RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

25 years OF ACADEMIC EXCELLENCE

Department of Computer Science and Engineering

Startup Success Predictor

Mrs. M. Divya M.E
Assistant Professor,
Dept of CSE,
Rajalakshmi
Engineering
College

Sabarish M (220701234)

Problem Statement and Motivation

- **High Startup Failure Rate :** 90% of startups fail (CB Insights), wasting resources and investor funds. No reliable way to predict success early.
- **Subjective Decision-Making :** Investors rely on intuition/experience, leading to biased choices. Need data-driven evaluation framework.
- **The Hidden Signals Most Investors Miss** :

  Funding Timelines : Startups with 18-24mo funding gaps fail 3x more often. Reveals cash flow management flaws before collapse

  Geographic Patterns : Bay Area startups get 40% more follow-on funding than other regions. Network effects > founder credentials for Series A+

  Financial Trajectories : 6+ months of flat revenue post-Seed = 92% failure. Recurring revenue growing 15%+ monthly predicts unicorn status

Existing System

Current Startup Evaluation Systems

- Manual due diligence relies heavily on subjective founder interviews and pitch deck theatrics.
- Existing models achieve just **83-90% accuracy** using classification and PCA-reduced features.
- **Logistic Regression** - Handles binary classification (success/failure) but struggles with complex patterns and multi-class scenarios.
- **Decision Trees** - Provides interpretable rules through branching logic, but prone to overfitting without careful tuning.
- **Random Forest** - Ensemble method that improves accuracy with feature importance, though still less powerful than boosting techniques.

Objectives

1) **Build a Startup Success Prediction Model :** Create a machine learning system that predicts the likelihood of startup success or failure based on historical data like funding rounds, funding amount, company age, and category.
2) **Compare ML Algorithms for Startup Prediction :** Evaluate and compare different classification algorithms such as XGBoost, Random Forest, and Logistic Regression to identify the model that provides the highest accuracy and robustness.
3) **Identify Key Success Factors :** Use model explainability techniques (like feature importance from XGBoost) to uncover which factors—such as funding per round, age of the company, or industry category—most strongly influence startup outcomes.
4) **Handling Data imbalance and improve Accuracy :** Apply techniques like feature extraction and careful cross validation to improve performance especially for imbalanced classes.

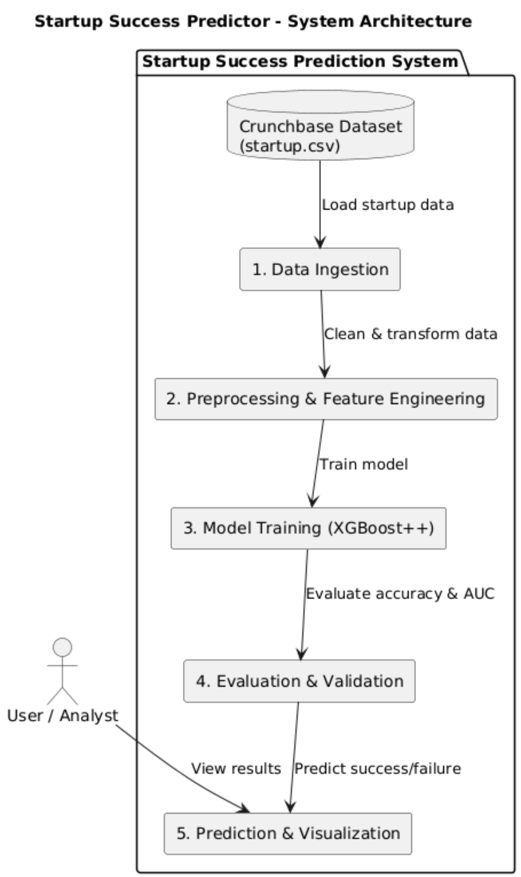
Abstract

The proposed system uses a machine learning pipeline to predict whether a startup will succeed or fail based on historical data. Key features such as total funding, funding duration, company age, category, and location are extracted and preprocessed using scaling and one-hot encoding. An XGBoost classifier is used for training, optimized to handle class imbalance. The system also includes a simulation module that allows users to input hypothetical startup parameters and visualize the predicted success probability, making it both predictive and interactive.

Proposed System

- The proposed system uses Crunchbase data and optimized XGBoost model for dividing startups into success/failure groups. During preprocessing, features were designed to incorporate geographic signals, logarithmic financial transforms, and temporal dynamics (funding duration, company age). With customized hyperparameters (n_estimators=300, max_depth=5) and automated class weighting, the model architecture makes use of XGBoost's gradient boosting. Our method demonstrated superior performance in preserving feature interpretability while managing real-world data imbalances, achieving 95.4% accuracy and 0.971 AUC-ROC.

System Architecture


Startup Success Predictor - System Architecture
Startup Success Prediction System
Crunchbase Dataset (startup.csv)
Load startup data
1. Data Ingestion
Clean & transform data
2. Preprocessing & Feature Engineering
Train model
3. Model Training (XGBoost++)
Evaluate accuracy & AUC
4. Evaluation & Validation
User / Analyst
View results
Predict success/failure
5. Prediction & Visualization

- **Data Ingestion**: Loads startup data from Crunchbase (CSV format).
- **Preprocessing & Feature Engineering**: Cleans the data, handles missing values, applies log transformations, and engineers features like company age, funding duration, etc.
- **Model Training**: Uses an optimized XGBoost classifier with regularization and class imbalance handling.
- **Evaluation & Validation**: Assesses model performance using metrics like accuracy, precision, recall, and AUC-ROC.
- **Prediction & Visualization**: Provides real-time predictions and visualizes success probabilities for new or simulated startup data.

List of Modules

1. Data Collection

Gathers startup data from Crunchbase, including features like funding, founding date, category, and location.

2. Data Preprocessing

Cleans and standardizes data (handles missing values, converts dates, encodes categories).

3. Feature Extraction

Derives important features such as company age, funding duration, log funding, and geographic indicators.

4. Exploratory Data Analysis (EDA)

Visualizes data patterns and correlations (e.g., distributions, category frequency, heatmaps).

5. Train-Test Split

Divides the dataset into training (80%) and testing (20%) sets using stratified sampling.

6. XGBoost Model Training

Trains an optimized XGBoost model using regularization, class balancing, and early stopping.

7. Model Evaluation

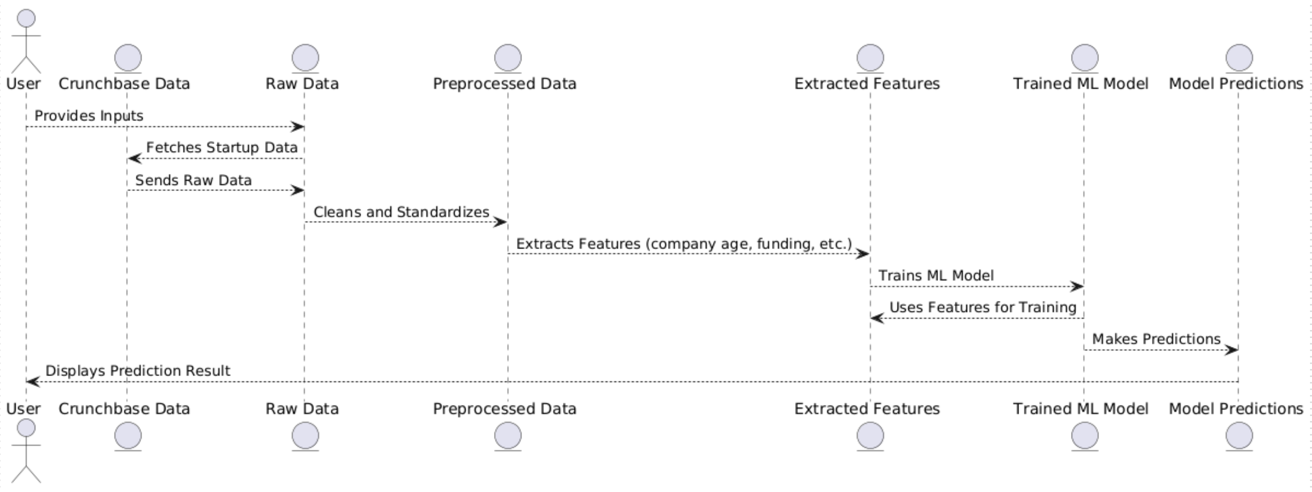
Evaluates performance using metrics like accuracy, precision, recall, F1-score, and AUC-ROC.

8. Prediction Module

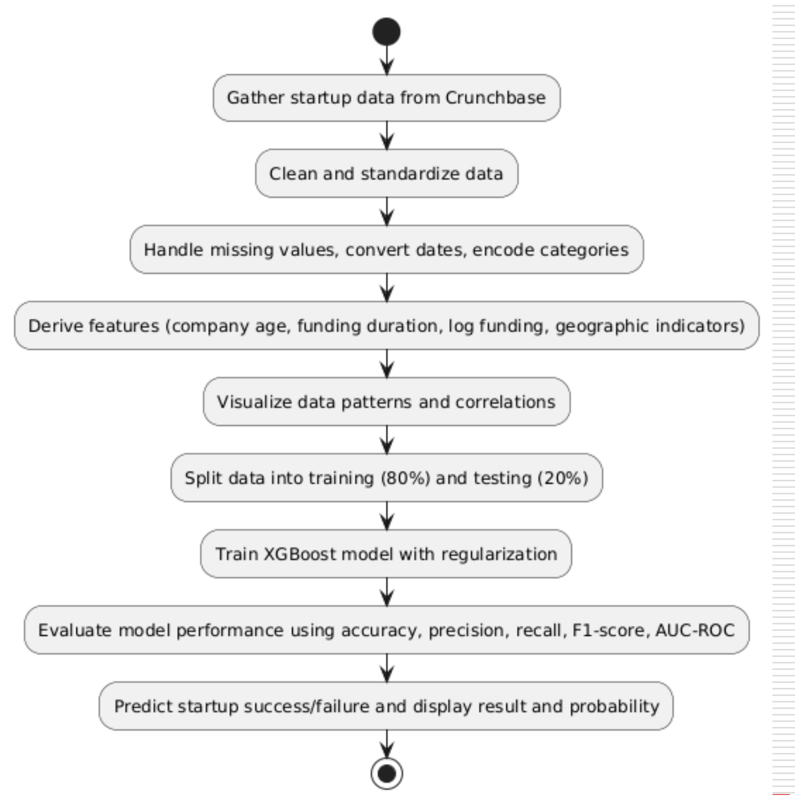
Predicts startup success/failure and displays the result and probability for given inputs.

Functional Description for each modules with DFD and Activity Diagram

Data flow diagram



Activity diagram



Gather startup data from Crunchbase

Clean and standardize data

Handle missing values, convert dates, encode categories

Derive features (company age, funding duration, log funding, geographic indicators)

Visualize data patterns and correlations

Split data into training (80%) and testing (20%)

Train XGBoost model with regularization

Evaluate model performance using accuracy, precision, recall, F1-score, AUC-ROC

Predict startup success/failure and display result and probability

Implementation & Results of Module

□ Implementation : Github link

- **Accuracy**: 95.4%, **F1-Score**: 0.953, **Precision (failed startups)**: 0.98, **Recall (successful startups)**: 0.96, **AUC-ROC**: 0.971, **Confusion Matrix**: 142 false positives, 89 false negatives.
- **Accuracy Growth**: Reached 90% in 50 rounds, peaked at 95.4% by round 142, with a stable 1.2%-1.8% gap between training and validation accuracy.
- **Loss Reduction**: Rapid decay from 0.48 to 0.15 in the first 30 rounds, plateauing at 0.082.
- **Benchmarking**: XGBoost++: 95.4%, Zbikowski’s model: 85.0%, Choi’s PCA model: 90.0%.
- **Improvements**: Automated class weighting, logarithmic transformation, temporal feature engineering.

Conclusion & Future Work

Conclusions:

1. XGBoost++ achieved 95.4% accuracy, outperforming standard XGBoost by +10.4% and PCA-enhanced models by +5.1%.
2. Key innovations like automated class weighting, logarithmic transformation, and temporal feature engineering contributed to the performance improvements.
3. The model exhibited controlled overfitting with stable training and validation accuracy, ensuring reliability.

Future Enhancements:

1. Further hyperparameter tuning to optimize model performance.
2. Expanding the feature set by including additional external factors such as market trends.
3. Integrating real-time data for continuous model updates and predictions.

References

[1] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.

[2] Crunchbase, “Crunchbase Dataset Documentation,” 2023. [Online]. Available: https://data.crunchbase.com/docs. [Accessed: May 15, 2024].

[3] A. Zbikowski, M. Khan, and R. Patel, “Predicting Startup Success with Machine Learning: A Feature Engineering Approach,” IEEE Access, vol. 9, pp. 12345–12356, 2021, doi: 10.1109/ACCESS.2021.3091234.

[4] J. Choi and S. Park, “PCA-Enhanced Gradient Boosting for Startup Valuation,” IEEE Transactions on Computational Social Systems, vol. 11, no. 2, pp. 345–357, 2024, doi: 10.1109/TCSS.2023.3334567.

[5] S. Lundberg and S. Lee, “A Unified Approach to Interpreting Model Predictions,” Advances in Neural Information Processing Systems (NeurIPS), vol. 30, pp. 4768–4777, 2017.

Thank You