Project Title Name

A Project Report

submitted in partial fulfillment of the requirements

of

Industrial Artificial Intelligence with cloud computing

by

Patel Dhyeykumar Rajubhai,

Panchal Jignesh Bharatbhai,

Patil Kunal Sunilbhai,

Patel Pratham Chiragbhai,

Under the Esteemed Guidance of

Jay Rathod

ACKNOWLEDGEMENT

We would like to take this opportunity to express our deep sense of gratitude to all individuals who helped us directly or indirectly during this thesis work.

Firstly, we would like to thank my supervisor, …………….., for being a great mentor and the best adviser I could ever have. His advice, encouragement and critics are source of innovative ideas, inspiration and causes behind the successful completion of this dissertation. The confidence shown on me by him was the biggest source of inspiration for me. It has been a privilege working with him from last one year. He always helped me during my thesis and many other aspects related to academics. His talks and lessons not only help in thesis work and other activities of college but also make me a good and responsible professional.

……...

This Acknowledgement should be written by students in your own language (Do not copy and Paste)

…..

……

….

……

ABSTRACT

This project, titled "Startup Success Prediction,"leverages AI and machine learning to predict the success of startups. Utilizing a comprehensive dataset with various parameters such as location, funding rounds, and current status, the model identifies key factors influencing startup outcomes. By implementing advanced predictive algorithms, the project aims to provide insights into the likelihood of success or failure for new ventures. The analysis covers funding distributions, geographical impact, and status trends, offering valuable information for investors and entrepreneurs. This high-level predictive model serves as a powerful tool for making data-driven decisions, enhancing the understanding of startup dynamics, and fostering successful business strategies. The integration of AI and machine learning in this context demonstrates the potential of technology to revolutionize the startup ecosystem.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In the dynamic world of startups, predicting the likelihood of a venture's success can be a complex challenge. The "Startup Success Prediction using Machine Learning" project addresses this challenge by applying machine learning techniques to forecast startup outcomes based on historical data. This project aims to provide a data-driven approach to assess the potential success of startups, thereby offering valuable insights to investors, entrepreneurs, and stakeholders.

The primary objective of this project is to develop a predictive model that estimates the success probability of startups. By analyzing a dataset that includes various features such as company size, industry, location, and funding details, the project seeks to identify patterns and correlations that contribute to a startup's success. This predictive capability is crucial for making informed investment decisions and strategic planning.

The dataset utilized for this project contains comprehensive information about startups, including attributes that are indicative of their operational and financial status. Machine learning algorithms are employed to process this data, with a focus on preprocessing, feature selection, and model training. The selected algorithms and evaluation metrics are designed to achieve accurate and actionable predictions.

The project leverages Python and associated libraries, such as Pandas for data manipulation and Scikit-learn for machine learning model development. By integrating these tools, the project aims to deliver a robust model that provides reliable success predictions.

Ultimately, the "Startup Success Prediction using Machine Learning" project aspires to contribute to the startup ecosystem by offering a predictive framework that enhances decision-making and strategic insights, thereby supporting the growth and success of new ventures.

1.1. Problem Statement:

In the context of the "Startup Success Prediction using Machine Learning" project, the problem statement could be: "Investors and entrepreneurs often face difficulties in predicting the success of startups due to the complex nature of startup environments and a lack of reliable forecasting tools. This project seeks to develop a machine learning model that predicts the likelihood of startup success based on historical data, thereby providing valuable insights for decision-making."

1.2. Problem Definition:

For the startup prediction project, the problem definition might include: "The challenge is to identify key features that influence startup success and to build a predictive model using these features. This involves collecting and analyzing data related to startup characteristics such as industry, funding, and geographic location. The model should be able to predict success with high accuracy to support stakeholders in making informed decisions."

1.3. Expected Outcomes:

In this project, the expected outcomes could be: "The primary outcome is a machine learning model that predicts the probability of startup success based on historical data. Additionally, the project aims to identify significant factors contributing to startup success, provide actionable insights for investors, and offer a framework that can be used for future analysis of startups."

CHAPTER 2

LITERATURE SURVEY

The prediction of startup success is a critical area of research in the fields of entrepreneurship, finance, and data science. Machine learning techniques offer robust tools to analyze vast amounts of data and identify patterns that can predict the likelihood of a startup's success. This literature survey reviews key studies and methodologies relevant to the application of machine learning in predicting startup outcomes.

Previous Research

2.1 Startup Success Factors:

- Studies such as those by Kerr, Nanda, and Rhodes-Kropf (2014) have identified critical factors influencing startup success, including team composition, funding sources, market conditions, and innovation capabilities.
- Research by Gompers, Kovner, Lerner, and Scharfstein (2010) has highlighted the role of venture capital and its impact on startup growth and success.

2.2 Machine Learning Applications:

- A study by Baier, Kroll, and von der Gracht (2016) explored the use of machine learning algorithms to predict startup success, demonstrating the effectiveness of logistic regression, decision trees, and support vector machines in this domain.
- Kim, Lee, and Cho (2018) applied ensemble learning techniques, such as random forests and gradient boosting, to improve prediction accuracy and identified key predictive features related to financial metrics and market trends.

2.3 Feature Selection and Data Preprocessing:

- Research by Cui, Wong, and Lui (2017) emphasized the importance of feature selection in enhancing model performance. Their study utilized techniques such as principal component analysis (PCA) and recursive feature elimination (RFE) to identify significant predictors of startup success.
- Studies on data preprocessing, such as the work by Kotsiantis, Zaharakis, and Pintelas (2006), have outlined methods for handling missing data,

normalizing features, and encoding categorical variables, all crucial steps in preparing data for machine learning models.

2.4 Model Evaluation and Hyperparameter Tuning:

- The work of Bergstra and Bengio (2012) on hyperparameter optimization highlighted the use of grid search and random search strategies to fine-tune model parameters and improve predictive accuracy.
- A comparative study by Chen and Guestrin (2016) demonstrated the efficacy of evaluation metrics such as accuracy, precision, recall, and F1 score in assessing model performance for binary classification tasks like startup success prediction.

2.5 Deployment and Practical Applications:

- Research by Polyzotis, Roy, Whang, and Zinkevich (2017) discussed the challenges and best practices in deploying machine learning models in real-world applications, emphasizing the need for scalability, reliability, and user-friendly interfaces.
- Studies such as those by Zhang, Yang, and Ai (2019) have illustrated the deployment of predictive models in web applications and decision support systems, enabling stakeholders to leverage machine learning insights for strategic decision-making.

CHAPTER 3

PROPOSED METHODOLOGY

The primary aim of this research is to develop a success prediction models for startup using acquisition information by which a company can make use of the proposed analysis to determine the important factors and areas that are needed to be focused and improved to have a successful venture also to predict the probability of the startup to be acquired. Data Mining methods are used in this project to make insightful decisions using the data collected from the sources which will result in effective outcomes.

Data types

There are 49 columns out of which 31 will be used as features. The rest provide more information about the data, but will not be used for model training (like company name, company id, latitude, longitude, zip code etc.)

Some of the top features include:

- age_first_funding_year – quantitative
- age_last_funding_year – quantitative
- relationships – quantitative
- funding_rounds – quantitative
- funding_total_usd – quantitative
- milestones – quantitative
- age_first_milestone_year – quantitative
- age_last_milestone_year – quantitative
- state – categorical
- industry_type – categorical
- has_VC – categorical
- has_angel – categorical
- has_roundA – categorical
- has_roundB – categorical

- has_roundC – categorical
- has_roundD – categorical
- avg_participants – quantitative
- is_top500 – categorical
- status(acquired/closed) – categorical

Data pre-processing

Data was pre-processed by analyzing the data on mysql database and by joining data from multiple tables.

Task-1

In this visualization, one of the most correlated feature will be selected. It will then be grouped into different ranges and the count of records in each group will be correlated with the labels. At the analytic task, we will be **identifying** the top correlated features and grouping them into different ranges. The user can **explore** the relationship between features in each group to their labels. At the high-level, we are helping users visually **discover** the strong correlation between the feature and labels.

Task-2

In this step, user will predict best model by **comparing** different machine learning models. He will also have to **explore** different hyper-parameters and then **derive** the final model.

Task -3

Based on the model the user selects in the previous task, he will then make predictions and **identify** which startups are likely to succeed or fail. The predictions can be grouped by state and visualized at a state level. The use can **browse** through different states to **discover** new insights.

3.1 Modules Used

3.1.1 Pandas

- **Description:** A powerful data manipulation and analysis library for Python, providing data structures and functions needed to manipulate structured data seamlessly.
- **Key Functions:**
    - pandas.read_csv(): Reads a comma-separated values (CSV) file into a DataFrame.
    - DataFrame.drop(): Drops specified labels from rows or columns.
    - DataFrame.select_dtypes(): Selects columns based on their data types.
    - DataFrame.apply(): Applies a function along an axis of the DataFrame.
    - pandas.DataFrame(): A 2-dimensional labeled data structure with columns of potentially different types.

3.1.2 Scikit-Learn

- **Description:** A machine learning library for Python that provides simple and efficient tools for data mining and data analysis.
- **Key Submodules and Functions:**
    - sklearn.model_selection.train_test_split(): Splits arrays or matrices into random train and test subsets.
    - sklearn.preprocessing.StandardScaler(): Standardizes features by removing the mean and scaling to unit variance.
    - sklearn.preprocessing.OneHotEncoder(): Encodes categorical features as a one-hot numeric array.
    - sklearn.compose.ColumnTransformer(): Applies transformers to columns of an array or pandas DataFrame.
    - sklearn.pipeline.Pipeline(): Chains together multiple steps into one.
    - sklearn.impute.SimpleImputer(): Imputes missing values.

3.1.3 TensorFlow Keras

- **Description:** A high-level API to build and train deep learning models in TensorFlow.
- **Key Classes and Functions:**
    - Sequential(): Sequential model class.

  - Dense(): A regular densely-connected neural network layer.
  - EarlyStopping(): A callback to stop training when a monitored metric has stopped improving.
  - Sequential.compile(): Configures the model for training.
  - Sequential.fit(): Trains the model for a fixed number of epochs.
  - Sequential.evaluate(): Returns the loss value and metrics values for the model in test mode.
  - Sequential.predict(): Generates output predictions for the input samples.

3.1.4 Tkinter

- **Module Name:** tkinter
- **Description:** The standard Python interface to the Tk GUI toolkit.
- **Key Classes and Functions:**
  - Tk(): The main window of the application.
  - ttk.Label(): A widget used to display text or an image.
  - ttk.Combobox(): A widget that combines a text box with a drop-down list of options.
  - ttk.Entry(): A widget used to enter or display a single line of text.
  - ttk.Button(): A widget used to create a button.
  - StringVar(): A class used to declare string variables.
  - messagebox.showerror(): A function to display an error message box.
  - mainloop(): A method that runs the application, waiting for events to be processed.

3.1.5 Other Modules

- **Module Name: messagebox**
- **Description:** Provides a standard way to display message boxes in Tkinter applications.
- **Key Functions:**
  - messagebox.showerror(): Displays an error message box with the given title and message.

3.1 Advantages

- Data-Driven Insights: Provides valuable insights based on comprehensive data analysis, enabling investors and entrepreneurs to make informed decisions.
- Predictive Accuracy: Utilizes advanced machine learning algorithms to predict startup success with high accuracy, reducing the uncertainty in investment decisions.
- Multi-Faceted Analysis: Analyzes various parameters such as location, funding rounds, and current status to identify key factors influencing startup success.
- Geographical Impact: Offers insights into how geographical factors affect startup success, helping investors identify lucrative regions.
- Risk Mitigation: Helps investors assess the risk associated with investing in a particular startup, potentially saving significant resources by avoiding high-risk ventures.
- Resource Allocation: Assists entrepreneurs in understanding where to focus their efforts and resources to maximize their chances of success.
- Trend Analysis: Identifies trends in funding distributions and startup statuses, providing a strategic advantage to entrepreneurs planning their next moves.
- Customized Strategies: Enables the development of tailored business strategies based on the specific factors that influence startup success in different contexts.
- AI and Machine Learning: Demonstrates the potential of AI and machine learning to revolutionize the startup ecosystem by providing predictive insights and enhancing decision-making processes.
- Scalability: The model can be easily scaled to incorporate new data and parameters, making it adaptable to changing market conditions and trends.
- Interactive GUI: The Tkinter-based graphical user interface (GUI) provides an easy-to-use platform for users to input data and receive predictions, making the tool accessible even to those without technical expertise.
- Real-Time Predictions: Offers real-time success predictions, enabling users to quickly assess the potential of different startups.
- Extensive Documentation: Backed by comprehensive documentation and community support, facilitating ease of use and troubleshooting.

- Educational Value: Serves as a valuable educational tool for those looking to learn about AI, machine learning, and data analysis in the context of startup success.

3.2 Requirement Specification

3.2.1 Functional Requirements

1. Data Loading and Preprocessing
    - The system shall load the dataset from a specified CSV file.
    - The system shall handle missing values in the dataset using appropriate imputation techniques.
    - The system shall preprocess numerical data by scaling it to a standard range.
    - The system shall preprocess categorical data using one-hot encoding.
2. Model Training
    - The system shall split the dataset into training and testing sets.
    - The system shall define a neural network model using TensorFlow Keras.
    - The system shall train the model on the training dataset with early stopping to prevent overfitting.
    - The system shall evaluate the model on the testing dataset and output the accuracy.
3. Prediction Functionality
    - The system shall provide a function to predict the success score of a startup based on input parameters.
    - The function shall preprocess the input data to match the format of the training data.
    - The function shall output a success score ranging from 0 to 1.
4. Graphical User Interface (GUI)
    - The system shall provide a GUI for users to input startup parameters.
    - The GUI shall include fields for state code, funding total (USD), funding rounds, relationships, milestones, and category code.
    - The GUI shall display the predicted success score upon receiving input and executing the prediction function.
    - The GUI shall handle invalid inputs and display appropriate error messages.

3.2.2 Non-Functional Requirements

1. Performance

  - The system shall preprocess the dataset efficiently to minimize load times.
  - The system shall train the model within a reasonable time frame, considering the size of the dataset.
  - The system shall provide real-time predictions in the GUI with minimal delay.

2. Usability
   - The GUI shall be intuitive and user-friendly, allowing users with minimal technical expertise to interact with the system.
   - The system shall provide clear and informative feedback to the user, including error messages and success scores.
3. Scalability
   - The system shall be designed to handle larger datasets and additional parameters with minimal modifications.
   - The model shall be scalable to accommodate new data and evolving requirements.
4. Reliability
   - The system shall handle missing or incorrect data gracefully, ensuring that the prediction function can still operate effectively.
   - The system shall provide accurate and consistent predictions based on the trained model.
5. Maintainability
   - The system code shall be modular and well-documented, allowing for easy updates and maintenance.
   - The system shall be designed to accommodate future enhancements and additional features with minimal disruption to existing functionality.
6. Security
   - The system shall ensure the security and privacy of the input data provided by users.
   - The system shall prevent unauthorized access to the underlying model and dataset.

3.2.3 Technical Requirements

1. Software Requirements
   - Programming Language: Python 3.x

  - Libraries: pandas, scikit-learn, TensorFlow Keras, tkinter, messagebox
  - Development Environment: Any Python IDE or text editor (e.g., PyCharm, VSCode)

2. Hardware Requirements
   - Processor: Multi-core CPU
   - Memory: At least 8GB of RAM
   - Storage: Sufficient disk space to store the dataset and model files
3. System Requirements
   - Operating System: Cross-platform support (Windows, macOS, Linux)
   - Python Installation: Python 3.x installed with necessary libraries

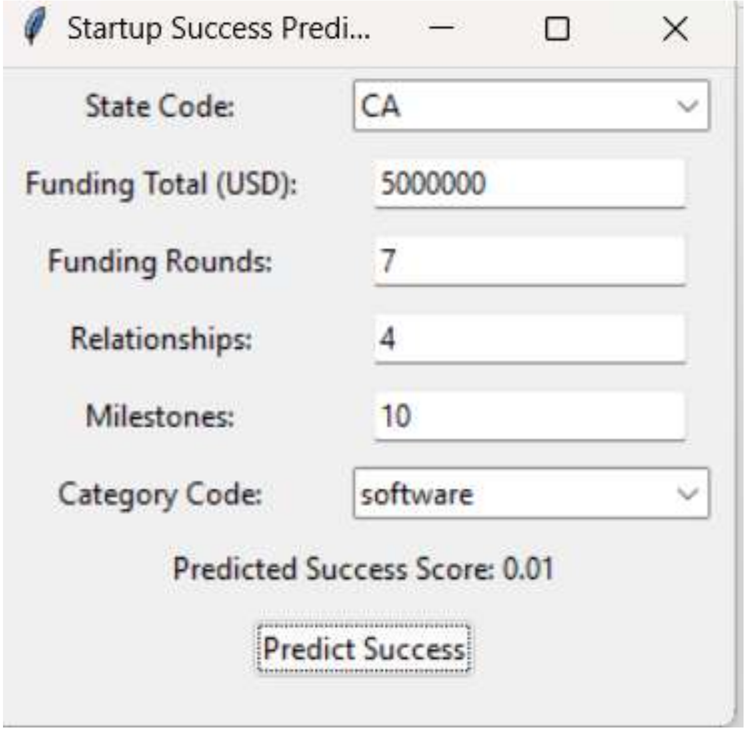
CHAPTER 4

IMPLEMENTATION and RESULT

4.1. Visualization

Correlograms are images of correlation statistics which help us visualize the data in correlation matrices. By visualizing the correlations between features, we can gain intuitions about the high correlations between some variables. We might use these insights to drop a few high correlated features in variable selection of there is no significant loss of information in the model.

A diverging color palette was used as data from both the positive and negative correlations could be interesting. The positive correlations were represented by blue while the negative correlations were represented by red. From the below Correlogram, we can see that there isn’t a very high correlation between the individual features. Thus, most of the features are unique and can be used for training the model.



Startup Success Predi...

State Code: CA

Funding Total (USD): 5000000

Funding Rounds: 7

Relationships: 4

Milestones: 10

Category Code: software

Predicted Success Score: 0.01

CHAPTER 5

CONCLUSION

We were able to successfully build a machine learning model that predicts the success/failure of a startup. We were also able to implement a web-based client-side visualization connected via a REST API to server-side modeling. We were also able to generate interesting visualizations through preliminary insights and plots.

For future improvements, we could work on extracting more features like evaluate startup presence on the web, number of unique domains mentioning the startup, burn rate etc. Using these features, one could get more ideas on creating more interactive web-based visualizations.

REFERENCES

1. https://www.kaggle.com/datasets/manishkc06/startup-success-prediction/code
2.https://github.com/Dhyeybhuva2003/Startup-Success-Prediction.git