

# 1. Overview

The Financial Score Engine is used to assess an individual's financial risk based on his// her income, risk sum assured, annual premiums to be paid, renewal rates of existing insurance policies with the company. The Financial score would be a normalized financial score, which is a crucial input in the insurance underwriting process for certain products to evaluate risk and or eligibility.

---

## 2. Objectives

- Quantify the applicant's financial status as a numeric score ( 1, 2, 3, 4, 5); the values being assigned as per the risk ; a) 1 - Safe , 2 - Low Risk, 3 - Medium Risk, 4 - High Risk, 5 - Reject
  - Enable risk-based decision-making during underwriting.
  - Standardize financial assessments across different data sources.
  - Allow explanation or traceability of the score (transparency for underwriters).
- 

## 3. Inputs

We will have input from proposal form and external inputs:

1. Inputs from proposal form.  
Example : Age, Occupation, Income (Annual), Sum Assured, Premium, Other Insurance Policy Sum Assured
  2. External Inputs  
ITR, Monthly Salary Credits
- 

## 4. Outputs

Output Field	Description
Financial Score	A score between 1 and 5 indicating overall financial risk (higher is riskier)

**Risk Category**      1 - Safe, 2 - Low Risk, 3 - Medium Risk, 4 - High Risk, 5 - Rejected

**Score Factors**      Top 3 contributing features to the score (for explainability)

**Underwriting Flag**    Pass / Manual Review / Decline

---

## 5. Processing & Logic

### 5.1 Preprocessing

- Data normalization & cleaning
- Imputation of missing fields (if feasible)
- Standardization across sources (e.g., units, formats)

### 5.2 Scoring Algorithm

- Start with a Rule-based model and move to ML model as the data gets accumulated.

## 6. Functional Requirements

ID	Requirement
----	-------------

FR1	Ingest applicant data from underwriting portal/API
-----	--

FR2	Validate required fields and flag missing critical data
-----	---

FR3	Generate financial score using configured logic
-----	---

FR4	Classify score into risk buckets
-----	----------------------------------

FR5	Log scoring decision and input snapshot for audit
-----	---

---

## 7. Non-Functional Requirements

ID	Requirement
----	-------------

NFR1	Response time < 1s for single applicant scoring
------	---

NFR2	Scalable to support bulk scoring (e.g., 10,000 applicants/day)
------	--

NFR3	Model retraining support every 3-6 months
------	---

NFR4 Secure handling of sensitive financial data ( IRDAI compliant)

NFR5 High availability with failover strategy

---

## 8. Integrations

- **Underwriting Portal / 3rd Party Integrations for Data Access**
  - **Policy Admin System** (optional)
- 

## 9. Assumptions

- Minimal dataset size available to train initial model.
  -
- 

## 10. Future Enhancements

- Risk Assessment ML models

## Simple Rule Engine Example

### What to Solve:

Simple weighted scoring approach combining **financial** inputs into a single **1 - 5 financial score**

### Design overview (high level)

- Output: single **Financial Score (1–5)**.
- Data Inputs:
  - **Sum Assured - Company (A)**
  - **Sum Assured - Others (B)**
  - **Annual Premium Amount (C )**
  - **Annual Income (D)**
- The various components are as under :
  - SAR/ Income Ratio (A/D) - 50%
  - TSAR/ Income Ratio ((A+B)/D) - 25%

- Premium Amount/ Income Ratio (C/D) - 25%
- Each component is the sum of sub-scores computed by simple rules (thresholds, boolean checks).
- Final score = Addition of all numbers

## Scoring Mechanism

- SAR/ Income Ratio (S)
    - $< 2 = 1$
    - $2 - 3 = 2$
    - $3 - 4 = 3$
    - $4 - 5 = 4$
    - $> 5 = 5$
  - TSAR/ Income Ratio (T)
    - $< 2 = 1$
    - $2 - 3 = 2$
    - $3 - 4 = 3$
    - $4 - 5 = 4$
    - $> 5 = 5$
  - Premium Amount/ Income Ratio (P)
    - $< 0.05 = 1$
    - $0.05 - 0.1 = 2$
    - $0.1 - 0.15 = 3$
    - $0.15 - 0.2 = 4$
    - $> 0.2 = 5$
- 

## Aggregation & interpretation

- Final numeric score = SAR/ Income Ratio + TSAR/ Income Ratio + Premium Amount/ Income Ratio → round to nearest integer.
- Interpretation example:
  - 1 → Proceed
  - 2 → Proceed, if P = 1, 2, 3, 4; else Review
  - 3 → Proceed, if P = 1, 2, 3 & S = 1, 2, 3; else Review
  - 4 → Review,
  - 5 → Reject,