IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

Characterization and Prediction of Popular Projects on GitHub

P. Srinivasa Rao[1], Pathakala Lavanya[2], Pochampally Shivani[3], S S Nandini[4], Kakkeri Venkat Sai[5]
Asst.Professor,ComputerScienceEngineering[1]
Students, Computer Science Engineering[2,3,4,5]
ACE Engineering College, Ghatkesar, India

Abstract: *GitHub is a widely used platform for hosting open-source and private repositories, where the popularity of a repository is often determined by the number of stars it receives. This project, Characterization and Prediction of Popular Projects on GitHub, aims to analyze repository features and predict their popularity using machine learning techniques. The system extracts data from GitHub using its API, including key attributes such as stars, forks, watchers, issues, contributors, tags, programming language, and description. The collected data undergoes preprocessing, feature engineering, and model training using multiple regression algorithms, including Linear Regression, Random Forest, SGDRegressor, Ridge Regression, Lasso Regression, and Elastic Net Regression.*

Keywords: GitHub, GitHub API, RESTAPI, Flask Web Application

I. INTRODUCTION

GitHub has emerged as one of the most influential platforms for software development, hosting millions of repositories ranging from open-source projects to enterprise-level applications. A repository's success is often measured by the number of stars, forks, watchers, and contributors, which indicate its popularity, relevance, and impact within the developer community. However, predicting whether a repository will gain popularity is a complex challenge, as multiple factors influence its visibility and success.

The Characterization and Prediction of Popular Projects on GitHub project addresses this challenge by leveraging machine learning techniques to analyze repository attributes and predict their popularity. By extracting key features such as stars, forks, issues, programming language, project description, contributors, and tags from GitHub using the GitHub API, the system processes and trains machine learning models to forecast a repository's future star count.

This project aims to:

[1] Collect and analyse repository data from GitHub, considering various factors that contribute to repository popularity.

[2] Train machine learning models to predict the number of stars a repository may receive based on historical data.

[3]Develop a web-based prediction system using Flask, allowing users to input a GitHub username to obtain output.

[4] Provide insights into repository success factors, helping developers optimize their projects to enhance visibility.

Feature engineering techniques are applied to process text-based fields such as project descriptions and tags using NLTK for natural language processing. The trained models are then used for real-time predictions on new repositories.

Implementation: The project includes a Flask-based web application that allows users to enter a GitHub username and repository name. The system:

- Fetches repository data using GitHub API.
- Preprocesses and converts extracted features into numerical form.
- Uses the trained machine learning model to predict the repository's potential number of stars.
- Displays the predicted popularity to the user in an intuitive interface.

Significance of the Project: This system is valuable for open-source developers, researchers, and organizations aiming to improve their repository's visibility. It helps:

- Developers understand key success factors for repositories.

DOI: 10.48175/IJARSCT-27632

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

- Organizations track trends in GitHub repositories.
- Researchers analyze open-source project growth patterns.

Future enhancements may include deep learning models, real-time GitHub trend analysis, and recommendation systems for improving repository performance. This project contributes to data-driven decision-making in open-source software development and repository management.

II. LITERATURESURVEY

Several studies have investigated factors influencing GitHub repository popularity.

- Borges et al. [1] highlighted that attributes such as the number of forks, contributors, programming language, and the quality of project documentation significantly impact repository visibility and engagement.
- Lima et al. [2] observed that repositories owned by well-known developers or organizations often gain more stars due to their pre-established reputation while Kalliamvakou et al. [3] emphasized that active maintenance, frequent updates, and detailed documentation positively affect the number of followers over time.
- Munaiah et al. [4] classified repositories into categories like open-source, personal, and educational, analyzing their differing contribution patterns. In the area of feature engineering, researchers have widely applied Natural Language Processing (NLP) to extract insights from repository descriptions, README files, and issue discussions. This project applies NLTK-based techniques, including tokenization and stopword removal, to process text and generate meaningful features.
- On the predictive modeling side, Jiang et al. [5] implemented regression models using features such as forks, watchers, issues, and contributors, concluding that ensemble methods like Random Forest and Gradient Boosting outperform basic linear models.
- Zhu et al. [6] introduced time-series forecasting using LSTM models, focusing on predicting popularity trends from historical data. Building on this, the current project employs Random Forest, Linear Regression, and SGDRegressor models, achieving an R^2 score of 0.91.
- Additionally, Hindle et al. [7] explored NLP-based keyword extraction from commit messages and repository metadata to classify projects by domain. This work adopts a tag frequency approach along with NLTK processing to evaluate the influence of textual metadata on popularity.
- For data collection, Gousios et al. [8] demonstrated the utility of GitHub's REST API in retrieving repository metrics, contributor activity, and commit history. Leveraging this, the current project integrates real-time GitHub API data with a Flask web application that allows users to input repository names and obtain predicted popularity scores. In conclusion, this research builds upon previous literature by combining machine learning models, text-based analysis, and real-time data extraction to form a robust popularity prediction system. Future directions include integrating deep learning models, conducting real-time trend analysis, and applying sentiment analysis to issue discussions for improved prediction accuracy.

III. PROPOSEDWORK

Methodology

The Characterization and Prediction of Popular Projects on GitHub project follows a structured methodology divided into three key phases: data collection and preprocessing, machine learning model training, and web application deployment.

A. Data Extraction:

Repository data is collected using the **GitHub API**, including attributes such as stars, forks, watchers, contributors, tags, descriptions, issues, and programming language.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

Dataset description: The dataset for the project "Characterization and Prediction of Popular Projects on GitHub" was sourced primarily from the GitHub API, augmented with additional information to enhance the feature set. The key attributes captured in the dataset include the following:
Repository Name, Stars, Forks, Watchers, Issues, Tags, Description, Contributors
The dataset combines these attributes to create a comprehensive view of what factors influence a GitHub repository's popularity. Each feature was carefully preprocessed—missing values were filled, textual fields were tokenized, and numerical transformations were applied—to ensure consistency and quality for training predictive models.

B. Data Cleaning:
Null values are handled appropriately—missing programming language entries are replaced with "no language," and empty descriptions are set to an empty string. This ensures consistency and completeness in the dataset.

C. Feature Engineering
Text-based features, such as repository descriptions and tags, are processed using NLTK for tokenization and stopword removal.
Numerical transformations are applied to columns with non-standard numeric formats (e.g., "k" for thousands).
Derived features like tag ratio and description-to-numeric representations are computed to capture the influence of topics and content length on repository popularity.

D. Machine Learning Model Training
Algorithm Selection: Multiple regression algorithms are evaluated, including SGDRegressor, Linear Regression, Random Forest Regressor, Lasso Regression, Ridge Regression, and Elastic Net Regression.
Training and Validation: The dataset is split into training and testing sets. Each model is trained using the processed feature set, and performance is assessed using mean squared error (MSE) and R^2 score.
Model Evaluation and Selection: After training, the Random Forest Regressor is identified as the best-performing model based on its R^2 score of 0.91. This model is then saved as a serialized object for future use.

E. Web Application Deployment
Web Framework and User Interface: A Flask-based web application is developed to provide an interactive interface where users can input a GitHub username and repository name.
Real-Time Data Integration: The application queries the GitHub API in real-time for the specified repository, processes the retrieved data, and applies the trained model to predict the number of stars.
Results Display: The predicted popularity is displayed to the user in a clear and accessible format, allowing developers to quickly gain insights into their repository's potential success.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

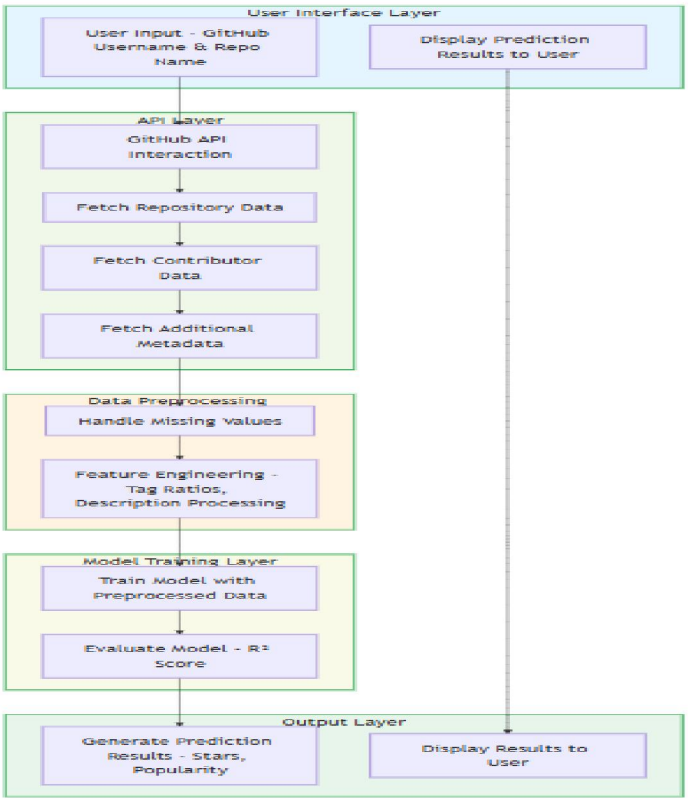

Fig.1.System Architecture

IV. RESULTS

The high R^2 score of the Random Forest model highlights its robustness in handling diverse features, including numerical attributes like stars and forks, as well as engineered text-based features such as tag ratios and description-based numerical values. The inclusion of these advanced features and the use of a comprehensive preprocessing pipeline allowed the model to capture intricate patterns that simpler models, like Linear Regression, could not fully exploit. Furthermore, Random Forest's ability to handle non-linear relationships and interactions between features likely contributed to its superior performance.

Methods	MSE	R2 Score
Gradient Descent	3.303	-4.217
Linear Regression	5681.986	0.875
Random Forest	4801.79	0.910
Lasso Regression	5680.798	0.8753
Ridge Regression	5689.72	0.8749
Elastic Net Regression	5848.99	0.868

Fig.3.1Model Results Comparison

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

While the simpler models offered quicker training times and easier interpretability, they were less effective at capturing complex feature interactions. The success of the Random Forest model demonstrates the importance of feature richness and model flexibility. These findings suggest that more advanced machine learning models, when combined with proper feature engineering, can significantly improve the prediction of GitHub repository popularity.

Fig.3.2This is the landing page of the website.

REGISTRATION PAGE: This is Registration page. In here, user can register with their credentials such as email, username, password.

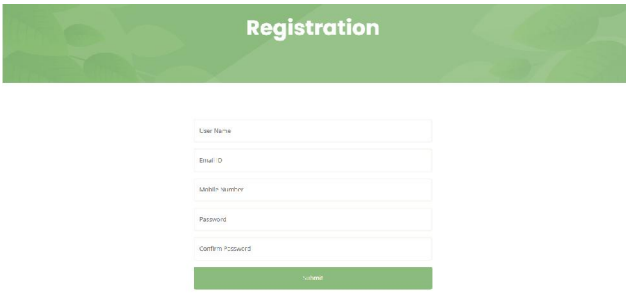

Fig.3.3This is the registration page of the website.

LOGIN PAGE: This is login page. In here user can login with their registered credentials such as email, password.

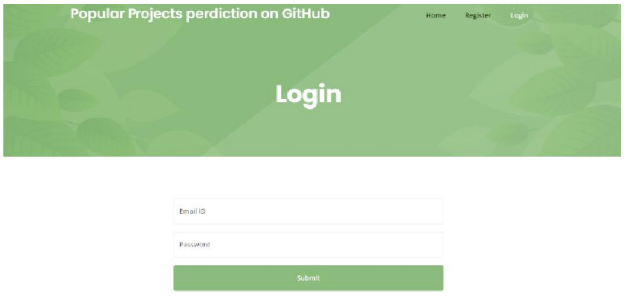

Fig.3.3This is the login page of the website.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

HOME PAGE: This is the user home page. After user successfully login, this page will be display.

Fig.3.4This is the home page of the website.

ABOUT PAGE: This is about section which contains information about our project.

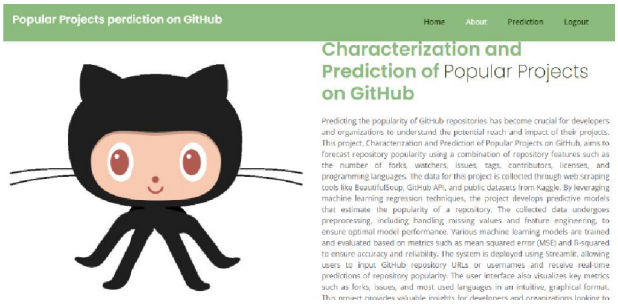

Fig.3.5This is the About page of the website.

Prediction PAGE: In here user can Give their GitHub account username and repository name.

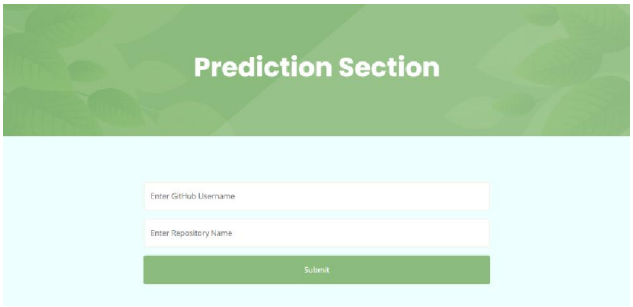

Fig.3.6This is the About page of the website.

Result PAGE: In here user can get the Repository information and predicted popularity.

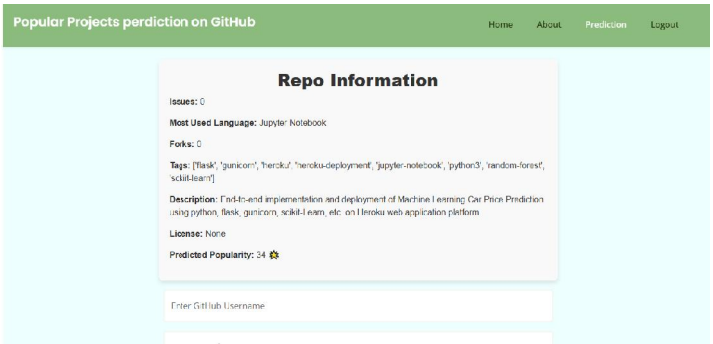

Fig.3.7This is the Result page of the website.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

V. CONCLUSION

The project Characterization and Prediction of Popular Projects on GitHub successfully leverages machine learning to predict the popularity of GitHub repositories based on various features such as stars, forks, watchers, contributors, and programming language. By using a combination of data extraction from the GitHub API and feature engineering techniques, the project provides a robust system capable of making real-time predictions with high accuracy. The Random Forest Regressor was identified as the most accurate model with an R^2 score of 0.91, highlighting its ability to handle complex, non-linear relationships in the data.

Through the web application, users can easily input repository details and receive instant predictions on the repository's potential success, enhancing decision-making for developers and repository maintainers. The system also provides valuable insights into the most popular tags, programming languages, and other trends, which can guide developers in improving their repositories' visibility.

In conclusion, this project demonstrates the power of machine learning in repository analysis, offering a time-saving, cost-effective, and accurate tool for predicting the success of GitHub projects. The system has the potential to be further enhanced by integrating additional features, improving the prediction model, and providing more personalized recommendations for users.

VI. ACKNOWLEDGE

We are also verythankful to P. Srinivasa Rao, Assistant Professor, Department of Computer Science Engineering, ACE Engineering College, for his thoughtful guidance, advice, and valuable suggestions all through this project. We also appreciate our institution for theresourcesandsupportwereceived.Aboveall,wewouldliketoextendour sincere appreciation to the editorial team of IJARSCT for allowing us to publish our work.

REFERENCES

[1] https://ieeexplore.ieee.org/xpl/conhome /7807393
[2] https://ieeexplore.ieee.org/document/97 34131
[3] https://ieeexplore.ieee.org/document/10 529252
[4] https://ieeexplore.ieee.org/document/10 795110/
[5] https://ieeexplore.ieee.org/document/10 555756
[6] https://link.springer.com/article/10.118 6/s40064-016-2897-7
[7] https://ieeexplore.ieee.org/abstract/docu ment/7816479/
[8] https://www.sciencedirect.com/science/ article/pii/S0957417418302793
[9] https://repositorio.ufmg.br/handle/1843/ BIRC-BBLN2S
[10] https://dl.acm.org/doi/abs/10.1145/297 2958.2972966
[11] https://www.sciencedirect.com/scienc e/article/pii/S095058491730304X
[12] https://ieeexplore.ieee.org/abstract/doc ument/9588891/
[13] https://ieeexplore.ieee.org/abstract/doc ument/7884605/
[14] https://link.springer.com/chapter/10.10 07/978-3-030-20883-7_8
[15] https://www.sciencedirect.com/scienc e/article/pii/S095058490600067X
[16] https://dl.acm.org/doi/abs/10.1145/338 3583.3398578
[17] https://www.sciencedirect.com/scienc e/article/pii/S0950584921000902
[18] https://www.mdpi.com/2078- 2489/13/2/73
[19] https://doi.org/10.1109/TSE.2020.29 91267
[20] https://doi.org/10.1145/3468264.346855 0
[21] https://doi.org/10.1016/j.jss.2021.11109 7
[22] https://doi.org/10.1016/j.infsof.2020.106 360
[23] https://doi.org/10.1007/s10664-014 9323-9
[24] https://doi.org/10.1145/2702123.270254 9
[25] https://doi.org/10.1145/2635868.263592 2

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

[26] https://doi.org/10.1145/2884781.28847 93
[27] https://octoverse.github.com
[28] https://doi.org/10.1016/j.cose.2020.101 754
[29] https://doi.org/10.1016/j.jss.2021.111120
[30] https://doi.org/10.1109/TSE.2019.2934417