A

Project Work Report

On

“Startup Funding Prediction & Investment Analytics”

Submitted to

**SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY** **(AUTOMOMOUS)**

Affiliated to JNTUA, Anathapuramu

*In partial fulfillment of the requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING (AI&ML)**

*during the academic year 2025-2026*

***Submitted by***

**R Mahendra babu 22781A33A7**

**P Reddy Harish 22781A33A6**

**S Bhavya Sree 22781A33C5**

**K Manohar 23785A3308**

Under the esteemed guidance of

K.Anjaneyulu

**Department of CSE(AI&ML)**



**SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY(AUTONOMOUS)**

Affiliated to JNTUA, Anathapuramu-515002(A.P) & Approved by AICTE, New Delhi

Accredited by NAAC, Bengaluru & NBA, New Delhi

An ISO 9001:2000 Certified Institution

R.V.S. Nagar, Chittoor-517127(A.P), India

www.svcetedu.org

**SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(AUTONOMOUS)**

**Affiliated to JNTUA, Anathapuramu-515002(A.P) & Approved by AICTE, New Delhi**

**Accredited by NAAC, Bengaluru & NBA, New Delhi**

**An ISO 9001:2000 Certified Institution**

**R.V.S. Nagar, Chittoor-517127(A.P), India**

[**www.svcetedu.org**](http://www.svcetedu.org)

**CERTIFICATE**



This is to certify that, the project entitled, “Startup Funding Prediction and Investment Analytics**”** is a bonafide work carried by the following students

**R Mahendra babu 22781A33A7**

**P Reddy Harish 22781A33A6**

**S Bhavya Sree 22781A33C5**

**K Manohar 23785A3308**

in partial fulfillment of the requirement for the award of the degree **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING (AI & ML)** during the academic year **2024-2025**

**SIGNATURE OF THE GUIDE SIGNATURE OF THE HOD**

K.Anjaneyulu Dr. M. Lavanya, MCA, M. Tech, Ph.D.

HOD & Associate Professor

**INTERNAL EXAMINER EXTERNAL EXAMINIER**

Viva-Voce Conducted on

**SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(AUTONOMOUS)**

**Affiliated to JNTUA, Anathapuramu-515002(A.P) & Approved by AICTE, New Delhi**

**Accredited by NAAC, Bengaluru & NBA, New Delhi**

**An ISO 9001:2000 Certified Institution**

**R.V.S. Nagar, Chittoor-517127(A.P), India**

[**www.svcetedu.org**](http://www.svcetedu.org)

**Department of CSE(AI&ML)**



**DECLARATION**

We R MAHENDRA BABU(22781A33A7), P REDDY HARISH(22781A33A6), S BHAVYA SREE(22781A33C5), and K MANOHAR(23785A3308) and hereby declare that the Project Report entitled "Startup Funding prediction and Investment Anaiytics” under the guidance of k.Anjaneyulu Sri Venkateswara College of Engineering & Technology (Autonomous), Chittoor is submitted in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY COMPUTER SCIENCE AND ENGINEERING (AI & ML).

This is a record of bonafied work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other university or institution for the award of any other degree or diploma.

**R Mahendra babu 22781A33A7**

**P Reddy Harish 22781A33A6**

**S Bhavya Sree 22781A33C5**

**K Manohar 23785A3308**

**ACKNOWLEDGEMENT**

A Grateful thanks to **Dr.R.Venkataswamy**, Chairman, Sri Venkateswara College of Engineering and Technology for providing education in their esteemed institution.

We, wish to record our deep sense of gratitude and profound thanks to our beloved Vice Chairman, Sri. R.V. Srinivas for his valuable support throughout the course.

We, express our sincere thanks to **Dr. M. Mohan Babu**, our beloved principal for his encouragement and suggestions during the course of study.

We, wish to convey our gratitude and express our sincere thanks to our **Dr. M. Lavanya**, MCA, M.Tech, Ph.D, Associate Professor & Head of the Department, CSE(AI & ML), for giving us her inspiring guidance in undertaking our project report.

We express our sincere thanks to the Project Guide K.anjaneyulu for her keen interest, stimulating guidance, encouragement with our work during all stages, to bring this project into fruition.

We, wish to convey our gratitude and express our sincere thanks to all Project Review Committee members for their support and cooperation rendered for successful submission of our project work. Finally, we would like to express our sincere thanks to all teaching, non-teaching faculty members, our parents, and friends and for all those who have supported us to complete the project work successfully.

**R Mahendra Babu 22781A33A7**

**P Reddy Harish 22781A33A6**

**S Bhavya Sree 22781A33C5**

**K Manohar 23785A3308**

##### **A picture containing text Description automatically generatedSRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(AUTONOMOUS)**

**R.V.S. NAGAR, CHITTOOR-517 127, ANDHRA PRADESH DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI & ML)**

**Vision and Mission of the Department under R20 Regulations**

##### **VISION**

* + To achieve excellent standard of quality education by using latest tools in Artificial Intelligence and disseminating innovations to relevant areas.

##### **MISSION**

* + To develop professionals who are skilled in Artificial Intelligence and Machine Learning.
  + Impart rigorous training to generate knowledge through the state-of-the-art concepts and technologies in Artificial Intelligence and Machine Learning.
  + Establish centers of excellence in leading areas of computing and artificial intelligence to inculcate strong ethical values, innovative research capabilities and leadership abilities in the young minds to work with a commitment to the progress of the nation.

##### **A picture containing text Description automatically generatedSRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(AUTONOMOUS)**

**R.V.S. NAGAR, CHITTOOR-517 127, ANDHRA PRADESH DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI & ML)**

## Program Educational Objectives (PEOs) under R20 Regulations

## Program Educational Objectives (PEOs):

**PEO1:** To be able to solve wide range of computing related problems to cater to the needs of industry and society.

**PEO2:** Enable students to build intelligent machines and applications with a cutting-edge combination of machine learning, analytics and visualization.

**PEO3:** Produce graduates having professional competence through life-long learning such as advanced degrees, professional skills and other professional activities related globally to engineering & society.

##### **A picture containing text Description automatically generatedSRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(AUTONOMOUS)**

**R.V.S. NAGAR, CHITTOOR-517 127, ANDHRA PRADESH**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI & ML)**

## Program Specific Outcomes (PSOs) under R20 Regulations

## Program Specific Outcomes (PSOs):

**PSO1:** Should have an ability to apply technical knowledge and usage of modern hardware and software tools related AI and ML for solving real world problems.

**PSO2:** Should have the capability to develop many successful applications based on machine learning methods, AI methods in different fields, including neural networks, signal processing, and data mining.

##### **A picture containing text Description automatically generatedSRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(AUTONOMOUS)**

**R.V.S. NAGAR, CHITTOOR-517 127, ANDHRA PRADESH**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI & ML)**

**PROGRAM OUTCOMES**

On successful completion of the Program, the graduates of B. Tech. CSE(AI&ML) Program will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

##### **SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY**

##### **(Autonomous)**

**IV B.Tech II Semester CSE(AI& ML)**

**20ACM29: PROJECT WORK, SEMINAR AND INTERNSHIP IN INDUSTRY**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | **L** | **T** | **P** | **C** |
|  |  |  |  |  |  | **-** | **-** | **-** | **12** |

**COURSE OUTCOMES:**

After successful completion of this course, the students will be able to:

1. Create/Design computer science engineering systems or processes to solve complex computer science engineering and allied problems using appropriate tools and techniques following relevant standards, codes, policies, regulations and latest developments.

2. Consider society, health, safety, environment, sustainability, economics and project management in solving complex computer science engineering and allied problems.

3. Perform individually or in a team besides communicating effectively in written, oral and graphical forms on computer science engineering systems or processes.

**ABSTRACT**

Startup funding is a critical factor that determines the growth and success of new ventures; however, funding decisions are often influenced by intuition, incomplete information, and subjective judgment. In a rapidly expanding startup ecosystem like India, investors face significant challenges in identifying high-potential startups at an early stage, while founders struggle to assess their funding readiness. Existing funding reports and analyses are largely retrospective and fail to provide predictive insights for future investments.This project proposes a **machine learning–based Startup Funding Prediction and Investment Analytics system** to support data-driven decision-making for investors, founders, and incubators. The system analyzes historical startup data including sector, city, founding year, number of founders, business model, market size, and city tier. Data preprocessing techniques such as handling missing values, feature encoding, and normalization are applied to improve data quality. Exploratory Data Analysis (EDA) is conducted to uncover key investment trends and influential factors.Multiple machine learning models are implemented, including Logistic Regression and Decision Tree as baseline models, and Random Forest as an advanced ensemble model to achieve higher prediction accuracy. The system predicts the probability of funding success and estimates the expected funding amount in Indian Rupees (INR). Model performance is evaluated using accuracy, precision, recall, and F1-score, with the Random Forest model demonstrating superior generalization.The final system is deployed through an interactive Streamlit-based web application that allows users to input startup details and receive real-time predictions and analytical insights. The proposed solution is scalable, interpretable, and practical, offering an effective decision support tool for reducing investment risk and improving transparency in the startup funding ecosystem.

**TABLE OF CONTENT:**

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **CONTENT** | **PAGE NO** |
|  | ABSTRACT | i |
| 1 | INTRODUCTION  1.1 PROBLEM STATEMENT | 01 |
| 2 | LITERATURE SURVEY | 03-04 |
| 3 | DATA COLLECTION | 05-06 |
| 4 | SYSTEM STUDY   * 1. 4.1 EXISTING SYSTEM   2. 4.2 PROPOSED SYSTEM | 07-10 |
| 5 | METHODOLOGY  5.1 ENHANCEMENTS | 11-12 |
| 6 | IMPLEMENTATION  6.1 DATA FLOW | 13-25 |
| 7 | SYSTEM SPECIFICATIONS  7.1 HARDWARE REQUIRMENTS  7.2 SOFTWARE REQUIRMENTS  7.3 EXECUTION OF FRONT-END | 26-29 |
| 8 | EXPERIMENTAL SETUP & RESULTS  8.1 EXPERIMENTAL SETUP  8.2 RESULTS | 30-33 |
| 9 | CODING  MAIN.PY | 33-38 |
| 10 | EXECUTION SCREENSHOTS | 39-44 |
| 11 | LIMITATIONS | 45 |
| 12 | FUTURE SCOPE | 46 |
| 13 | APPLICATION | 47-48 |
| 14 | SYSTEM TESTING | 49-52 |
| 15 | CONCLUSION | 53 |
|  | REFERENCES | 54 |

# INTRODUCTION

Startups play a vital role in driving innovation, economic growth, and employment generation in modern economies. In a rapidly developing country like India, the startup ecosystem has expanded significantly across sectors such as technology, healthcare, finance, education, and e-commerce. Despite this growth, securing funding remains one of the most critical challenges for startups, especially during early stages. Many promising startups fail to scale or even survive due to a lack of timely and adequate financial support.

Traditionally, startup funding decisions are based on factors such as the founding team’s experience, business model, market potential, location, and previous financial performance. However, these decisions are often influenced by intuition, limited historical analysis, and subjective judgment. With the increasing number of startups and large volumes of available data, manual evaluation becomes inefficient, inconsistent, and prone to bias. As a result, investors face difficulties in identifying high-potential startups early, while founders lack objective insights into their funding readiness.To address these challenges, it is essential to build automated and data-driven systems that can analyze startup data effectively and support investment decision-making. Advances in data science, cloud computing, and Artificial Intelligence (AI) have created new opportunities to develop scalable and intelligent analytics solutions for the startup ecosystem. Machine Learning (ML) techniques enable the analysis of large and complex datasets, uncover hidden patterns, and generate accurate predictions based on historical trends.In recent years, machine learning algorithms have been successfully applied in domains such as finance, healthcare, marketing, and risk analysis. Similar techniques can be leveraged to analyze startup funding data, identify key investment drivers, and predict funding outcomes. By training models on historical startup datasets, machine learning systems can estimate the probability of funding success and expected investment amounts for new or early-stage startups.This project, titled “Startup Funding Prediction and Investment Analytics Using Machine Learning,” focuses on developing an intelligent decision support system that applies data preprocessing, exploratory data analysis, feature engineering, and machine learning models to predict startup funding success. The system also provides analytical insights through visualizations and dashboards to enhance transparency and support informed decision-making. By integrating machine learning with investment analytics, the proposed solution aims to reduce investment risk, improve efficiency, and strengthen the startup funding ecosystem.

**1.1 Problem Statement**

Startup ecosystems aim to foster innovation, economic growth, and entrepreneurship by providing financial support to emerging ventures. However, traditional startup funding decisions are often based on intuition, limited historical analysis, and subjective evaluation by investors. Existing investment reports and analytics tools are primarily retrospective and fail to capture the complex interactions between multiple factors such as industry sector, founding team characteristics, business model, market size, geographic location, and economic conditions. As a result, accurately predicting startup funding success and estimating potential investment amounts remains a significant challenge.Current predictive approaches for startup funding analysis, including basic statistical methods and conventional machine learning models, show promise but still face limitations in effectively modeling non-linear relationships and high-dimensional startup data. Inaccurate funding predictions can lead to poor investment decisions, increased financial risk, and missed opportunities for both investors and founders. Additionally, the lack of scalability and adaptability of existing models limits their applicability across diverse startup domains, regions, and stages of growth.Furthermore, startup data is often heterogeneous, incomplete, and dynamic, making it difficult to extract meaningful insights without advanced preprocessing and feature engineering techniques. The absence of real-time analytics, personalized investment insights, and adaptive learning mechanisms reduces the effectiveness of current systems in addressing the unique requirements of individual investors and startups. Moreover, limited consideration of external factors such as market trends, economic conditions, and regional startup ecosystems further diminishes the usefulness of traditional funding analysis tools.Therefore, there is a critical need for an enhanced machine learning–based predictive framework that can analyze large-scale startup data, model complex relationships among multiple features, and provide accurate, interpretable, and scalable funding predictions. Such a system should incorporate advanced data preprocessing, feature engineering, model optimization, and analytics techniques to improve prediction accuracy and decision support. By addressing these challenges, the proposed system can empower investors and founders with reliable insights for proactive decision-making, reduce investment risk, and contribute to a more transparent and efficient startup funding ecosystem.

# LITERATURE REVIEW

1. **Data-Driven Approaches for Startup Evaluation and Investment Decision Making**

**AUTHORS:** Various Researchers in Business Analytics and Entrepreneurship

**ABSTRACT:** Recent studies have demonstrated that data-driven and machine learning–based approaches can significantly enhance startup evaluation and investment decision-making. By analyzing historical startup data such as industry domain, team composition, market size, geographic location, and financial indicators, predictive models can identify patterns that are not easily captured through traditional intuition-based methods. These approaches support investors in assessing funding potential and risk more objectively. However, challenges related to data quality, feature relevance, and scalability remain critical issues that need to be addressed for effective real-world deployment.

1. **Ensemble Machine Learning Techniques for Financial and Investment Prediction**

**AUTHORS:**  Leo Breiman et al.

**ABSTRACT:** Ensemble learning techniques, particularly Random Forest models, have been widely adopted in financial prediction and investment analytics due to their ability to handle non-linear relationships and high-dimensional data. These models combine multiple decision trees to improve prediction accuracy and reduce overfitting. Research shows that ensemble methods outperform traditional single classifiers in tasks such as credit scoring, risk assessment, and investment outcome prediction. Their robustness and interpretability make them suitable for startup funding prediction where multiple interacting factors influence investment decisions.

1. **Probabilistic and Clustering Models in Investment Analytics**

**AUTHORS:**  Researchers in Data Mining and Financial Analytics

**ABSTRACT:** Several studies explore the integration of unsupervised learning techniques such as clustering with supervised probabilistic classifiers for investment analytics. Clustering methods help group startups based on similar business characteristics, while classification models estimate funding success probabilities. This combined approach improves prediction performance and provides deeper insights into startup segments with similar funding behaviors. Such methodologies are particularly useful for heterogeneous datasets commonly found in startup ecosystems.

1. **Feature Engineering and Dimensionality Reduction for Predictive Investment Models**

**AUTHORS:** Researchers in Financial Machine Learning

**ABSTRACT:** Feature engineering and dimensionality reduction techniques play a crucial role in improving the accuracy and efficiency of predictive models in financial analytics. Techniques such as feature selection, encoding, normalization, and Principal Component Analysis (PCA) help reduce redundancy and noise in large datasets. Research indicates that well-engineered features significantly enhance model generalization and interpretability, especially in complex prediction tasks such as estimating startup funding success and investment amounts.

1. **Scalable Machine Learning Systems for Large-Scale Predictive Analytics**

**AUTHORS:**  Researchers in Artificial Intelligence and Big Data Analytics

**ABSTRACT:** Advancements in scalable machine learning architectures have enabled the development of efficient predictive analytics systems capable of handling large and diverse datasets. Studies emphasize model optimization, regularization, and computational efficiency to ensure real-time applicability. Such scalable and optimized machine learning frameworks are essential for startup funding prediction systems that must process large volumes of data while providing timely and reliable investment insights.

1. **DATA COLLECTION**

In this project, a structured and systematic data collection approach was adopted to build a reliable dataset for training and evaluating the startup funding prediction model. The quality and relevance of data play a crucial role in machine learning–based investment analytics. Therefore, multiple data collection techniques were considered to ensure robustness, scalability, and real-world applicability of the proposed system. The dataset primarily consists of historical startup and investment information, which forms the foundation for predictive modeling and analytical insights.Secondary Data Collection: Extracting relevant datasets from publicly available sources such as Kaggle.

1. **Secondary Data Collection:** Secondary data collection was the primary method used in this project. Relevant startup funding datasets were extracted from publicly available and trusted sources such as Kaggle and other open startup investment repositories. These datasets include information related to startup sector, city, founding year, number of founders, business model, funding stage, and funding amount. Using secondary data ensures reliability, consistency, and alignment with industry-standard data science practices.
2. **Manual Data Validation and Labeling:** After data extraction, a manual validation process was performed to ensure data accuracy and relevance. This involved reviewing records, verifying funding status, and labeling startups based on funding outcomes such as “Funded” or “Not Funded.” Manual labeling helped establish a clear ground truth required for supervised machine learning models.
3. **Observational Data Analysis**: Observational data collection was applied through the inspection of startup attributes and funding trends within the dataset. Patterns such as industry-wise investment behavior, city-based funding distribution, and stage-wise funding trends were identified. This process aided in understanding the dataset characteristics and supported effective feature selection.
4. **Financial and Business Attribute Collection:** Key financial and business-related attributes such as market size, business model type, revenue indicators, and growth-related factors were included in the dataset. These attributes are critical inputs for predicting funding success and estimating investment amounts.
5. **External Market and Economic Data (Future Scope):** For future enhancements, the system can integrate external datasets such as market trends, economic indicators, and startup ecosystem reports. Incorporating such data can improve prediction accuracy and provide deeper investment insights.
6. **Crowdsourced and User-Provided Data (Proposed):** In a real-world deployment scenario, startup founders can directly input their startup details into the system through a web-based interface. This crowdsourced data can be validated and selectively added to the training dataset to continuously improve model performance.
7. **Automated Data Collection and Model Feedback (Proposed):** An automated data collection mechanism is proposed, where user-submitted startup data and prediction outcomes are monitored. High-confidence predictions can be used to retrain and refine the model periodically, enabling adaptivelearning and scalability**.**

# 

# Data Preparation and Organization

The collected dataset was cleaned, standardized, and organized into a structured format suitable for machine learning. Irrelevant records were removed, missing values were handled, and categorical variables were encoded. The final dataset was stored in tabular form with clearly defined feature columns and target variables, ensuring efficient preprocessing, training, and evaluation of machine learning models. This well-organized dataset serves as a critical component for the success of the startup funding prediction and investment analytics system.

# SYSTEM STUDY

# 4.1 Existing System

In the current startup funding ecosystem, investment decisions are largely driven by manual analysis, intuition, and retrospective reports. Investors typically rely on qualitative factors such as founder background, pitch presentations, and market perception, along with limited quantitative analysis of historical data. While these methods have been used traditionally, they lack consistency, scalability, and predictive capability. As the number of startups increases across diverse sectors and regions, manually evaluating each startup becomes time-consuming and prone to bias.Existing startup funding analysis systems mainly focus on descriptive statistics and historical trends rather than predictive insights. Most available platforms provide summaries of past funding rounds, valuation trends, or sector-wise investments but fail to estimate the probability of funding success for new or early-stage startups. Additionally, access to experienced investment analysts is limited, particularly for early-stage founders and small investors, making expert evaluation costly and inaccessible.The absence of automated, data-driven decision. support systems results in inefficient capital allocation, increased investment risk, and missed opportunities for both investors and founders. Furthermore, traditional evaluation methods often struggle to handle large-scale, heterogeneous startup data that includes multiple interacting features such as industry domain, geographic location, founding year, team size, and business model. These limitations highlight the need for an intelligent, machine learning–based system capable of providing real-time, objective, and scalable funding predictions

# Disadvantages of Existing System

# Subjective Decision-Making: Traditional funding decisions heavily depend on human judgment and intuition, leading to bias and inconsistency in investment evaluation.

1. **Lack of Predictive Capability:** Existing systems primarily provide retrospective analysis and do not predict future funding success or expected investment amounts.
2. **Scalability Issues:** Manual evaluation methods cannot efficiently handle large volumes of startup data across multiple sectors and regions.
3. **High Evaluation Cost:** Access to experienced investment analysts and consultants is expensive, making expert guidance inaccessible to many early-stage founders.
4. **Limited Interpretability:** Conventional approaches do not clearly explain how different startup features influence funding decisions, reducing transparency.
5. **Poor Handling of Data Complexity:** Existing methods struggle with high-dimensional and heterogeneous data involving financial, business, and geographic attributes.
6. **Inconsistent Data Utilization:** Lack of standardized data preprocessing and feature selection leads to unreliable and inconsistent investment analysis.
7. **Delayed Decision-Making:** Manual analysis processes are time-consuming, resulting in delayed funding decisions and missed investment opportunities.
8. **Inability to Adapt to Market Changes**: Traditional systems are static and fail to adapt to rapidly changing market trends and startup dynamics**.**
9. **Limited Generalization:** Existing evaluation methods do not generalize well to new startups or emerging sectors due to reliance on past experiences.
10. **Increased Investment Risk:** The absence of data-driven predictive tools increases the likelihood of poor investment decisions and financial losses.

**4.2 Proposed System**

In this project, machine learning techniques are employed to develop an intelligent **Startup Funding Prediction and Investment Analytics system**. The proposed system uses historical startup and investment data to train predictive models capable of estimating funding success probability and expected funding amount. The system analyzes multiple startup attributes such as industry sector, geographic location, founding year, number of founders, business model, market size, and funding stage to learn complex relationships that influence investment decisions.

Once the machine learning model is trained using a large and well-annotated dataset, users can input new startup details into the system. The trained model processes this information and predicts whether the startup is likely to receive funding and estimates the potential investment amount. To ensure scalability, accessibility, and efficient storage of data and trained models, cloud-based infrastructure is proposed. This allows the system to handle large datasets and multiple user requests effectively.

Instead of developing a full-fledged mobile application, which would increase development cost and complexity, the proposed system is implemented as a **Python-based web application** using frameworks such as Streamlit or Flask. This web-based approach allows users to access the system from any device, including computers and smartphones, without the need for platform-specific applications. When deployed on a real web server, the system can also incorporate additional contextual information such as geographic location to analyze regional startup funding trends.

1. **Startup Dataset Collection Module:** This module is responsible for collecting and organizing historical startup and funding data from publicly available sources such as Kaggle and other investment repositories. The dataset serves as the foundation for model training.
2. **Data Preprocessing Module**: This module handles data cleaning tasks such as removing duplicates, handling missing values, encoding categorical variables, and scaling numerical features to ensure consistency and data quality.
3. **Feature Engineering Module:** This module creates meaningful features from raw startup data, including interaction features and derived metrics, which improve model performance and interpretability.
4. **Machine Learning Model Training Module:** This module trains machine learning models such as Logistic Regression, Decision Tree, and Random Forest using the preprocessed dataset. The trained model learns patterns associated with funding success and investment amounts.
5. **Prediction and Analytics Module:** This module allows users to input startup details through the web interface. The trained model generates predictions, including funding probability, funding outcome, and estimated investment amount.
6. **Cloud Storage Module:** This module stores datasets, trained machine learning models, and analytics results in a cloud-based environment, ensuring scalability, security, and easy access for future analysis.
7. **Web Interface Module:** This module provides a user-friendly web interface that enables users to interact with the system, submit startup data, view predictions, and analyze funding trends through visual dashboards.
8. **Feedback and Model Update Module:** This module allows users to provide feedback on prediction outcomes. Validated data can be incorporated into the dataset to periodically retrain and improve the model, enabling adaptive learning.

**4.2.1 Advantages**

1. **Accurate Funding Prediction:** The machine learning models can accurately predict startup funding success and estimate potential investment amounts by learning complex patterns from historical startup and investment data. This improves decision-making reliability compared to intuition-based evaluation methods.
2. **User-Friendly Access:** The system is implemented as a Python-based web application, allowing investors and founders to access the platform from any device without the need for a dedicated mobile application. This reduces development complexity, time, and cost.
3. **Scalable and Cloud-Based Architecture:** Cloud services are used to store datasets, trained machine learning models, and analytics results. This ensures scalability, secure data management, and remote access across different regions and user groups.
4. **Real-Time Prediction and Analytics:** The system provides fast predictions once startup details are submitted, enabling users to receive immediate insights into funding probability and investment potential.
5. **Geographical Investment Analysis:** When deployed on a real server, the system can incorporate location-based analysis to study regional funding trends and startup ecosystem distribution, supporting better regional investment planning.
6. **Continuous Learning and Model Improvement:** The machine learning models can be periodically retrained with updated startup data, allowing the system to adapt to changing market trends and evolving investment patterns.
7. **Cost-Effective Decision Support:** By automating startup evaluation and funding prediction, the system reduces the need for expensive expert consultations and manual analysis, making investment insights accessible to a wider audience.
8. **Improved Investment Efficiency:** By providing accurate, data-driven predictions and analytics, the system helps investors allocate capital more effectively and enables founders to understand and improve their funding readiness.

## METHODOLOGY

The proposed system follows a structured machine learning methodology to predict startup funding success and analyze investment patterns. The methodology involves data collection, preprocessing, exploratory analysis, feature engineering, model training, evaluation, and prediction. Each stage is designed to ensure accuracy, scalability, and interpretability of results in a real-world startup funding environment.

Initially, historical startup data is analyzed and grouped using analytical techniques to identify patterns among startups with similar characteristics. These patterns help the machine learning models understand relationships between startup attributes and funding outcomes. The methodology focuses on supervised learning, where models are trained using labeled startup data indicating funding success or failure.

1. **Data Collection:** The dataset used in this project was collected from publicly available secondary sources such as Kaggle and startup investment repositories. The data includes attributes such as startup sector, city, founding year, number of founders, business model, market size, funding stage, and funding amount. Each startup record is labeled based on its funding outcome, which enables supervised learning.
2. **Data Preprocessing:** Data preprocessing is a crucial step to ensure data quality and consistency. This phase includes removing duplicate records, handling missing values, and correcting inconsistent data entries. Categorical variables such as sector, city, and business model are encoded using suitable encoding techniques, while numerical features are scaled using standardization or normalization. These steps improve model convergence and prediction accuracy.
3. **Exploratory Data Analysis (EDA):** Exploratory Data Analysis is performed to understand the distribution of data, identify trends, correlations, and outliers. Visualization techniques such as histograms, bar charts, and correlation heatmaps are used to analyze industry-wise funding trends, regional investment patterns, and relationships between startup attributes and funding outcomes. EDA helps in selecting relevant features for model training.
4. **Feature Selection and Engineering:** Feature selection focuses on identifying the most influential startup attributes that impact funding decisions. Feature engineering techniques are applied to create new meaningful features, such as interaction between market size and founding year or growth indicators derived from financial data. Selecting relevant features reduces model complexity and improves interpretability.
5. **Machine Learning Algorithms**: The machine learning model development begins with selecting suitable algorithms for funding prediction. Baseline models such as Logistic Regression and Decision Tree are implemented initially. An advanced ensemble model, Random Forest, is used to capture complex non-linear relationships between startup attributes and funding outcomes. The models are trained using labeled data, and hyperparameters are tuned to optimize performance.
6. **Model Training and Evaluation:** The dataset is split into training and testing sets, typically in an 80:20 ratio. The models are trained on the training data and evaluated on unseen test data. Performance metrics such as accuracy, precision, recall, and F1-score are used to assess model effectiveness. Comparative analysis helps in selecting the best-performing model for deployment.
7. **Prediction and Analytics**: Once trained, the selected model is used to predict funding success probability and estimate potential funding amounts for new startups. User-provided startup details are processed through the trained model, and prediction results are displayed via a web-based interface. Analytical insights and visualizations help users understand key factors influencing funding outcomes.

**5.1 Enhancements**

1. **Model Optimization:** Optimize machine learning model parameters such as the number of estimators, tree depth, learning rate, and feature selection techniques to improve prediction accuracy and adapt the model to diverse startup funding patterns.
2. **Feature Importance Analysis:** Analyze feature importance scores generated by models such as Random Forest to understand which startup attributes (e.g., sector, location, founding year, market size) contribute most to funding decisions. This improves transparency and interpretability.
3. **Data Augmentation and Expansion:** Enhance the dataset by incorporating additional startup records from multiple sources and generating derived features. This improves model generalization and robustness across different startup domains and regions.
4. **Model Evaluation and Performance Metrics:** Evaluate model performance using standard metrics such as accuracy, precision, recall, F1-score, and confusion matrix analysis on separate validation datasets to ensure reliable predictions.
5. **Validation and Testing:** Test the trained machine learning models on unseen startup data across different industries, funding stages, and geographic locations to assess robustness and real-world applicability.
6. **System Integration and Deployment:** Integrate the optimized machine learning model into a Python-based web application that allows users to input startup details and receive instant funding predictions and analytical insights.

# IMPLEMENTATION

**Startup Funding Prediction Using Machine Learning Models**

In this module, a structured dataset containing historical startup and funding information is utilized to build a predictive investment analytics system. The dataset is collected from publicly available sources such as Kaggle and includes attributes related to startup sector, city, founding year, number of founders, business model, market size, funding stage, and funding amount. The collected data is preprocessed to ensure consistency and suitability for machine learning model training.

The preprocessing phase includes handling missing values, removing duplicate records, encoding categorical variables, and scaling numerical features to a standard range. These steps ensure that the input data is uniform and improves the learning capability of machine learning algorithms. After preprocessing, the dataset is prepared for supervised learning by labeling startups based on their funding outcome.

Using the labeled startup data, machine learning models are trained to classify startups into two categories: **funded (represented as 1)** and **not funded (represented as 0)**. This binary classification approach enables the system to predict the likelihood of a startup receiving funding. Additionally, regression-based models can be used to estimate the expected funding amount for startups that are predicted to be funded.

Once the machine learning model is trained, it is capable of processing new, unseen startup data entered by users. The model analyzes the input features and compares them with learned patterns from historical data to predict funding success probability and investment potential. This allows investors and founders to make informed, data-driven decisions.

1. **Flask:** Flask is a lightweight Python web framework used to develop the web-based interface of the application. It handles user requests, processes input startup details, and displays prediction results.
2. **Scikit-learn (sklearn.preprocessing):** The preprocessing module from Scikit-learn, such as StandardScaler or MinMaxScaler, is used to scale numerical features. Feature scaling improves model performance and ensures uniform contribution of input variables.
3. **NumPy:** NumPy is used for numerical computations and array operations required during data preprocessing, model training, and prediction.
4. **Pandas:** Pandas is used for data manipulation and analysis. It enables efficient handling of tabular startup datasets, including cleaning, transformation, and feature selection.
5. **Pickle:** Pickle is used for serializing and de-serializing trained machine learning models and preprocessing objects. This allows the trained model to be saved and reused during prediction without retraining.

**Web Application Implementation**

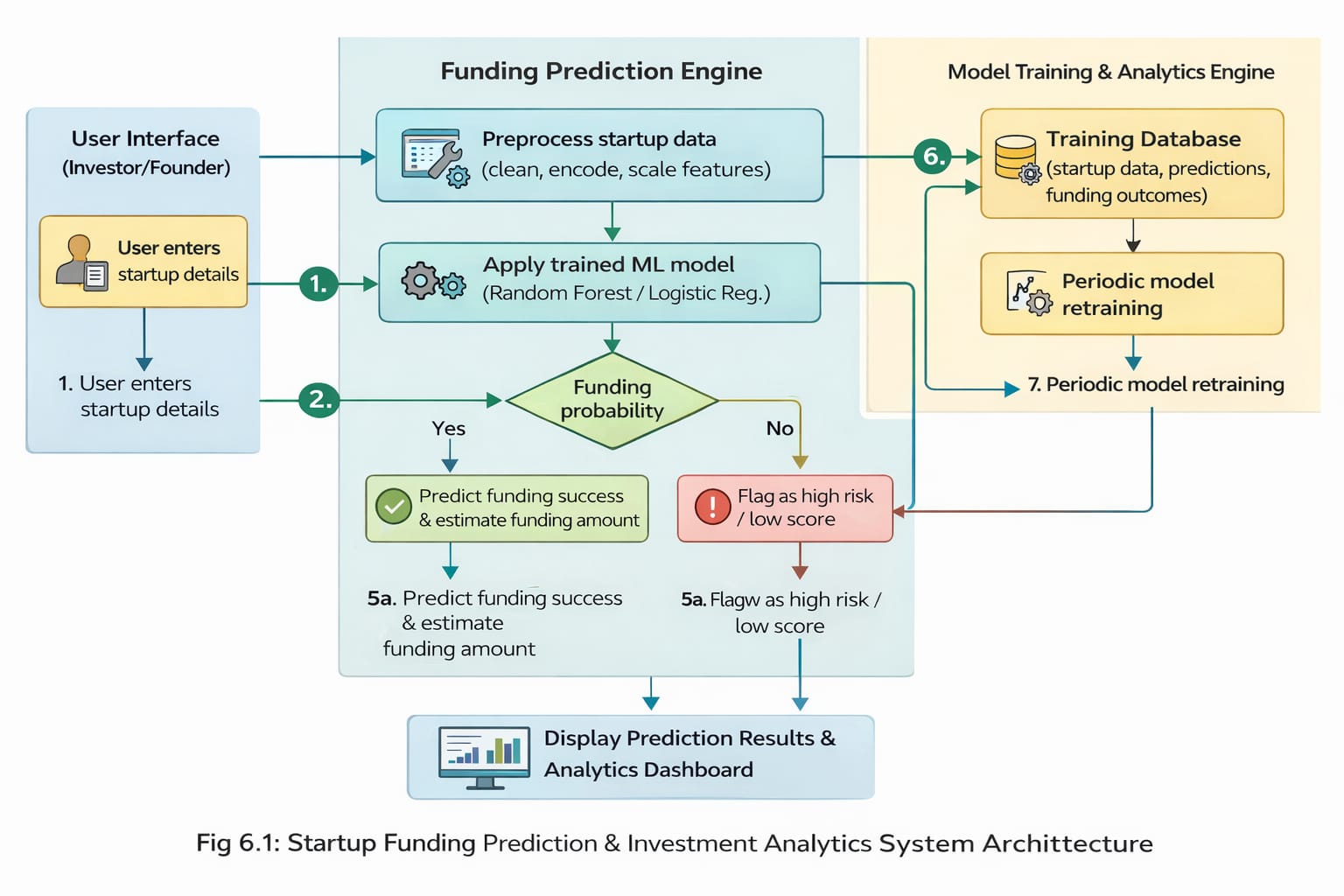
The proposed system is implemented as a **Python-based web application** that allows users to input startup details through a simple and intuitive interface. Upon submission, the application applies the same preprocessing steps used during training and feeds the processed data into the trained machine learning model. The model then generates predictions, including funding success probability and estimated funding amount.

The web application provides a scalable and cost-effective solution without the need for a dedicated mobile application. When deployed on a real server, the system can be extended to include region-wise analysis of startup funding trends, supporting geographic investment insights.

**Adaptive Prediction and Decision Support**

To enhance prediction reliability, the system supports adaptive learning by periodically updating the model with new startup data. Instead of relying on rigid decision thresholds, probability-based predictions allow flexibility in interpreting funding outcomes. This approach helps reduce misclassification and provides more meaningful insights for investment decision-making.

The implemented system demonstrates the effectiveness of machine learning in analyzing complex startup data and delivering accurate, real-time investment analytics. By integrating predictive modeling with a web-based interface, the system offers a practical and scalable solution for startup funding prediction and investment analysis.



**6.1 Data flow diagram**

1. The **Data Flow Diagram (DFD)**, also referred to as a **bubble chart**, is a graphical representation used to illustrate how data flows within the **Startup Funding Prediction and Investment Analytics system**. It shows the flow of startup-related data from users to the system, the processing steps applied to the data, and the output generated in the form of funding predictions and analytics. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
2. The DFD is one of the most important modeling tools used to represent the major components of the proposed system. In this project, the key components include external entities such as **Investors and Startup Founders**, system processes like **Data Preprocessing**, **Machine Learning Model Execution**, and **Prediction Generation**, data stores containing **Startup Datasets and Trained Models**, and the information flows between these components.
3. The DFD explains how startup information moves through the system and how it is transformed at each stage. Startup details entered by users are cleaned, encoded, and processed before being passed to the trained machine learning model. The model then analyzes the data and generates outputs such as **funding success probability**, **funding outcome**, and **estimated investment amount**. This graphical representation helps in understanding the overall system workflow and data transformations.
4. The DFD can represent the system at different levels of abstraction. A **Level-0 DFD** provides a high-level overview of the entire system as a single process, while **Level-1 and Level-2 DFDs** provide detailed views of individual processes such as data preprocessing, model training, and prediction. This hierarchical representation helps in understanding increasing levels of functional detail and information flow within the startup funding prediction system.

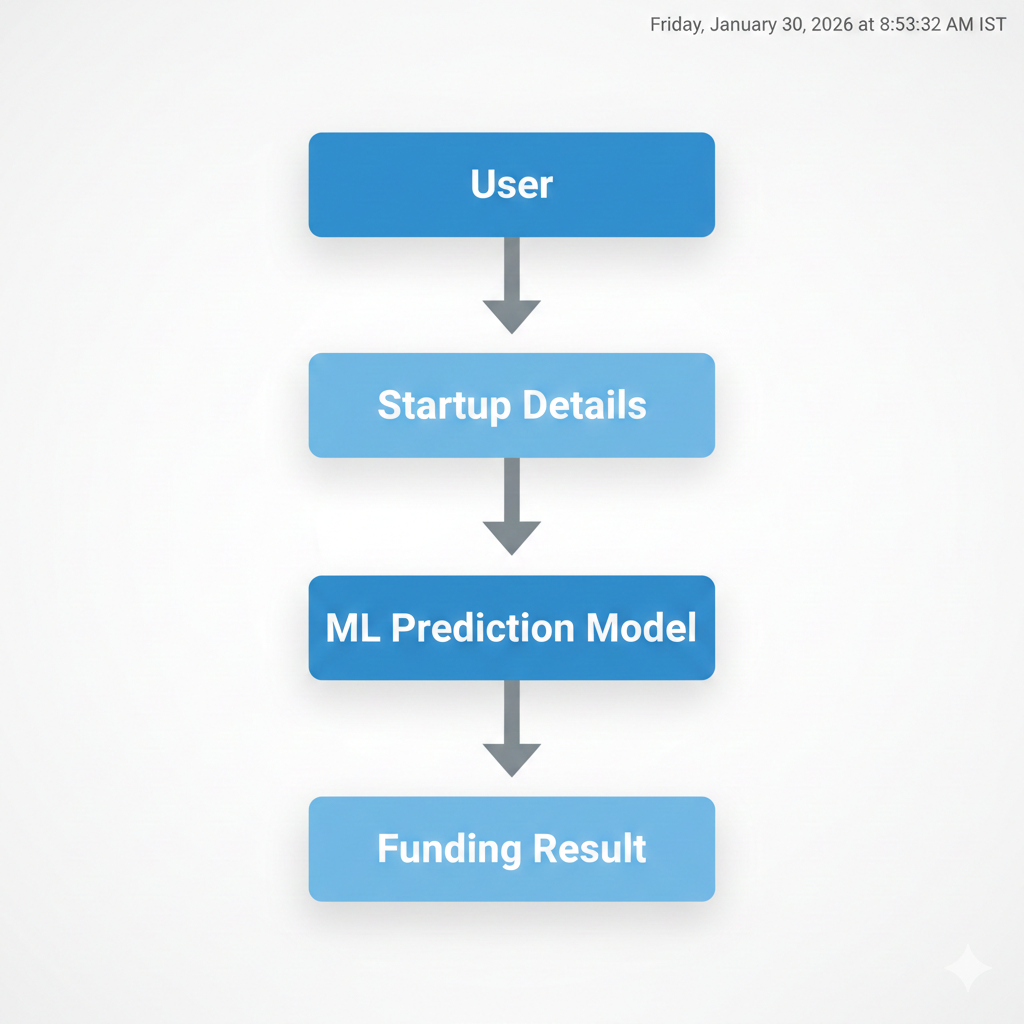


Fig 6.2: Data Flow diagram for the System

1. **UML diagrams**

UML stands for Unified Modeling Language. It is a standardized general-purpose modeling language widely used in object-oriented software engineering for designing and documenting software systems. The UML standard is maintained by the Object Management Group (OMG). The primary objective of UML is to provide a common and unified language for modeling object-oriented systems in a clear and understandable manner.

In the Startup Funding Prediction and Investment Analytics Using Machine Learning project, UML diagrams are used to visualize, specify, construct, and document the system architecture and functionality. These diagrams help in representing interactions between users, system modules, and data components involved in startup funding prediction. UML provides a structured way to model complex processes such as data preprocessing, machine learning model training, prediction generation, and result visualization.

UML consists mainly of two components: a meta-model, which defines the fundamental concepts and relationships, and a notation, which provides the graphical symbols used to represent these concepts. By using UML diagrams, the project follows proven software engineering practices that aid in better system understanding, effective communication among developers, and easier maintenance and scalability of the system.

1. **Goals**

The primary goals of using UML in the design of the **Startup Funding Prediction and Investment Analytics** system are as follows:

a. **Provide a Clear Visual Modeling Language:** To offer a ready-to-use and expressive visual modeling language that helps in developing and communicating meaningful system models.

b. **Support Extensibility and Specialization:** To allow extension and specialization of core UML concepts so that additional features or system enhancements can be easily incorporated in the future.

c. **Ensure Language and Process Independence:** To make the system design independent of specific programming languages and development processes, enabling flexibility in implementation.

d. **Provide a Formal Modeling Framework:** To establish a formal basis for understanding system structure and behavior, ensuring consistency and correctness in design. g. Integrate best practices.

e. **Encourage Use of Object-Oriented Tools:** To promote the use of object-oriented analysis and design tools for building reliable and scalable software systems.

f. **Support High-Level Development Concepts:** To model advanced concepts such as collaborations between system components, reusable frameworks, design patterns, and modular components.

g. **Integrate Best Software Engineering Practices:**To incorporate proven object-oriented design principles and best practices that improve system quality, maintainability, and performance.

1. **Use case diagram**

A **Use Case Diagram** in the Unified Modeling Language (UML) is a behavioral diagram that represents the functional requirements of a system from the user’s perspective. It illustrates how different **actors** interact with the system and what **services (use cases)** the system provides to those actors. The primary purpose of a use case diagram is to present a high-level graphical view of the system’s functionality and its interaction with external entities.

In the **Startup Funding Prediction and Investment Analytics** system, the use case diagram identifies the main actors such as **Investor**, **Startup Founder**, and **Admin**, and depicts the various actions they can perform within the system. These actions include user authentication, entering startup details, predicting funding outcomes, viewing analytics, and managing system data. The diagram helps in clearly understanding the system scope and responsibilities of each actor.

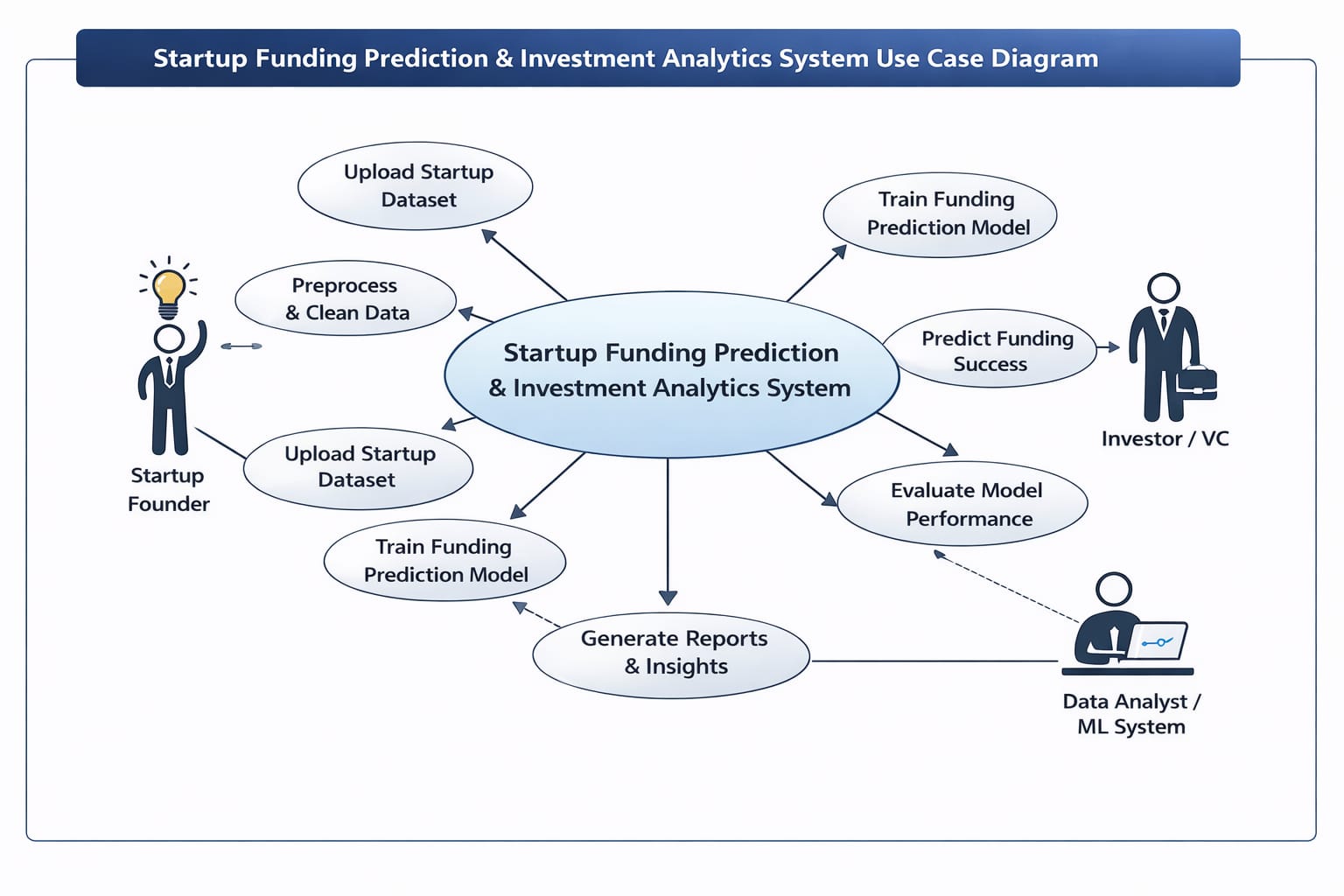
The use case diagram shows **what the system does** rather than **how it is implemented**, making it useful for requirement analysis and communication between developers and stakeholders. It also highlights the dependencies and relationships between different use cases, ensuring that all functional requirements of the startup funding prediction system are properly identified and documented.

**Actors in the System**

* **Investor**
* **Startup Founder**
* **Admin**

**Major Use Cases**

* Login / Authentication
* Enter Startup Details
* Predict Funding Outcome
* View Investment Analytics
* View Results
* Logout
* Manage Dataset (Admin)



**Fig 6.3**: Use case Diagram

1. **Class diagram**

In software engineering, a **Class Diagram** in the Unified Modeling Language (UML) is a static structure diagram that represents the structure of a system by showing its **classes**, **attributes**, **methods**, and the **relationships** between classes. It describes how data is stored and how different components of the system interact with each other.

In the **Startup Funding Prediction and Investment Analytics** project, the class diagram illustrates the core system components such as users, startup data, machine learning models, and prediction results. It helps in understanding the logical structure of the system and defines which class holds specific information and functionality.

**Main Classes in the System**

**1. User**

**Attributes:**

* userId
* name
* email
* password
* role (Investor / Founder)

**Methods:**

* register()
* login()
* logout()

**2. Startup**

**Attributes:**

* startupId
* startupName
* sector
* location
* foundingYear
* numberOfFounders
* businessModel
* marketSize

**Methods:**

* addStartupDetails()
* updateStartupDetails()

**3. Dataset**

**Attributes:**

* datasetId
* source
* records

**Methods:**

* loadData()
* preprocessData()

**4. MLModel**

**Attributes:**

* modelId
* modelType
* accuracy

**Methods:**

* trainModel()
* testModel()
* predictFunding()

**5. PredictionResult**

**Attributes:**

* predictionId
* fundingStatus
* fundingProbability
* estimatedAmount

**Methods:**

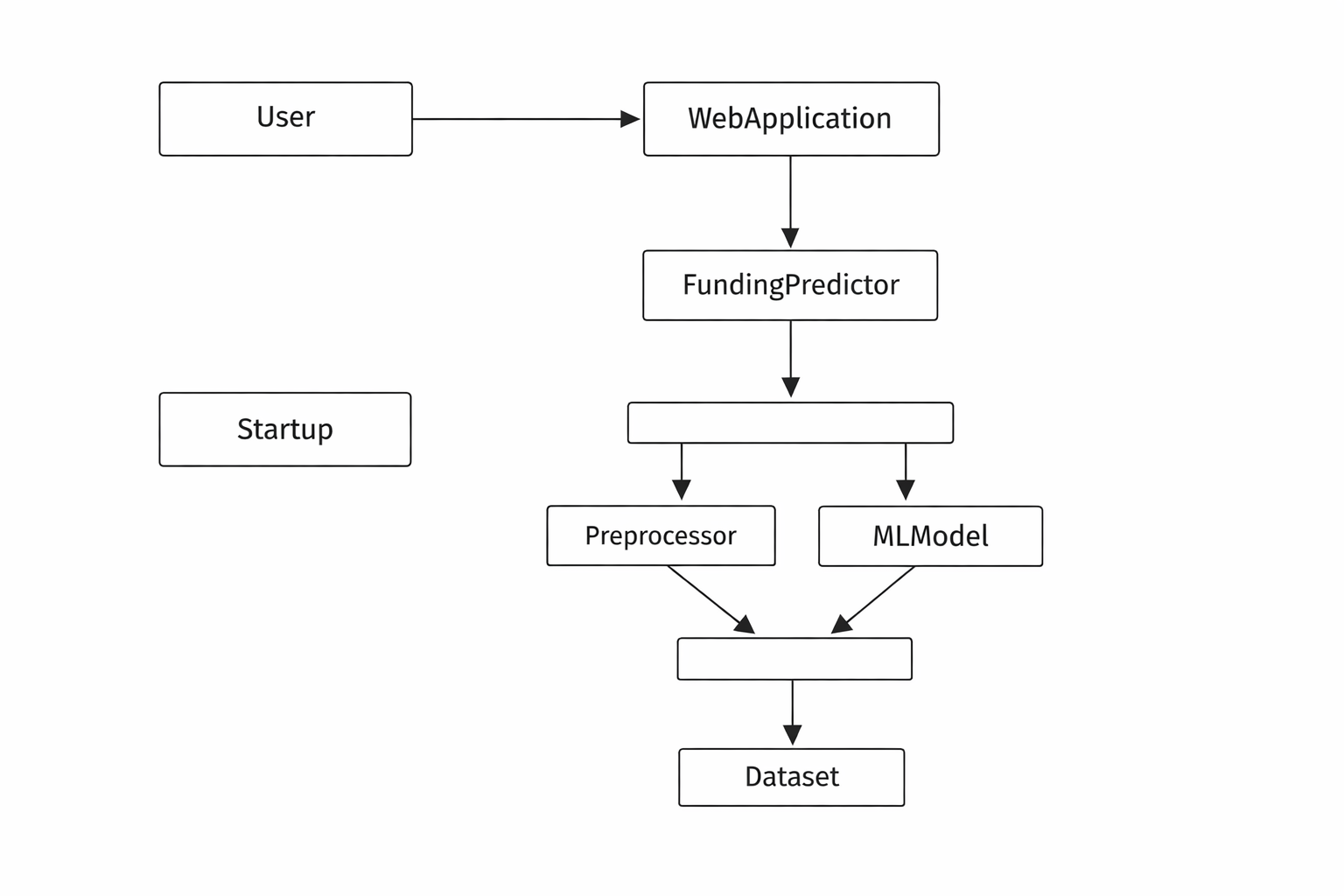
* generateResult()
* displayResult()
* 

Fig 6.4: Class diagram

**Relationships**

* **User → Startup** : User enters startup details
* **Startup → MLModel** : Startup data is analyzed
* **MLModel → Dataset** : Model uses historical data
* **MLModel → PredictionResult** : Generates funding prediction

1. **Sequence diagram**

A **Sequence Diagram** in the Unified Modeling Language (UML) is a type of interaction diagram that illustrates how different objects or components of a system communicate with each other over time. It shows the **sequence of messages** exchanged between system components in a specific order to accomplish a particular functionality. Sequence diagrams are also referred to as **event diagrams** or **timing diagrams**, as they emphasize the order of operations and interactions.

In the **Startup Funding Prediction and Investment Analytics** system, the sequence diagram represents the step-by-step interaction between the **User**, **Web Application**, **Machine Learning Model**, **Dataset**, and **Prediction Result** components. It clearly shows how user input is processed, how the machine learning model is invoked, and how the prediction results are returned to the user.

**Sequence Diagram – Startup Funding Prediction System**

**Participants**

* User
* Web Application
* ML Model
* Dataset
* Prediction Result

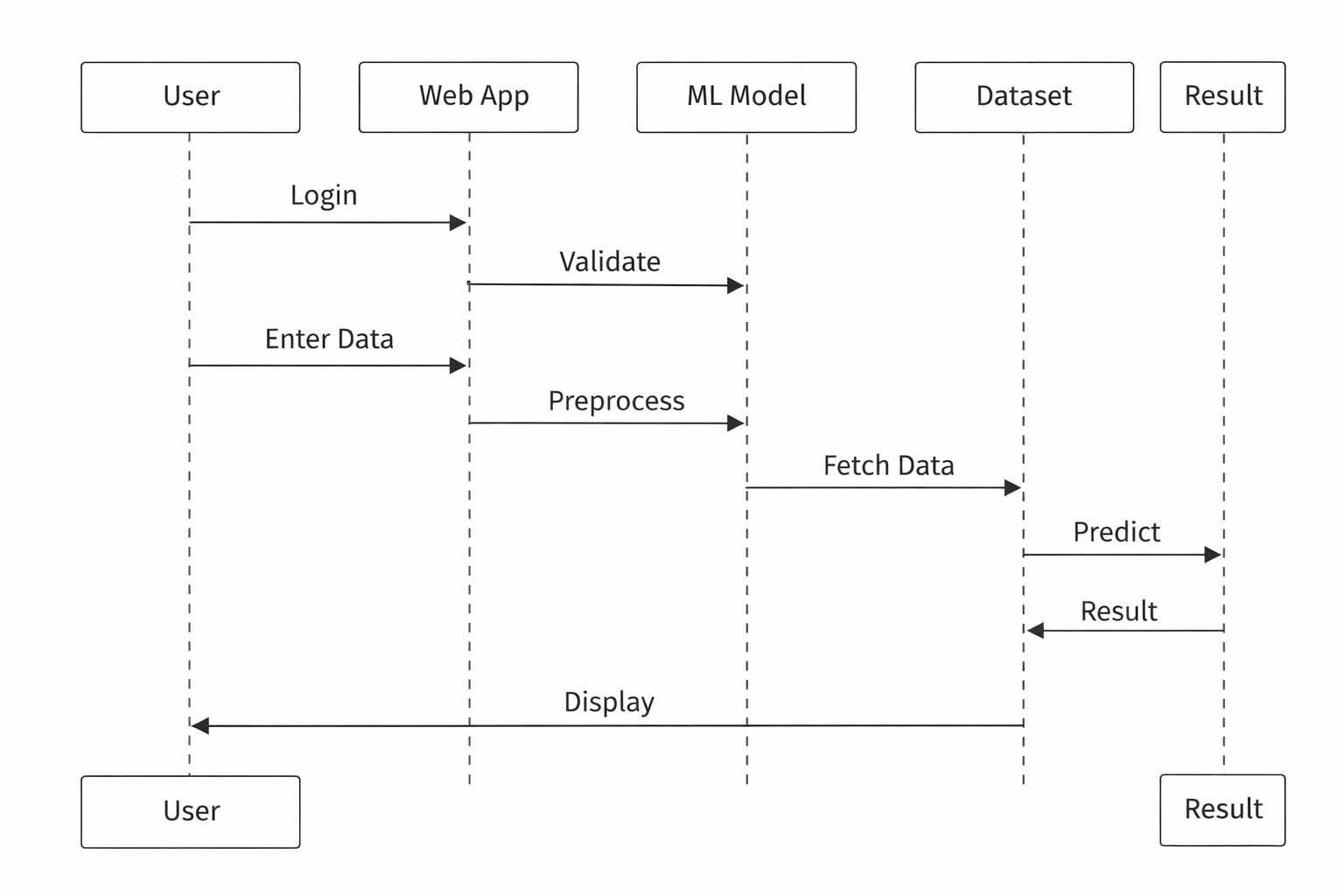
****

Fig 6.5: Sequence diagram

**Explanation**

1. The user logs into the system through the web application.
2. The system validates the user credentials.
3. The user enters startup details into the application.
4. The web application preprocesses the data and sends it to the trained machine learning model.
5. The machine learning model accesses historical data if required and performs prediction.
6. The prediction result is generated and returned to the web application.
7. The result is displayed to the user.
8. The user logs out, completing the interaction sequence.
9. **Activity diagram**

An **Activity Diagram** in the Unified Modeling Language (UML) represents the workflow of a system by illustrating the sequence of activities, decisions, and control flow from start to end. It supports modeling of **decision points**, **loops**, and **concurrent activities**, making it suitable for describing business and operational workflows.

In the **Startup Funding Prediction and Investment Analytics** system, the activity diagram describes the step-by-step workflow followed when a user interacts with the system to obtain funding predictions. It shows the overall control flow from user login to prediction generation and logout, providing a clear understanding of system behavior.

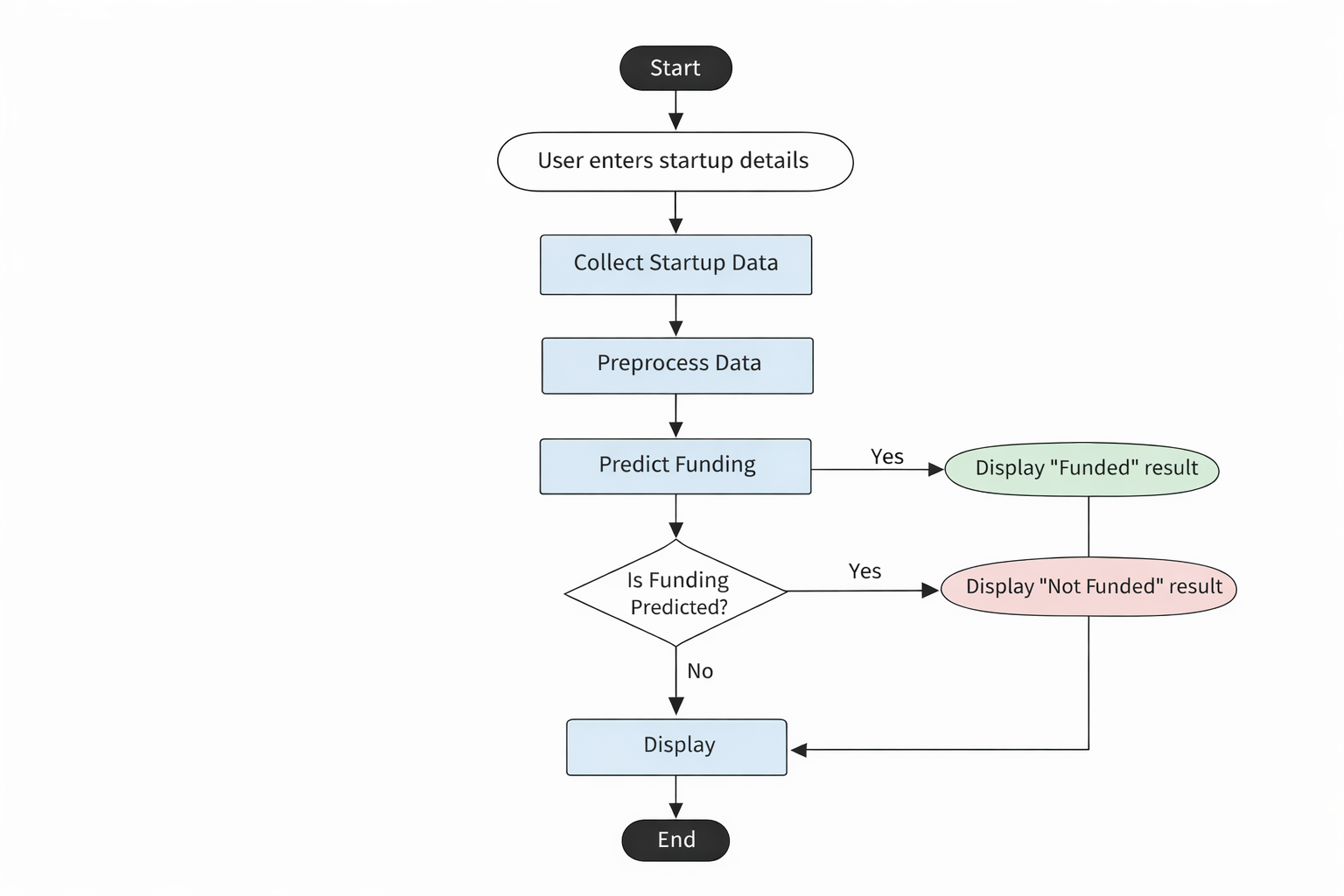


Fig 6.6: Activity Diagram

**Explanation**

1. The activity starts when the user accesses the system.
2. The user logs into the application.
3. The system verifies user authentication.
4. Upon successful authentication, the user enters startup details.
5. The system preprocesses the input data.
6. The machine learning model predicts funding outcome and investment amount.
7. The results are displayed to the user.
8. The user logs out and the process ends.
9. **Collaboration diagram**

A **Collaboration Diagram** in the Unified Modeling Language (UML) represents the interactions between different objects in a system by showing how they are connected and how messages are exchanged among them. Unlike sequence diagrams that emphasize time order, collaboration diagrams focus on **object relationships and message flow**, with **numbered interactions** used to indicate the sequence of communication.

In the **Startup Funding Prediction and Investment Analytics** system, the collaboration diagram illustrates how various objects such as the **User**, **Web Application**, **Machine Learning Model**, **Dataset**, and **Prediction Result** collaborate with each other to complete the funding prediction process. This diagram helps in identifying all possible interactions that each object has with other objects in the system.

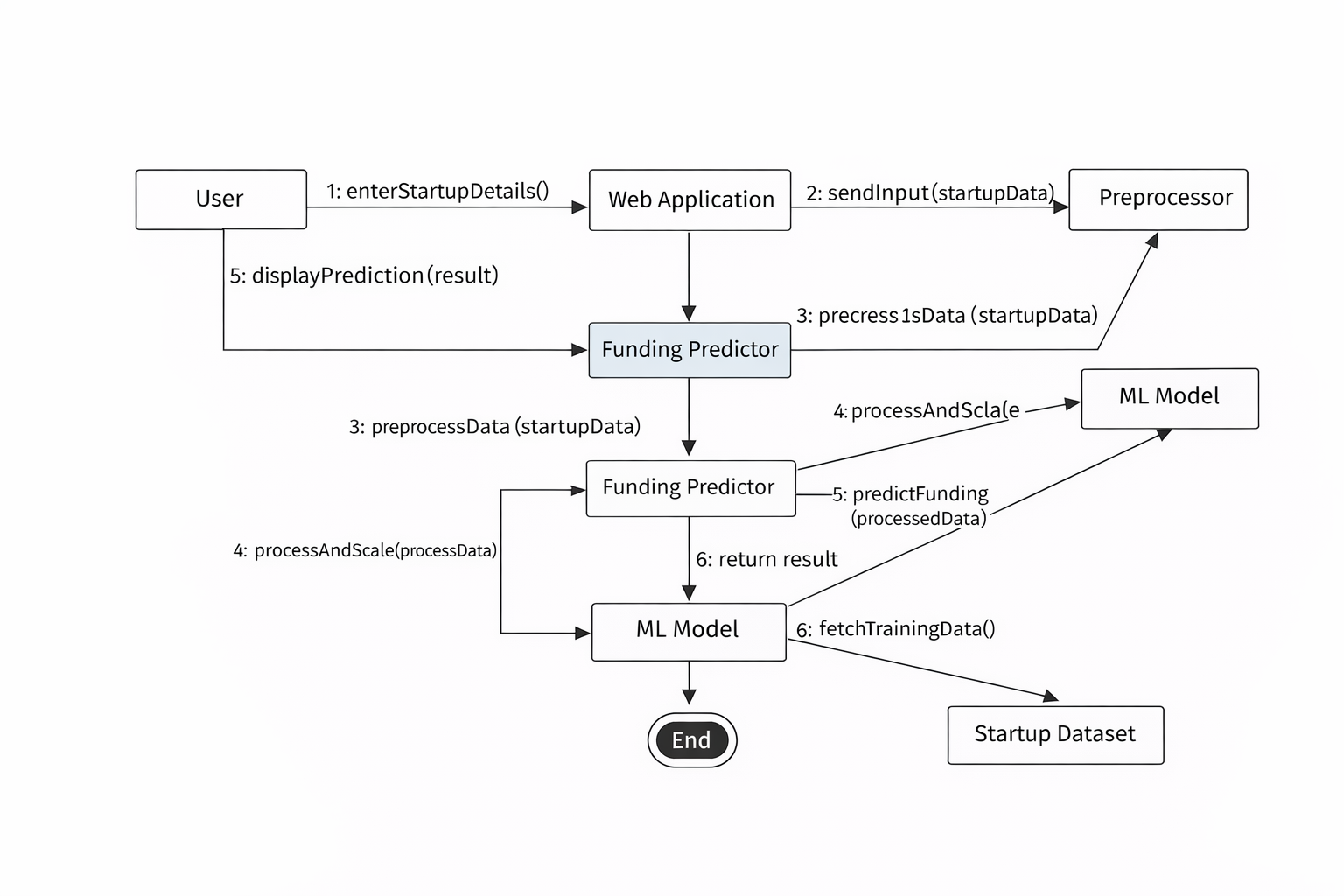
****

Fig 6.7: Collaboration diagram

# SYSTEM SPECIFICATION

# 7.1 Hardware requirements

**Computing System**

1. **Processor:** Multi-core processor (e.g., Intel Core i5/i7 or AMD Ryzen) to efficiently handle data preprocessing, machine learning model training, and prediction tasks.
2. **RAM:** Minimum 8 GB RAM (16 GB recommended) to manage large startup datasets and perform complex analytical computations smoothly.
3. **Storage:** Solid State Drive (SSD) for faster access to datasets, trained machine learning models, and application files.
4. **Graphics Processing Unit (GPU):** Optional. A GPU can improve performance during model training for large datasets but is not mandatory for traditional machine learning algorithms used in this project.

**7.2 Software requirements**

### Operating System

* Linux (Ubuntu/CentOS): Preferred for stability, security, and compatibility with data science and machine learning frameworks.
* Windows or macOS: Suitable alternatives for development and testing.

### Programming Languages

1. **Python:** Primary programming language used for data preprocessing, machine learning model development, prediction, and web application implementation.
2. **Integrated Development Environment (IDE):** Tools such as PyCharm, Jupyter Notebook, or Visual Studio Code (VS Code) are used for coding, debugging, and experimentation.

### Version Control: Git is used to manage the codebase and support collaboration and version tracking.

1. **Scikit-learn:** Used for implementing machine learning algorithms such as Logistic Regression, Decision Tree, and Random Forest.
2. **TensorFlow / PyTorch (Optional):** Can be used for advanced deep learning experiments if required in future enhancements
3. **Matplotlib and Seaborn**: Used for visualizing startup funding trends, correlations, and model performance metrics.
4. **Plotly:** Used for creating interactive dashboards and web-based visual analytics**.**
5. **PostgreSQL / MySQL:** Relational database systems used to store structured startup data, user inputs, and prediction results securely**.**

**7.3 EXECUTION FOR FRONT-END**

**HTML & CSS**

The front-end of the **Startup Funding Prediction & Investment Analytics** system is developed using **HTML and CSS**. The front-end provides an interactive and user-friendly interface that allows users to input startup details and visualize the prediction results generated by the machine learning model.

**HTML (HyperText Markup Language)**

HTML is used to design the **structure of the web application**. It defines the layout of the pages such as input forms, headings, buttons, and result sections. HTML elements are arranged in a structured and hierarchical manner to ensure clarity and usability.

In this project, HTML is used to:

* Create **input forms** for startup details such as sector, location, founding year, and business model
* Design buttons for **submitting data** for prediction
* Display **prediction results and analytics output**
* Organize content using headings, paragraphs, and images

A basic HTML structure used in the project is shown below:

<!DOCTYPE html>

<html>

<head>

<title>Startup Funding Prediction</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1>Startup Funding Prediction & Investment Analytics</h1>

<form action="/predict" method="post">

<label>Sector:</label>

<input type="text" name="sector"><br>

<label>Founding Year:</label>

<input type="number" name="year"><br>

<label>Location:</label>

<input type="text" name="location"><br>

<button type="submit">Predict Funding</button>

</form>

<p id="result">Prediction Result will appear here</p>

</body>

</html>

**Description of key HTML tags used:**

* <!DOCTYPE html>: Declares the document type and HTML version
* <html>: Root element of the document
* <head>: Contains metadata and stylesheet links
* <title>: Specifies the title shown on the browser tab
* <body>: Contains visible web content
* <form>: Collects startup input data from the user
* <input>: Accepts user input values
* <button>: Submits data for prediction

**CSS (Cascading Style Sheets)**

CSS is used to define the **visual appearance and layout** of the web pages. It separates content from design, making the interface more attractive and easier to maintain.

In this project, CSS is used to:

* Improve **layout alignment and spacing**
* Style forms, buttons, and text
* Enhance **user experience and readability**
* Maintain a consistent theme across pages

CSS is applied using an **external stylesheet**, which improves reusability and maintainability.

Basic CSS syntax:

selector {

property: value;

}

Example CSS used in the project:

body {

font-family: Arial, sans-serif;

background-color: #f4f6f8;

}

h1 {

color: #2c3e50;

text-align: center;

}

form {

width: 40%;

margin: auto;

padding: 20px;

background-color: #ffffff;

border-radius: 8px;

}

button {

background-color: #3498db;

color: white;

padding: 10px;

border: none;

}

**Front-End Role in the Project**

HTML and CSS together ensure that:

* Users can easily **enter startup-related data**
* Outputs such as **funding prediction results** are clearly displayed

**8. EXPERIMENTAL SETUP AND RESULTS**

**8.1 Experimental Setup**

**a. Dataset Selection:**  
A structured dataset containing relevant startup-related attributes is used for this project. The dataset includes information such as startup sector, founding year, location, business model, founders’ background, funding stages, and funding amounts. The dataset comprises both funded and non-funded startups, ensuring balanced representation for training and testing. The data reflects diverse industries and regions to accurately represent the real-world startup ecosystem.

**b. Data Preprocessing:**  
Data preprocessing is performed to enhance data quality and model performance. This includes handling missing values, removing duplicate records, encoding categorical variables, and normalizing numerical features. Outliers in funding amounts are treated appropriately to prevent bias. The dataset is standardized to ensure consistency across all features, enabling effective model training.

**c. Feature Selection:**  
Relevant features influencing startup funding outcomes are selected using domain knowledge and statistical analysis. Key factors such as sector, founding year, geographic location, team experience, and business model are identified as important predictors. Feature engineering techniques are applied to extract meaningful insights and improve the predictive power of the machine learning models.

**d. Experimental Design:**  
The dataset is divided into training and testing sets using stratified sampling to maintain class balance. Typically, a major portion of the data is used for training, while the remaining portion is reserved for evaluation. This split ensures that the model learns effectively while being objectively assessed on unseen data.

**e. Model Training:**  
Multiple machine learning models such as Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting are implemented to predict startup funding outcomes. The models are trained using the training dataset, and parameters such as learning rate, number of trees, and maximum depth are adjusted to optimize performance. Ensemble models are explored to enhance predictive accuracy.

**f. Cross-Validation:**  
k-fold cross-validation is employed to evaluate the generalization capability of the models. The dataset is divided into multiple folds, and each model is trained and validated across different subsets. Performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC are calculated to ensure model robustness and reduce overfitting.

**g. Model Evaluation:**  
The trained models are evaluated using the independent testing dataset. Predicted outcomes are compared with actual funding status to compute evaluation metrics and confusion matrices. This evaluation provides insight into the effectiveness of the models in predicting real-world startup funding outcomes.

**h. Parameter Tuning:**  
Hyperparameter tuning is performed using techniques such as grid search and random search. Various combinations of parameters like learning rate, number of estimators, and maximum depth are tested to determine the optimal configuration that maximizes prediction accuracy and minimizes error.

**8.2 Results**

**a. Prediction Accuracy:**  
The machine learning models demonstrate high prediction accuracy in identifying startups likely to receive funding. Performance is measured using accuracy, precision, recall, F1-score, and ROC-AUC metrics, confirming the models’ effectiveness in funding prediction and investment analysis.

**b. Comparison with Baseline Models:**  
The performance of advanced models such as Random Forest and Gradient Boosting is compared with baseline models like Logistic Regression and Decision Tree. Experimental results show that ensemble-based models outperform traditional techniques in terms of accuracy, stability, and prediction reliability.

**c. Effect of Optimization Techniques:**  
Hyperparameter tuning, feature engineering, and data preprocessing significantly improve model performance. Optimized models achieve higher accuracy and reduced prediction error, demonstrating the importance of systematic optimization in startup funding prediction tasks.

**d. Robustness and Generalization:**  
The models exhibit strong robustness and generalization by maintaining consistent performance across startups from different sectors, regions, and founding periods. Testing on unseen data confirms that the models can adapt to diverse startup scenarios without overfitting.

**e. Scalability:**  
Scalability analysis shows that the models efficiently handle large datasets with increasing numbers of startup records. Training time and prediction latency remain within acceptable limits, making the system suitable for large-scale investment analytics and decision support applications.

**f. Practical Utility:**  
The proposed system provides practical value by assisting investors, venture capitalists, and startup founders in data-driven decision-making. By predicting funding likelihood and analyzing investment trends, the system supports early opportunity identification, risk assessment, and strategic planning in the startup ecosystem.

**9.Coding:-**

**9.1 main.py**

import warnings

warnings.filterwarnings("ignore")

import sqlite3

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.pipeline import Pipeline

from sklearn.linear\_model import LogisticRegression, LinearRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor

from sklearn.metrics import accuracy\_score, f1\_score, confusion\_matrix

from scipy.stats import ttest\_rel

from fpdf import FPDF

from datetime import datetime

class PDFLogger:

    def \_\_init\_\_(self, filename):

        self.lines = []

        self.filename = filename

    def log(self, text=""):

        print(text)

        self.lines.append(str(text))

    def save(self):

        pdf = FPDF()

        pdf.set\_auto\_page\_break(auto=True, margin=15)

        pdf.add\_page()

        pdf.set\_font("Arial", size=10)

        for line in self.lines:

            pdf.multi\_cell(0, 6, line)

        pdf.output(self.filename)

# Initialize logger

logger = PDFLogger(

    filename=f"Startup\_Funding\_Report\_{datetime.now().strftime('%Y%m%d\_%H%M%S')}.pdf"

)

df = pd.read\_csv("data/processed/startup\_funding\_cleaned.csv")

logger.log("Available columns:")

logger.log(", ".join(df.columns.tolist()))

# Encode categorical features

categorical\_cols = df.select\_dtypes(include="object").columns

label\_encoders = {}

logger.log("\nEncoding categorical columns...")

for col in categorical\_cols:

    le = LabelEncoder()

    df[col] = le.fit\_transform(df[col].astype(str))

    label\_encoders[col] = le

logger.log(f"Encoded columns: {list(categorical\_cols)}")

# Targets

y\_reg = df["amount\_in\_inr"]

df["funding\_level"] = pd.qcut(df["amount\_in\_inr"], q=4, labels=[0, 1, 2, 3])

y\_cls = df["funding\_level"]

logger.log("\nFunding level distribution:")

logger.log(str(y\_cls.value\_counts()))

# Features

X = df.drop(columns=["amount\_in\_inr", "funding\_level"])

# Train-test split

X\_train, X\_test, y\_train\_cls, y\_test\_cls = train\_test\_split(

    X, y\_cls, test\_size=0.2, random\_state=42, stratify=y\_cls

)

\_, \_, y\_train\_reg, y\_test\_reg = train\_test\_split(

    X, y\_reg, test\_size=0.2, random\_state=42

)

logger.log(f"\nTrain shape: {X\_train.shape}")

logger.log(f"Test shape : {X\_test.shape}")

lr\_pipeline = Pipeline([

    ("scaler", StandardScaler()),

    ("model", LogisticRegression(max\_iter=3000))

])

lr\_pipeline.fit(X\_train, y\_train\_cls)

lr\_pred = lr\_pipeline.predict(X\_test)

dt = DecisionTreeClassifier(random\_state=42)

dt.fit(X\_train, y\_train\_cls)

dt\_pred = dt.predict(X\_test)

rf = RandomForestClassifier(n\_estimators=300, random\_state=42)

rf.fit(X\_train, y\_train\_cls)

rf\_pred = rf.predict(X\_test)

logger.log("\nLogistic Regression")

logger.log(f"Accuracy : {accuracy\_score(y\_test\_cls, lr\_pred):.4f}")

logger.log(f"F1 Score : {f1\_score(y\_test\_cls, lr\_pred, average='weighted'):.4f}")

logger.log("\nDecision Tree")

logger.log(f"Accuracy : {accuracy\_score(y\_test\_cls, dt\_pred):.4f}")

logger.log(f"F1 Score : {f1\_score(y\_test\_cls, dt\_pred, average='weighted'):.4f}")

logger.log("\nRandom Forest")

logger.log(f"Accuracy : {accuracy\_score(y\_test\_cls, rf\_pred):.4f}")

logger.log(f"F1 Score : {f1\_score(y\_test\_cls, rf\_pred, average='weighted'):.4f}")

logger.log("\nObservation:")

logger.log("Random Forest performs best among baseline models")

logger.log("\nClassification Evaluation:")

logger.log(f"Accuracy : {accuracy\_score(y\_test\_cls, rf\_pred):.4f}")

logger.log(f"F1 Score : {f1\_score(y\_test\_cls, rf\_pred, average='weighted'):.4f}")

logger.log("\nConfusion Matrix:")

logger.log(str(confusion\_matrix(y\_test\_cls, rf\_pred)))

cv\_scores\_rf = cross\_val\_score(rf, X, y\_cls, cv=5, scoring="accuracy")

logger.log("\nCross Validation Accuracy Scores:")

logger.log(str(np.round(cv\_scores\_rf, 4)))

logger.log(f"Mean CV Accuracy : {cv\_scores\_rf.mean():.4f}")

# Regression

rf\_reg = RandomForestRegressor(n\_estimators=300, random\_state=42)

rf\_reg.fit(X\_train, y\_train\_reg)

reg\_pred = rf\_reg.predict(X\_test)

mae = np.mean(np.abs(y\_test\_reg - reg\_pred))

r2 = rf\_reg.score(X\_test, y\_test\_reg)

logger.log("\nRegression Evaluation:")

logger.log(f"MAE : {mae:.2e}")

logger.log(f"R2  : {r2:.4f}")

baseline\_scores = cross\_val\_score(

    lr\_pipeline, X, y\_cls, cv=5, scoring="accuracy"

)

logger.log(f"Baseline Mean Accuracy : {baseline\_scores.mean():.4f}")

logger.log(f"Optimized Mean Accuracy: {cv\_scores\_rf.mean():.4f}")

feature\_importance = pd.DataFrame({

    "Feature": X.columns,

    "Importance": rf.feature\_importances\_

}).sort\_values(by="Importance", ascending=False)

logger.log("\nTop 5 Important Features:")

logger.log(str(feature\_importance.head(5)))

t\_stat, p\_value = ttest\_rel(cv\_scores\_rf, baseline\_scores)

logger.log(f"t-statistic : {t\_stat:.4f}")

logger.log(f"p-value     : {p\_value:.6f}")

if p\_value < 0.05:

    logger.log("Decision: Reject Null Hypothesis (Significant improvement)")

else:

    logger.log("Decision: Fail to Reject Null Hypothesis")

conn = sqlite3.connect("data/startup\_funding.db")

X\_db = pd.read\_sql("SELECT \* FROM X\_features", conn)

y\_level\_db = pd.read\_sql("SELECT \* FROM y\_funding\_level", conn).values.ravel()

y\_amount\_db = pd.read\_sql("SELECT \* FROM y\_funding\_amount", conn).values.ravel()

conn.close()

# Align feature order

X\_db = X\_db[X.columns]

X\_tr, X\_te, y\_lvl\_tr, y\_lvl\_te, y\_amt\_tr, y\_amt\_te = train\_test\_split(

    X\_db, y\_level\_db, y\_amount\_db,

    test\_size=0.2,

    random\_state=42,

    stratify=y\_level\_db

)

lvl\_pred\_db = rf.predict(X\_te)

amt\_pred\_db = rf\_reg.predict(X\_te)

logger.log("\nFunding Level Evaluation (DB):")

logger.log(f"Accuracy : {accuracy\_score(y\_lvl\_te, lvl\_pred\_db):.4f}")

logger.log(f"F1 Score : {f1\_score(y\_lvl\_te, lvl\_pred\_db, average='weighted'):.4f}")

logger.log("Confusion Matrix:")

logger.log(str(confusion\_matrix(y\_lvl\_te, lvl\_pred\_db)))

mae\_db = np.mean(np.abs(y\_amt\_te - amt\_pred\_db))

logger.log(f"\nFunding Amount MAE (DB): {mae\_db:.2e}")

def get\_valid\_input(name, encoder):

    logger.log(f"\nAvailable {name} options:")

    for v in encoder.classes\_:

        logger.log(f"- {v}")

    while True:

        val = input(f"Enter {name}: ").strip()

        if val in encoder.classes\_:

            return val

        print("Invalid input, try again.")

industry = get\_valid\_input("industry", label\_encoders["industry"])

city = get\_valid\_input("city", label\_encoders["city"])

investment\_type = get\_valid\_input("investment\_type", label\_encoders["investment\_type"])

city\_tier = get\_valid\_input("city\_tier", label\_encoders["city\_tier"])

market\_size = get\_valid\_input("market\_size\_category", label\_encoders["market\_size\_category"])

founded\_year = int(input("Enter founded\_year: "))

no\_of\_founders = int(input("Enter no\_of\_founders: "))

startup\_encoded = {

    "industry": label\_encoders["industry"].transform([industry])[0],

    "city": label\_encoders["city"].transform([city])[0],

    "investment\_type": label\_encoders["investment\_type"].transform([investment\_type])[0],

    "founded\_year": founded\_year,

    "no\_of\_founders": no\_of\_founders,

    "city\_tier": label\_encoders["city\_tier"].transform([city\_tier])[0],

    "market\_size\_category": label\_encoders["market\_size\_category"].transform([market\_size])[0],

}

startup\_df = pd.DataFrame([startup\_encoded])[X.columns]

funding\_level\_pred = rf.predict(startup\_df)[0]

funding\_amount\_pred = rf\_reg.predict(startup\_df)[0]

level\_map = {

    0: "Low Funding",

    1: "Medium Funding",

    2: "High Funding",

    3: "Very High Funding"

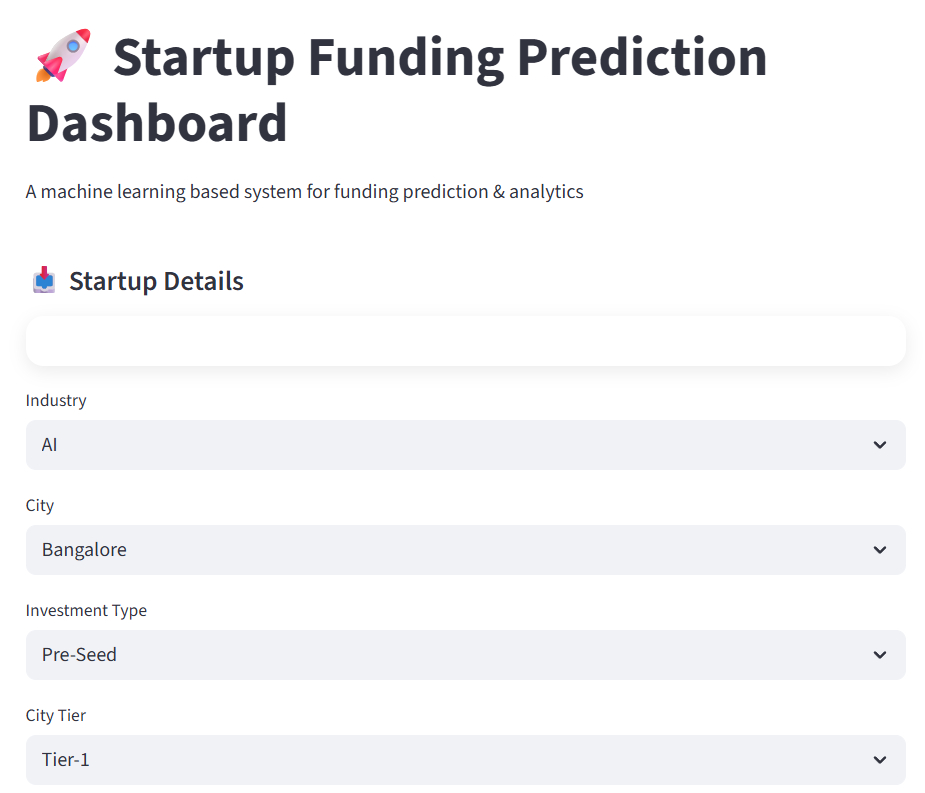
}

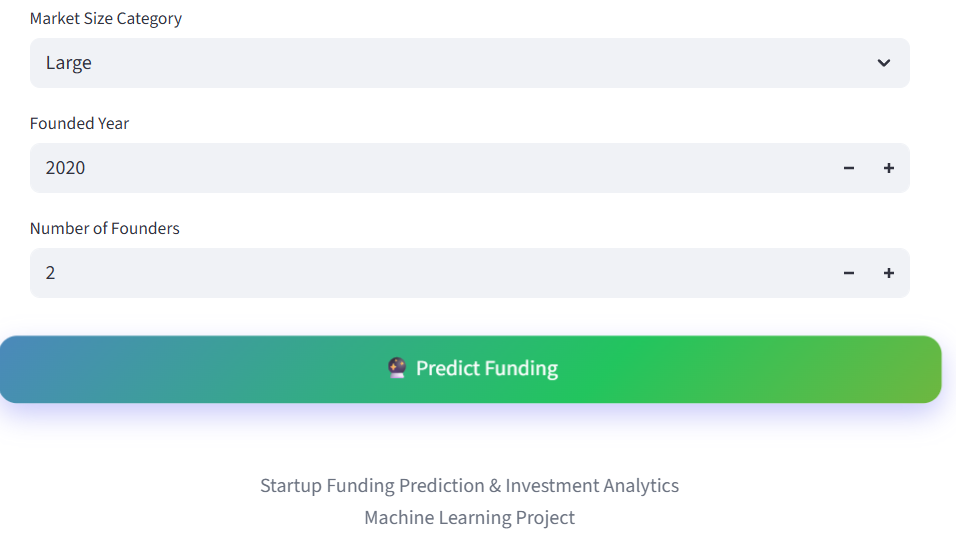
logger.log("\nPrediction Result:")

logger.log(f"Funding Level    : {level\_map[funding\_level\_pred]}")

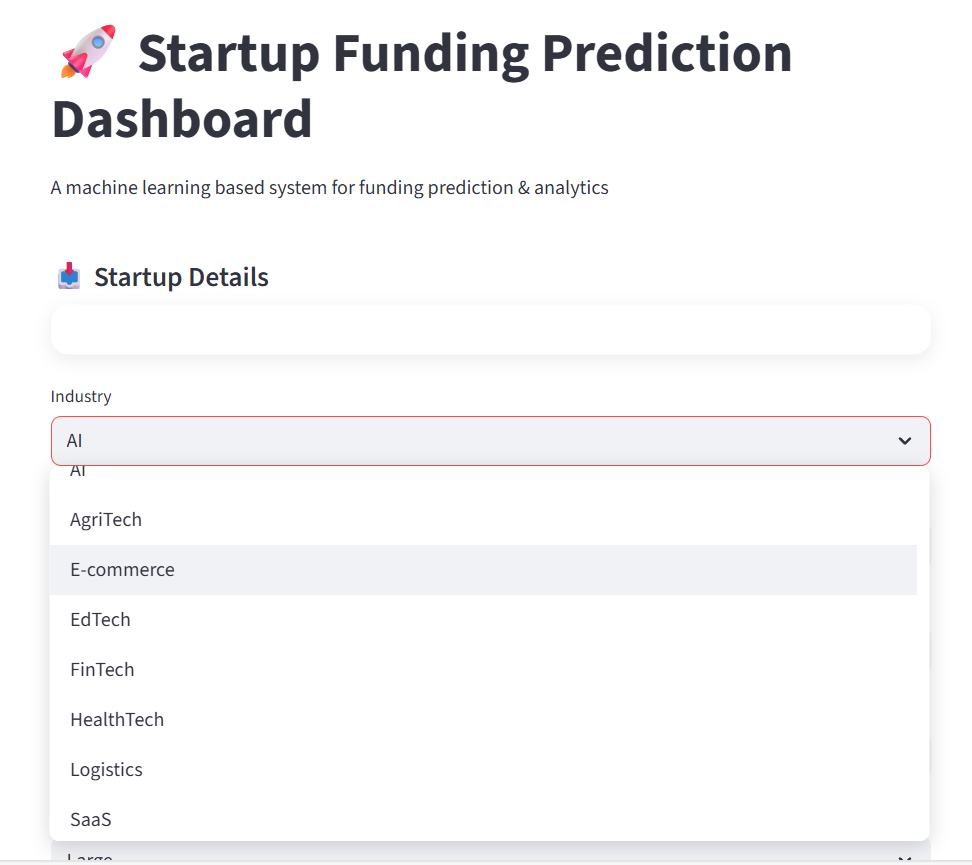
logger.log(f"Estimated Amount : ₹ {funding\_amount\_pred:,.0f}")

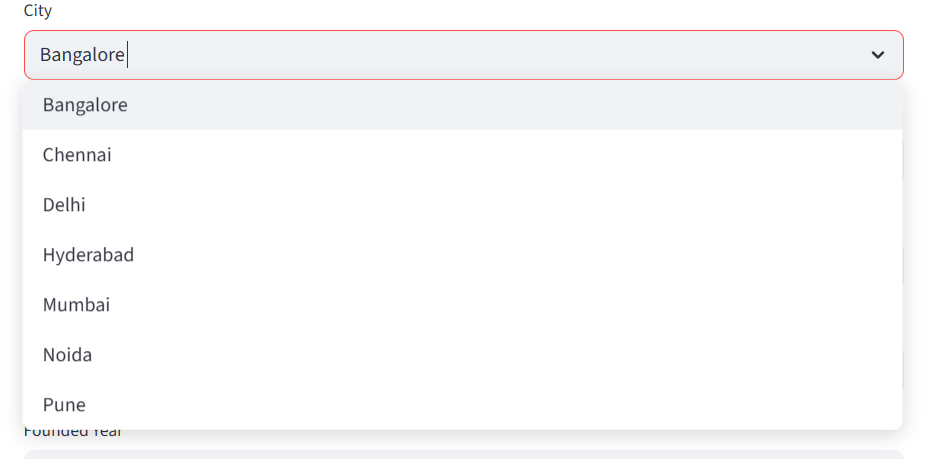
1. **EXECUTIONSCREENSHOTS**

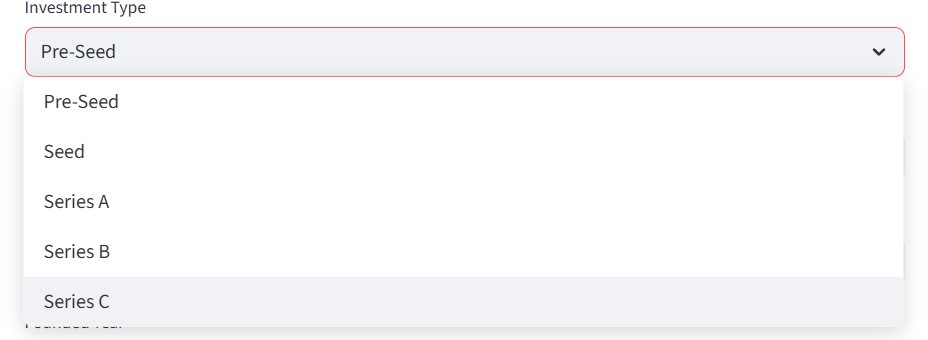
****

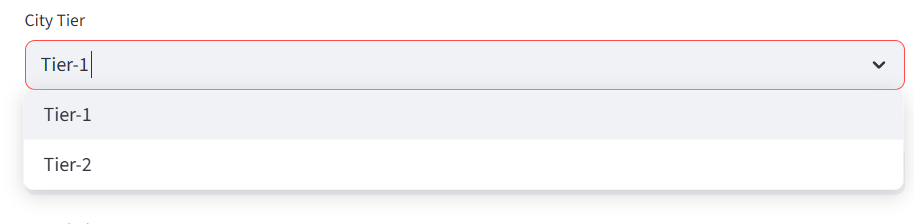


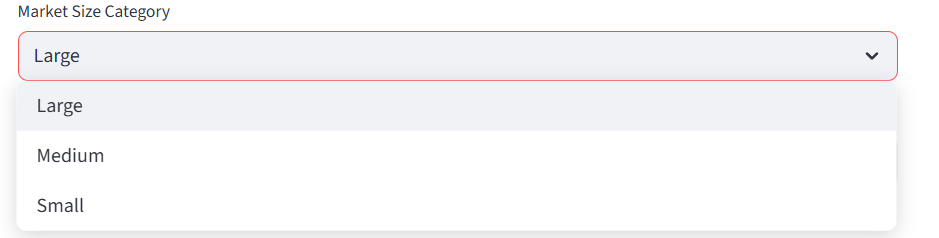
Screenshot 10.1 home page

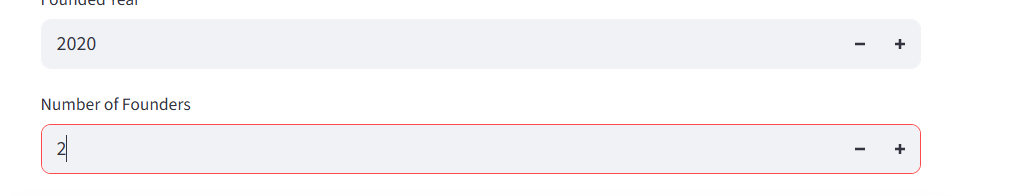
****

****

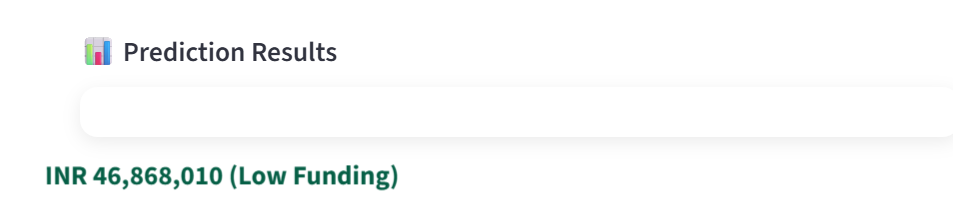




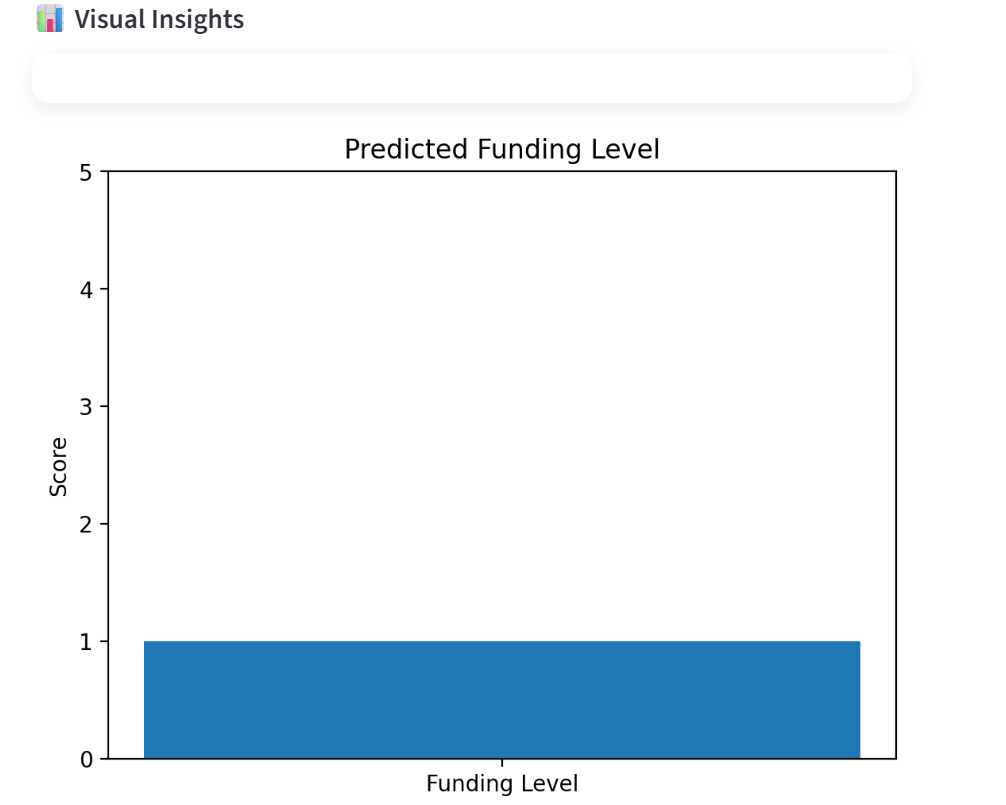




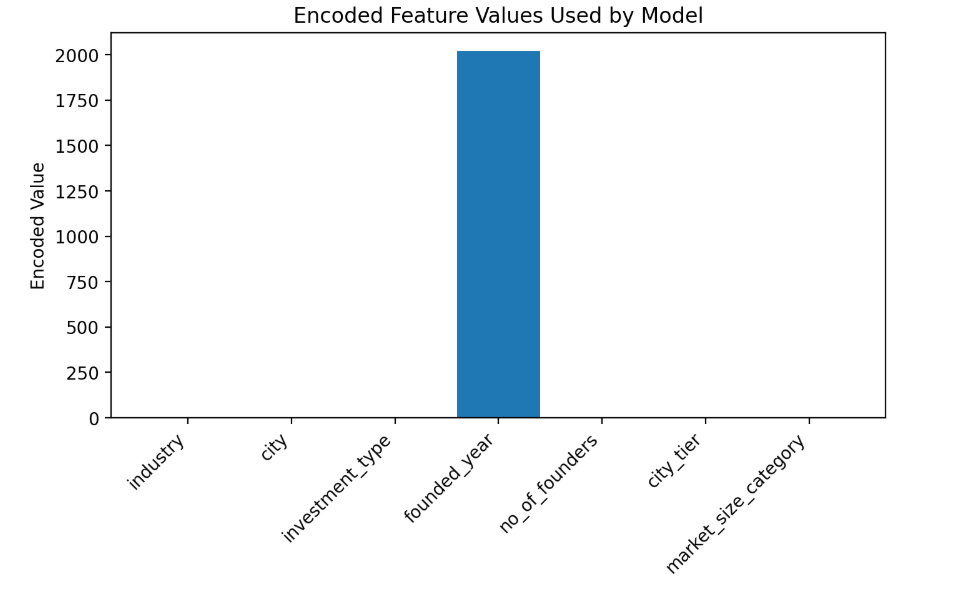
Screenshot 10.2: enter startup details

****

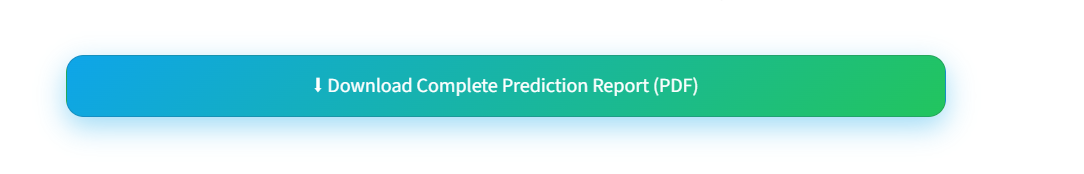
Screenshot 10.3: startup funding prediction



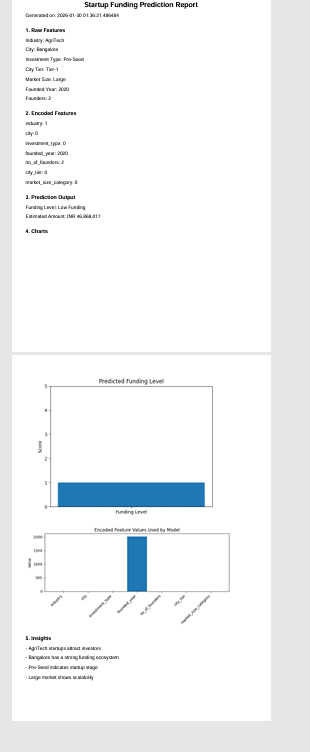
Screenshot 10.4: visual insights

****

Screenshot 10.5: features



Screenshot 10.6 download button to download complete report

****

Screenshot 10.7: complete report

**10. LIMITATIONS**

**a. Limited Dataset Scope:**  
The dataset used in this project is derived from historical startup funding information, which may not fully capture the rapidly evolving dynamics of the startup ecosystem. New business models, emerging sectors, and recent market trends may not be adequately represented, limiting the model’s applicability over time.

**b. Data Quality Dependence:**  
The accuracy of the prediction model heavily depends on the quality, completeness, and correctness of the input data. Incomplete records, missing financial details, or inaccurate founder information can negatively affect prediction performance.

**c. Bias in Historical Data:**  
Historical funding data may contain inherent biases based on geographic regions, industry preferences, or investor behavior. These biases can influence the model’s predictions and may lead to unfair or skewed outcomes for certain types of startups.

**d. Inability to Predict Novel Trends:**  
The model is trained on past data and may not accurately predict funding outcomes for entirely new startup sectors or innovative business models that were not present during training.

**e. Limited Feature Coverage:**  
Although key features are considered, some important qualitative factors such as founder networking strength, investor sentiment, market competition, and macroeconomic conditions are not included, which could improve prediction accuracy if available.

**f. Generalization Challenges:**  
A model trained on a specific dataset or regional startup ecosystem may not generalize effectively to other countries or markets with different investment behaviors and regulatory environments.

**g. No Real-Time Market Integration:**  
The current system does not incorporate real-time market indicators, investor activity feeds, or financial news, which could significantly impact funding decisions and improve prediction reliability.

**h. Interpretability Constraints:**  
While advanced machine learning models provide higher accuracy, they may lack full transparency in explaining individual predictions, which can reduce user trust and limit practical adoption for critical investment decisions.

**10. FUTURE SCOPE**

Future work on the Startup Funding Prediction & Investment Analytics system involves expanding the model by incorporating a broader range of parameters that can improve the correlation with startup funding success. In addition to the current features such as sector, location, founding year, and business model, future versions of the system can include financial indicators like revenue growth, burn rate, customer acquisition cost, valuation trends, and funding stage progression. Including these factors would significantly enhance prediction accuracy and provide a more holistic view of startup performance.

The dataset can be further enriched by integrating external and macroeconomic data sources, such as market conditions, industry growth rates, investor sentiment indices, and economic indicators. Publicly available data on venture capital activity, interest rates, and startup ecosystems can be leveraged to enable more accurate funding forecasts and trend analysis. This expansion would allow the model not only to predict funding likelihood but also to estimate potential funding amounts and investment timing.

Another key future enhancement is the ability to adapt to emerging startup trends. By continuously updating the training dataset with newly funded startups and failed ventures, the model can learn from recent patterns and reduce dependency on historical assumptions. Semi-automated data validation mechanisms can be implemented to selectively accept new user-provided data into the training pipeline, minimizing human intervention while maintaining data quality.

The system can also be extended to support time-based analytic

**10. APPLICATIONS**

a. Startup Funding Prediction:  
The primary application of the proposed system is the prediction of startup funding outcomes. By analyzing historical and current startup data, the machine learning model determines the likelihood of a startup receiving funding. This enables founders and investors to make informed decisions at early stages.

b. Investment Decision Support:  
The system acts as a decision support tool for investors, venture capitalists, and angel investors. Based on predicted funding probabilities and key influencing factors, users can assess investment risks and opportunities more effectively.

c. Portfolio Risk Management:  
By identifying startups with higher or lower funding potential, the platform helps investors manage investment risk across portfolios. Early prediction allows optimization of capital allocation and diversification strategies.

d. Strategic Planning for Startups:  
Startup founders can use the system to evaluate their funding readiness and identify areas for improvement. Insights derived from the prediction model can guide strategic decisions related to business models, market selection, team composition, and growth planning.

e. Market Trend Analysis:  
The system supports sector-wise and region-wise investment trend analysis. By aggregating prediction results, users can identify emerging industries, high-growth regions, and changing investor preferences within the startup ecosystem.

f. Performance Benchmarking:  
Startups can benchmark their profiles against funded and non-funded startups in the dataset. This comparison helps evaluate competitiveness and readiness for investment in a data-driven manner.

g. Financial Planning and Forecasting:  
Accurate funding prediction supports better financial planning by helping startups estimate the likelihood and timing of investments. This enables improved budgeting, cash flow management, and operational planning.

h. Policy and Ecosystem Development:  
Government agencies, incubators, and accelerators can use insights from the system to support entrepreneurship programs, identify high-potential startups, and design policies that strengthen the startup ecosystem.

**14. SYSTEM TESTING**

The purpose of system testing is to identify errors and ensure that the developed software meets its specified requirements and user expectations. Testing is the process of executing a system with the intent of detecting faults, validating correctness, and ensuring reliable performance under expected conditions.

In the **Startup Funding Prediction & Investment Analytics** system, testing ensures that all components—including data preprocessing, machine learning model, API integration, and user interface—function correctly both individually and as an integrated system. Various types of testing are performed, each addressing specific validation requirements.

**14.1 TYPES OF TESTS**

**a. Unit Testing**

Unit testing involves designing test cases to verify that individual modules of the application function correctly. Each unit is tested independently after implementation and before integration with other modules.

In this project, unit testing is applied to:

* Data preprocessing functions (handling missing values, encoding, scaling)
* Feature engineering modules
* Model training and prediction functions
* Input validation logic in the web application

This testing ensures that each component performs as per the documented specifications and produces correct outputs for valid inputs.

**b. Integration Testing**

Integration testing is conducted to verify that different modules of the system interact correctly when combined. Although individual components may function properly, integration testing identifies issues arising from data flow or interface mismatches between components.

For this project, integration testing validates:

* Interaction between the front-end and Flask API
* Flow of user inputs from web forms to the prediction model
* Model output rendering on the result page

This testing ensures that all integrated modules operate as a unified system.

**c. Functional Testing**

Functional testing verifies that the system’s functionalities meet both business and technical requirements. It focuses on evaluating the system behavior against predefined functional specifications.

Functional testing is centered on:

* **Valid Input:** Correct startup details are accepted and processed
* **Invalid Input:** Incorrect or missing inputs are rejected with appropriate messages
* **Functions:** All core functions such as prediction and result display are executed
* **Output:** Correct funding prediction results are generated
* **System Procedures:** Backend processing and page navigation work as expected

This ensures that the application performs all intended operations accurately.

**d. System Testing**

System testing validates the complete and fully integrated system to confirm compliance with functional and non-functional requirements. It focuses on overall system behavior under realistic usage scenarios.

In this project, system testing ensures:

* Smooth end-to-end workflow from data input to prediction output
* Predictable and reliable system responses
* Proper handling of edge cases and user errors

System testing confirms the readiness of the application for deployment.

**e. White Box Testing**

White Box Testing is performed with knowledge of the internal structure and logic of the application. It focuses on verifying internal code paths, decision logic, and execution flow.

In this system, white box testing validates:

* Conditional logic in prediction workflows
* Data processing pipelines
* Internal model invocation and preprocessing steps

This ensures correctness at the code level and reduces hidden logical errors.

**f. Black Box Testing**

Black Box Testing evaluates the system without any knowledge of its internal implementation. The system is tested purely based on input-output behavior.

For this project, black box testing includes:

* Submitting startup details via the UI
* Verifying correctness of prediction outcomes
* Checking page navigation and response handling

This approach validates user-facing functionality and overall usability.

**g. Acceptance Testing**

User Acceptance Testing (UAT) is performed to ensure the system meets user requirements and expectations. End users validate the system’s functionality in real-world scenarios.

Acceptance testing verifies:

* Ease of use of the interface
* Accuracy and relevance of prediction results
* Overall user satisfaction

**Test Results:**  
All acceptance test cases were executed successfully, and no defects were encountered.

**14.2 TEST OBJECTIVES**

* All input fields must accept valid data correctly
* Invalid or incomplete inputs must be handled gracefully
* Pages and links must navigate to the correct views
* System responses must be timely and accurate

**14.3 FEATURES TESTED**

* Validation of input data formats
* Prevention of duplicate or inconsistent entries
* Correct prediction generation
* Proper rendering of outputs and navigation

**Test Results Summary**

All the test cases across unit testing, integration testing, functional testing, system testing, and acceptance testing were executed successfully. No critical defects were identified, and the system performed reliably as per the specified requirements.

**15. CONCLUSION**

This project presents an automated, data-driven, and user-friendly end-to-end solution for one of the most critical challenges in the startup ecosystem—accurately predicting funding outcomes and supporting informed investment decisions. By leveraging machine learning techniques on historical startup data, the proposed system enables founders, investors, and stakeholders to evaluate funding potential efficiently and objectively.

The system advances traditional analytical approaches by integrating robust data preprocessing, feature engineering, and ensemble-based machine learning models to analyze key startup attributes such as sector, location, business model, founding year, and founder background. A high-performance Random Forest model is employed to deliver reliable funding predictions through a web-based platform that allows real-time interaction and accessibility.

The modular and scalable design of the system supports continuous improvement in prediction accuracy. As new startup data becomes available, the model can be retrained to reflect recent funding trends and evolving market conditions. This adaptability ensures that the system remains relevant in a dynamic investment landscape. Experimental results demonstrate strong predictive performance, validating the effectiveness of the proposed approach in identifying high-potential startups.

In conclusion, the **Startup Funding Prediction & Investment Analytics** system shows significant potential for real-world deployment. Its scalability, predictive accuracy, and adaptability make it a valuable decision support tool for investors and entrepreneurs alike. By combining machine learning with systematic data analysis and intuitive user interfaces, the project contributes toward reducing uncertainty in funding decisions, optimizing investment strategies, and fostering sustainable growth within the startup ecosystem.

## 

## REFERENCES

## [1] T. M. Kakkar and S. Bansal, “Predicting Startup Success using Machine Learning Techniques,” *International Journal of Computer Applications*, vol. 182, no. 44, pp. 1–6, 2019.

## [2] C. G. Bandiera and E. R. de Paula, “Startup Valuation and Funding Prediction using Data Analytics,” *Journal of Business Research*, vol. 118, pp. 414–424, 2020.

## [3] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

## [4] I. Guyon and A. Elisseeff, “An Introduction to Variable and Feature Selection,” *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

## [5] Scikit-learn Documentation. [Online]. Available: <https://scikit-learn.org>

## [6] P. Domingos, “A Few Useful Things to Know About Machine Learning,” *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

## [7] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014.

## [8] Kaggle. “Startup Funding Datasets.” [Online]. Available: <https://www.kaggle.com>

## [9] A. Agrawal, J. Gans, and A. Goldfarb, *Prediction Machines: The Simple Economics of Artificial Intelligence*, Harvard Business Review Press, 2018.

## [10] F. Provost and T. Fawcett, *Data Science for Business*, O’Reilly Media, 2013.

## [11] CB Insights, “Global Venture Capital Trends and Startup Funding Analysis,” CB Insights Research Reports, 2021.

## [12] M. Porter, “The Competitive Advantage of Nations,” *Harvard Business Review*, pp. 73–93, 1990.