International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

A Smart Decision Support Platform for Venture Idea Inspection and Feasibility Analysis

Mrs. Sushma T N
[GUIDE], Assistant Professor Department of Computer Science and Engineering
G Madegowda Institute of Technology (GMIT)
Bharathinagara, Mandya, Karnataka, India

Anushree R
Department of Computer Science and Engineering
G Madegowda Institute of Technology (GMIT)
Bharathinagara, Mandya, Karnataka, India

Chaithrashree K V
Department of Computer Science and Engineering
G Madegowda Institute of Technology (GMIT)
Bharathinagara, Mandya, Karnataka, India

Harshitha H D
Department of Computer Science and Engineering
G Madegowda Institute of Technology (GMIT)
Bharathinagara, Mandya, Karnataka, India

Namitha B
Department of Computer Science and Engineering
G Madegowda Institute of Technology (GMIT)
Bharathinagara, Mandya, Karnataka, India

Abstract - Early-stage founders must decide whether an idea deserves time and money before reliable market data exists, yet most validation tools either need paid data or give opaque conclusions. This paper presents VentureIQ, a locally run, low-cost web application that helps a founder examine an idea and decide what to test next. The system combines a 100-point readiness rubric that measures how completely an idea is specified and evidenced, TF-IDF cosine similarity against a catalogue of 108 source-linked companies, rule-based funding-route guidance supported by 3,044 cleaned historical funding records, a three-scenario cash-flow and break-even model, a location matrix scored from the user's own ratings, and a prioritised recommendation list. An optional AI layer, off by default, writes a narrative only from retrieved web sources, and code removes statements that lack a valid citation. No trained prediction model is used, no success probability is claimed, and no market size, rate or investor record is invented; simulated demonstration data are labelled and excluded from every score. The software is built with Python, Flask and SQLite and passes 106 automated tests. On six illustrative inputs the pipeline gave sensible sector labels and similar ventures, and a sparse idea scored 17 to 26 while a fully specified one scored 88, showing that the score tracks completeness. No user study was carried out, so results are preliminary.

Keywords - Startup validation; decision support; TF-IDF; cosine similarity; rule-based scoring; financial scenarios; Flask

I. INTRODUCTION

Startup ideas should be examined before time, money and effort are committed. Many ventures struggle for reasons that can be tested cheaply: an unclear customer, an untested price, unrecognised competitors, or a funding plan that does not match the stage [5], [6]. Manual evaluation is subjective and usually treats market, competition, finance and location separately. Student founders in particular lack structured tools, and web-based AI validators generally do not show how their conclusions were produced.

VentureIQ addresses this gap with one pipeline: the founder enters an idea, rule-based and text-similarity modules analyse it locally, and a report shows a readiness score, similar ventures, funding routes, financial scenarios and a validation roadmap, with the method behind every result. The main contributions are:

- A transparent 100-point readiness rubric that explains each point awarded and is explicitly not a success probability.
- A TF-IDF and cosine similarity module that ranks similar ventures from a source-linked catalogue and shows the overlapping terms.
- Rule-based funding guidance and a cleaned historical funding dataset, with generic guidance kept separate from verified facts.
- A deterministic three-scenario financial model and a location matrix based on the user's own evidence, so no regional statistics are fabricated.
- A prioritised recommendation list, business-plan draft and 12-week roadmap derived from the actual results, plus an optional, source-validated AI layer.

II. LITERATURE REVIEW

Belgaum et al. compared several machine-learning algorithms for predicting startup success and reported an accuracy of 88.48 %, but noted that scarce startup-failure data limits real-world performance [1]. Zuo et al. studied a deep-learning decision-support system for investment and financing and reported 91.7 % accuracy in their comparison, with results that depend on input data quality [2]. Gracy et al. compared machine-learning methods for software-startup success prediction, again depending on dataset coverage and chosen features [3]. For text comparison, Liu et al. discuss cosine-distance similarity between texts, which can support comparison of venture descriptions but does not by itself establish feasibility [4].

These studies mainly predict success or investment outcomes and need labelled historical outcomes that are not available for a student project. VentureIQ therefore does not predict success. It uses classical term weighting [7], [8], [9] to compare an idea with real ventures, and it turns validation advice from lean-startup practice [5], [6] into specific, checkable steps. For the optional AI layer it follows the retrieval-grounded approach of [10] and adds a code-level citation check.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

III. PROPOSED SYSTEM

VentureIQ is a server-rendered web application. Routes validate input, call an orchestrator that runs independent analysis modules, and store one JSON result per idea. Fig. 1 shows the architecture. Every output carries a label stating whether it is rule-based, statistical text similarity, historical data, arithmetic, simulated or AI-generated.

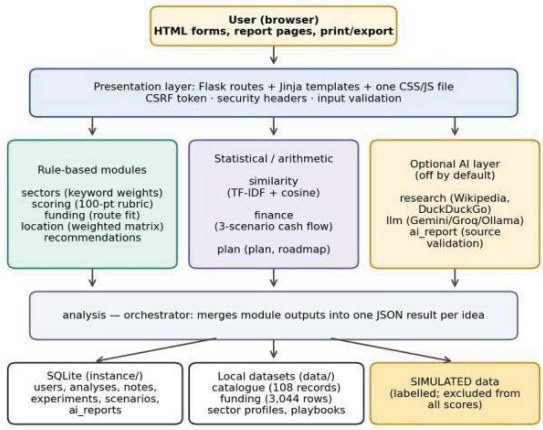


Fig. 1. Architecture of VentureIQ. The AI layer is off unless configured.

A. Software and Persistence

The application uses Python, Flask and Jinja templates with one stylesheet and a small JavaScript file; charts are server-generated SVG so no external script is needed. SQLite stores users, analyses (inputs and full results), research notes, experiments, finance scenarios and AI reports. Identical resubmissions are detected by a content fingerprint with a uniqueness constraint. Work done without an account is private to the browser and is moved into an account at sign-in. Security measures are CSRF tokens on all state-changing requests, HttpOnly cookies, a Content-Security-Policy, escaped templates, input length limits, salted password hashing, login throttling and parameterised SQL [13]. Only Flask is required to run the core.

B. Sector and Business-Model Inference

The idea text is matched against keyword lists for fifteen sectors. Each distinct matched term adds a weight by tier (core 8, strong 4, normal 2, weak 1) and the highest-scoring sector is chosen. Confidence is high when the top score is at least 6 with a margin of at least 3, medium when it is at least 3 with a margin of at least 1, and low otherwise. The user may override the sector. The business model is taken from the user or inferred from keywords. This is a heuristic, not a trained classifier.

C. Readiness Score

The score is the sum of nine capped dimensions (Table I). Each has fixed thresholds on the text and numbers entered, and the report shows the points, the reason and an improvement hint for each. The score measures how completely and coherently the idea is specified; it does not predict success.

TABLE I. READINESS RUBRIC

Dimension	Max	Basis
Problem and solution clarity	20	Text
Target customer defined	12	Text
Revenue model	10	Choice/infer
Location specificity	10	Text
Competitor awareness	10	List size
Financial inputs supplied	14	Numbers
Basic numbers consistency	8	Arithmetic
Stage and funding need	8	Fields
Evidence gathered	8	Text, digits

D. TF-IDF and Cosine Similarity

Each catalogue record joins its name, category, description, target customers, problem keywords and business model. Text is lower-cased, stop-words removed, lightly stemmed, passed through a short hand-written synonym list and extended with adjacent word pairs. With N records and document frequency df(t):

$$tf(t,d) = 1 + \ln c(t,d) \quad (1)$$

$$idf(t) = \ln[(1 + N)/(1 + df(t))] + 1 \quad (2)$$

$$sim(q,d) = (v_q \cdot v_d) / (||v_q|| \, ||v_d||) \quad (3)$$

where v = tf × idf. The score is shown as a percentage with the shared terms that contributed most. Bands are: 40 or more high, 20 to 39 moderate, 8 to 19 low, below 8 negligible. No bonus is added for category agreement, so the value is a pure cosine. The report states that wording overlap does not prove competition or predict outcomes.

E. Financial Model

Given price p, month-1 customers n1, monthly growth g, fixed cost F, variable cost per customer v, opening cash K and one-time spend U, month m gives:

$$n_m = n1\ (1+g)^{(m-1)}; \quad profit_m = (p - v)\ n_m - F \quad (4)$$

$$cash_m = cash_(m-1) + profit_m, \quad cash_0 = K - U \quad (5)$$

$$n^* = ceil(F / (p - v)), \quad p > v \quad (6)$$

Conservative and optimistic scenarios scale customers, price and costs by editable multipliers (defaults 0.6, 0.9, 1.15 and 1.3, 1.05, 0.95). These are illustrative judgements, not market data. The model reports break-even customers, first profitable month, payback of the one-time spend, lowest cash and the extra funding needed to keep cash non-negative. Inputs are validated so that division by zero cannot occur; if p is not above v the report says break-even is unreachable.

F. Funding Guidance

Six routes (bootstrapping, grants and incubators, government schemes, loans, angel investors, venture capital) receive one of four fit labels from rules over stage, stated amount band, evidence, business model and currency. For example, venture capital is marked probably premature unless the idea is at revenue stage with a large amount and a scalable model. Labels are not eligibility

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

decisions; scheme names are pointers to look up, and no terms, rates or links are asserted. A supplied dataset of 3,044 historical funding rows is also cleaned and searchable, and every look-up is labelled historical.

G. Location Analysis

The system holds no rent, footfall or demographic statistics and therefore shows none. For a proposed location it reports how specific it is and, where the city appears in the dataset, how many historical funding records name it. A comparison tool lets the user rate two to four candidates from 1 to 5 on six criteria, and the score is

$$S = 100 \cdot sum_k\ w_k\ (r_k - 1)/4 \qquad (7)$$

with sector-specific weights w_k that sum to one. The result is only as reliable as the user's own field research.

H. Prioritised Recommendations

Recommendations are generated from weak rubric dimensions, similarity outcome, unit economics, location specificity and funding fit. Each is rated 1 to 5 for impact, urgency and feasibility by fixed rules and ranked by

$$priority = 0.5\ impact + 0.3\ urgency + 0.2\ feasibility \qquad (8)$$

I. Optional Grounded AI Layer

When switched on, free web search (Wikipedia and DuckDuckGo results) returns numbered sources with URL, snippet and date. A free language model (Gemini, Groq or a local Ollama model) writes structured JSON using only those sources [10]. Code then removes any market statement or competitor that cites a non-existent source, any number not found in a cited snippet, and any competitor name not found in a cited source. SWOT, personas and the verdict are labelled unsourced opinion. The layer is rate-limited and discloses what is sent to third parties.

IV. DATASETS

TABLE II. DATASETS LOADED BY THE APPLICATION

Dataset	Size and nature
Venture catalogue	108 records, each with a source URL
Funding records	3,044 rows, historical, 0 exact duplicates
Sector profiles	15 hand-written profiles
Simulated planning data	About 3,000 random rows, never used in scores

The funding file is cleaned before use: missing-value markers are removed, city and industry spellings are merged (for example Bangalore to Bengaluru), round types are normalised and unknown amounts are kept as unknown instead of zero. The catalogue keeps only records with a name, description and http(s) source, and merges duplicates. The simulated files were generated by a script with a fixed seed and are shown only in panels marked Simulated.

V. RESULTS AND DISCUSSION

A. Illustrative Runs

Six inputs were run through the unmodified pipeline (Table III). Inputs A to E contain a one-sentence description and a location; F is the farm idea of A with customers, model, stage, investment, price, costs, competitors and evidence added. These runs are demonstrations written by the authors, not an accuracy evaluation, because no labelled ground truth exists.

TABLE III. ILLUSTRATIVE OUTPUTS (REAL RUNS)

ID	Sector (conf.)	Score	Top similar ventures (cosine %)
A	Agriculture (medium)	20	Aibono 29, Ninjacart 17, BigBasket 16
B	Education (high)	23	Vedantu 63, Physics Wallah 49, Unacademy 47
C	Healthcare (high)	17	Apollo 24|7 45, Practo 39, PharmEasy 28
D	Food and Grocery (high)	17	Licious 32, BigBasket 24, Zomato 22
E	Technology and SaaS (medium)	26	Rebel Foods 23, Tally Solutions 19, Zomato 16
F	Agriculture (high)	88	Aibono 28, Ninjacart 26, BigBasket 22

Sector labels matched the intended domain in all six cases, although confidence was only medium for A and E, whose text mixes sectors. For B and C the closest records are sensible. For E the top matches (a restaurant brand, an accounting-software firm, a delivery platform) share vocabulary rather than being true competitors, which shows the limit of lexical similarity on a small catalogue. The score rose from 17 to 26 to 88 when details were added to the same farm idea, so it tracks information completeness and should never be read as idea quality. For A to E, loans, angels and venture capital were marked probably premature; for F loans and angels became possible later while venture capital stayed premature.

B. System Behaviour

Fig. 2 shows the report overview for idea F: the score with its verdict, the inferred sector with the matched keyword, and the stated assumptions. Fig. 3 lists the closest catalogue ventures with similarity, band, overlapping terms and a source link; the page repeats that overlap is a reason to investigate, not proof of competition.

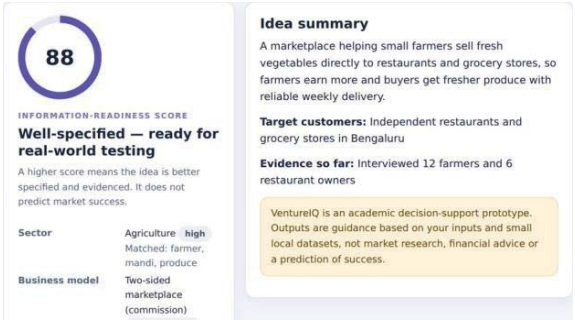


Fig. 2. Report overview with readiness score and sector inference.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026



Fig. 3. Similar ventures from the catalogue (TF-IDF cosine).

For F (price Rs. 500, 40 customers, Rs. 15,000 fixed and Rs. 100 variable cost per customer, Rs. 300,000 opening cash), Eq. (6) gives a break-even level of 38 customers per month, so the baseline sits just above break-even while the conservative case (24 customers, break-even 52) does not (Fig. 4). Because the model is arithmetic, it was verified by unit tests against hand-computed values.

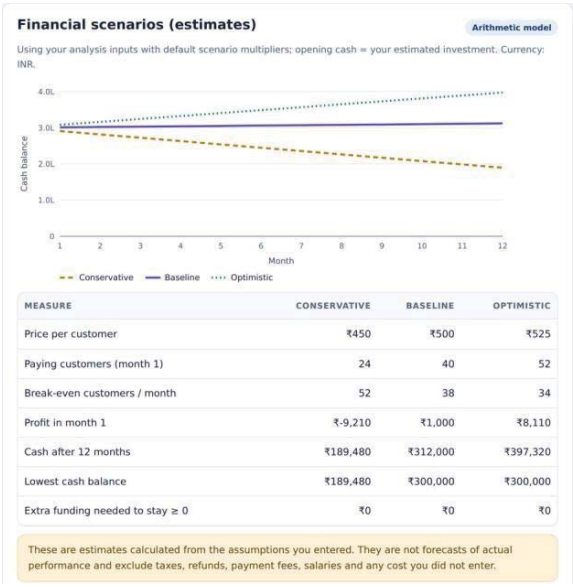


Fig. 4. Financial scenarios for idea F (estimates from user inputs).

Fig. 5 shows the prioritised recommendations: the first asks for a small pilot with a written success threshold, and the second asks the founder to compare the idea with the three closest catalogue ventures. Each item shows its reason, steps and impact, urgency and feasibility ratings.

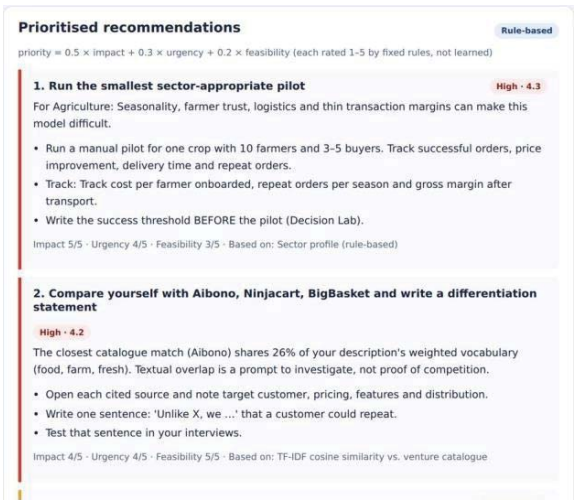


Fig. 5. Prioritised recommendations generated from the analysis.

C. Software Verification

The application has 106 automated tests and all of them pass: 77 core tests (classification, TF-IDF edge cases, validation, finance arithmetic including division by zero, score bounds, funding rules, dataset cleaning, every route, CSRF, malformed input, escaping, privacy between visitors, duplicate prevention, persistence across restart and the account flows), 18 tests of the AI layer using mocked providers, and 11 tests of the public pages. Internal links were crawled without errors, and headless-browser sessions showed no console errors and no horizontal overflow at 390 pixels. The AI layer was tested only with mocked providers and search; it has not yet been run against live services.

VI. LIMITATIONS AND FUTURE SCOPE

No user study was carried out, so usefulness to founders is unmeasured. Rubric thresholds, funding-fit rules, location weights and scenario multipliers are judgement calls and are not calibrated. The catalogue is small and India-centred, so low similarity does not mean no competition. The funding data are historical, unaudited and incomplete. The illustrative cases were chosen by the authors. The system does not predict success, and the prediction methods in [1]-[3] were not reproduced. Future work includes a usability study with student founders, a larger and refreshed catalogue, calibration of the rubric against expert ratings, measuring the citation precision of the AI layer on real output, embedding-based similarity compared with TF-IDF, and deployment hardening such as HTTPS and email verification.

VII. CONCLUSION

VentureIQ shows that a useful and explainable first-pass validation tool can be built with transparent rules, classical text similarity and simple financial arithmetic, without inventing market data or claiming to predict success. The scoring, similarity, funding, finance, location and recommendation modules work together in one web application that passes 106 automated tests. Its main weaknesses are the small knowledge base and the lack of empirical validation, which are the next steps.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering, GMIT, for support and guidance.

REFERENCES

[1] M. R. Belgaum *et al.*, "Comparative analysis of startup success rate prediction using machine learning techniques," in *Proc. IEEE ICSTSN*, 2024.

[2] Z. Peiwen, L. Qian, and Z. Ning, "Algorithm research on investment and financing decision support system based on deep learning," in *Proc. IEEE EESPE*, 2025.

[3] Gracy *et al.*, "Software startup success prediction: Machine learning of comparative analysis," in *Proc. IEEE InCoWoCo*, 2024.

[4] Liu *et al.*, "Text similarity calculation method based on optimized cosine distance," IEEE, 2022

[5] E. Ries, *The Lean Startup*. New York, NY, USA: Crown Business, 2011.

[6] A. Maurya, *Running Lean: Iterate from Plan A to a Plan That Works*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2012.

[7] K. Spärck Jones, "A statistical interpretation of term specificity and its application in retrieval," *J. Documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[8] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.

[9] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[10] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. NeurIPS*, vol. 33, 2020, pp. 9459–9474.

[11] Pallets Projects, "Flask documentation."

[12] SQLite Consortium, "SQLite documentation

[13] OWASP Foundation, "Cross-Site Request Forgery Prevention Cheat Sheet."

(This work is licensed under a Creative Commons Attribution 4.0 International License.)