STARTUP SUCCESS PREDICTION

A UG PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

RACHAMALLA RATHNAKAR **20UK1A0571**

Under the esteemed guidance of

Dr. R. NAVEEN KUMAR

(Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD BOLLIKUNTA, WARANGAL – 506005

2020-2024

STARTUP SUCCESS PREDICTION

A UG PROJECT PHASE-1 REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

RACHAMALLA RATHNAKAR **20UK1A0571**

Under the esteemed guidance of

Dr. R. NAVEEN KUMAR

(Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD BOLLIKUNTA, WARANGAL – 506005

2020-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

BOLLIKUNTA , WARANGAL – 506005

CERTIFICATE OF COMPLETION UG PHASE-1 PROJECT REPORT

This is to certify that the UG Project Phase-1 Report entitled **"STARTUP SUCCESS PREDICTION USING MACHINE LEARNING "** is being submitted by **RACHAMALLA RATHNAKAR (2OUK1AO571)**in partial fulfilment of the requirements for the award of the degree **of Bachelor of Technology in Computer Science and Engineering** to **Jawaharlal Nehru Technological University Hyderabad** during the academic year 2023-2024, is a record of work carried out by them under the guidance and supervision.

Project guide	**Head of the department**
Dr. R. Naveen Kumar	**Dr . R. Naveen Kumar**
(professor)	(professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. Prasad Rao**, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase1 in the institute.

We extend our heartfelt thanks to **Dr. R. Naveen Kumar**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to the guide, **Dr. R. Naveen Kumar**, Assistant professor, Department of CSE for her constant support and giving necessary guidance for completion of this UG Project Phase-2

Finally,We express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

RACHAMALLA RATHNAKAR 20UK1A0571

ABSTRACT

The ability to predict the success of startups is a critical need in the investment and entrepreneurial ecosystem. This project explores the application of machine learning techniques to forecast the success of startups by analyzing various factors that influence their growth and sustainability. Utilizing a comprehensive dataset comprising financial metrics, market conditions, founder backgrounds, and industry-specific variables, we develop and evaluate several machine learning models, including logistic regression, random forests, and gradient boosting machines.

Our methodology involves data preprocessing, feature selection, and model training using historical data from both successful and unsuccessful startups. We employ techniques such as cross-validation and hyperparameter tuning to optimize model performance. The key performance indicators include accuracy, precision, recall, and the F1-score, with a particular focus on minimizing false negatives to ensure promising startups are not overlooked.

Preliminary results indicate that machine learning models can significantly improve the accuracy of startup success predictions compared to traditional heuristic methods. By identifying the most critical predictors of startup success, our models offer valuable insights for investors, incubators, and entrepreneurs. Future work will involve integrating real-time data feeds and expanding the model to incorporate macroeconomic factors to enhance its predictive power.

This project demonstrates the potential of machine learning to transform the startup investment landscape, enabling more informed decision-making and fostering a more robust entrepreneurial environment.

TABLES OF CONTENT

1. INTRODUCTION

1.1OVERVIEW

A startup success prediction project utilizing machine learning techniques aims to harness the power of advanced analytics to forecast the potential success or failure of emerging ventures. By leveraging diverse datasets encompassing financial records, market trends, team profiles, and product attributes, the project seeks to identify patterns and predictors indicative of startup outcomes. Key components of the project include data collection, preprocessing, feature engineering, model development, evaluation, interpretation, and deployment. Through the deployment of trained models into production environments, stakeholders gain access to realtime predictions and insights, enabling informed decision-making and strategic planning. The project's outcomes include accurate predictions, identification of influential factors, enhanced decision-making capabilities, and contributions to the development of responsible AI practices within the startup ecosystem. Ultimately, the project aims to empower stakeholders to allocate resources effectively and support the growth and success of startups in a dynamic and competitive landscape.

1.2PURPOSE

One of the primary challenges in predicting startup success lies in the scarcity and quality of available data. Startup datasets often suffer from limited information, particularly for nascent ventures, which complicates the development of accurate predictive models. Moreover, the data may be plagued by inaccuracies, inconsistencies, or missing values, further undermining the reliability of predictions. Another significant hurdle is the high dimensionality of startup datasets, with numerous features complicating the identification of relevant predictors of success. Additionally, imbalanced class distributions between successful and failed startups pose a challenge, as models may struggle to effectively capture minority class patterns. Furthermore, the dynamic nature of startup environments introduces temporal complexities, with success influenced by evolving market trends, economic conditions, and technological advancements that may not be adequately captured in static datasets. These challenges underscore the need for innovative approaches to data collection, preprocessing, and modeling to enhance the accuracy and robustness of startup success prediction systems.

2.LITERATURE SURVEY

2.1EXISTING PROBLEM

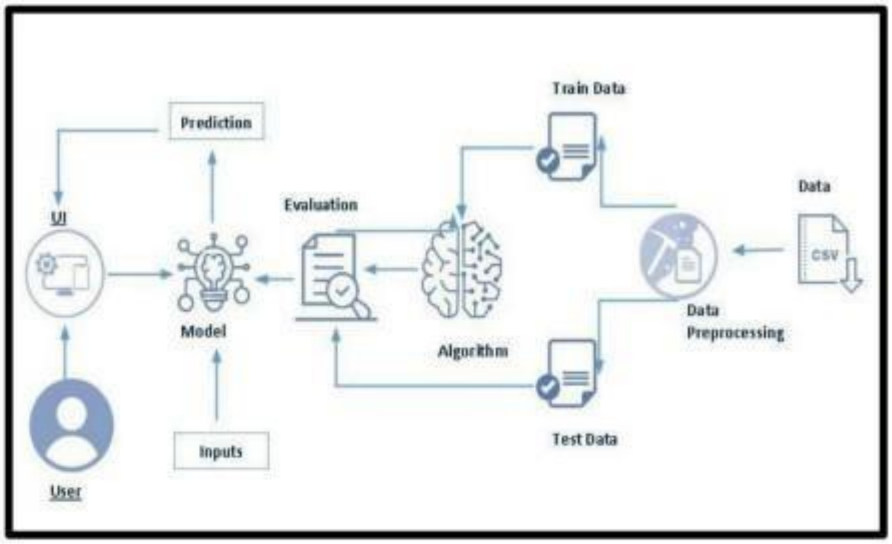
One of the predominant challenges in developing a startup success prediction project using machine learning lies in the complex and multifaceted nature of startup success. Data availability and quality present significant hurdles, as startups often operate in volatile environments with sparse or unreliable data. Moreover, selecting relevant features for prediction poses a dilemma, as determining which factors—such as team expertise, market dynamics, or funding history—most strongly correlate with success requires careful consideration. Imbalanced datasets further complicate matters, potentially biasing models and hindering their ability to accurately predict outcomes. Defining success itself is subjective and can vary widely, leading to ambiguity in model evaluation and performance metrics. Additionally, temporal dynamics and the influence of external factors introduce further complexity, necessitating robust techniques for capturing and adapting to changing conditions over time. Ensuring model interpretability and generalization to unseen data is crucial for practical application, yet achieving this balance remains a persistent challenge in the field of startup success prediction using machine learning.

2.2PROPOSED SOLUTION

A promising solution to the challenges of startup success prediction using machine learning involves adopting a comprehensive and iterative approach that integrates advanced techniques from various domains. Leveraging ensemble learning methods can help mitigate the effects of imbalanced datasets and enhance model robustness by combining multiple models' predictions. Feature engineering, informed by domain expertise and supplemented by techniques such as automated feature selection and dimensionality reduction, can improve model performance by identifying and prioritizing relevant predictors. Incorporating time-series analysis and dynamic modeling approaches allows for capturing temporal dynamics and adapting to changing conditions over time. Moreover, developing interpretable models, such as decision trees or rulebased systems, alongside complex models like neural networks, enables stakeholders to gain insights into the factors driving predictions while maintaining predictive accuracy. Furthermore, integrating external data sources, such as economic indicators or industry trends, enhances model comprehensiveness and predictive power. Continuous model evaluation and refinement, guided by feedback from domain experts and realworld performance metrics, are essential for ensuring the model's relevance and effectiveness in predicting startup success accurately. By embracing a holistic and adaptive methodology, startup success prediction projects can overcome existing challenges and deliver actionable insights to support decisionmaking in the dynamic startup ecosystem.

3.THEORETICAL ANALYSIS

3.1BLOCK DIAGRAM



3.2HARDWARE/SOFTWARE DESIGNING

Firstly, in terms of hardware design, emphasis should be placed on selecting scalable and efficient computing infrastructure capable of handling large volumes of data and complex computational tasks. Cloud-based solutions offer flexibility and scalability, allowing startups to scale their computing resources according to their needs while minimizing upfront infrastructure costs. In terms of software design, flexibility and modularity are essential to accommodate diverse data sources, modeling techniques, and analytical workflows. Adopting a microservices architecture can facilitate modular development, enabling independent scaling and deployment of individual components such as data ingestion, preprocessing, modeling, and prediction.

1. User Requirements

1.Data Input and Integration: Users should be able to input various types of data relevant to startup success prediction, including financial data, market metrics, funding history, team expertise, and product information. The system should support integration with external data sources for enriching and updating the dataset
2.Data Preprocessing and Cleaning: Users require tools for preprocessing and cleaning the data to ensure its quality and consistency. This includes handling missing values, outlier detection, feature scaling, and encoding categorical variables.

3.Model Selection and Training: Users need the ability to select and train different machine learning models to predict startup success. The system should provide a range of algorithms, such as regression, classification, and ensemble methods, along with hyperparameter tuning options for model optimization.

4.Evaluation and Performance Metrics: Users should have access to evaluation metrics to assess the performance of trained models accurately. Common metrics include accuracy, precision, recall, F1-score, and ROC-AUC. The system should also facilitate model comparison and visualization of results.

5.Interpretability and Explainability: Users require insights into the factors driving model predictions to understand the rationale behind the predicted outcomes. The system should provide interpretability tools, such as feature importance analysis, SHAP values, or decision explanations, to make the predictions more transparent.

3.1.2 SOFTWARE REQUIREMENTS

1.Programming Languages

2.Machine Learning Frameworks

3.Data Processing Tools

4.Development Environments

5.Version Control

6.Database Management Systems

7.Web Frameworks

8.Documentation and Collaboration Tools

3.1.3 HARDWARE REQUIREMENTS

1.Compute Infrastructure: Depending on the size of the dataset and the complexity of the machine learning algorithms, you may require powerful computing resources. This can range from standard CPUs to Graphics Processing Units (GPUs) or Tensor Processing Units (TPUs) for accelerated training of deep learning models.

2.Central Processing Unit (CPU): A multicore CPU with sufficient processing power is essential for data preprocessing, feature engineering, and model training tasks. Higher clock speeds and multiple cores can speed up computation-intensive tasks.

3.Graphics Processing Unit (GPU): GPUs are particularly useful for accelerating the training of deep learning models, which often involve matrix operations and parallel computations. NVIDIA GPUs are commonly used for machine learning tasks due to their CUDA support and optimized libraries.

4.DESIGN

4.1INTROUDUCTION

In today's fast-paced and ever-evolving business landscape, startups are emerging as powerful drivers of innovation, disruption, and economic growth. However, the inherent uncertainty and risks associated with launching a new venture make predicting their success an incredibly challenging task. Enter Algorithmic Alchemy: ML-Driven Startup Predictions. Algorithmic Alchemy is a groundbreaking approach that harnesses the power of machine learning (ML) to provide insightful and data-driven predictions about the success rate of startups. By analyzing a range of parameters and variables, such as market trends, funding history, team composition, and product-market fit, Algorithmic Alchemy aims to uncover patterns and correlations that can forecast a startup's trajectory. Machine learning algorithms excel at processing vast amounts of data and identifying hidden patterns that human analysts may overlook. By training these algorithms on historical data from successful and failed startups, Algorithmic Alchemy can learn from past outcomes and develop predictive models that significantly enhance the accuracy of startup success predictions. The ML-driven approach of Algorithmic Alchemy offers several advantages over traditional methods of startup evaluation. It eliminates bias and subjectivity by relying on objective data-driven insights, ensuring more consistent and reliable predictions.

4.2 PROCEDURE

Define Objectives and Success Metrics:

Clearly define the objectives of the startup success prediction project, such as predicting future success indicators (e.g., revenue growth, user acquisition) based on historical data. Identify success metrics to evaluate the performance of the predictive models, such as accuracy, precision, recall, F1-score, or area under the ROC curve (ROC-AUC). Data Collection and Integration: Gather relevant data sources that may influence startup success, including financial data, market metrics, funding history, team expertise, product information, and external factors (e.g., economic indicators). Integrate and preprocess the collected data, handling missing values, outliers, and inconsistencies, and ensuring data quality and consistency. Exploratory Data Analysis (EDA): Conduct exploratory data analysis to gain insights into the characteristics and relationships within the dataset. Visualize distributions, correlations, and trends in the data to identify patterns and potential predictors of startup success.

Feature Engineering:

Engineer new features or transform existing features to enhance the predictive power of the models. Utilize domain knowledge and data analysis techniques to extract relevant features that capture the underlying dynamics of startup success.

Model Selection and Training:

Select appropriate machine learning algorithms based on the problem domain, dataset characteristics, and objectives. Split the dataset into training, validation, and test sets for model training, tuning, and evaluation. Train multiple models using different algorithms, hyperparameters, and feature subsets to compare their performance. Model Evaluation and Validation:

Evaluate the trained models using appropriate evaluation metrics and cross-validation techniques to assess their generalization performance. Tune model hyperparameters and feature selections based on validation results to improve predictive accuracy and robustness.

Interpretation and Explanation: Interpret model predictions and identify the most influential features driving the predictions. Utilize explainable AI techniques, such as feature importance analysis, SHAP values, or decision explanations, to provide insights into the rationale behind the predictions.

Model Deployment and Integration: Deploy the trained models into production environments, either as standalone applications or as part of existing systems. Integrate the predictive models with other business processes and workflows, such as customer relationship management (CRM) systems or decision support tools.

Monitoring and Maintenance: Implement monitoring and logging mechanisms to track model performance, detect anomalies, and ensure continuous reliability. Regularly update and retrain the predictive models with new data to adapt to changing conditions and improve accuracy over time.

Documentation and Knowledge Sharing: Document the entire procedure, including data sources, preprocessing steps, model architectures, and deployment strategies. Share insights, findings, and best practices with stakeholders and team members to facilitate knowledge sharing and collaboration.

4.3Algorithms and Techniques:

There are various algorithms used in this project

1.Logistic Regression: Logistic regression is a simple yet effective algorithm for binary classification tasks, where the goal is to predict whether a startup will succeed or fail based on input features. It models the probability of success as a logistic function of the input features.

2.Random Forest: Random forest is an ensemble learning technique that combines multiple decision trees to make predictions. It is effective for both classification and regression tasks and offers robustness to overfitting and noise in the data.

3.Gradient Boosting Machines (GBM): Gradient boosting machines, such as XGBoost, LightGBM, and CatBoost, sequentially train multiple weak learners (usually decision trees) to correct the errors of the previous models. GBM is known for its high predictive accuracy and flexibility.

4.Support Vector Machines (SVM): Support vector machines are powerful supervised learning algorithms used for classification and regression tasks. SVM aims to find the hyperplane that best separates the classes in the feature space by maximizing the margin between the classes.

5.Neural Networks: Deep learning techniques, including artificial neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs), can capture complex patterns and relationships in the data. They are suitable for tasks with large, highdimensional datasets and can automatically learn feature representations from raw data.

6.K-Nearest Neighbors (KNN): K-nearest neighbors is a simple and intuitive algorithm for classification and regression tasks. It predicts the label of a data point by averaging the labels of its k nearest neighbors in the feature space.

7.Ensemble Learning: Ensemble learning techniques, such as bagging, boosting, and stacking, combine multiple base models to improve predictive performance. By leveraging the diversity of individual models, ensemble methods can often achieve better generalization and robustness.

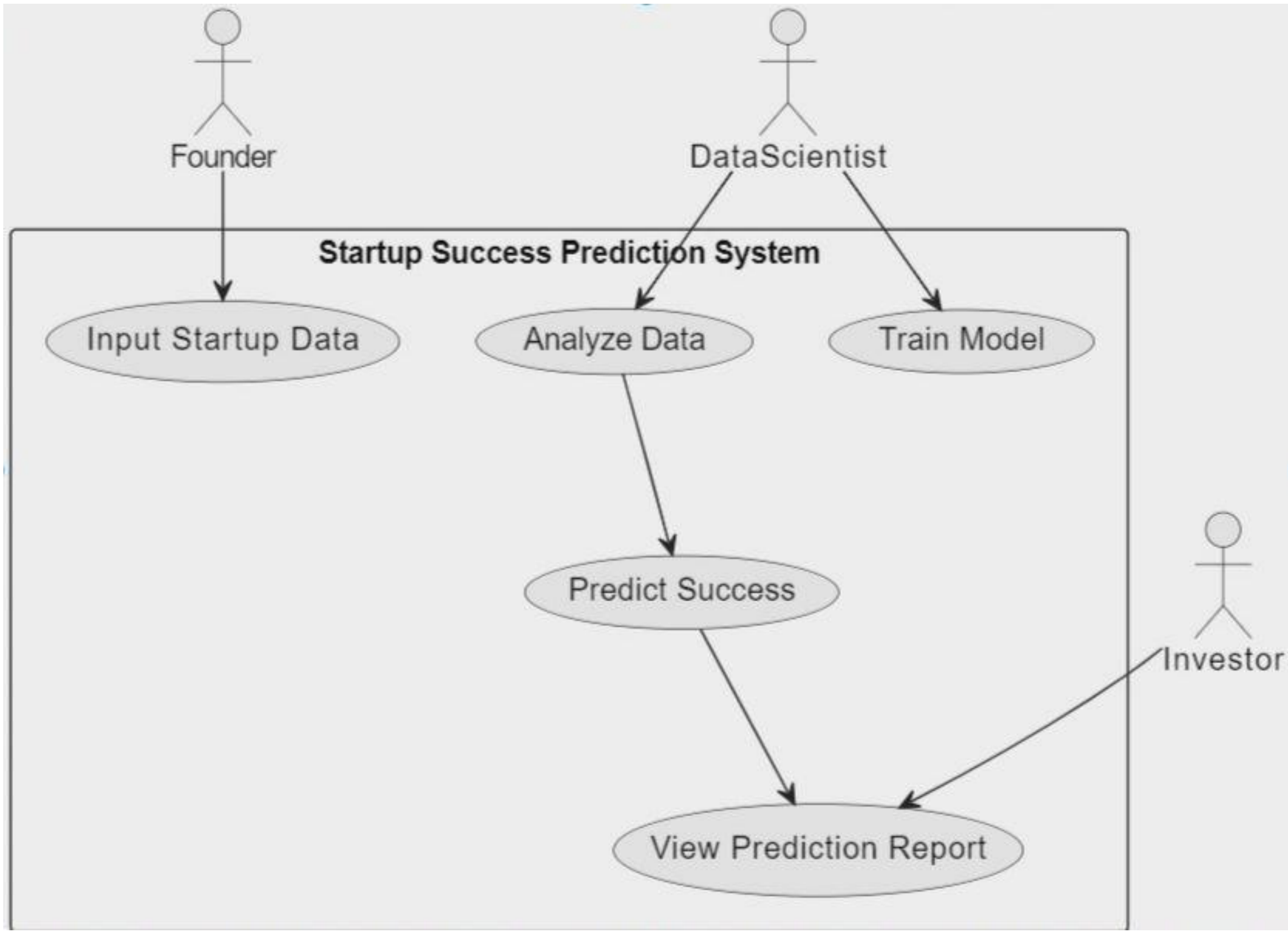
8.Feature Engineering: Feature engineering techniques involve creating new features or transforming existing ones to improve model performance. This includes techniques such as onehot encoding, feature scaling, dimensionality reduction (e.g., PCA), and feature selection.

9.Time-Series Analysis: For startups with temporal data (e.g., time-stamped financial data, user activity over time), time-series analysis techniques, such as autoregressive models, moving averages, and seasonal decomposition, can capture temporal patterns and trends.

5.PROJECT DESIGN

5.1UML DIAGRAMS

5.1.1 USE CASE DIAGRAMS



6.CONCLUSION

In conclusion, leveraging machine learning for startup success prediction projects holds immense potential to revolutionize how stakeholders in the startup ecosystem make decisions and allocate resources. By analyzing large volumes of data and identifying patterns and correlations, machine learning models can offer valuable insights into the factors that contribute to startup success. Through predictive analytics, stakeholders such as investors, incubators, accelerators, and policymakers can make more informed decisions, mitigate risks, and support high-potential startups more effectively. However, it's crucial to recognize the challenges and limitations associated with machine learning projects, including data quality issues, model interpretability concerns, and ethical considerations. Addressing these challenges requires a holistic approach that incorporates domain expertise, ethical frameworks, and responsible AI practices. Despite these challenges, the benefits of startup success prediction projects using machine learning are significant. From guiding investment decisions to fostering innovation and ecosystem development, predictive analytics has the potential to drive positive outcomes for startups, investors, and society as a whole. In conclusion, the future of startup success prediction lies at the intersection of data science, entrepreneurship, and innovation. By harnessing the power of machine learning, we can unlock new opportunities, mitigate risks, and pave the way for a more prosperous and sustainable startup ecosystem.

STARTUP SUCCESS PREDICTION

A UG PHASE-2 PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

RACHAMALLA RATHNAKAR 20UK1A0571

Under the esteemed guidance of

Dr. R. NAVEEN KUMAR

(Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD BOLLIKUNTA, WARANGAL – 506005

2020-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

BOLLIKUNTA , WARANGAL – 506005

2020-2024

CERTIFICATE OF COMPLETION MAJOR PROJECT REPORT

This is to certify that the UG Project Phase-1 Report entitled **"STARTUP SUCCESS PREDICTION USING MACHINE LEARNING "** is being submitted by **RACHAMALLA RATHNAKAR(2OUK1AO571)**in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2023-24, is a record of work carried out by them under the guidance and supervision.

Project guide	**Head of the department**
Dr. R. Naveen Kumar	**Dr . R. Naveen Kumar**
(professor)	(professor)

External

<u>ACKNOWLEDGEMENT</u>

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. Prasad Rao**, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase1 in the institute.

We extend our heartfelt thanks to **Dr. R. Naveen Kumar**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to the guide, **Dr. R. Naveen Kumar**, Assistant professor, Department of CSE for her constant support and giving necessary guidance for completion of this UG Project Phase-2

Finally,We express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

RACHAMALLA RATHNAKAR 20UK1A0571

TABLES OF CONTENT

1. INTRODUCTION

The startup ecosystem is inherently risky, with a significant proportion of new ventures failing within the first few years. Identifying the factors that contribute to a startup's success and predicting outcomes with high accuracy are paramount for investors, entrepreneurs, and other stakeholders. Traditional methods of assessing startup viability often rely on heuristic approaches and subjective judgment, which can lead to inconsistent and suboptimal decision-making.

Advancements in machine learning (ML) offer promising opportunities to enhance the predictive accuracy of startup success. By leveraging large datasets and sophisticated algorithms, ML can uncover complex patterns and relationships that are not immediately apparent through conventional analysis. This project aims to harness the power of machine learning to develop predictive models that can forecast the success of startups, thereby aiding stakeholders in making data-driven decisions.

Our approach involves assembling a comprehensive dataset that includes financial metrics, market conditions, founder backgrounds, and other relevant variables. We will preprocess this data to handle missing values, normalize features, and engineer new variables that may contribute to the predictive power of our models. Several machine learning techniques, such as logistic regression, random forests, and gradient boosting machines, will be employed and rigorously evaluated to determine their effectiveness in predicting startup outcomes.

2.CODE SNIPPETS

2.1 DATA COLLECTION

We obtain the dataset from Kaggle

(link: https://www.kaggle.com/datasets/manishkc06/startup-success-prediction) to train and test our system

Importing The Libraries

```
In [1]: import numpy as np # linear algebra
        import pandas as pd # data processing
        pd.set_option('display.max_columns', None)
        import os
        import matplotlib.pyplot as plt
        import seaborn as sns
        from sklearn.model_selection import train_test_split
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.model_selection import GridSearchCV
        from sklearn.metrics import classification_report
```

Read The Dataset

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

In pandas we have a function called read_csv() to read the dataset. As a parameter we have to give the directory of the csv file.

```
data = pd.read_csv('startup data.csv')
```

```
data.head()
```

	Unnamed: 0	state_code	latitude	longitude	zip_code	id	city	Unnamed: 6	name	labels	founded_at	closed_at	first_funding_at	last_fun
0	1005	CA	42.358880	-71.056820	92101	c:6669	San Diego	NaN	Bandsintown	1	1/1/2007	NaN	4/1/2009	
1	204	CA	37.238916	-121.973718	95032	c:16283	Los Gatos	NaN	TriCipher	1	1/1/2000	NaN	2/14/2005	12/
2	1001	CA	32.901049	-117.192656	92121	c:65620	San Diego	San Diego CA 92121	Plixi	1	3/18/2009	NaN	3/30/2010	3/
3	738	CA	37.320309	-122.050040	95014	c:42668	Cupertino	Cupertino CA 95014	Solidcore Systems	1	1/1/2002	NaN	2/17/2005	4/
4	1002	CA	37.779281	-122.419236	94105	c:65806	San Francisco	San Francisco CA 94105	Inhale Digital	0	8/1/2010	10/1/2012	8/1/2010	

2.2Exploratory Data Analysis

Visualisation

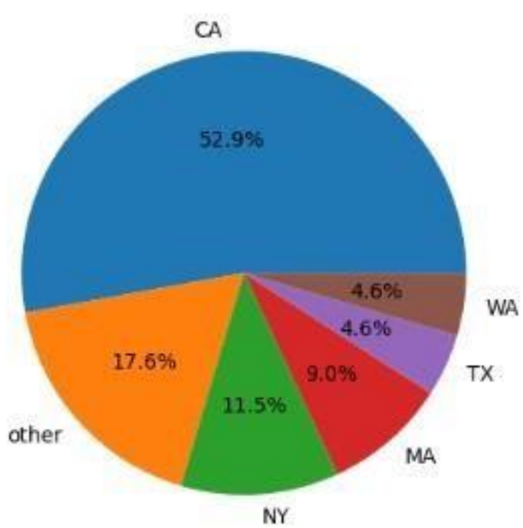
Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

Univariate & Multivariate Analysis

```
data['State'] = 'other'
data.loc[(data['state_code'] == 'CA'), 'State'] = 'CA'
data.loc[(data['state_code'] == 'NY'), 'State'] = 'NY'
data.loc[(data['state_code'] == 'MA'), 'State'] = 'MA'
data.loc[(data['state_code'] == 'TX'), 'State'] = 'TX'
data.loc[(data['state_code'] == 'WA'), 'State'] = 'WA'
```

```
state_count = data['State'].value_counts()
plt.pie(state_count, labels = state_count.index, autopct = '%1.1f%%')
plt.show()
```

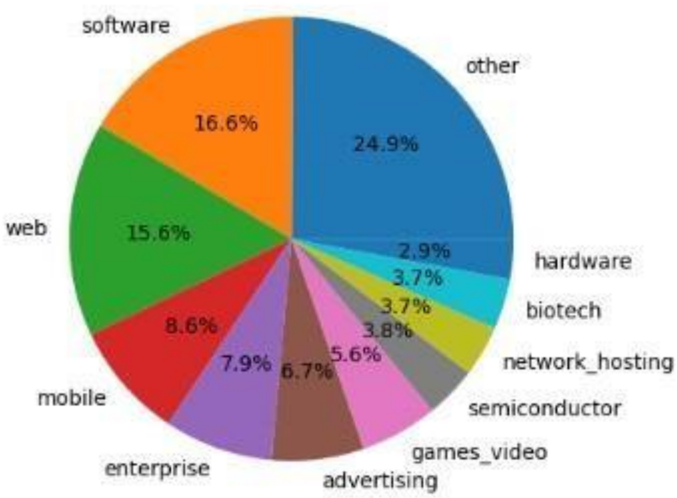
It assigns the value 'other' to the 'State' column for all rows in the 'data' DataFrame. Then, specific rows where the 'state_code' matches certain values (CA, NY, MA, TX, WA) are assigned their respective state names in the 'State' column. The 'value_counts()' method is used to count the occurrences of each unique value in the 'State' column, and a pie chart is created to visualize the distribution of states.



```
data['category'] = 'other'
data.loc[(data['category_code'] == 'software'), 'category'] = 'software'
data.loc[(data['category_code'] == 'web'), 'category'] = 'web'
data.loc[(data['category_code'] == 'mobile'), 'category'] = 'mobile'
data.loc[(data['category_code'] == 'enterprise'), 'category'] = 'enterprise'
data.loc[(data['category_code'] == 'advertising'), 'category'] = 'advertising'
data.loc[(data['category_code'] == 'games_video'), 'category'] = 'games_video'
data.loc[(data['category_code'] == 'semiconductor'), 'category'] = 'semiconductor'
data.loc[(data['category_code'] == 'network_hosting'), 'category'] = 'network_hosting'
data.loc[(data['category_code'] == 'biotech'), 'category'] = 'biotech'
data.loc[(data['category_code'] == 'hardware'), 'category'] = 'hardware'
```

```
category_count = data['category'].value_counts()
plt.pie(category_count, labels = category_count.index, autopct = '%1.1f%%')
plt.show()
```

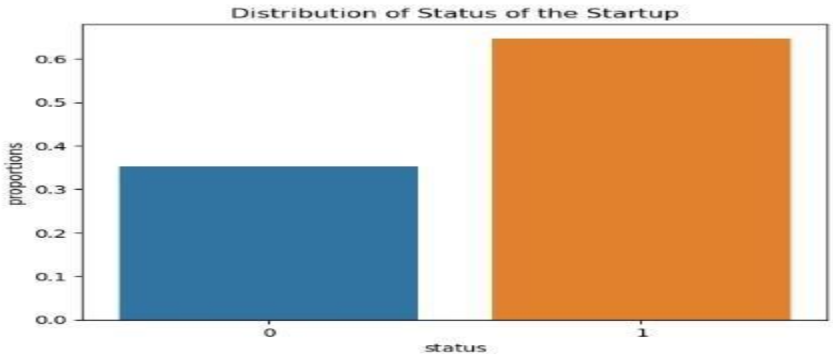
The given code snippet performs a univariate analysis. It assigns the value 'other' to the 'category' column for all rows in the 'data' DataFrame. Then, specific rows matching certain 'category_code' values are assigned corresponding categories in the 'category' column. The 'value_counts()' method is used to count the occurrences of each unique category in the 'category' column, and a pie chart is created to visualize the distribution of categories.



- **Distribution of Status of Startup**

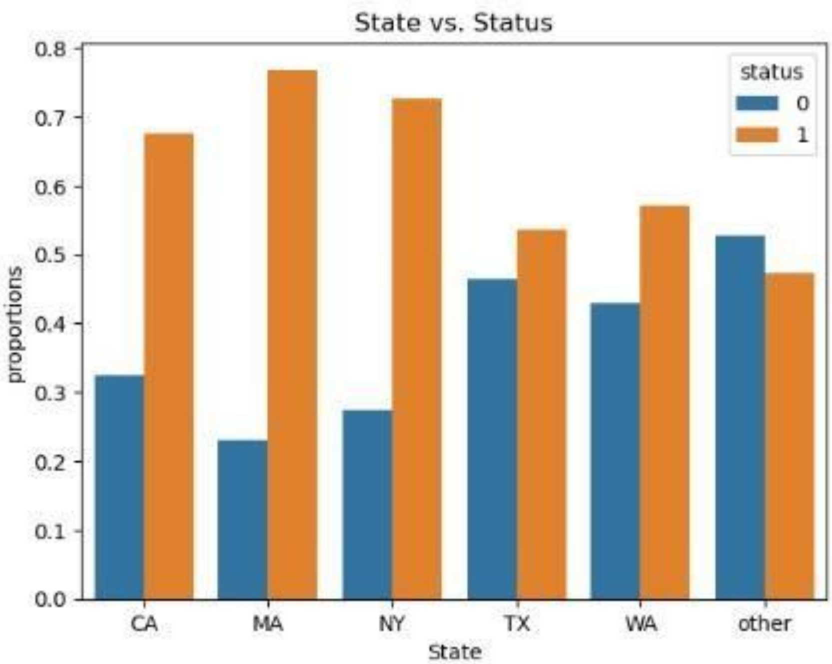
```
prop_df = data.groupby('status').size().reset_index(name = 'counts')
prop_df['proportions'] = prop_df['counts']/prop_df['counts'].sum()
```

```
sns.barplot(data = prop_df, x = 'status', y = 'proportions')
plt.title('Distribution of Status of the Startup')
```



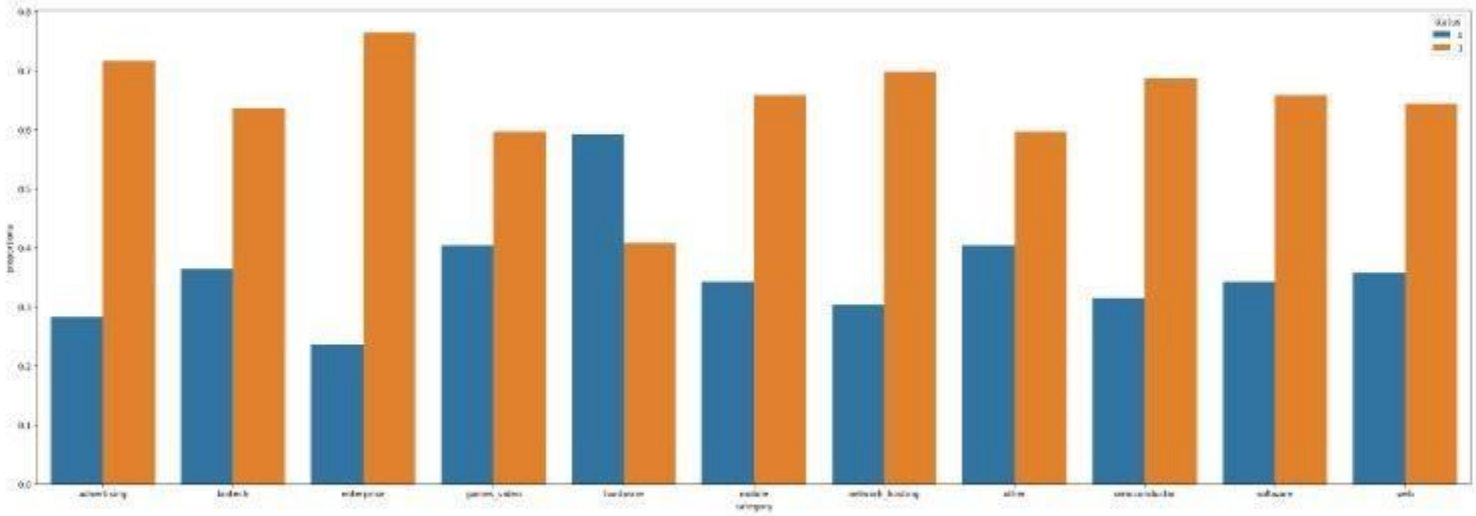
• **State Vs. Staus**

```
prop_df = data.groupby(['State','status'],group_keys = True).size().reset_index(name='count')
prop_df['proportions'] = prop_df.groupby('State')['count'].apply(lambda x: x/x.sum())
sns.barplot(data = prop_df, x = 'State', y = 'proportions',hue = 'status')
plt.title('State vs. Status')
```



- State Vs. Category

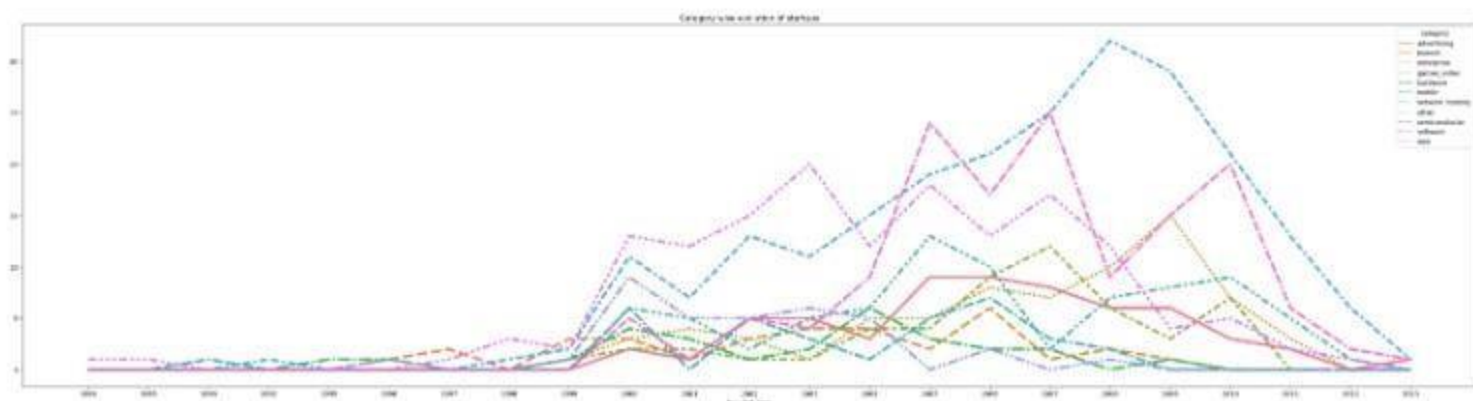
```
fig, ax = plt.subplots(figsize = (30,10))
prop_df = data.groupby(['category','status']).size().reset_index(name='counts')
prop_df['proportions'] = prop_df.groupby('category')['counts'].apply(lambda x: x/float(x.sum()))
sns.barplot(data = prop_df, x = 'category', y = 'proportions',hue = 'status')
```



- Category Vs. Founded- Year

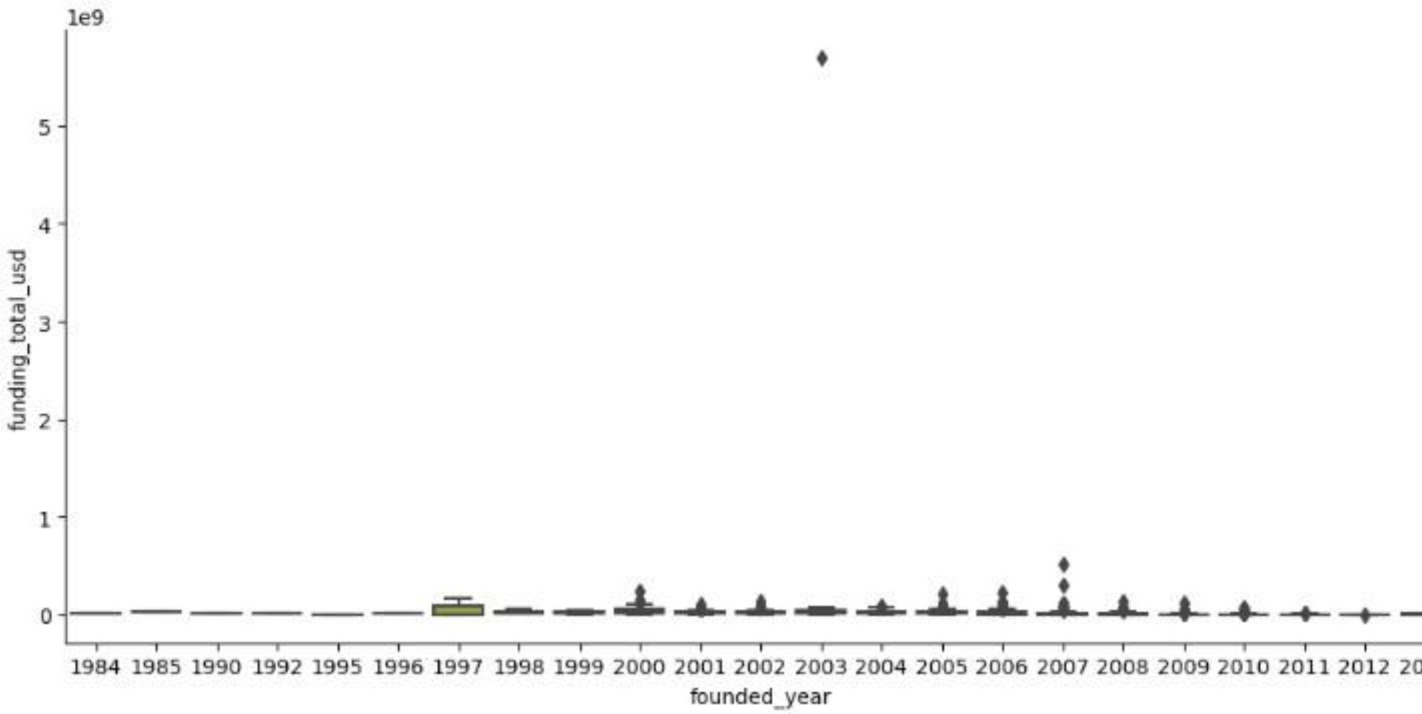
```
cat_year = pd.crosstab(index = data['founded_year'], columns = data['category'])
```

```
fig, ax = plt.subplots(figsize=(40,10))
sns.lineplot(data = cat_year, lw = 4)
plt.title('Category wise evolution of startups')
```



- Founded- Year Vs. Total Funding

```
sns.catplot(data=data, x="founded_year", y="funding_total_usd",
            kind="box",height=5, aspect=2,order = ['1984','1985','1990','1992','1995',
                                                  '1996','1997','1998','1999','2000',
                                                  '2001','2002','2003','2004','2005',
                                                  '2006','2007','2008','2009','2010',
                                                  '2011','2012','2013'])
```



- Statistical Analysis

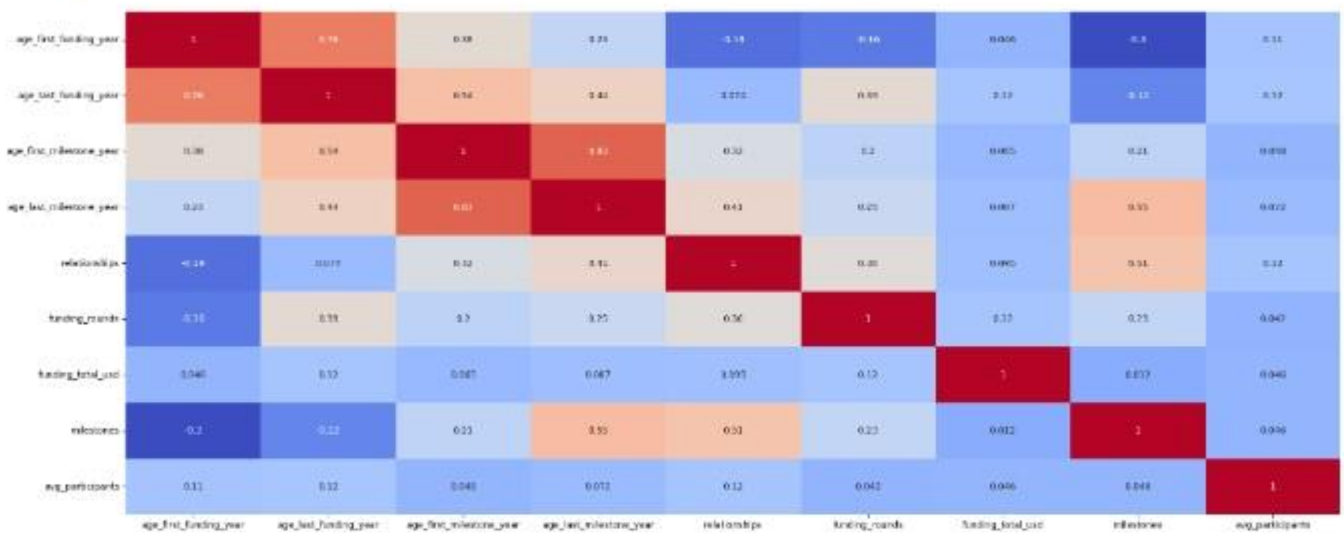
```
data.describe(include = ['float64','int64'])
```

	age_first_funding_year	age_last_funding_year	age_first_milestone_year	age_last_milestone_year	relationships	funding_rounds	funding_total_usd
count	914.000000	914.000000	914.000000	914.000000	914.000000	914.000000	9.140000e+02
mean	2.270677	3.947489	2.613675	3.985237	7.680525	2.312910	2.552193e+07
std	2.457724	2.930272	2.816996	3.382481	7.265480	1.394131	1.905590e+08
min	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.100000e+04
25%	0.568475	1.669175	0.000000	1.000000	3.000000	1.000000	2.712500e+06
50%	1.449350	3.545200	2.000000	3.754800	5.000000	2.000000	1.000000e+07
75%	3.578075	5.558900	4.002700	6.026050	10.000000	3.000000	2.465264e+07
max	21.895900	21.895900	24.684900	24.684900	63.000000	10.000000	5.700000e+09

- Correlation Plot

```
fig, ax = plt.subplots(figsize = (30,10))
corr = data.select_dtypes(include=['int64','float64']).corr()
sns.heatmap(corr, cmap = 'coolwarm', annot = True)
```

```
<Axes: >
```



Pre- Processing

- Reducing the Number of Categories

```
print(data['state_code'].equals(data['state_code.1']))
```

```
False
```

```
df = data.loc[data['state_code'] != data['state_code.1']]
df.style.set_properties(**{'background-color': 'yellow'}, subset=['state_code','state_code.1'])
```

	state_code	city	labels	founded_at	closed_at	first_funding_at	last_funding_at	age_first_funding_year	age_last_funding_year
515	CA	Menlo Park	0	1/1/2005	9/1/2010	3/1/2007	4/15/2008	2.161600	3.287700

```
state = data['state_code'].value_counts().to_frame()
state['proportion'] = state['state_code']/sum(state['state_code'])*100
state
```

2.3MODEL BUILDING

Define Pre-Processing And Modelling Function

- The code snippet performs a grid search using the GridSearchCV class to find the best combination of hyperparameters for a Random Forest classifier. The parameter grid is defined with different values for 'n_estimators', 'max_depth', 'min_samples_split', 'min_samples_leaf', and 'bootstrap'. The grid search is performed using 5-fold cross-validation (cv=5) and parallelized (-1 for n_jobs). After fitting the grid search object to the training data, it prints the best parameters found based on the evaluation of different parameter combinations.

```
#rf = RandomForestClassifier()
param_grid = {'n_estimators':[100,200,300],
              'max_depth':[10,20,30],
              'min_samples_split':[2,4,6],
              'min_samples_leaf':[1,2,3],
              'bootstrap':[True,False]}

grid_search = GridSearchCV(estimator=rf, param_grid = param_grid, cv = 5, n_jobs = -1, verbose = False)
grid_search.fit(x_train, y_train)

print('Best parameters:', grid_search.best_params_)
```

-

```
Best parameters: {'bootstrap': False, 'max_depth': 20, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 100}
```

- The _scale function takes in training and validation data along with a list of features. It applies the StandardScaler to standardize the numerical features in the training data. It then uses the computed scaler to transform both the training and validation data. The function returns the updated training and validation data with the scaled features. This ensures that the features have zero mean and unit variance, which can be beneficial for certain machine learning algorithms.

```
#rf = RandomForestClassifier(n_estimators=100,bootstrap=False,max_depth=20,min_samples_leaf=2, min_samples_split=2)
model_rf = rf.fit(x_train,y_train)
y_pred_rf = model_rf.predict(x_test)
cr_rf = classification_report(y_pred_rf, y_test)
print(cr_rf)
```

```
              precision    recall  f1-score   support

           0       0.59      0.78      0.67        68
           1       0.92      0.82      0.87       207

    accuracy                           0.81       275
   macro avg       0.75      0.80      0.77       275
weighted avg       0.84      0.81      0.82       275
```

The classification report provides an evaluation of the model's performance. For class 0, the precision is 0.59, recall is 0.78, and F1-score is 0.67. For class 1, the precision is 0.92, recall is 0.82, and F1-score is 0.87. The overall accuracy of the model is 0.81. The macro average of precision, recall, and F1-score is 0.75, 0.80, and 0.77, respectively. The weighted average of precision, recall, and F1-score is 0.84, 0.81, and 0.82, respectively.

2.4. MODEL DEPLOYMENT

Save And Load The Best Model

The provided code snippet demonstrates the usage of joblib library for training and saving a Random Forest Classifier model. First, the Random Forest Classifier is trained using the training data X_train and target labels y_train. The trained model is then saved to a file named 'random_forest_model.pkl' using the joblib.dump() function.

Later, to make predictions on new data, the saved model is loaded from the file using the joblib.load() function. The loaded model can then be used to make predictions on the test data X_test by calling the model.predict() method, which returns an array of predicted labels.

```
# Save the trained model to a file
joblib.dump(model, 'random_forest_model.pkl')
```

```
['random_forest_model.pkl']
```

We save the model using the pickle library into a file named random_forest_model.pkl

Integrate With Web Framework

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI. This section has the following tasks

? Building HTML Pages

? Building server-side script

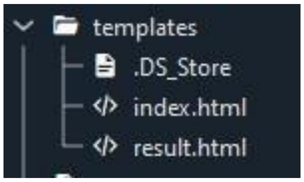
? Run the web application

Building HTML Pages:

For this project we create two HTML files namely

- Index.html
- Results.html
- home.html
- adaptivity.html

And we will save them in the templates folder.



Build Python Code

Create a new app.py file which will be store in the Flask folder.

- Import the necessary Libraries.

```python
from flask import Flask, render_template, request
import joblib
```

- This code uses the joblib module to load a saved machine learning model and a saved preprocessing object. The joblib.load function is used to load the model object from a file called model.pkl and the scaler object from a file called random_forest_model.pkl. These objects are then stored in the model and scaler variables, respectively. The loaded model object can be used to make predictions on new data and the loaded scaler object can be used to preprocess the data in the same way as it was done during training. This process of loading saved model and preprocessing objects can save time and resources when working on a new project or dataset.

```python
app = Flask(__name__)
model = joblib.load('random_forest_model.pkl')
```

- This code creates a new instance of a Flask web application using the Flask class from the Flask library. The __name__ argument specifies the name of the application's module or package.

```python
app = Flask(__name__)
```

- The code snippet defines a route '/' for the Flask application. When a user accesses the homepage of the application, it calls the 'home' function. This function renders the 'index.html' template, which will be displayed as the homepage of the application.

```python
@app.route('/')
def home():
    return render_template('index.html')
```

- The code snippet defines a route '/predict' for the Flask application, which expects a POST request. When this route is accessed, the 'predict' function is called. Inside the function, it retrieves input values from a form submitted with the POST request. The values are converted to float and stored in respective variables for further processing.

```python
@app.route('/predict', methods=['POST'])
def predict():
    # Get the input values from the form
    age_first_funding_year = float(request.form['age_first_funding_year'])
    age_last_funding_year = float(request.form['age_last_funding_year'])
    age_first_milestone_year = float(request.form['age_first_milestone_year'])
    age_last_milestone_year = float(request.form['age_last_milestone_year'])
    relationships = float(request.form['relationships'])
    funding_rounds = float(request.form['funding_rounds'])
    funding_total_usd = float(request.form['funding_total_usd'])
    milestones = float(request.form['milestones'])
    avg_participants = float(request.form['avg_participants'])
```

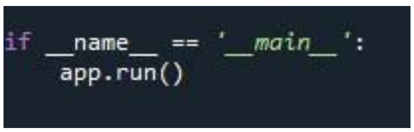
- The code snippet creates a list called 'input_data' containing the input values retrieved from the form. It then uses a loaded model to make a prediction based on the input data. The predicted label is mapped to a meaningful output ('Acquired' or 'Closed'). Finally, the 'result' variable is passed to the 'result.html' template for rendering the prediction result on a webpage.

```python
# Create a list with the input values
input_data = [
    age_first_funding_year,
    age_last_funding_year,
    age_first_milestone_year,
    age_last_milestone_year,
    relationships,
    funding_rounds,
    funding_total_usd,
    milestones,
    avg_participants
]

# Make a prediction using the loaded model
prediction = model.predict([input_data])[0]

# Map the predicted label to a meaningful output
if prediction == 1:
    result = 'Acquired'
else:
    result = 'Closed'

# Render the prediction result
return render_template('result.html', result=result)
```

-

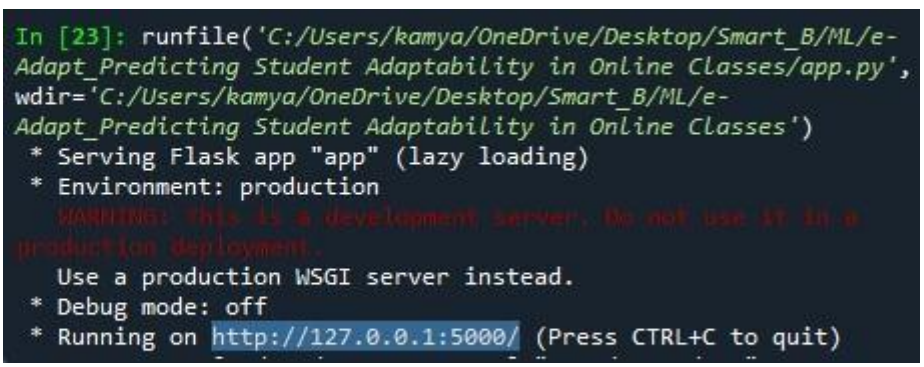
Main Function:

This code runs the Flask application if the script is being executed directly (i.e. not imported as a module in another script). The if __name__ == '__main__': line checks if the script is the main module being executed, and if so, runs the Flask application using the app.run() method. This method starts the Flask development server, allowing the application to be accessed via a web browser at the appropriate URL.



Run The Web Application

When you run the “app.py” file this window will open in the console or output terminal. Copy the URL given in the form http://127.0.0.1:5000 and paste it in the browser.



6.ADVANTAGES

1.Data-Driven Insights: Machine learning algorithms analyze large volumes of data to identify patterns and correlations that may not be apparent through traditional analysis methods. This datadriven approach provides valuable insights into the factors influencing startup success.

2.Predictive Accuracy: Machine learning models can make predictions with high accuracy by learning from historical data and identifying complex relationships between input features and startup outcomes. This allows stakeholders to make more informed decisions based on reliable predictions.

3.Scalability: Machine learning techniques can scale to handle large and diverse datasets, making them suitable for analyzing startup data, which often includes multiple variables and dimensions. As the startup grows and collects more data, machine learning models can adapt and continue to provide accurate predictions.

4.Automation: Machine learning automates the process of analyzing data and making predictions, reducing the need for manual intervention and saving time for stakeholders. This automation allows startups to focus on core business activities while leveraging predictive analytics for strategic decision-making.

5.Personalized Recommendations: Machine learning models can generate personalized recommendations for startups based on their unique characteristics and historical performance. These recommendations help startups identify areas for improvement and optimize their strategies for success.

6.Real-Time Insights: Machine learning models can analyze data in real-time, providing timely insights into market trends, customer behavior, and competitive landscape. This enables startups to adapt quickly to changing market conditions and make data-driven decisions in a dynamic environment.

7.Risk Mitigation: By predicting potential risks and challenges, machine learning models help startups proactively mitigate risks and make informed decisions to minimize negative outcomes. This risk-aware approach enhances the resilience and sustainability of startups in competitive markets.

8.Competitive Advantage: Startups that leverage machine learning for success prediction gain a competitive advantage by using advanced analytics to drive innovation, optimize resource allocation, and capitalize on market opportunities. This enables them to differentiate themselves and stay ahead of competitors.

9.Continuous Improvement: Machine learning models can continuously learn and improve over time as more data becomes available and as new insights are gained. This iterative learning process allows startups to refine their predictive models and adapt to evolving market dynamics.

10.Cost-Effectiveness: While initial investment in developing machine learning models may be required, the long-term benefits of improved decision-making and strategic planning outweigh the costs. By maximizing resources and minimizing risks, startups can achieve cost-effective growth and sustainable success.

7.DISADVANTAGES:

1.Data Quality Issues: Machine learning models are highly dependent on the quality and reliability of the input data. Startup data may be incomplete, noisy, or biased, leading to inaccurate predictions and unreliable insights.

2.Overfitting: Machine learning models may overfit the training data, capturing noise or irrelevant patterns that do not generalize well to new data. This can lead to poor performance on unseen data and inaccurate predictions for real-world scenarios.

3.Interpretability Challenges: Some machine learning models, particularly complex ones like deep learning models, lack interpretability, making it difficult to understand the underlying reasons behind predictions. This can reduce trust in the model and hinder stakeholders' ability to make informed decisions based on the predictions.

4.Model Complexity and Resource Requirements: Developing and training sophisticated machine learning models requires significant computational resources, including highperformance hardware and specialized expertise. Startups with limited resources may face challenges in implementing and maintaining such models.

5.Data Privacy and Security Risks: Machine learning projects involving sensitive data, such as financial records or customer information, may pose data privacy and security risks. Mishandling or unauthorized access to data could lead to legal and reputational consequences for the startup.

6.Bias and Fairness Issues: Machine learning models may inadvertently encode biases present in the training data, leading to unfair or discriminatory outcomes. This can perpetuate existing inequalities and disadvantage certain groups or demographics, undermining the ethical considerations of the project.

7.Continuous Model Maintenance: Machine learning models require continuous monitoring and maintenance to ensure their performance remains optimal over time. Changes in the data distribution, market conditions, or business environment may necessitate updates to the model, which can be time-consuming and resource-intensive.

8.Limited Domain Understanding: Machine learning models may lack domain-specific knowledge and context, leading to suboptimal predictions or misinterpretation of results. Incorporating domain expertise and human judgment into the modeling process is essential for enhancing the relevance and accuracy of predictions.

9.Regulatory Compliance: Machine learning projects may be subject to regulatory requirements, such as data protection laws or industry-specific regulations. Ensuring compliance with these regulations adds complexity and overhead to the project and may constrain the types of data and algorithms that can be used.

10.Ethical Considerations: Machine learning models have the potential to impact individuals and society in profound ways, raising ethical considerations around issues such as fairness, transparency, accountability, and unintended consequences. Startups must carefully consider the ethical implications of their predictive models and prioritize responsible AI practices.

8.APPLICATIONS:

1.Venture Capital and Investment:

Venture capital firms and investors can use machine learning models to evaluate startup investment opportunities and identify high-potential startups for funding. Predictive analytics can assist in assessing the likelihood of startup success based on various factors such as market potential, team expertise, and competitive landscape.

2.Business Incubators and Accelerators:

Business incubators and accelerators can leverage machine learning to support startup growth and success. Predictive models can assist in selecting startups for incubation programs, providing tailored support and resources based on each startup's unique characteristics and growth potential.

3.Startup Ecosystem Development:

Governments and organizations focused on fostering startup ecosystems can use machine learning to analyze ecosystem dynamics, identify emerging trends, and predict future success factors. This information can inform policy decisions, investment strategies, and ecosystem development initiatives.

4.Startup Support Services:

Startup support services, such as mentorship programs, co-working spaces, and networking events, can benefit from machine learning-driven insights. Predictive analytics can help match startups with relevant mentors, collaborators, and resources based on compatibility and growth potential.

5.Startup Consulting and Advisory Services:

Consulting firms and advisory services specializing in startups can offer machine learning-based predictive analytics as part of their service offerings. This includes providing strategic advice, risk assessment, and growth recommendations based on data-driven insights into startup success factors.

6.Entrepreneurship Education and Training:

Universities and educational institutions offering entrepreneurship programs can integrate machine learning-based startup success prediction projects into their curriculum. Students can gain practical experience in data analysis, predictive modeling, and decision-making in the context of startup ventures.

7.Startup Competitions and Challenges:

Startup competitions and challenges can incorporate machine learning-based evaluation criteria to assess the viability and potential of participating startups. Predictive models can help identify top-performing startups and allocate prizes or opportunities accordingly.

8.Startup Support Platforms:
Online platforms and communities that provide support and resources for startups can integrate machine learning-based recommendation systems. These systems can suggest relevant content, connections, and opportunities to startups based on their profile and goals, enhancing user engagement and satisfaction.

9.Corporate Innovation Programs:
Corporations engaging in open innovation and collaboration with startups can use machine learning to identify promising startup partners and innovation opportunities. Predictive analytics can assist in evaluating startups' alignment with corporate goals, innovation potential, and compatibility with existing business units.

10.Startup Survival Prediction:
In addition to predicting startup success, machine learning can also be applied to predict the likelihood of startup failure or survival. This information can help stakeholders, including founders, investors, and policymakers, make more informed decisions about resource allocation and risk management.

9.CONCLUSION:

In conclusion, leveraging machine learning for startup success prediction projects holds immense potential to revolutionize how stakeholders in the startup ecosystem make decisions and allocate resources. By analyzing large volumes of data and identifying patterns and correlations, machine learning models can offer valuable insights into the factors that contribute to startup success. Through predictive analytics, stakeholders such as investors, incubators, accelerators, and policymakers can make more informed decisions, mitigate risks, and support high-potential startups more effectively.

However, it's crucial to recognize the challenges and limitations associated with machine learning projects, including data quality issues, model interpretability concerns, and ethical considerations. Addressing these challenges requires a holistic approach that incorporates domain expertise, ethical frameworks, and responsible AI practices.

Despite these challenges, the benefits of startup success prediction projects using machine learning are significant. From guiding investment decisions to fostering innovation and ecosystem development, predictive analytics has the potential to drive positive outcomes for startups, investors, and society as a whole.

In conclusion, the future of startup success prediction lies at the intersection of data science, entrepreneurship, and innovation. By harnessing the power of machine learning, we can unlock new opportunities, mitigate risks, and pave the way for a more prosperous and sustainable startup ecosystem.

10.FUTURE SCOPE:

The future scope for startup success prediction projects using machine learning is vast and promising, with several emerging trends and opportunities on the horizon:

1.Advanced Machine Learning Techniques:
Future research and development efforts will focus on advancing machine learning techniques to improve predictive accuracy, scalability, and interpretability. This includes exploring deep learning architectures, reinforcement learning algorithms, and meta-learning approaches tailored to startup success prediction.

2.Integration of Multimodal Data:
As startups increasingly operate in complex and interconnected environments, there is a growing need to integrate multimodal data sources, including textual, visual, and sensor data. Future projects will leverage techniques such as natural language processing (NLP), computer vision, and sensor fusion to extract valuable insights from diverse data types.

3.Real-Time Prediction and Adaptation:
With the proliferation of IoT devices and real-time data streams, future startup success prediction projects will focus on developing models capable of making predictions in real-time and adapting to dynamic changes in the business environment. This enables stakeholders to respond promptly to emerging opportunities and threats.

4.Explainable AI and Fairness:
Addressing concerns around model interpretability and fairness will be a key focus area in future projects. Efforts will be made to develop explainable AI techniques that provide transparent insights into model predictions and mitigate biases and discrimination in predictive algorithms.

5.Personalized and Context-Aware Recommendations:
Future projects will aim to deliver personalized and context-aware recommendations to startups based on their unique characteristics, goals, and challenges. This involves leveraging techniques such as contextual bandits, reinforcement learning, and multi-armed bandit algorithms to optimize decision-making and resource allocation.

6.Integration with Blockchain and Decentralized Finance (DeFi):
The integration of machine learning with blockchain technology and decentralized finance (DeFi) platforms presents new opportunities for startup success prediction. Future projects may explore the use of smart contracts, tokenomics, and decentralized prediction markets to incentivize accurate predictions and allocate resources efficiently.

7.Collaborative and Federated Learning:

Collaborative and federated learning approaches will enable startups to leverage distributed data sources while preserving data privacy and security. Future projects will explore federated learning techniques that allow multiple parties to collaboratively train machine learning models without sharing sensitive data.

8.Hybrid Models and Ensemble Learning:

Hybrid models combining multiple machine learning techniques, such as symbolic AI and connectionist AI, will emerge as a powerful approach for startup success prediction. Ensemble learning methods will also gain popularity, combining the strengths of different algorithms to improve predictive performance and robustness.

9.Ethical and Responsible AI Practices:

Future projects will prioritize ethical and responsible AI practices, incorporating principles such as transparency, accountability, and fairness into the design and deployment of predictive models. This involves adopting ethical guidelines, regulatory frameworks, and governance mechanisms to ensure the responsible use of machine learning in startup success prediction.

10.Cross-Domain Collaboration and Knowledge Sharing:

Collaboration across domains, including academia, industry, and government, will drive innovation and knowledge sharing in the field of startup success prediction. Future projects will leverage interdisciplinary approaches, open-source initiatives, and collaborative platforms to accelerate progress and tackle complex challenges collectively.

11.BIBILOGRAPHY

1.Y.B. Altun ”Pandemic Fuels Global Growth Of Entrepreneurship And Startup.

2.S. Korreck. The Indian Startup Ecosystem: Drivers, Challenges and Pillars of Support. ORF Occasional Paper.

3.M. Van Gelderen, R. Thurik, and N. Bosma. Success and risk factors in the pre-startup phase. Small Business Economics, 24(4):365–380, 2005.

4.T. M. Begley and W.-L. Tan. The socio-cultural environment for entrepreneurship: A comparison between east asian and anglo-saxon countries. Journal of international business studies, pages 537–553, 2001. 5.R. Dickinson. Business failure rate. American Journal of Small Business, 6(2):17–25, 1981.

12.APPENDIX

home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Startup Parameters</title>
  <style>      body {        display:
flex;       justify-content: center;
align-items: center;         height:
100vh;        margin: 0;        font-
family:       Arial,       sans-serif;
background-color: #f0f0f0;
    }
    .container {       background-color:
#ffffff;      padding: 20px;      border-
radius: 8px;      box-shadow: 0 0 10px
rgba(0, 0, 0, 0.1);      max-width: 400px;
width: 100%;
    }    h1 {      text-align:
center;                margin-
bottom: 20px;
    }
    .form-group  {           margin-
bottom: 15px;
    }
    .form-group label {

      display: block;

      margin-bottom: 5px;
    }

    .form-group      input      {
width:  calc(100%   -   20px);
padding: 10px;
```

```
    border: 1px solid #ccc;          border-
radius: 4px;
   }
   .btn-predict {          display:
block;               width:  100%;
padding: 10px;     background-
color: #007bff;     color: white;
border: none;     border-radius:
4px;     cursor: pointer;     font-
size: 16px;
   }
   .btn-predict:hover {         background-
color: #0056b3;
   }
  </style>
</head>
<body>
  <div class="container">
   <h2>ENTER STARTUP PARAMETERS</h2>
   <form action="/predict" method="post">
    <div class="form-group">
      <label for="ageFirstFunding">Age at First Funding Year:</label>
      <input type="number" id="ageFirstFunding" name="ageFirstFunding">
    </div>
    <div class="form-group">
      <label for="ageLastFunding">Age at Last Funding Year:</label>
      <input type="number" id="ageLastFunding" name="ageLastFunding">
    </div>
    <div class="form-group">
      <label for="ageFirstMilestone">Age at First Milestone Year:</label>
      <input type="number" id="ageFirstMilestone" name="ageFirstMilestone">
    </div>
    <div class="form-group">
      <label for="ageLastMilestone">Age at Last Milestone Year:</label>
      <input type="number" id="ageLastMilestone" name="ageLastMilestone">
    </div>
    <div class="form-group">
      <label for="numRelationship">Number of Relationships:</label>
      <input type="number" id="numRelationship" name="numRelationship">
```

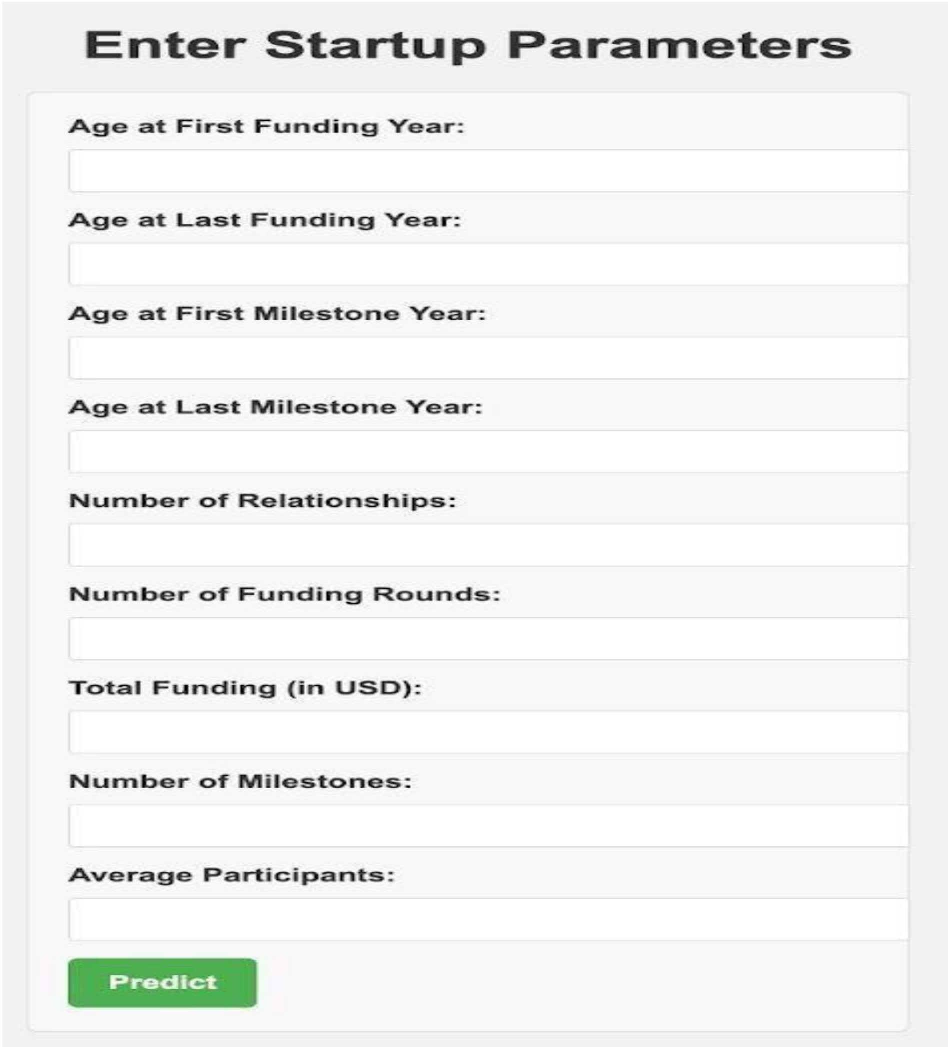
```
      </div>
      <div class="form-group">
        <label for="numFundingRounds">Number of Funding Rounds:</label>
        <input type="number" id="numFundingRounds" name="numFundingRounds">
      </div>
      <div class="form-group">
        <label for="totalFunding">Total Funding (in USD):</label>
        <input type="number" id="totalFunding" name="totalFunding">
      </div>
      <div class="form-group">
        <label for="numMilestones">Number of Milestones:</label>
        <input type="number" id="numMilestones" name="numMilestones">
      </div>
      <div class="form-group">
        <label for="numParticipants">Number of Participants:</label>
        <input type="number" id="numParticipants" name="numParticipants">
      </div>
      <button type="submit" class="btn-predict">Predict</button>
    </form>
  </div>
</body> </html>
```

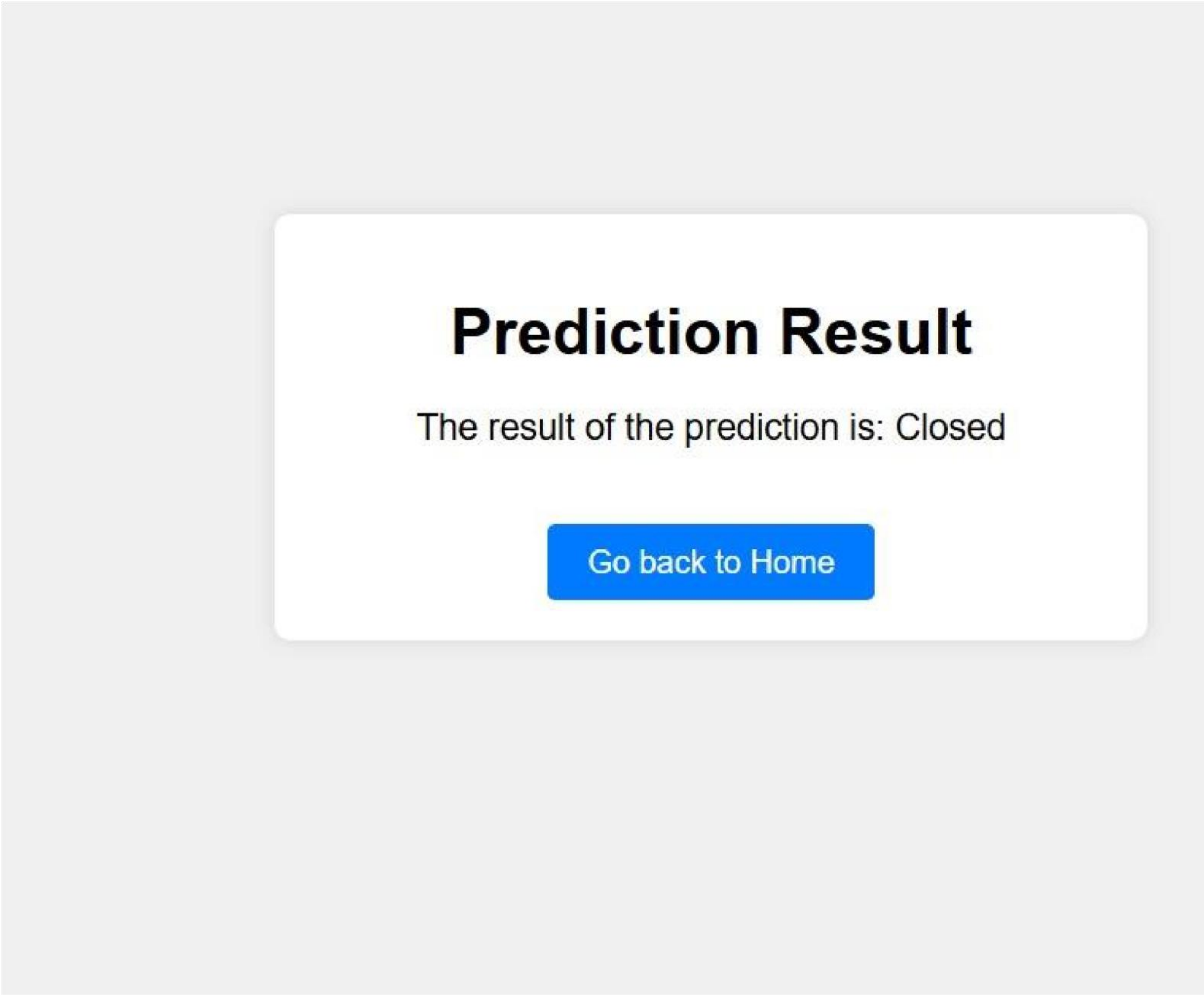
result.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Prediction Result</title>
  <style>     body {        display:
flex;        justify-content: center;
align-items: center;         height:
100vh;        margin: 0;        font-
family:        Arial,        sans-serif;
background-color: #f0f0f0;
    }
    .container {       background-color:
#ffffff;       padding: 20px;       border-
radius: 8px;       box-shadow: 0 0 10px
rgba(0, 0, 0, 0.1);       max-width: 400px;
width: 100%;       text-align: center;
    }      h1 {         margin-
bottom: 20px;
    }
p {
      font-size: 18px;
    }     a {       display: inline-
block;       margin-top: 20px;
padding: 10px 20px;
background-color: #007bff;
color: white;       text-
decoration: none;       border-
radius: 4px;
    }
    a:hover {
```

```
      background-color: #0056b3;

    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Prediction Result</h1>
    <p>The result of the prediction is: {{ result }}</p>
    <a href="/">Go back to Home</a>
  </div>
</body> </html>
```

FINAL OUTPUT




Prediction Result
The result of the prediction is: Closed
Go back to Home