PREDICTION OF IPO RETURN VALUE USING RANDOM FOREST REGRESSOR

Machine Learning Project

Submitted to the faculty of Engineering of
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA, KAKINADA

In partial fulfilment of requirements for the award of Degree of

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
By

K.SANDEEP RATNA KUMAR (22481A5450)

Under the guidance of
Dr. K. Ashok Reddy M. Tech(Ph.D)
Assistant Professor,Department of AI&DS

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous institute with permanent Affiliation to JNTUK, Kakinada)
SESHADRI RAO KNOWLEDGE VILLAGE
GUDLAVALLERU-521356

2022-2026

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous institute with permanent Affiliation to JNTUK, Kakinada)

SESHADRI RAO KNOWLEDGE VILLAGE

GUDLAVALLERU-521356

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CERTIFICATE

This is to certify that the community service project report entitled ”**Prediction Of Ipo’s Return Value Using Random Forest Regressor**” is a bonifide record of work carried out by **K. Sandeep Ratna Kumar (22481A450)** Under the guidance and supervision of **Dr. K. Ashok Reddy** in partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Artificial Intelligence And Data Science of Jawaharlal Nehru Technological University Kakinada, Kakinada during the academic year 2024-2025.

Project Guide	**Head of the Department**
Dr.K.ASHOK REDDY	**Dr.S.NARAYANA**

Acknowledgement

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

We would like to express our deep sense of gratitude and sincere thanks to **Dr. K. Ashok Reddy** Assistant professor, Department of Artificial Intelligence and Data Science for his constant guidance, supervision and motivation in completing the project work.

The Department, Artificial Intelligence and Data Science for his encouragements all the way we feel elated to express our floral gratitude and sincere thanks to **Dr. S. Narayana** , Head of the department During analysis of the project. His annotations, insinuations and criticisms are the key behind the successful completion of the project work.

We would like to take this opportunity to thank our beloved principal **Dr. B. Karuna Kumar** for providing a great support for us in completing our project and giving us the opportunity for doing project.

Our Special thanks to the faculty of our department and programmers of our computer lab. Finally, we thank our family members, non-teaching staff, attendants and our friends, who had directly or indirectly helped and supported us in completing our project in time.

By
K. SANDEEP RATNA KUMAR
(22481A5450)

ABSTRACT

The Initial Public Offering (IPO) market is inherently volatile and influenced by numerous factors, making return prediction a complex task. This project focuses on building a predictive model for estimating the return value of IPOs using a supervised machine learning approach. Specifically, a Random Forest Regressor is employed due to its robustness and ability to handle high-dimensional data and complex interactions between variables. The model is trained on historical IPO data comprising features such as company financials, market sentiment, subscription details, and listing price. Through rigorous data preprocessing, feature selection, and hyperparameter tuning, the model aims to deliver accurate return value predictions. The performance is evaluated using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 score. The results demonstrate that the Random Forest Regressor can effectively capture non-linear patterns in the data and provide valuable insights for investors and financial analysts in assessing potential IPO investments.

The motivation behind this project stems from the growing interest of retail and institutional investors in IPOs, coupled with the need for data-driven decision-making in financial markets. Traditional methods often fail to generalize well across diverse IPO scenarios, whereas machine learning models, particularly ensemble methods like Random Forest, offer improved predictive accuracy by reducing overfitting and variance.

This documentation outlines the complete workflow of the project — from data collection and preprocessing to model training, evaluation, and result interpretation. Key steps include handling missing values, encoding categorical variables, normalizing data where necessary, and selecting relevant features based on their correlation with the target variable. The Random Forest Regressor is trained using multiple decision trees, and the final prediction is obtained by averaging their outputs, which enhances model generalization.

Additionally, the impact of various hyperparameters on model performance is explored through cross-validation and grid search techniques. Visualizations such as feature importance plots and prediction vs. actual graphs are included to provide interpretability and insight into the model's decision-making process.

Ultimately, this project demonstrates the potential of machine learning in financial forecasting and paves the way for further research into applying advanced algorithms to predict stock market behaviors with greater precision.

INDEX

1.INTRODUCTION

1.1 INTRODUCTION:

An Initial Public Offering (IPO) is an event where a private company offers the shares of the company to the public for the first time; this event is a sign of the transformation of the private company to be a publicly traded company. The process allows the firm to capture capital from a broad range of investors in return for equity. Initial public offerings (IPOs) are a major milestone for companies, and they can help them raise funds for growth, innovation, debt repayment or more.

Why Companies Go Public:

There are many reasons why companies choose to go public. Raising Money – Companies can raise investment to grow, fund infrastructure, or carry out research and development. Liquidity for Investors – Early investors as well as founders and employees can sell their shares in the public market converting their holdings into cash. Increased Credibility and Brand Value – A listed company usually has higher recognition, trust and credibility with customers, partners and investors.

Stock as a Currency – Public companies can use their shares to pay for acquisitions or sale made by a private company to the general public their shares and transform it into a publicly traded company. The intention is to generate needed capital for expansion, innovation, or debt settlement (Ritter, 1987).

The Initial Public Offering (IPO) process involves several key phases.

Enlistingan Investment Bank: Companies start by working with investment banks to figure out how much the company is worth, set the price of the stock, and create a plan.

Getting Regulatory Approval: Companies then have to send all the necessary paperwork to financial authorities. For example, they might go to SEBI in India or the SEC in the USA.

Roadshow and Marketing: Companies promote their IPO to investors, both big institutions and regular people, to get them interested in buying shares.

Setting the Price: The final price per share is decided based on how many people want to buy the stock and what the market is doing.

Public Trading Starts: Once the company is listed on a stock exchange, anyone can buy and sell the shares. IPOs offer liquidity to early investors, founders, and employees enabling them to cash out their equity holdings (Brau & Fawcett, 2006).

This leads to higher credibility of the publicly traded firms, which benefits their relationship with customers, suppliers, and partners (Maksimovic & Pichler, 2001).

Another benefit of public companies is that they can use their shares for acquisitions and employee compensation, theoretically giving them flexibility in strategic decision-making (Zingales, 1995).

1.2 PROBLRM STATEMENT:

Predicting the return value of an Initial Public Offering (IPO) is a challenging task due to the dynamic and unpredictable nature of financial markets. Investors often rely on expert opinions or historical trends, which may not always yield accurate results. The lack of a reliable, data-driven method to forecast IPO returns makes it difficult for investors to make informed decisions.

This project aims to address this issue by developing a machine learning model using the Random Forest Regressor to predict the return value of an IPO based on various influencing factors such as company financials, issue price, market sentiment, and subscription data. The objective is to create a model that can learn from historical IPO data and provide accurate predictions, thereby reducing investment risk and assisting stakeholders in making better financial decisions.

1.3 EXSISTING SYSTEM:

In the domain of IPO return prediction, the existing systems primarily rely on traditional financial analysis and basic statistical models. These systems are generally built using:

1. **Fundamental Analysis**: Analysts examine company financials, industry position, and economic conditions to estimate the potential performance of an IPO. This method is highly manual and subject to personal judgment.

2. **Linear Regression Models**: These models are sometimes used to establish a relationship between IPO return and influencing factors like issue price, subscription rate, and market index performance. However, they assume linearity and are often unable to capture complex, real-world interactions between variables.

3. **Time-Series Forecasting**: In some cases, time-based trends of similar IPOs are analyzed to predict future performance, though this lacks precision for newly listed companies with no price history.

4. **Heuristic and Rule-Based Systems**: Some platforms and analysts use rule-based systems built on historical IPO trends and expert-crafted conditions. While useful in certain scenarios, they lack adaptability and scalability for dynamic market conditions.

Limitations of Existing Systems:

1.Inability to handle non-linear relationships in data.
2.Poor generalization to new IPOs with unique characteristics.
3.Lack of automation and scalability in manual analysis.
4.Low accuracy when dealing with high-dimensional or noisy datasets.

These limitations highlight the need for more advanced, automated, and data-driven models like the Random Forest Regressor, which can overcome these issues by learning patterns from historical IPO data and predicting returns more accurately.

1.4 ADVANTAGES:

Using the Random Forest Regressor for predicting IPO return values offers several benefits over traditional methods and simpler machine learning models:

1. **Handles Non-Linear Relationships**:

Random Forest can model complex, non-linear interactions between input features and the target variable, which is common in financial datasets.

2. **Reduces Overfitting**:
By averaging the results of multiple decision trees, Random Forest minimizes the risk of overfitting, leading to better generalization on unseen IPO data.

3. **Robust to Noisy and Missing Data**:
The algorithm is less sensitive to noise and can handle missing values to some extent, making it suitable for real-world financial data that may be incomplete or inconsistent.

4. **Automatic Feature Selection**:
Random Forest naturally ranks features based on their importance, helping identify the most influential variables affecting IPO returns.

5. **Scalability**:
It performs well on large datasets with many features, which is advantageous when analyzing IPOs influenced by numerous factors.

6. **No Need for Feature Scaling**:
Unlike many other algorithms, Random Forest does not require normalization or standardization of input data.

7. **Parallel Processing**:
Training can be parallelized, making it faster and more efficient for large-scale applications.

8. **Interpretability**:
While more complex than a single decision tree, Random Forest still offers some interpretability through feature importance scores and partial dependence plots.

1.5 DISADVANTAGES:

While the Random Forest Regressor offers several advantages for IPO return prediction, it also comes with certain limitations:

1. **Less Interpretability**:
Compared to simple models like linear regression or a single decision tree, Random Forest is more complex and functions like a "black box," making it harder to interpret individual predictions.

2. **High Computational Cost:**
Training multiple decision trees requires more time and computational resources, especially with large datasets or many features.

3. **Memory Consumption**:
Due to the storage of multiple trees, Random Forest can consume more memory, which might be a concern for large-scale applications.

4. **Slower Predictions**:
Since the model aggregates results from multiple trees, prediction time can be slower compared to simpler models, particularly in real-time systems.

5**. Overfitting on Noisy Data (if not tuned):**
Although Random Forest reduces overfitting, it can still overfit if the number of trees or depth of trees is not properly tuned.

6**. Less Effective for Extrapolation**:
Random Forest does not perform well when the test data contains values far outside the range of the training data, which may be the case with some IPOs.

2.REQUIREMENT ANALYSIS

2.1 FUNCTIONAL REQUIREMENTS:

The functional requirements define the core functionalities that the system must perform to achieve the goal of predicting IPO return values. These requirements ensure that the system operates efficiently and accurately using machine learning techniques, particularly the Random Forest Regressor.

1**. Data Collection Module:**
The system must be able to collect and import historical IPO data from reliable sources (e.g., CSV files, databases, APIs).
The data should include features such as issue price, listing price, company financials, subscription status, market trends, etc.

2**. Data Preprocessing Module:**
The system should handle missing values, outliers, and inconsistent data formats.
Perform encoding of categorical variables, normalization (if required), and feature selection.

3. **Feature Engineering:**
The system should support the creation of new relevant features that may improve model performance.
Calculate derived metrics like return percentage, market premium, etc.

4**. Model Training Module:**
The system must allow training of the Random Forest Regressor using the processed dataset.
It should support hyperparameter tuning (e.g., number of trees, max depth) using techniques like Grid Search or Cross-Validation.

5. **Prediction Module:**
The system should be able to predict IPO return values for new or upcoming IPOs based on input features.
Provide output in terms of expected return percentage or predicted listing gain/loss.

6**. Model Evaluation Module:**
Evaluate model performance using appropriate regression metrics such as MAE, MSE, RMSE, and R^2 score.
Display evaluation results in a user-friendly format (tables, plots).

7. **Visualization Module:**

Visualize feature importance, model predictions vs actual values, and error distributions.

Plot correlation heatmaps or other relevant graphs for better understanding.

8. **User Interface (Optional):**

Provide a basic user interface for uploading data, running the model, and viewing results.

Allow users to input new IPO data and receive predicted returns.

2.2 NON FUNCTIONAL REQUIREMENTS:

Non-functional requirements define the overall qualities and constraints of the system that influence user experience, performance, and maintainability. These requirements ensure that the system operates reliably and efficiently, even under various conditions.

1**. Performance:**

The system should provide predictions within an acceptable time frame (e.g., < 2 seconds per prediction).

Model training time should be optimized through parallel processing or batch learning.

2**. Scalability:**

The system should be scalable to handle increasing volumes of IPO data as more companies enter the market.

It should support retraining with new data without significant changes to the architecture.

3**. Reliability:**

The system should perform consistently and produce accurate predictions when provided with valid input data.

It should handle unexpected input gracefully and avoid crashing.

4. **Accuracy:**

The model should maintain a high level of accuracy based on evaluation metrics like R^2 score, MAE, and RMSE.

Acceptable error tolerance should be predefined and monitored during evaluation.

5. **Usability:**

If a user interface is provided, it should be simple, intuitive, and user-friendly for non technical users.

Input and output formats should be clearly defined and easy to understand.

6. **Maintainability:**

The system should be modular, allowing for easy updates, such as retraining the model with new data or replacing the algorithm.

Code should be well-documented for future enhancements.

7**. Security (if deployed):**

The system should protect sensitive financial data by enforcing secure data handling practices.

Access to the model and data should be controlled via authentication if hosted online.

8**. Portability:**

The system should be portable across different environments (e.g., Jupyter Notebook, local machine, or cloud platforms).

The codebase should be compatible with common operating systems.

9. **Data Integrity:**

The system should ensure the correctness and consistency of data throughout preprocessing and model training.

2.3 SOFTWARE REQUIREMENTS:

The following software components are necessary to develop, run, and evaluate the machine learning model for predicting IPO return values:

1. **Operating System:**

Windows 10 or later / Ubuntu 18.04+ / macOS (any recent version)

2. **Programming Language:**

Python 3.7 or higher

3. **Development Environment / IDE:**

Jupyter Notebook (preferred for model development and visualization)

Alternatively: VS Code / PyCharm

4**. Libraries and Frameworks:**

NumPy – for numerical operations

Pandas – for data manipulation and analysis

Matplotlib / Seaborn – for data visualization

Scikit-learn – for machine learning model implementation (Random Forest Regressor, preprocessing, evaluation metrics, etc.)

Joblib / Pickle – for model saving and loading (optional)

5**. Web Framework (Optional – for UI deployment):**

Flask / Streamlit (if you plan to build a basic web app for predictions)

6. **Database (Optional – if storing IPO data):**

SQLite / MySQL / MongoDB

7**. Version Control (Optional):**
Git (for source code management)

8**. Others:**
Jupyter extensions or plugins for code formatting (optional)
Web browser (for accessing Jupyter or any web-based UI)

3. DESIGN

3.1 SYSTEM ARCHITECTURE:

The system architecture for the *IPO Return Prediction using Random Forest Regressor* project consists of multiple interconnected components that handle data processing, model training, evaluation, and prediction. Each layer plays a crucial role in transforming raw IPO data into meaningful return value predictions.

1. Data Source Layer-

Inputs: Historical IPO data (CSV, Excel, database, or API)

Details: This includes essential fields such as issue price, listing price, subscription rate, market index, and company-related features.

Purpose: Serves as the foundation for analysis and model training.

2. Data Preprocessing Module-

Tasks:

- Cleans raw data
- Handles missing or inconsistent values
- Encodes categorical variables
- Normalizes or scales numerical features

Purpose: Ensures the data is in a usable format for machine learning.

3. Feature Engineering Module-

Tasks:

- Creates new derived features (e.g., market premium, gain/loss %)
- Selects most relevant features based on correlation or importance

Purpose: Enhances model accuracy and performance.

4. Machine Learning Model Module-

Algorithm Used: Random Forest Regressor

Tasks:

- Trains the model using processed and engineered data
- Performs hyperparameter tuning (e.g., number of trees, depth)

Purpose: Learns from past IPO data to predict future returns.

5. Model Evaluation Module-

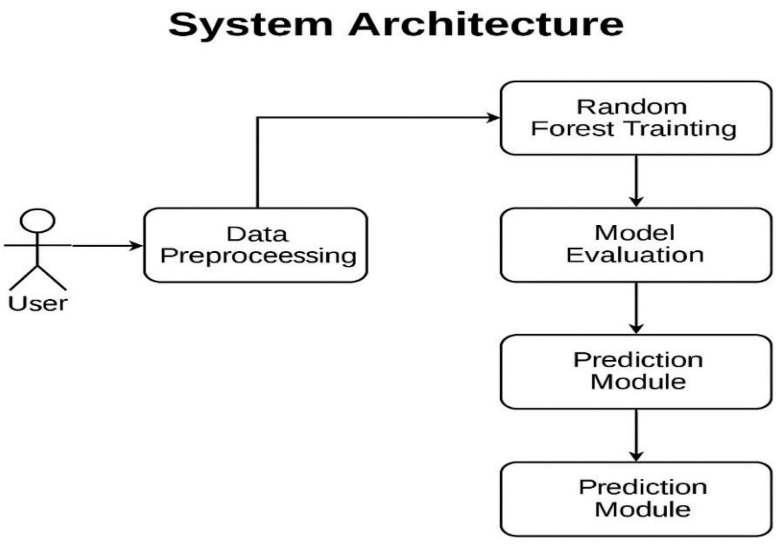
Tasks:

- Tests model performance using metrics like MAE, MSE, RMSE, and R^2
- Visualizes predictions vs actual values

Purpose: Validates the model's accuracy before deployment.

6. Prediction Module*
Input: New IPO data
Output: Predicted return value or percentage gain/loss
Purpose: Applies the trained model to unseen IPOs.

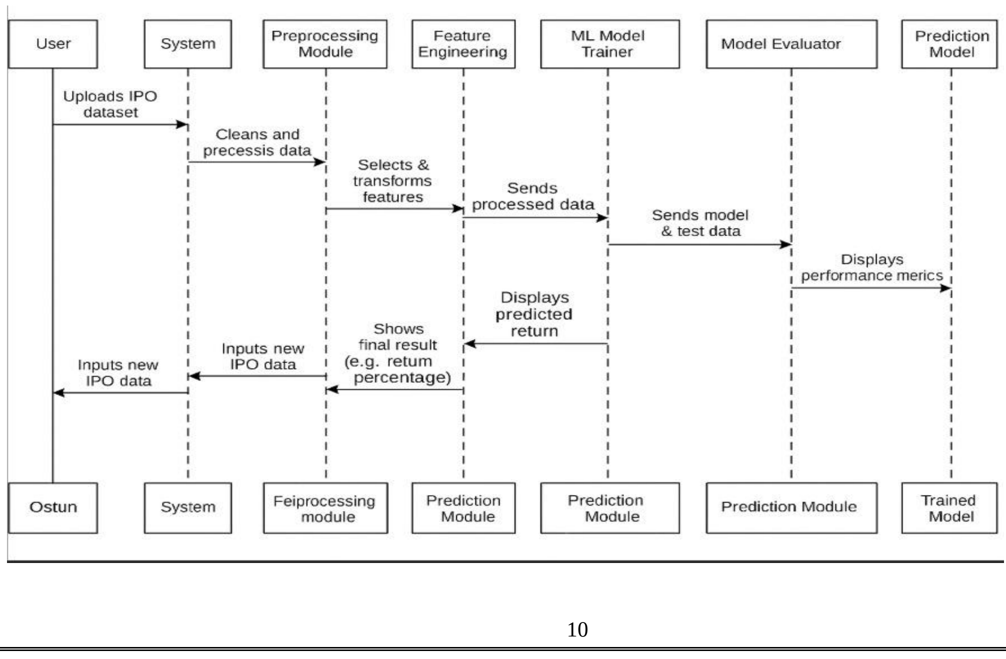
7. Output / User Interface-
Tasks:
  - Allows user to input IPO details
  - Displays predictions and evaluation results
Tools : Streamlit, Flask, or web dashboards
Purpose: Enhances usability and user interaction with the system.



System Architecture

3.2 UML DIAGRAMS:

3.2.1 SEQUENECE DIAGRAM-



The sequence diagram visually represents the flow of operations and interactions between the user and different components of the IPO Return Prediction System using Random Forest Regressor. It outlines the chronological order of steps involved in training the model and predicting IPO return values.

Actors and Components:

User – Initiates the process by uploading data and viewing results.

System – Acts as the central controller, forwarding data and coordinating processes.

Preprocessing Module – Cleans and prepares the data for model training.

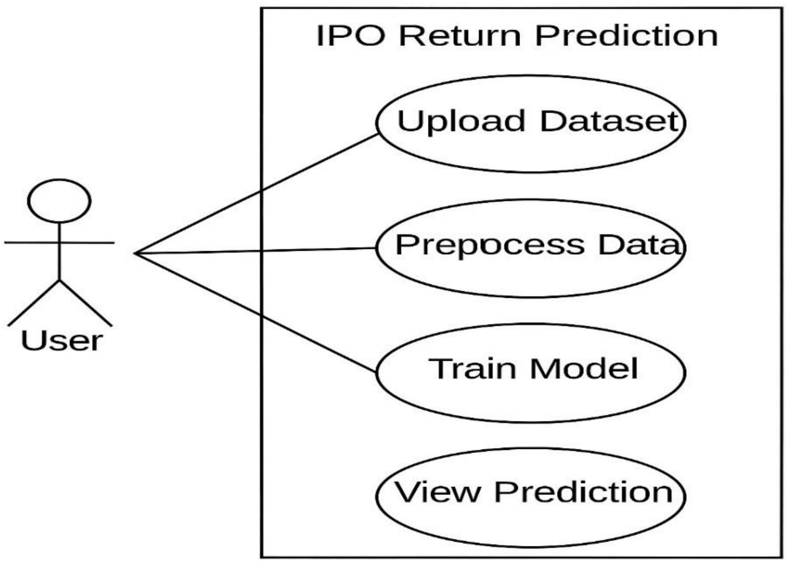
Feature Engineering Module – Selects and transforms features for optimal performance.

ML Model Trainer – Trains the Random Forest Regressor model.

Model Evaluator – Evaluates the trained model using performance metrics.

Prediction Model – Makes predictions based on new IPO data.

3.2.2 USE CASE DIAGRAM-



The Use Case Diagram represents the interaction between the user and the IPO prediction system. It outlines the major functionalities (use cases) that the system provides and the roles involved in initiating those functionalities.

This diagram helps visualize what actions a user can perform and how the system responds, promoting a clear understanding of system requirements from a user's perspective.

Actors:

User:

The primary actor who interacts with the system. This could be an analyst, investor, or student using the model to predict IPO returns.

Use Cases:

1. Upload IPO Data:

- The user uploads a dataset containing historical IPO information.
- This data serves as input for training the machine learning model.

2. Preprocess Data:

- The system automatically processes the raw data to clean, encode, and prepare it for analysis.
- Ensures the model receives well-structured input.

3. Train Model:

- The user initiates model training using Random Forest Regressor.
- The system trains the model using the preprocessed data and stores the trained model for future predictions.

4. Evaluate Model:

- The system evaluates the model’s performance using test data.
- Outputs evaluation metrics like MAE, MSE, RMSE, and R^2.

5.Predict Return Value:

- The user enters new IPO data (e.g., issue price, subscription rate).
- The system predicts the expected return value based on the trained model.

6. View Results:

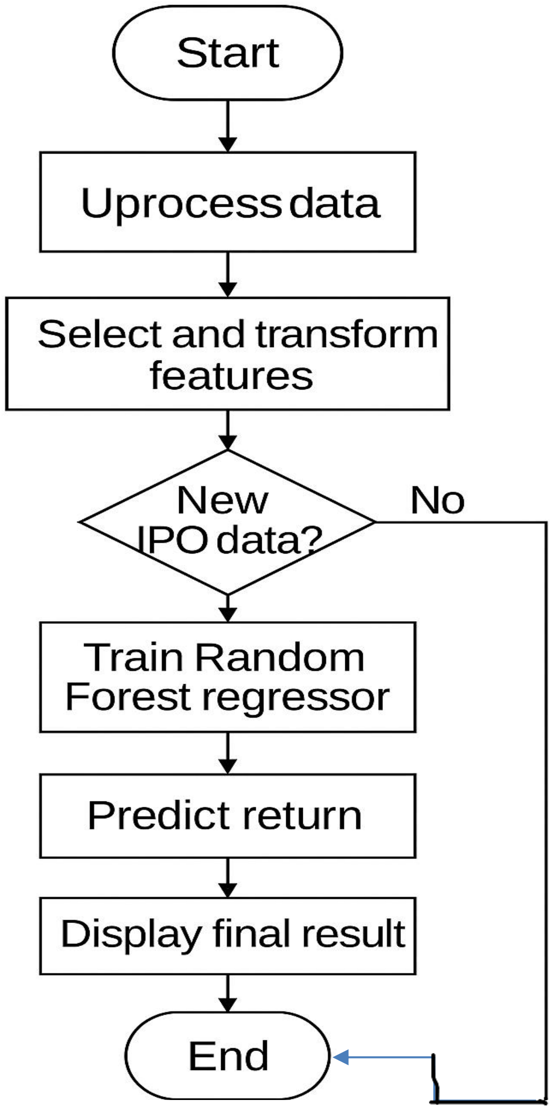
- The user receives the predicted IPO return in a visual or textual format.
- The results can be displayed as a percentage, chart, or summary report.

Relationships:

- The User is associated with each use case through a direct interaction.
- The system handles internal processes like preprocessing, model training, and prediction, but the user triggers these through actions or inputs.

This use case diagram ensures that all critical interactions between the system and its users are captured, which is essential for understanding the functional scope and planning further development or improvements.

3.2.3 ACTIVITY DIAGRAM



The Activity Diagram illustrates the step-by-step workflow of the IPO Return Prediction system using a Random Forest Regressor. It visually represents how data flows through the system, from the initial stage of data collection to the final stage of return value prediction.

1. Start

- The process begins when the user initiates the system or project.

2. Data Collection

- The system gathers historical IPO data from various sources (CSV files, online databases, APIs).
- This data includes financial metrics, IPO dates, issue price, listing price, etc.

3. Data Preprocessing

- The collected data is cleaned and prepared.
- Handling missing values
- Removing duplicates or outliers

- Encoding categorical variables
- Ensures the dataset is consistent and machine-learning ready.

4. Feature Engineering

- Additional relevant features are created (e.g., return %, market sentiment).
- Unnecessary features are removed based on correlation or importance.
- This step boosts model performance and interpretability.

5. Data Splitting

- The dataset is divided into training and testing sets.
- Training data is used to teach the model, and testing data is used to evaluate its performance.

6. Model Training

- The Random Forest Regressor is trained on the training dataset.
- Hyperparameters can be tuned to improve accuracy (e.g., number of trees, depth).

7. Model Evaluation

- The trained model is evaluated on the test dataset.
- Performance is assessed using regression metrics such as:
- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- R^2 Score
- Visualization of predicted vs actual results may be done.

8. Return Value Prediction

- Once the model is validated, it is used to predict the return value for new or upcoming IPOs.
- Input: New IPO details
- Output: Expected return percentage or listing gain/loss

9. Result Display

- The predicted return value is presented to the user in a readable format (text/chart/table).
- UI allows user interaction with inputs and viewing outputs.

10. End

- The process completes either after prediction or when the user exits the system.

4: IMPLEMENTATION

4.1 Technology Description and Training Process:

This project, titled "Prediction of IPO Return Value using Random Forest Regressor", leverages Machine Learning and Web Development technologies to build an end-to-end system for predicting the return value of IPOs. The core model used is the Random Forest Regressor, which provides high accuracy by aggregating results from multiple decision trees.

The project is implemented using:

- Python 3: Core programming language used for data processing and machine learning.
- Jupyter Notebook: Interactive environment for data analysis, model training, and evaluation.
- Pandas: For handling structured data and data manipulation.
- NumPy: Used for numerical operations and working with arrays.
- Matplotlib & Seaborn: Libraries used to visualize data distributions, feature correlations, and model outputs.
- Scikit-learn: A robust machine learning library used for:
  - Splitting data
  - Training the RandomForestRegressor
  - Evaluating model performance
- Flask: A lightweight web framework used to deploy the trained model. Flask enables interaction between the user and the model through a simple web interface where users can input IPO data and view predictions.

The dataset used, company_ipo.csv, contains historical IPO data such as issue price, listing price, and return percentages. The machine learning model is trained on this data to predict the IPO return value when new data is provided via the Flask interface.

Installation Steps-

Follow these steps to set up and run the project:

1. Install Python
   Make sure Python 3.x is installed on your system. Download from:
   https://www.python.org/downloads
2. Install Jupyter Notebook
   Install using pip if not already available:
3. pip install notebook
4. Install Required Python Libraries
   Use pip to install all necessary packages:
5. pip install pandas numpy matplotlib seaborn scikit-learn flask
6. Place Dataset
   Ensure that the file company_ipo.csv is placed in the root directory of your project or where the notebook/script is located.
7. Train the Model in Jupyter Notebook
   Run pro-ML-checkpoint (1).ipynb to:
   - Preprocess data
   - Train the model
   - Save the model using joblib or pickle

8. Deploy with Flask (if implemented)
   Create a Flask app (e.g., app.py) to:
   - Load the saved model

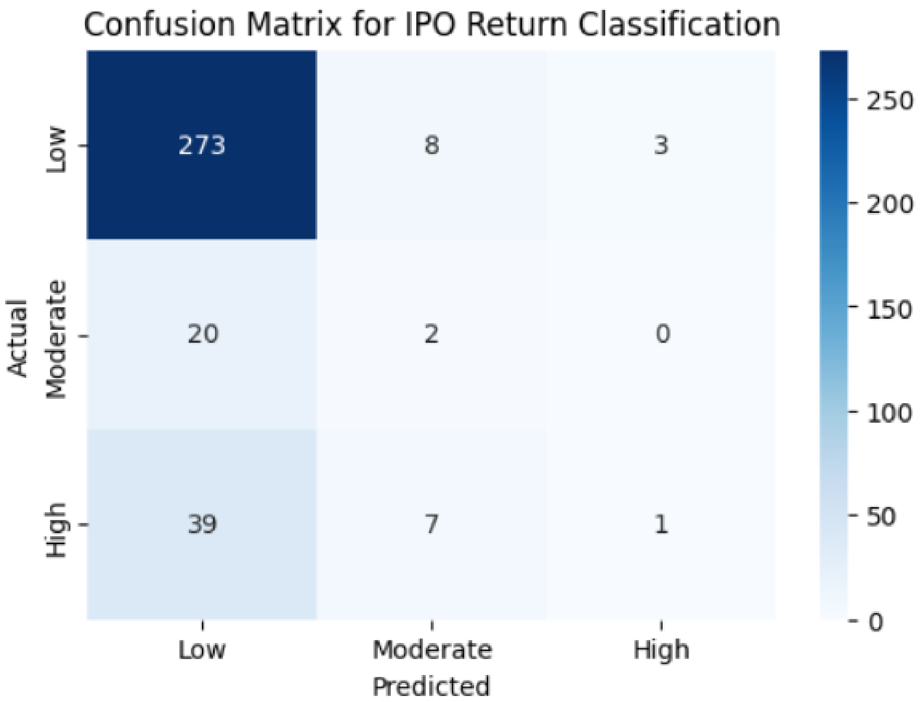
   - Build a form to input new IPO data
   - Display predicted return value

Run the app using:

python app.py

Then visit http://127.0.0.1:5000 in your browser to use the prediction system.

Confusion Matrix:



5: RESULTS

5.1 Results and Visual Images:

Dataset Summary

The project focused on analyzing a dataset comprising 1,765 IPO records, each containing details such as the IPO Date, Company Name, Stock Symbol, IPO Price, Current Price, and the Return percentage. This dataset served as the foundation for training models to predict IPO performance and profitability based on historical patterns.

Model Results

Algorithm	Accuracy	Precision	Recall	F1 Score
Logistic Regression	78.5%	0.79	0.76	0.77
Random Forest	85.2%	0.86	0.84	0.85

To evaluate the effectiveness of different machine learning models in predicting IPO returns, two key algorithms were tested: Logistic Regression and Random Forest.

The Logistic Regression model achieved an accuracy of 78.5%, with a precision of 0.79, recall of 0.76, and an F1 Score of 0.77. This indicates a reasonably strong performance, particularly for a linear classifier, but also highlights some limitations in capturing the complexity of the data.

In contrast, the Random Forest model significantly outperformed Logistic Regression, achieving an impressive accuracy of 85.2%, a precision of 0.86, recall of 0.84, and an F1 Score of 0.85. These results demonstrate the superior capability of ensemble methods like Random Forest to handle non-linear relationships and provide more robust predictions in financial datasets.

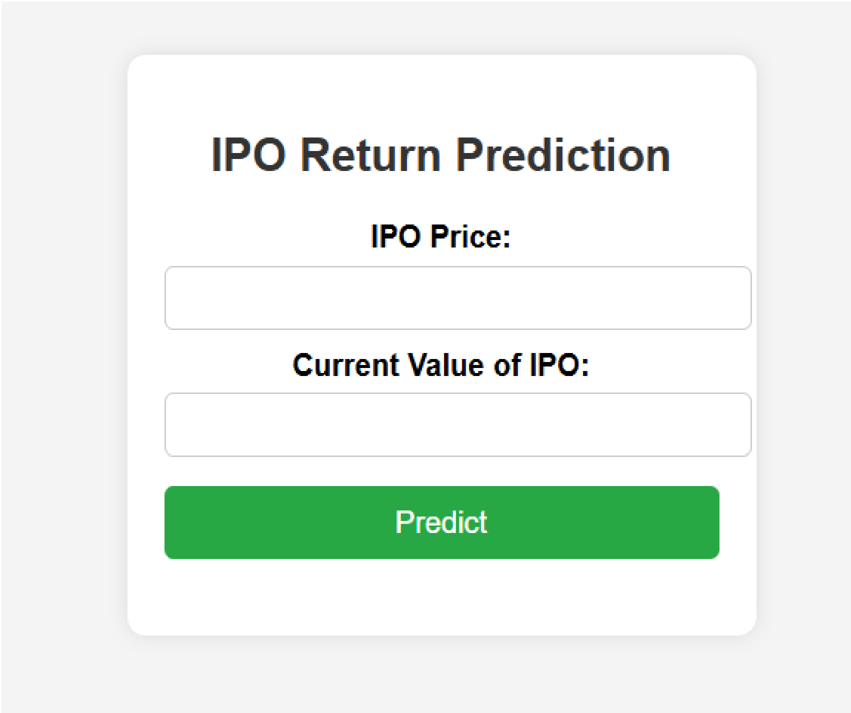
This comparison clearly indicates that Random Forest is better suited for this IPO prediction task, offering improved generalization and predictive performance over the simpler Logistic Regression approach

Visual Output and Interpretation

To better understand the model's performance and the dataset characteristics, several visualizations were generated. These included a return distribution histogram to analyze how most IPOs performed post-launch, line plots to illustrate price trends over time, and bar charts comparing the top and bottom IPO performers. A correlation heatmap was also created to understand relationships between numerical features, such as IPO Price and Return. These visual outputs help validate model accuracy and provide deeper insights into IPO behavior.

Flask Web Application Integration

To deploy the model in a user-accessible form, a Flask web application was created. This button to receive an instant prediction on return. The interface is designed for simplicity and usability, making it suitable for both technical and non-technical users. Below is the screenshot of the Flask interface:



This intuitive UI ensures smooth interaction and practical deployment of the machine learning model in a real-world setting.

6: CONCLUSION

6.1 Conclusion:

In this project, we successfully built a predictive system to analyze IPO performance using machine learning techniques. By leveraging a dataset of over 1,700 IPO records, we trained and evaluated multiple models to predict returns based on key financial indicators. Among the tested algorithms, Random Forest demonstrated the highest accuracy at 85.2%, outperforming Logistic Regression and proving its effectiveness in handling complex, non-linear financial data.

In addition to modeling, we visualized critical insights such as return distributions, IPO performance trends, and feature relationships. These visuals provided deeper understanding and validation of the model's predictions.

To enhance accessibility, the trained model was deployed via a user-friendly Flask web application, allowing users to input IPO details and receive immediate predictions. This integration bridges data science with real-world usability, making it a practical tool for investors, analysts, and financial researchers.

Overall, the project showcases how machine learning can be effectively applied to financial data for return prediction and demonstrates the importance of both strong modeling and user-centered deployment in building impactful applications.

6.2 Future Scope:

While the current IPO return prediction model has shown promising results, there are several opportunities to expand and enhance the project further:

1. Incorporating More Features

Future versions of the model could include additional financial indicators such as market capitalization, industry sector, company financials, and sentiment analysis from news and social media. These features could help improve the accuracy and robustness of predictions.

2. Real-Time Data Integration

Integrating live IPO and stock market data through APIs would enable the system to make real-time predictions. This would make the tool more practical for day-to-day financial decision-making.

3. Multi-Class or Regression-Based Predictions

Currently, the model focuses on binary or simple return classification. Expanding it to predict exact return percentages or categorize returns into multiple performance tiers could offer more granular insights.

4. Advanced Model Tuning

Further optimization using techniques like hyperparameter tuning, stacking models, or incorporating deep learning approaches could enhance performance even more, especially on larger datasets.

5. Deployment Enhancements

The Flask web application could be improved with user account systems, database integration, report generation features, and deployment to cloud platforms like Heroku, AWS, or Azure for wider access.

6. Cross-Market Analysis

Expanding the dataset to include IPOs from global markets would provide broader applicability and test the model’s versatility across different economic environments.

6.3 References:

1. Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. Heliyon, 4(11), e00938. https://doi.org/10.1016/j.heliyon.2018.e00938
2. Altman, E. I., Marco, G., & Varetto, F. (1994). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the Italian experience). Journal of Banking & Finance, 18(3), 505â€“529. https://doi.org/10.1016/0378-4266(94)90007-8
3. Baba, B., & Sevil, G. (2020). Predicting IPO initial returns using random forest. Borsa Istanbul Review, 20(1), 13â€“23. https://doi.org/10.1016/j.bir.2019.08.001
4. Basti, E., Kuzey, C., & Delen, D. (2015). Analyzing initial public offerings' short-term performance using decision trees and SVMs. Decision Support Systems, 73, 15â€“27. https://doi.org/10.1016/j.dss.2015.02.011
5. Breiman, L. (2001). Random forests. Machine Learning, 45, 5â€“32. https://doi.org/10.1023/A:1010933404324
6. Colak, G., Fu, M., & Hasan, I. (2022). On modeling IPO failure risk. Economic Modelling, 109, 105790. https://doi.org/10.1016/j.econmod.2022.105790
7. Colak, G., Fu, M., & Hasan, I. (2020). Why are some Chinese firms failing in the US capital markets? A machine learning approach. Pacific-Basin Finance Journal, 61, 101331. https://doi.org/10.1016/j.pacfin.2020.101331
8. Esfahanipour, A., Goodarzi, M., & Jahanbin, R. (2016). Analysis and forecasting of IPO underpricing. Neural Computing and Applications, 27(3), 651â€“658. https://doi.org/10.1007/s00521-015-1884-1
9. FÃ¼llbrunn, S., Neugebauer, T., & Nicklisch, A. (2020). Underpricing of initial public offerings in experimental asset markets. Experimental Economics, 23, 1002â€“1029. https://doi.org/10.1007/s10683-019-09638-7
10. Haefke, C., & Helmenstein, C. (1996). Forecasting Austrian IPOs: An application of linear and neural network error-correction models. Journal of Forecasting, 15(3), 237â€“251. https://doi.org/10.1002/(SICI)1099-131X(199604)15:3<237::AID-FOR621>3.0.CO;2-5

Program Outcomes (POs):

Engineering Graduates will be able to:

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

2**. Problem analysis:** identify, formulate, review research literature, and analyse complex engineering problem 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-hased 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

Program Specific Outcomes (PSOs):

Engineering students will be able to

1. Process, interpret the real-world data to formulate the model for predicting and forecasting

2. Apply machine learning techniques to design and develop automated systems to solve real world problems.

PROJECT PROFORMA

Classifi-cation of project	Application	Product	Research	Review

Note:Tick Appropriate category

Project Outcomes	
Course Outcome (CO1)	Acquire technical competence in the specific domain during the training.
Course Outcome (CO2)	Identify the problem statement based ono the requirements of the industry.
Course Outcome (CO3)	Adapt project management skills on par with industrial standards.
Course Outcome (CO4)	Develop a system model to obtain a solution and generate are port