustainability. Despite the overall growth,
    regulatory uncertainty poses a potential risk.
Product Viability: 7.36/10 - The product introduces an innovative use of
    AI in renewable energy management, which is patent-pending. However,
    it faces competition from established players with deeper market
    penetration and brand recognition.
Founder Competency: 9.1/10 - The founding team comprises industry
    veterans with prior successful exits and a strong network in the
    energy sector. Their track record includes scaling similar startups
    and navigating complex regulatory landscapes.

Recommendation: Invest. The team's deep industry expertise and innovative
     product position it well to capitalize on the market's regulatory
    changes. Although competition is stiff, the founders' experience and
    network provide a competitive edge crucial for market adoption and
    navigating potential regulatory hurdles.

Example 2:
Market Viability: 5.31/10 - The market for wearable tech is saturated,
    with slow growth projections. However, there exists a niche but
    growing interest in wearables for pet health.
Product Viability: 6.5/10 - The startup's product offers real-time health
     monitoring for pets, a feature not widely available in the current
    market. Yet, the product faces challenges with high production costs
    and consumer skepticism about the necessity of such a device.
Founder Competency: 6.39/10 - The founding team includes passionate pet
    lovers with backgrounds in veterinary science and tech development.
    While they possess the technical skills and passion for the project,
    their lack of business and scaling experience is a concern.
```

```
Recommendation: Hold. The unique product offering taps into an emerging
    market niche, presenting a potential opportunity. However, the
    combination of a saturated broader market, challenges in justifying
    the product's value to consumers, and the team's limited experience
    in business management suggests waiting for clearer signs of product-
    market fit and strategic direction.

Now, analyze the following:

Market Viability: {market_info}
Product Viability: {product_info}
Founder Competency: {founder_info}
Founder-Idea Fit: {founder_idea_fit}
Founder Segmentation: {founder_segmentation}
Random Forest Prediction: {rf_prediction}

Some context here for the scores:
1. Founder-Idea-Fit ranges from -1 to 1, a stronger number signifies a
    better fit.
2. Founder Segmentation outcomes range from L1 to L5, with L5 being the
    most "competent" founders, and L1 otherwise.
3. Random Forest Prediction predicts the expected outcome purely based on
     a statistical model, with an accuracy of around 65%.

Provide an overall investment recommendation based on these inputs. State
     whether you would advise 'Invest' or 'Hold', including a
    comprehensive rationale for your decision. Consider all provided
    predictions and analyses, but do not over-rely on any single
    prediction.
```

**QuantDecision Prompt**

```
You are a final decision-maker. Think step by step. You need to consider
    all the quant metrics and make a decision.

You are now given Founder Segmentation. With L5 very likely to succeed
    and L1 least likely. You are also given the Founder-Idea Fit Score,
    with 1 being most fit and -1 being least fit. You are also given the
    result of prediction model (which should not be your main evidence
    because it may not be very accurate).

This table summarises the implications of the Level Segmentation:

Founder Level & Success & Failure & Success Rate & X-Time Better than L1
    \\
\midrule
L1 & 24 & 75 & 24.24\% & 1 \\
L2 & 83 & 223 & 27.12\% & 1.12 \\
L3 & 287 & 445 & 39.21\% & 1.62 \\
L4 & 514 & 249 & 67.37\% & 2.78 \\
L5 & 93 & 8 & 92.08\% & 3.79 \\

Regarding the Founder-Idea-Fit Score. Relevant context are provided here:
The previous sections show the strong correlation between founder's
    segmentation level and startup's outcome, as L5 founders are more
    than three times likely to succeed than L1 founders. However, looking
     into the data, one could also see that there are L5 founders who did
     not succeed, and there are L1 founders who succeeded. To account for
     these scenarios, we investigate the fit between founders and their
    ideas.

To assess quantitatively, we propose a metric called Founder-Idea Fit
    Score (FIFS). The Founder-Idea Fit Score quantitatively assesses the
    compatibility between a founder's experience level and the success of
```

```
     their startup idea. Given the revised Preliminary Fit Score ($PFS$)
    defined as:
\[PFS(F, O) = (6 - F) \times O - F \times (1 - O)\]
where $F$ represents the founder’s level ($1$ to $5$) and $O$ is the
    outcome ($1$ for success, $0$ for failure), we aim to normalize this
    score to a range of $[-1, 1]$ to facilitate interpretation.

To achieve this, we note that the minimum $PFS$ value is $-5$ (for a
    level $5$ founder who fails), and the maximum value is $5$ (for a
    level $1$ founder who succeeds). The normalization formula to scale
    $PFS$ to $[-1, 1]$ is:
\[Normalized\;PFS = \frac{PFS}{5}\]

Now use all of these information, produce a string of the predicted
    outcome and probability, with one line of reasoning.

Your response should be in the following format:
{
  "outcome": "<Successful or Unsuccessful>",
  "probability": <probability as a float between 0 and 1>,
  "reasoning": "<One-line reasoning for the decision>"
}

You will also receive a categorical prediction outcome of the prediction
    model (which should not be your main evidence because it may not be
    very accurate, just around 65% accuracy).

Ensure that your response is a valid JSON object and includes all the
    fields mentioned above.

f"You are provided with the categorical prediction outcome of {
    rf_prediction}, Founder Segmentation of {Founder_Segmentation},
    Founder-Idea Fit of {Founder_Idea_Fit}."
```

## G External Knowledge Block Details

The External Knowledge Block implements a multi-stage Retrieval-Augmented Generation (RAG) pipeline that enriches agent analyses with real-time market intelligence. The process operates as follows:

**Stage 1: Keyword Generation.** The system processes startup descriptions through specialized LLM prompts to generate targeted search queries. For market analysis, keywords capture the core market segment with augmentation terms ("Growth," "Trend," "Size," "Revenue"). For product analysis, keywords combine company name with "News" to retrieve recent developments and sentiment data.

**Stage 2: Web Search via SerpAPI.** Generated keywords are submitted to the Google Search API through SerpAPI, configured to retrieve the top 10-20 organic search results. The API returns structured JSON containing titles, snippets, sources, publication dates, related questions, and top stories where available.

**Stage 3: Information Extraction and Structuring.** Raw search results undergo systematic processing to extract relevant fields: (1) *Organic Results* provide title, snippet, source, and date for each result; (2) *Related Questions* capture common queries and associated answers from knowledge panels; (3) *Top Stories* aggregate breaking news with publication metadata. Sitelinks are recursively processed to extract additional context from expanded search result sections. This structured data is concatenated into a comprehensive knowledge document.

**Stage 4: LLM-Based Synthesis.** The compiled search data is submitted to an LLM with synthesis prompts tailored to each domain:

**Market Synthesis Prompt:**

```
As a market research analyst, synthesize the following market data into a
structured report. Focus on:
1. Market size and growth rates (include specific numbers)
2. Industry trends and developments
3. Competitive dynamics
4. Market timing and sentiment

Use specific data points from the research where available.
Format your response as a clear, data-driven market report.
```

**Product Synthesis Prompt:**

```
You will assist me in summarising the latest information and news about the
company. After google search, you are given important context information
and data (most of the time). Now please summarise the information as a
report to highlight the latest information and public sentiment towards the
company and its product, alongside with your existing knowledge. Make your
response structured and in detail.
```

**Stage 5: Integration into Agent Analysis.** The synthesized market and product reports are injected into the respective agent prompts as "Additional Information" or "External Research" context blocks. Agents reference these reports when generating viability scores, competitive assessments, and risk evaluations, ensuring that LLM reasoning is grounded in current, verifiable external data rather than relying solely on parametric knowledge.

### G.1 Technical Implementation Details

- **API Configuration**: SerpAPI configured with `engine="google"`, `num=5-20` results per query
- **Rate Limiting**: 1-second delays between sequential requests to comply with API quotas
- **Error Handling**: Fallback to empty knowledge blocks if API fails, with detailed logging
- **Result Processing**: Top N organic results (N=10-20) filtered for content quality (non-empty snippets)

- **Knowledge Injection**: Synthesized reports appended to agent prompts under explicit section headers

This RAG architecture ensures that SSFF evaluations incorporate real-time market conditions, recent product developments, and competitive intelligence unavailable in pre-trained LLM parameters, significantly improving the framework's ability to assess startup viability under current market dynamics.

# H Results & Case Studies

## H.1 Sample Numerical Output

A table is listed here as a sample summary.

| Metric | Value |
|---|---|
| Overall Prediction | Successful, 85% (overall positive) |
| Founder Segmentation | L5 (indicating high likelihood of success) |
| Founder-Idea Fit | 0.58861464 (indicating a good fit) |
| Categorical Prediction | Successful |
| Market Viability Score | 8 (strong market viability) |
| Product Viability Score | 8 (strong product viability) |
| Founder Competency Score | 9.25 (strong founder competency) |

Table 7: Sample Numerical SSFF Outputs & Implications

## H.2 Case Studies Summary

The below case is for a specific series-A company and its founders. The same input is given to both SSFF and Vanilla GPT.

### H.2.1 Version 1: SSFF Analysis Summary

**Founder's Competency & Rating**

- redacted: Co-Founder & CEO at Company (redacted), MBA from Cambridge, featured in business magazines, active user engagement in 50 countries. Rating: L3 - Moderate likelihood of success.
- redacted: CTO at Company (redacted), Ph.D. from UC Berkeley, extensive experience in tech. Rating: L3 - Moderate likelihood of success.
- Combined Rating: L3, indicating moderate likelihood of success for both founders.

**Product Viability & Competency**

- Viability Score: 9/10.
- High user satisfaction and active engagement in 50 countries.
- Market Fit: Addresses sleep deprivation among parents, uses AI and expert insights for personalized solutions.
- Scalability: Potential expansion into broader child wellness sectors.

**Market Viability**

- Viability Score: 8.5/10.
- Growth Rates: Positive trends in children's wellness markets.
- Market Timing: Growing demand for digital sleep solutions.
- Strategy: Strong digital presence, content marketing, subscription models.

**News & Market Background**

- Recent funding of $12.5 million.
- High public sentiment and user satisfaction.
- Market Sentiment: Strong, driven by technological advancements in child wellness.

**Quantifiable Modelling**

- Founder-Idea Fit Score: 0.41 (Moderate).
- Predicted Outcome: 35% success probability.
- Recommendation: Hold on investments; monitor market and founders' performance.

**Recommendation:** *Hold*

- **Strengths:** Strong market and product viability, innovative approach, positive user feedback, and substantial funding.
- **Concerns:** Moderate founder competency and fit with the startup idea, potential gaps in child wellness expertise.
- **Action:** Monitor the startup's progress, particularly in addressing identified gaps and demonstrating traction in the market. Reconsider investment after close monitoring and ensuring the founders' readiness to bridge the gaps.

#### H.2.2 Version 2: Vanilla GPT-4o Version Analysis Summary

**Founder's Competency**

**redacted (Co-Founder & CEO)**

- **Experience:** Over seven years leading Company (redacted), with prior roles in project management and client solutions.
- **Education:** MBA from the University of Cambridge

**redacted (Co-Founder & CTO)**

- **Experience:** Co-founder of Company (redacted) and Company (redacted), with a background in low-power wireless IP implementation.
- **Education:** Ph.D. in Electrical Engineering and Computer Sciences from UC Berkeley.

**Rating:** Strong technical and managerial expertise, with significant experience in relevant fields.

**Product Viability**

- Strong product-market fit, leveraging AI and expert consultation for unique value.
- High user satisfaction, indicating acceptance and efficacy.

**Market Viability**

- Large potential in child sleep solutions.
- Well-positioned in the digital health sector, and aligned with the current trends.

**News & Market Background**

- Growing interest in digital health solutions.

- Need to navigate regulatory landscapes and competitive pressures.

**Quantifiable Modeling**

- Strong foundational backgrounds of founders.
- Significant market opportunities in child wellness.
- Recommendations: Invest in R&D, market expansion, and strategic partnerships.

**Overall Investment Recommendation** Company (redacted) presents a compelling investment opportunity. Strategic investments in AI, market expansion, and partnerships are advised to capitalize on its innovative approach and strong market presence.

# I Data Integrity and Contamination Mitigation

To ensure experimental validity and address potential language model training data contamination concerns, we employ a multi-layered safeguard protocol grounded in our dataset's structure and temporal characteristics.

## I.1 UUID-Based Anonymization

All 9,745 startups in our source corpus are identified via Crunchbase UUID v4 identifiers (e.g., `a2323b6c-29b9-4750-905c-cdcbd9ce92b8`), which serve as non-semantic tracking keys throughout our experimental pipeline. These 5,843 unique identifiers provide stable cross-split tracking without exposing semantically interpretable company information during model evaluation. During inference, records are matched via UUID rather than name, preventing direct lookup-based contamination where language models might recall memorized outcomes for well-known entities.

**Acknowledged Limitation**: While UUID-based tracking prevents *accidental* data leakage across experimental splits, it does not eliminate contamination risk during evaluation. Our current implementation includes company names in prompts sent to language models (e.g., in market research synthesis tasks), which may allow models to access memorized knowledge about high-profile startups. For well-known entities achieving unicorn status or major acquisitions, the model may recall outcomes from training data rather than perform genuine causal reasoning. We estimate that this affects fewer than 5% of our test set (primarily successful companies with significant media coverage), but cannot quantify the exact impact without controlled ablation studies.

## I.2 Stratified Sampling with Reproducibility Guarantees

**Strength**: Our sampling methodology ensures reproducibility and prevents selection bias. Test sets are constructed via stratified random sampling with a fixed seed (`random_state=42` in scikit-learn), drawing independently from successful and unsuccessful company pools before shuffling post-concatenation. We generate five test configurations with varying class distributions (10%, 20%, 30%, 40%, 50% success rates) to validate model robustness across imbalance scenarios, though our primary evaluation uses the realistic 10% distribution. Each test set comprises exactly 1,000 companies, ensuring consistent evaluation scale while maintaining 940+ unique UUIDs per configuration. The fixed seed guarantees bit-identical reproduction: independent researchers using `random_state=42` will generate identical test sets from the same source corpus, enabling precise replication and sensitivity analysis.

## I.3 Temporal Validation and Outcome Determination

**Strong Natural Protection**: Our dataset exhibits robust temporal separation from language model training cutoffs, providing our strongest contamination safeguard. Companies were founded between 2010 and 2022, with successful startups spanning the full range and unsuccessful startups concentrated in 2010–2016. Critically, success/failure labels—defined by the $500M+ threshold for funding, or acquisition or IPO valuation—were determined in 2020–2023 based on exit events, late-stage funding rounds, and operational status. This determination occurred *after* major language model training cutoffs: GPT-4 (April 2023), GPT-4o (October 2023), and Claude (early 2023).

This temporal stratification ensures that while company *descriptions* and founding narratives may overlap with training corpora (Crunchbase is publicly accessible), their ultimate *outcomes* reflect post-training real-world events. For the 2010–2016 founding cohort—representing the majority of unsuccessful startups—outcomes were determined 4–10 years post-founding, well beyond the information horizon available during language model pre-training. Consequently, models cannot rely on memorized outcomes and must instead perform genuine analysis of startup success factors.

**Caveat**: This temporal protection is strongest for the unsuccessful startup cohort (concentrated in 2010–2016, outcomes finalized 2020–2023). However, successful startups span 2010–2022, with some achieving prominence before LLM training cutoffs. Companies like those mentioned in our dataset descriptions (e.g., "working with brands including AirBnB") may be indirectly identifiable through contextual clues even without explicit naming.

### I.4 Strict Train-Test Isolation

**Strength**: The 1,000-startup test set remains completely isolated from all training procedures. Test data reside in separate file paths distinct from the 2,000-startup training corpus used for Random Forest calibration and founder segmentation. Training scripts never import test set files, and evaluation scripts operate in read-only mode, extracting predictions without model parameter updates. All records are cross-referenced via UUID to prevent accidental inclusion in training folds, and test set creation occurred temporally after training completion, enforcing procedural isolation.

### I.5 Summary of Contamination Risk Profile

We provide an honest assessment of contamination risks and mitigation status:

| Risk Factor | Level | Mitigation | Status and Impact |
|---|---|---|---|
| **Company name visibility** | HIGH | Not addressed | Names in prompts allow LLM recall of known outcomes for high-profile companies (<5% of test set) |
| **Temporal leakage (2021–2022 companies)** | MODERATE | Partial | Limited to successful startups; outcomes still post-training |
| **Temporal leakage (2010–2020 companies)** | LOW | Strong | Outcomes determined 2020–2023, after LLM training |
| **Founder LinkedIn leakage** | MODERATE | Partial | Detailed profiles may enable indirect inference through founder associations |
| **UUID-based tracking** | — | Implemented | Prevents cross-split leakage; does not prevent name-based recall |
| **Stratified sampling** | — | Implemented | Ensures reproducibility and prevents selection bias |
| **Train-test isolation** | — | Implemented | Complete separation of training and evaluation data |

Table 8: Contamination risk assessment and mitigation status. Green indicates effective mitigation, orange indicates partial mitigation, red indicates unaddressed risk.

**Recommended Future Mitigation**: To fully eliminate contamination concerns, future work should implement the following: (1) *Name anonymization in prompts*, replacing specific company names with generic descriptors (“a social media marketing platform founded in 2011”); (2) *Ablation studies* comparing model performance on named vs. anonymized company presentations to empirically quantify contamination effects; (3) *Synthetic data generation* preserving statistical properties while eliminating all training data overlap; and (4) *Temporal holdout experiments* evaluating models exclusively on companies with outcomes determined in 2024 or later (post all LLM training cutoffs).

## J Limitations and Future Directions

This work has several limitations that motivate future research. We organize these into areas where **we have made progress** and **areas requiring further development**.

### J.1 Dataset Scale and Geographic Diversity

**Current Strength**: Our dataset demonstrates substantial breadth across dimensions relevant to startup evaluation. The source corpus spans 5,126 distinct industry categories, 90 countries, and 1,115 cities, providing diverse coverage of global startup ecosystems. The 1,000-startup test set yields 95% confidence intervals of approximately $\pm 3.1\%$ for proportion estimates at 50% prevalence, sufficient for statistically rigorous method validation.

**Acknowledged Limitation**: Despite this breadth, representation remains weighted toward U.S.-based (estimated 60–70%), venture-backed, English-documented technology startups, reflecting Crunchbase's inherent coverage biases. Our evaluation has not systematically examined performance across: (1) *non-English markets* (Asia-Pacific, Latin America, Middle East); (2) *bootstrapped ventures* lacking institutional funding; (3) *non-technology sectors* (manufacturing, agriculture, traditional services); or (4) *founder demographics* (gender, ethnicity, educational background).

**Path Forward**: Expansion to 10,000+ companies would enable robust subgroup analysis. Partnerships with regional databases (Tracxn for India, 36Kr for China) would improve geographic diversity. Explicit demographic data collection—with appropriate privacy safeguards and bias auditing—would enable fairness analysis currently impossible with our anonymized dataset.

### J.2 Prospective Validation and Human-in-the-Loop Integration

**Current State**: Our evaluation constitutes retrospective analysis using historical outcome labels. This approach enables controlled method development and comparison across architectures but does not demonstrate utility in actual decision-making workflows.

**What We Have Not Done**: We have not conducted:

- **Live pilot studies** with venture capital firms evaluating real deal flow
- **A/B testing** comparing VC investment decisions with vs. without SSFF assistance
- **Longitudinal tracking** of portfolio outcomes when SSFF recommendations are followed vs. ignored
- **Cognitive task analysis** measuring whether SSFF outputs actually inform or merely confirm VC intuitions
- **Active learning integration** where expert feedback refines model predictions over time

**Why This Matters**: Offline metrics (accuracy, F1, calibration) correlate imperfectly with real-world utility. A model achieving 70% F1 but with errors that concentrate on easily-detected failures provides less value than one achieving 60% F1 with insights on non-obvious opportunities. Prospective validation would establish whether SSFF improves *decisions* rather than merely *predictions*.

**Path Forward**: We are initiating collaborations with early-stage investment firms for pilot deployment in Q4 2025. Planned studies include: (1) parallel evaluation where VCs and SSFF independently assess opportunities, with outcome tracking; (2) wizard-of-oz experiments where VCs receive SSFF analyses but believe they come from human associates; and (3) think-aloud protocols capturing how VCs interpret and use multi-agent outputs.

### J.3 Robustness to Data Quality Degradation

**What We Have Built**: Our framework employs defensive error handling throughout the pipeline. Missing company descriptions trigger fallback to minimal founder-only analysis; external API failures for market research gracefully degrade to parametric LLM knowledge; incomplete founder profiles use available fields without imputation. Error handling prevents cascade failures where one missing field crashes the entire analysis.

**What We Have Not Quantified**: We have not conducted systematic *data ablation studies* measuring performance degradation as features are progressively removed. Critical unaddressed questions include the following:

- What is the minimum viable data profile for reliable analysis? (Is founder background alone sufficient? Can we predict without product descriptions?)
- How does prediction confidence degrade with data sparsity? (Do calibration metrics deteriorate faster than raw accuracy?)
- Which features are most critical for each outcome class? (Are different features needed to identify successes vs. failures?)

**Why This Matters**: Real-world deployment encounters incomplete data. Early-stage startups lack the extensive public profiles of Series B companies. Data quality requirements must be explicitly characterized to set appropriate deployment boundaries.

**Path Forward**: Controlled ablation experiments are tractable with our existing test set. For each company, we can systematically mask the following information: (1) all product information; (2) all founder background; (3) all market context; (4) combinations thereof. Measuring performance across these conditions establishes minimum data requirements and informs data collection priorities.

### J.4 Computational Efficiency and Scalability

**Current Performance**: Per-startup analysis averages 2–3 seconds, dominated by language model API calls (1.5–2.5 seconds across three agents) rather than ML inference ($<$100ms). For 1,000-startup portfolio screening, total runtime is 45–60 minutes with mandatory rate limiting.

**Optimization Opportunities**: The modular architecture enables parallelization—market, product, and founder agents operate independently and can execute concurrently, reducing wall-clock time to 15–20 minutes with 3-way parallelism. Batch processing with asynchronous API calls could further reduce latency. However, we have not implemented production-grade optimizations.

**Scaling Challenges Beyond Proof-of-Concept**: Large-scale deployment (10,000+ companies) requires infrastructure we have not built:

- **Cost management**: At current GPT-4o-mini pricing ($0.50–$1.00 per startup), analyzing 10,000 companies costs $5,000–$10,000—acceptable for institutional investors but requiring budget justification
- **Caching strategies**: Repeated analyses of the same company over time (quarterly updates) waste resources without caching prior LLM outputs
- **Incremental updates**: When only founder background changes (new hire), re-running full market analysis is inefficient
- **Priority queuing**: Real-time deal flow screening requires sub-minute response times

**Path Forward**: Moving to production will demand engineering work beyond the research prototype, including result caching with invalidation policies, incremental analysis triggering only affected agents, and priority queuing for time-sensitive evaluations.

### J.5 Temporal Adaptability and Market Regime Modeling

**What We Have Implemented**: The External Knowledge Block retrieves real-time market intelligence—market size, growth trajectories, competitive dynamics, sentiment trends—at inference time, enabling adaptation to current market conditions rather than relying solely on static training data.

**What We Have Not Modeled**: Startup success criteria vary systematically across macroeconomic regimes, but SSFF lacks explicit regime awareness:

- **Bull vs. bear markets**: During expansions, aggressive growth and high burn rates may be rewarded; during contractions, profitability and capital efficiency typically dominate
- **Interest rate environments**: Zero-rate regimes (2010–2021) generally favor long-term bets and market share acquisition; rising rates (2022+) tend to favor near-term cash generation

- **Sector rotations**: Investor preferences periodically shift among segments as we have seen in consumer (2010–2015), enterprise SaaS (2016–2019), cryptocurrency (2020–2021), and AI (2023+)
- **Regulatory changes**: Evolving frameworks such as GDPR (2018), CCPA (2020), and AI regulations (2024+) fundamentally alter the viability of data-intensive business models

Current SSFF predictions implicitly reflect training data spanning 2010–2023, averaging across multiple market cycles. A startup evaluated in 2024's AI boom may receive overly optimistic assessments, while one evaluated during 2022's correction may be unfairly penalized.

**Why This Is Hard**: Regime modeling requires: (1) *labeled regime data* (when did markets shift from bull to bear?); (2) *regime-specific training sets* (sufficient data within each regime for robust estimation); and (3) *real-time regime detection* (algorithms classifying current conditions). We have none of these components.

**Path Forward**: Minimum viable approach incorporates temporal features into prompts: "Evaluate this startup considering current market conditions: Q4 2024, rising interest rates, AI investment boom, Series A median $15M (up 40% YoY)." More sophisticated approaches require regime-switching models with explicit state variables, demanding substantial additional research.

### J.6 Fairness, Bias, and Ethical Considerations

**What We Have Not Done**: Our evaluation has not examined differential performance across demographics. We cannot answer critical fairness questions:

- Does SSFF exhibit higher false negative rates for female founders, holding business fundamentals constant?
- Do model recommendations systematically favor geographic clusters (e.g. San Francisco, New York) over emerging ecosystems (e.g. Austin, Miami or non-US hubs)?
- Does founder segmentation penalize individuals from non-traditional backgrounds such as bootcamp graduates or serial entrepreneurs from failed ventures?
- When demographic attributes are varied in prompts while business descriptions are held constant, do model predictions change?

**Risks of Unchecked Deployment**: If SSFF encodes historical biases, deployment could have the following problems:

- **Entrench inequality**: Recommending against underrepresented founders with objectively strong ventures
- **Narrow innovation**: Favoring incremental ideas resembling past successes over paradigm-shifting approaches from non-traditional sources
- **Feedback loops**: If SSFF-guided investments concentrate on familiar patterns, training data for future models becomes increasingly homogeneous
- **Legitimacy concerns**: Algorithmic recommendations may be perceived as objective, obscuring subjective value judgments embedded in training data

**Prerequisites for Responsible Deployment**:

1. **Bias auditing protocols**: Systematic testing for disparate impact before production use
2. **Adversarial debiasing**: Training procedures penalizing correlation between predictions and protected attributes
3. **Transparency requirements**: Disclosing model limitations to end-users, particularly regarding underrepresented populations
4. **Human-in-the-loop safeguards**: Ensuring SSFF augments rather than replaces human judgment, with special scrutiny for decisions affecting marginalized founders
5. **External audits**: Independent fairness assessments by domain experts and affected communities

We view these as essential future work before responsible real-world deployment, not optional enhancements.

## K Final Remark

We have introduced the SSFF, a novel multi-agent system that addresses systematic over-prediction bias in LLMs for VC decision support. Through structured integration of specialized agents, quantitative grounding via LLM-enhanced ML models, and real-time market intelligence retrieval, SSFF achieves 67.8% weighted-F1 on realistic imbalanced test data. This is nearly 6× the performance of GPT baselines.

Our comprehensive evaluation across 9,745 startups demonstrates that structured reasoning architectures fundamentally outperform direct language model prompting under class imbalance, offering both higher predictive accuracy and superior calibration. The framework's modular design enables interpretable analysis across market viability, product innovation, and founder competency dimensions, providing actionable insights beyond binary success/failure predictions.

This work establishes three key technical contributions: (1) demonstration that LLM optimism bias can be systematically corrected through agent specialization and quantitative grounding; (2) validation that retrieval-augmented generation significantly improves startup evaluation by incorporating real-time market conditions; and (3) evidence that founder-idea fit scoring provides complementary signal to traditional ML features. These contributions advance the state-of-the-art in using LLMs for high-stakes financial decision-making.

However, we acknowledge substantial limitations requiring ongoing research before responsible real-world deployment. Name-based contamination in prompts, lack of prospective validation, absence of fairness auditing, and gaps in temporal regime modeling all demand further investigation. We have been transparent about what we achieved and what remains unaddressed, providing honest assessment to guide future work.

By combining the contextual understanding of LLMs with the statistical rigor of traditional ML and the grounding of external knowledge retrieval, SSFF charts a methodological path forward. Yet the path from research prototype to deployable system remains long, requiring engineering investment, prospective validation, fairness guarantees, and domain partnership. This work provides the technical foundation; realizing its practical potential depends on addressing limitations we have explicitly acknowledged.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract/intro state the contributions (multi-agent SSFF, bias of vanilla LLMs, empirical gains) match the methods and results reported in Secs. 2–3 and Apps. A–C.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The conclusion and the Appendix sections include all the limitations in detail.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   [NA]

   Justification: The paper is empirical/methodological (no new theorems); we provide algorithmic specifications and ablations (Secs. 3–4, Apps. B–C) rather than formal proofs.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in the Appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   [Yes]

   Justification: We release code with exact configs, seeds, and scripts.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a LLM), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a LLM), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is released with a reproducibility script; dataset access is provided via licensed sources (e.g., Crunchbase/Dealroom) with step-by-step construction, plus an anonymized/synthetic subset and aggregated labels to mirror results (App. D–E).

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and test details are shared in the paper. Hyperparameters and type of optimizers are shared in the released code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in the Appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Statistical details of the set-up of the paper is described in more details in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   We provide time of execution and cost of experiments in the paper.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   All complied.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    [Yes]

    Justification: The conclusion and introduction sections address societal impact. Also, we touched these points in the Limitations and Future Directions section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: We do not release proprietary labels; we provide an anonymized/synthetic dataset and gated scripts with usage guidelines, investment disclaimers, and evaluation-only checkpoints.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All third-party datasets/APIs and libraries are cited with versions and licenses; we follow provider ToS and include license files in the supplement.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release the SSFF codebase, configs, and documentation (readme, data prep guide, evaluation scripts), plus a synthetic benchmark with schema, limitations, and license (App. D–E).

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing or human-subjects studies were conducted.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human-subjects research is involved; IRB approval is not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer:[Yes]

    Justification: Yes, any usage was described throughout the paper, both in the main body and in the Appendix section.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.