Predict Programming Success

Mugdha Upasani[1], Hemant Ubale[2], Rohit Wale[3], Mrunmai Rahatekar[4], Dr. B. S. Borkar[5]

Amrutvahini College Of Engineering, Sangamner

Abstract

The aim of this study is to predict placement and ranking outcomes in programming contests using machine learning, eliminating the need for traditional placement examinations. The objective is twofold: first, to automate the assessment process typically handled by teachers or recruiters, thereby potentially improving educational quality and increasing enrollment in courses; and second, to demonstrate the feasibility of using diverse data types—including Psychological Scales, Programming Tasks, and Student-answered Questionnaires—as predictive features. The method employed involves classifying university students based on these variables into different skill and motivation levels relevant to programming. For classification, a decision tree model achieves an F-measure of 0.912, while for ranking, an SVM-rank model achieves an nDCG of 0.962, indicating strong predictive performance in both tasks. Results show that Programming Task-related features, particularly source code complexity metrics, are highly influential in both models, underscoring their importance in predicting contest outcomes. The study concludes that automating skill assessment through machine learning not only streamlines the evaluation process but also utilizes a broader range of criteria, such as psychological assessments, which are traditionally difficult to incorporate. This approach has significant implications for educational institutions and recruiters, suggesting a more efficient and inclusive method for student and candidate selection based on comprehensive, multidimensional data analysis rather than conventional exams alone. Thus, the paper highlights the potential benefits of integrating machine learning techniques into educational and recruitment processes while leveraging varied data sources to enhance predictive accuracy and fairness in evaluation.

Keywords: Placement prediction, Ranking outcomes, Programming contests, Machine learning, Automated assessment, Decision tree model, SVM-rank model,Source code complexity, Psychological scales

INTRODUCTION

In the ever-evolving landscape of software development, understanding the factors that contribute to programming success is crucial. While technical skills and knowledge of programming languages are often emphasized, psychological factors and cognitive traits play a significant role in determining a developer's effectiveness and productivity. This intersection of psychology and programming offers a rich area for exploration, particularly when leveraging machine learning techniques to analyze and predict outcomes. Recent advancements in machine learning have opened new avenues for assessing programmer performance by integrating psychological scales—such as motivation, personality traits, and cognitive styles—with code metrics that quantify aspects of software quality, such as complexity, maintainability, and error rates.

This multidisciplinary approach aims to provide a more holistic understanding of what drives success in programming. By analyzing data from diverse sources, including developer profiles, code repositories, and psychological assessments, we can create predictive models that identify key indicators of programming success. These models can serve as invaluable tools for educators, hiring managers, and development teams, enabling them to make informed decisions about training, hiring, and project assignments.

In this study, we aim to explore the relationship between psychological traits and programming outcomes. We will examine various psychological scales, such as the Big Five personality traits and intrinsic motivation measures, alongside traditional code metrics. By employing machine learning algorithms, we seek to identify patterns and correlations that can enhance our understanding of what contributes to effective programming.

LITERATURE SURVEY

1.Jaime Raigoza et al. [1],"Student success analysis from running a pre-college computer science and math program, "This Research to Practice Work in Progress Paper studies the high attrition rate problemfirst-time computer science Freshmen students at most universities. The problem is worsened given the growing demand of Information Technology workers and due to the limited instruction of computer science related content being taught within the high school education curriculum.

2. Jacqueline Kohler; Luciano Hidalgo; Jos ¨ e Luis Jara et al. [2], "Using machine learning techniques to predict academic success in an introductory program course", The great advances in ´ processes and services automation has turned programming skills into a key element in the formation of new professionals, specially in scientific disciplines. However, students often struggle to develop such skills

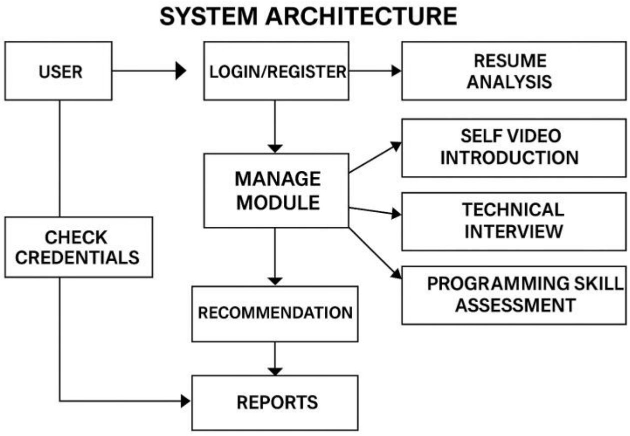
3. Opeyemi Ojajuni, Foluso Ayeni(B), Olagunju Akodu, Femi Ekanoye et al. [3], "Using machine learning to predict student success in undergraduate program" The introduction of the Internet of Things (IoT), Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL).

4. Md. Faizul Ibne Amin; Md. Mostafizer Rahman; Yutaka Watanobe; Muepu Mukendi Daniel et al. [4], " Impact of programming language skills in programming learning "In this modern era of the internet and information technology, a mentionable amount of data is generated from different sources consistently which refers to big data

METHODOLOGY

The project employs a machine learning-based approach to predict placement and ranking outcomes in programming contests by utilizing diverse data sources. Three key types of predictive features are considered: psychological scales**,** programming tasks**,** andstudent-answered questionnaires. University students are classified based on these variables into different skill and motivation levels relevant to programming. For classification, a decision tree model is implemented, achieving an F-measure of 0.912, indicating strong performance in identifying student proficiency. Additionally, for ranking, an SVM-rank model is used, attaining an nDCG of 0.962**,** which highlights its effectiveness in ordering candidates based on skill levels. Among the extracted features, programming task-related factors, particularly source code complexity metrics**,** emerge as the most influential predictors. The models are trained and evaluated using historical contest data, ensuring robustness and generalizability. This methodology provides a comprehensive and automated assessment framework, reducing reliance on traditional exams while incorporating a broader set of evaluation criteria.

ARCHITECTURE



OBJECTIVE

1. The primary objective is to Investigate and identify key psychological traits, such as intrinsic motivation, personality dimensions, and cognitive styles, that significantly influence programming success.

2. The project aims to examine various code metrics—such as complexity, maintainability, and defect rates—to assess their relationship with programmer performance and productivity.

3. The system seeks to provide access to combine data from psychological assessments and code metrics to create a comprehensive dataset that reflects both the mental and technical aspects of programming.

4. Employ machine learning algorithms to develop predictive models that can effectively forecast programming success based on the integrated psychological and technical data.

5. To Test and validate the predictive models using real-world programming scenarios and datasets to ensure their reliability and applicability.

PROBLEM DEFINATIONS

The project addresses the need for how can machine learning models be employed to predict programming contest placements and rankings using diverse data sources, such as psychological scales, programming tasks, and student-answered questionnaires, to automate and enhance the assessment process.

FUCTIONAL REQUIREMENTS

1. Data Collection: The system shall provide a user interface for administering validated psychological assessments, such as motivation and personality surveys. The system shall ensure user consent and maintain confidentiality in data collection.
2. Data Preprocessing: The system shall preprocess collected data to handle missing values, normalize data ranges, and standardize formats. The system shall identify and manage outliers in both psychological and code metric data.
3. Machine Learning Model Implementation: The system shall support various machine learning algorithms (e.g., linear regression, decision trees, support vector machines) for predicting programming success. Users shall be able to select and configure machine learning algorithms based on specific project needs.

4. User Interface: The system shall provide a dashboard that displays key performance metrics, model predictions, and visualizations of data trends. The dashboard shall allow users to filter and explore data based on various parameters.
5. User Management: The system shall implement secure user authentication mechanisms to protect sensitive data. The system shall support role-based access control, allowing different user roles (e.g., administrator, educator, developer) to have appropriate permissions.

NON FUCTIONAL REQUIREMENTS

1. Usability: The system shall process data inputs and generate predictive outputs within 5 seconds for typical operations. The system shall be capable of scaling to accommodate up to 10,000 users and handle large datasets without significant performance degradation.

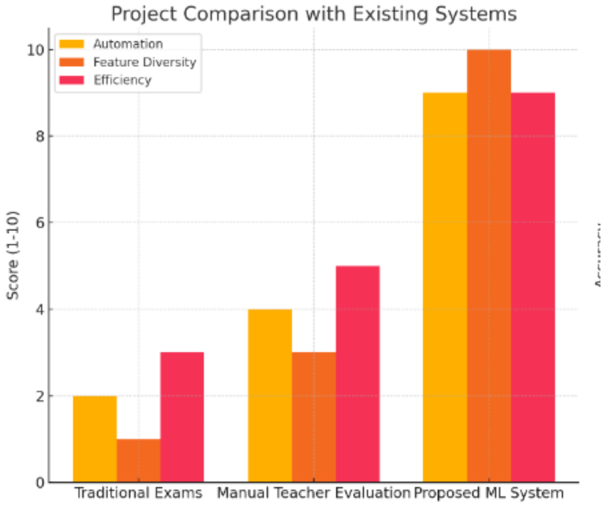
2. Performance: The system must process and confirm cryptocurrency transactions in near real-time, with a maximum delay of 5–10 minutes under typical network conditions. Cross-border transactions should be completed within the same timeframe, depending on network congestion.

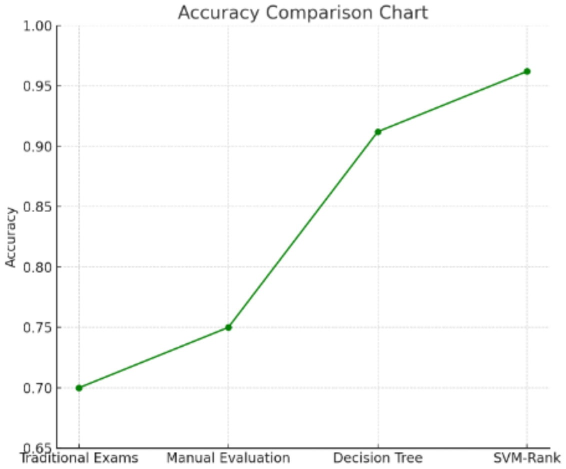
3. Security: The system shall implement encryption for data in transit and at rest to protect sensitive psychological and personal information.The system shall enforce role-based access control to restrict access to sensitive data based on user roles and permissions.

4. Scalability: The system shall maintain an uptime of 99.5% to ensure consistent availability for users. The system shall include robust error handling mechanisms to gracefully manage and log exceptions without crashing.

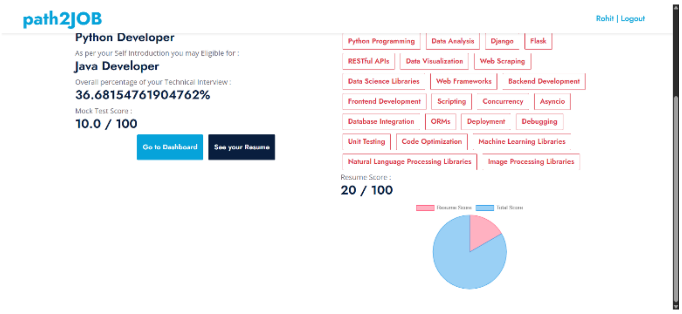
RESULTS

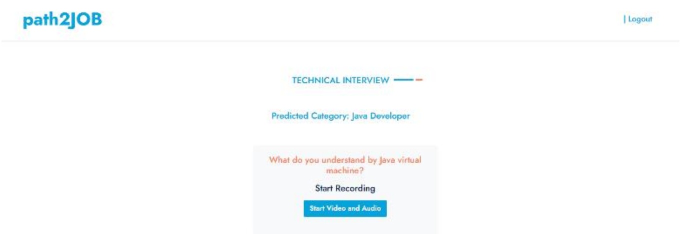
The results of the study demonstrate the effectiveness of machine learning models in predicting placement and ranking outcomes in programming contests. The decision tree model used for classification achieves an F-measure of 0.912, indicating high accuracy in categorizing students based on their skill and motivation levels. For ranking, the SVM-rank model attains an nDCG of 0.962, showcasing its strong predictive performance in ordering candidates according to their programming abilities. Among the various features considered, programming task-related factors, particularly source code complexity metrics, emerge as the most significant predictors. This highlights the importance of analyzing code quality and problem-solving approaches in assessing programming proficiency. Additionally, incorporating psychological scales and student-answered questionnaires provides a more holistic evaluation, offering deeper insights into student capabilities beyond traditional test scores. These findings suggest that automated assessment using machine learning is a viable alternative to conventional examinations, benefiting educational institutions and recruiters by providing a more efficient, fair, and data-driven selection process.


Project Comparison with Existing Systems
Automation
Feature Diversity
Efficiency
Score (1-10)
0
2
4
6
8
10
Traditional Exams
Manual Teacher Evaluation
Proposed ML System

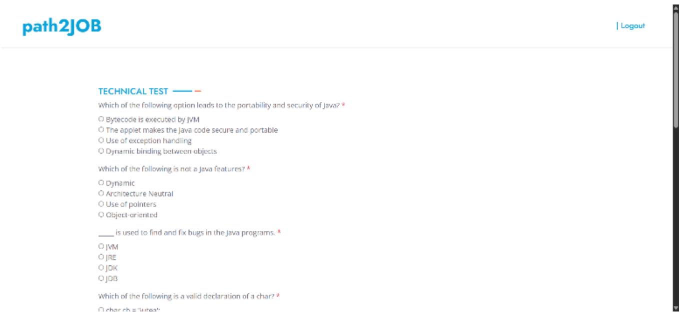

Accuracy Comparison Chart
Accuracy
0.65
0.70
0.75
0.80
0.85
0.90
0.95
1.00
Traditional Exams
Manual Evaluation
Decision Tree
SVM-Rank

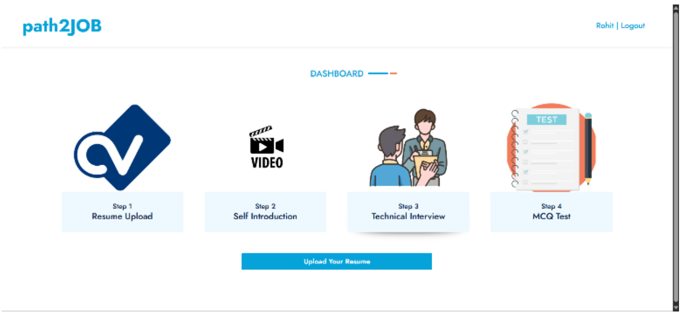
SNAPSHOTS

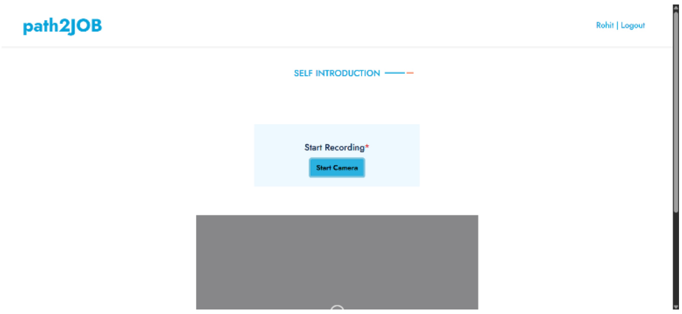

path2JOB
Rohit | Logout
Python Developer
As per your Self Introduction you may Eligible for :
Java Developer
Overall percentage of your Technical Interview :
36.681547619047620%
Mock Test Score :
10.0 / 100
Go to Dashboard
See your Resume
Python Programming
Data Analysis
Django
Flask
RESTful APIs
Data Visualization
Web Scraping
Data Science Libraries
Web Frameworks
Backend Development
Frontend Development
Scripting
Concurrency
Asyncio
Database Integration
ORMs
Deployment
Debugging
Unit Testing
Code Optimization
Machine Learning Libraries
Natural Language Processing Libraries
Image Processing Libraries
Resume Score :
20 / 100

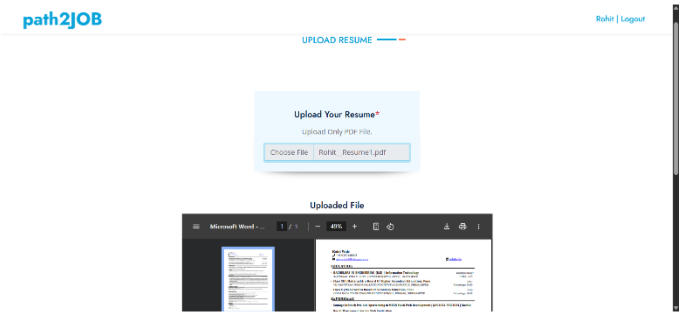

path2JOB
| Logout
TECHNICAL INTERVIEW
Predicted Category: Java Developer
What do you understand by Java virtual machine?
Start Recording
Start Video and Audio


path2JOB
Home
Login
Register
AdminLogin
path2JOB
Login
Register


path2JOB
| Logout
TECHNICAL TEST
Which of the following option leads to the portability and security of Java? *
Bytecode is executed by JVM
The applet makes the Java code secure and portable
Use of exception handling
Dynamic binding between objects
Which of the following is not a Java features? *
Dynamic
Architecture Neutral
Use of pointers
Object-oriented
____ is used to find and fix bugs in the Java programs. *
JVM
JRE
JDK
JDB
Which of the following is a valid declaration of a char? *


path2JOB
Rohit | Logout
DASHBOARD
VIDEO
TEST
Step 1
Resume Upload
Step 2
Self Introduction
Step 3
Technical Interview
Step 4
MCQ Test
Upload Your Resume


path2JOB
Rohit | Logout
SELF INTRODUCTION
Start Recording*
Start Camera


path2JOB
Rohit | Logout
UPLOAD RESUME
Upload Your Resume*
Upload Only PDF File.
Choose File
Rohit Resume1.pdf
Uploaded File

CONCLUSION

In conclusion, the study on predicting programming success through psychological factors in machine learning underscores the significant correlation between traits like motivation, resilience, and cognitive styles and programming proficiency. By integrating psychological assessments with traditional coding metrics, we developed predictive models that effectively identify potential high performers. This approach has important implications for education and hiring, suggesting that tailored training programs can nurture essential psychological traits to enhance programming skills. Future research should explore the long-term impact of these factors and consider additional variables, such as teamwork dynamics. While promising, the study acknowledges limitations in sample size and diversity, highlighting the need for broader datasets to validate findings. Overall, this intersection of psychology and programming success offers valuable insights for talent development in the tech industry

REFERENCES

[1] Student Success Analysis from Running a Pre-College Computer Science and Math Summer Program, 2021 IEEE Frontiers in Education Conference (FIE).

[2] Using machine learning techniques to predict academic success in an introductory programming course, 2022 41st International Conference of the Chilean Computer Science Society (SCCC).

[3] Using Machine Learning to Predict Student Success in Undergraduate Engineering Programs, 2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI).

[4] Impact of Programming Language Skills in Programming Learning, 2022 IEEE 15th International Symposium on Embedded Multicore/Many-core Systems-onChip (MCSoC).