

Animation Game Engine For Interactive Presentation Of Educational Media

Ashwin Ravi (3122215001014)
Matli Mithilesh Reddy (3122215001050)

Guided by
Dr. Sakaya Milton R
Professor
SSN College of Engineering

April 08, 2025

Outline

- 1 Motivation
- 2 Problem Statement
- 3 Literature Review
- 4 Purpose
- 5 Functionality
- 6 Architecture

Outline

- 7 List of Modules
- 8 Output Screenshots
- 9 Performance Analysis
- 10 Impacts Pertaining to Society, Health, Safety, Legal, Environment and Culture
- 11 Plan for submitting Project Work

Motivation

- Traditional teaching methods often struggle to effectively convey dynamic concepts, making it essential to develop interactive solutions that enhance comprehension and engagement.
- Visualization plays a crucial role in learning, and by integrating real-time animations, students can better understand complex theories in mathematics, science, and computer science.
- Interactive learning fosters deeper engagement by allowing students to explore concepts at their own pace, reinforcing retention and improving problem-solving skills.
- The accessibility and ease of use of this engine empower educators to create high-quality educational content without requiring advanced technical expertise.
- Adaptive learning through real-time feedback and personalized pathways ensures that students receive tailored support, providing various options.
- With advancements, integrating interactive media in education paves the way for a more engaging, effective learning experience.

Problem Statement

The goal is to develop a flexible and interactive animation engine for educational media using the Godot Engine and GDScript. Traditional educational tools lack interactivity, adaptability, and real-time feedback, leading to passive learning experiences. Static materials struggle to convey dynamic processes, making it difficult for learners to grasp abstract concepts.

To address these challenges, the animation engine will:

- Provide keyframe-based animations for smooth visual representation.
- Offer an intuitive scripting interface for easy content creation.
- Integrate interactive features like quizzes and variable manipulation.
- Enable real-time user tracking and personalized feedback.
- Ensure lightweight performance for broad accessibility.

Literature Review

- **Algorithm Visualization:** Animations enhance conceptual understanding and engagement in computational processes (Kogan et al., 2024; Végh et al., 2017). AR-based visualizations further improve cognitive load management (Paredes-Velasco et al., 2022).
- **Game-Based Learning:** Gamification boosts student motivation and assessment-driven learning (Bandura et al., 2020; Shute et al., 2012). Godot Engine is recognized for developing interactive learning experiences (Manzur et al., 2018; Ranaweera et al., 2024).
- **Student-Driven Visualization:** Encouraging students to create algorithm animations enhances understanding and problem-solving (Stasko et al., 1997; Hansen et al., 2002).
- **Cognitive Load & Adaptability:** Interactive media reduces cognitive load, improving retention and adaptability in learning (Mayer et al., 2001; Wilson et al., 2019). Adaptive tools refine problem-solving strategies (Wu et al., 2021; Carneiro et al., 2022).

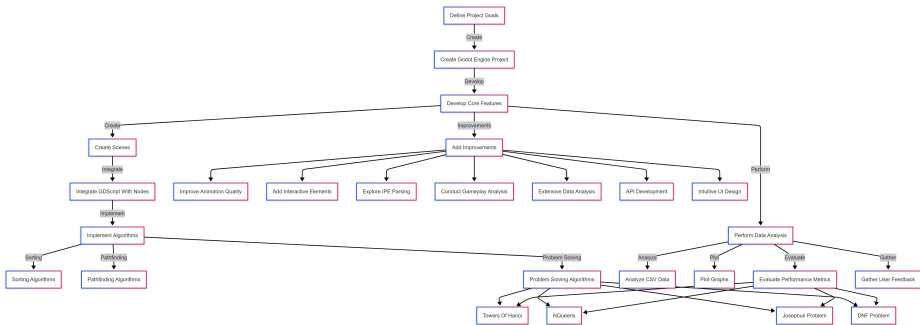
Purpose

- Design an accessible animation engine using Godot Engine and GDScript for dynamic educational media.
- Simplify the creation of keyframe-based animations to illustrate complex concepts effectively.
- Foster active learner engagement with integrated quizzes, variable manipulation, and real-time feedback mechanisms.
- Enable real-time tracking and analysis of learner interactions to personalize learning paths.
- Ensure compatibility with low-end devices, democratizing access to interactive learning technologies.

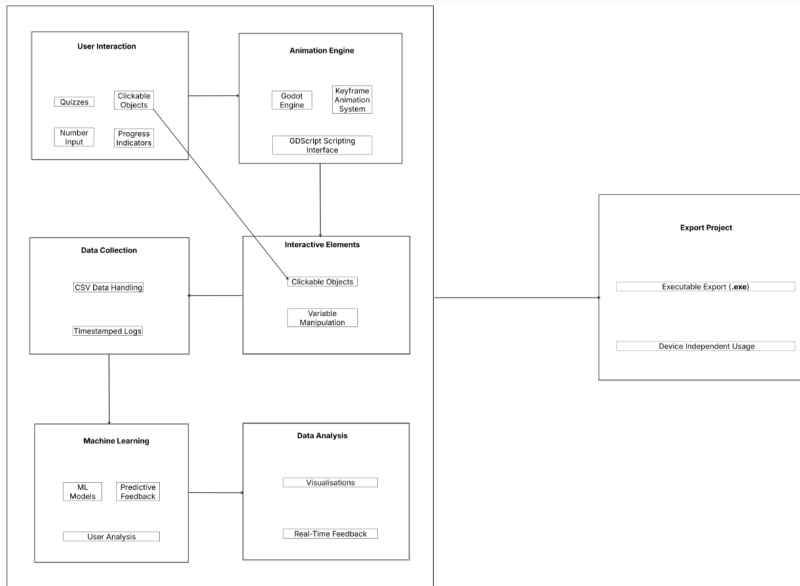
Functionality

- Develops an interactive animation engine using Godot Engine and GDScript.
- Facilitates creation of keyframe-based animations to visualize dynamic, complex processes.
- Embeds interactive components like quizzes, clickable objects, and variable manipulation.
- Tracks learner interactions in real time and records data in CSV format for analysis.
- Integrates machine learning models to classify learner performance and provide predictive feedback.
- Exports the educational content as standalone executable files compatible with multiple platforms.
- Ensures smooth performance on low-end hardware, promoting broad accessibility.
- Allows educators to create engaging content without deep programming expertise.

Architecture Diagram - Version 1



Architecture Diagram - Version 2



List of Modules

- Project Initialization
- Godot Engine Setup
- Core Feature Development
- Algorithm Implementation
- Data Collection and Analysis
- Performance Optimization
- User Interface and Experience
- Executable Export

Project Initialization Module

- Defines project objectives, scope, and deliverables.
- Structures development phases and resource distribution.
- Identifies pedagogical and technical challenges early to refine execution strategy.

Godot Engine Setup Module

- Configures Godot's scene hierarchy, node structure, and project settings.
- Sets up UI components, input handlers, and autoload scripts.
- Organizes project assets and workflow standards for maintainability.

Core Feature Development Module

- Designs scenes, UI layouts, and learner interaction models.
- Implements keyframe-based animations using AnimationPlayer.
- Integrates GDScript logic for controlling animations, quizzes, and variable manipulation.

Algorithm Implementation Module

- Implements algorithm visualizations like Towers of Hanoi, N-Queens, Josephus Problem, and Sorting Algorithms.
- Provides real-time demonstrations and variable manipulation for deeper understanding.
- Encourages user-driven exploration of algorithm behavior.

Data Collection and Analysis Module

- Records user interactions like quiz responses, engagement patterns, and player-specific analysis.
- Stores structured data in CSV format for analysis.
- Integrates machine learning models for classification and predictive insights.

Performance Optimization Module

- Optimizes animation rendering, memory usage, and UI responsiveness.
- Reduces latency and computational overhead for hardware compatibility.
- Ensures smooth, efficient operation without sacrificing user experience.

User Interface and Experience Module

- Designs intuitive navigation, clear feedback mechanisms, and easy-to-use layouts.
- Supports customizable themes, layouts, and multilingual content.
- Prioritizes accessibility to suit educators and learners from varied backgrounds.

Executable Export Module

- Utilizes Godot's export system to compile the project into standalone executables (.exe, .apk, etc.).
- Ensures compatibility across multiple platforms without requiring Godot installation.
- Facilitates easy distribution and usage in varied educational setups.

Output Screenshots

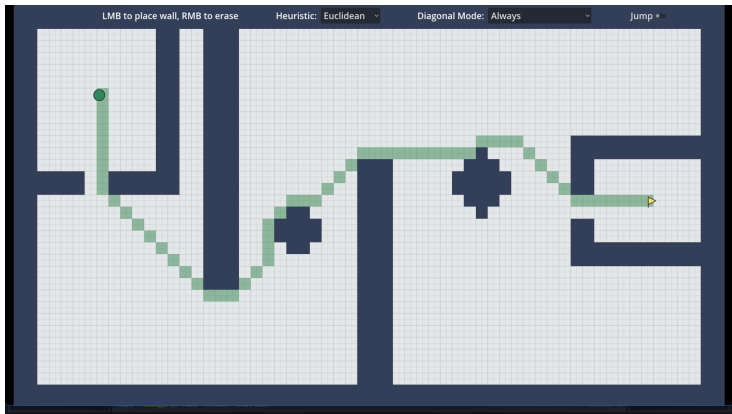


Figure: A-Star Implementation

Output Screenshots

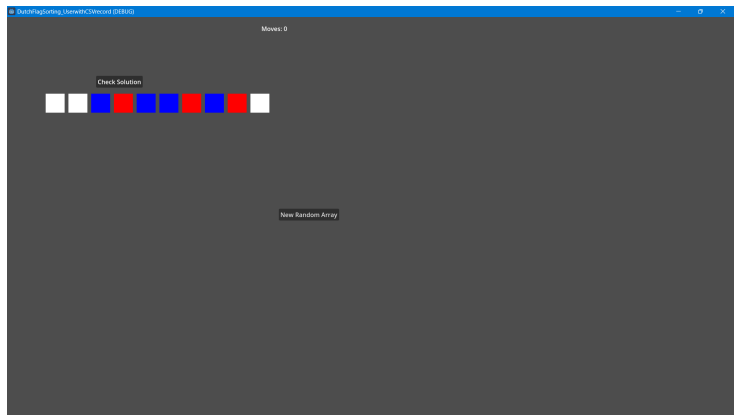


Figure: DNF Implementation

Output Screenshots

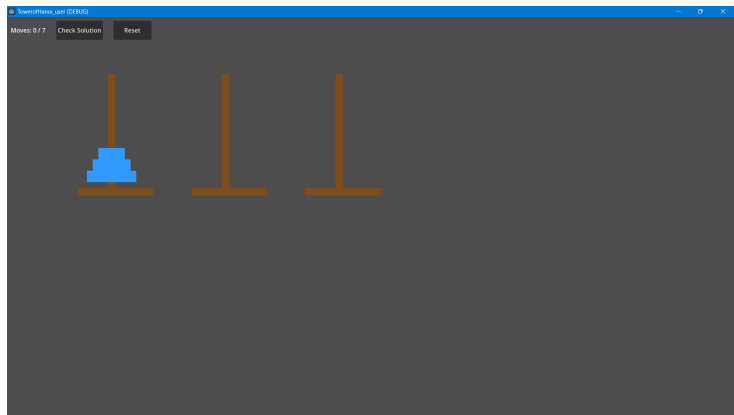


Figure: Towers of Hanoi Implementation

Output Screenshots

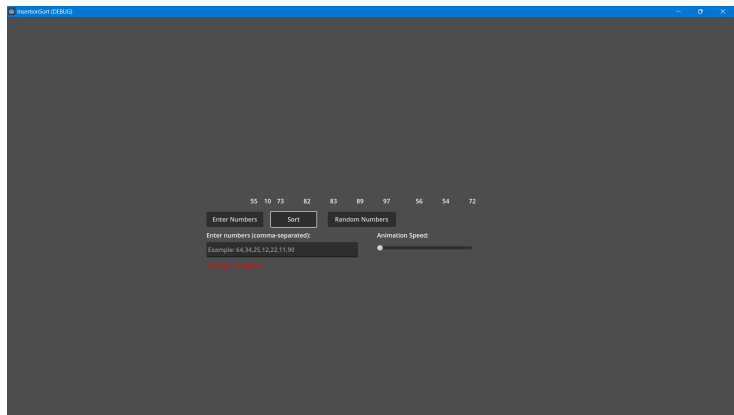


Figure: Insertion Sort Implementation

Output Screenshots

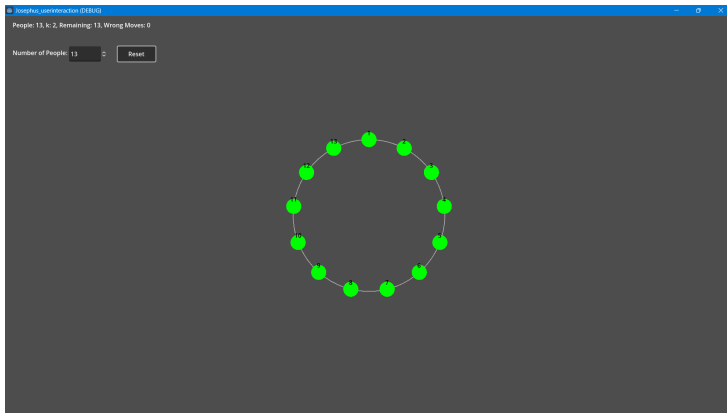


Figure: Josephus Implementation

Output Screenshots

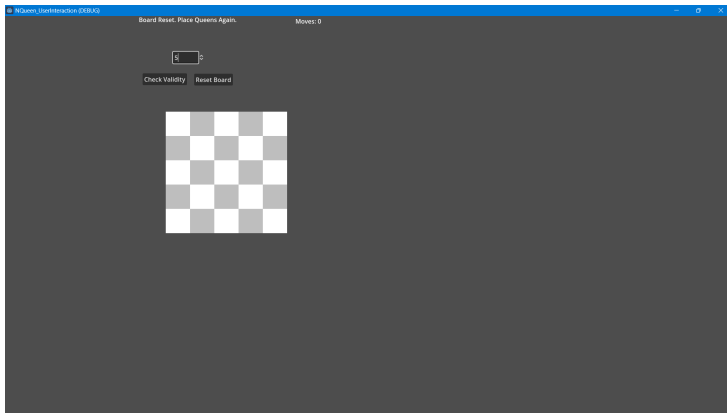


Figure: N-Queens Implementation

Output Screenshots

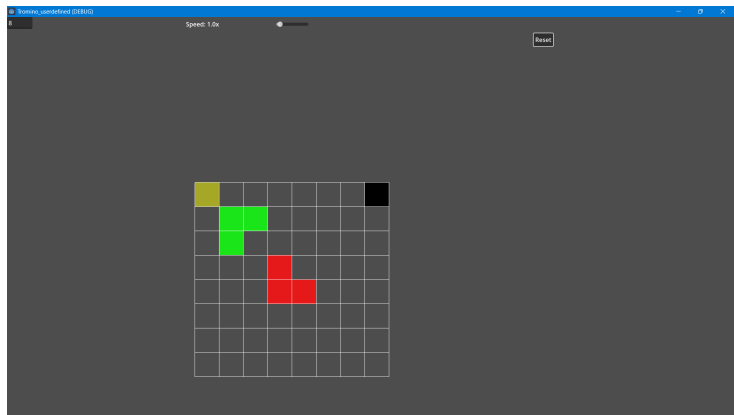


Figure: Tromino Implementation

Output Screenshots

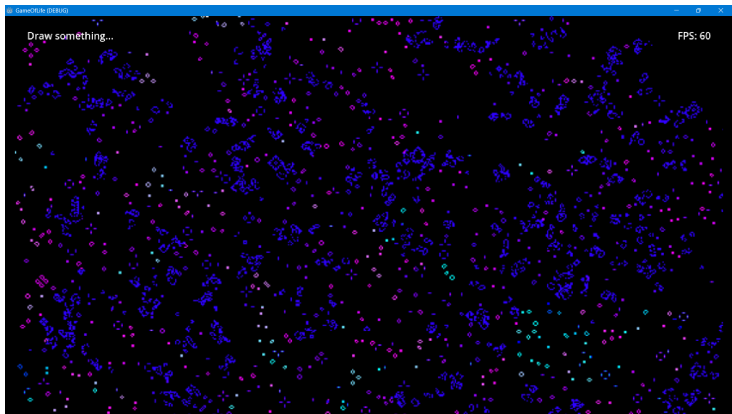


Figure: Cellular Automaton Implementation

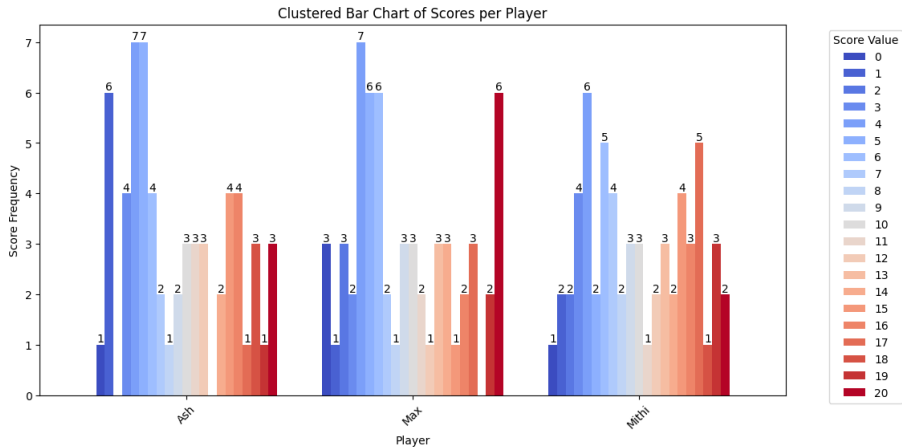
Performance Analysis

- Dutch National Flag Algorithm
- Tower of Hanoi Algorithm
- Josephus Problem Algorithm
- N-Queens Problem Algorithm

Dutch National Flag Algorithm

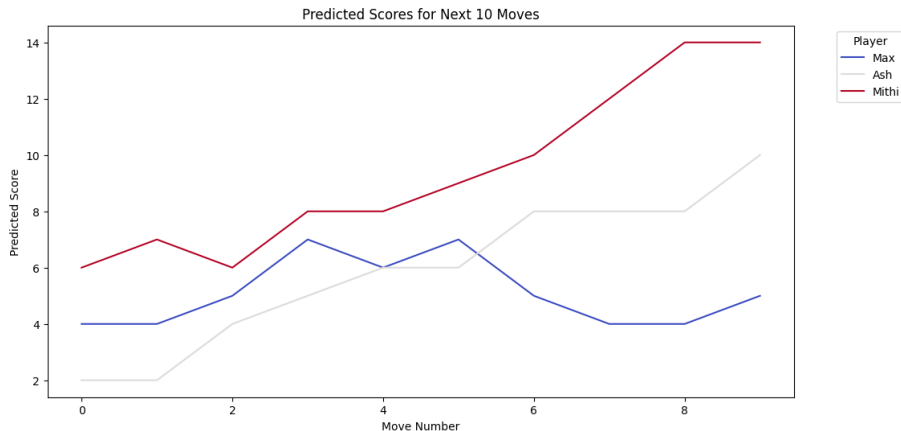
- Achieved excellent accuracy with an R-squared score of 1.00, indicating near-perfect prediction capability.
- Random Forest Regressor effectively identifies whether the color sorting was successful.
- Error metrics: Mean Absolute Error (MAE) of 0.06 and Root Mean Squared Error (RMSE) of 0.12, demonstrating high precision.
- Predicted scores range between 2 and 9, accompanied by remarks such as "Correct! The colors are properly sorted!" and "Not quite right. Keep trying!"
- Despite minor fluctuations, the model consistently offers valuable insights into sorting accuracy and performance trends.

Dutch National Flag Algorithm - Metrics



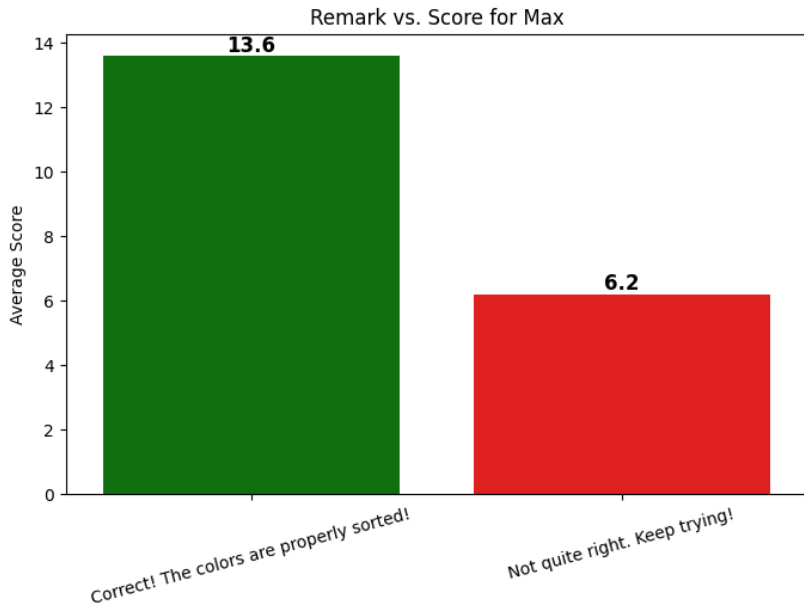
Clustered Bar Chart Of Scores Per Player In DNF

Dutch National Flag Algorithm - Metrics

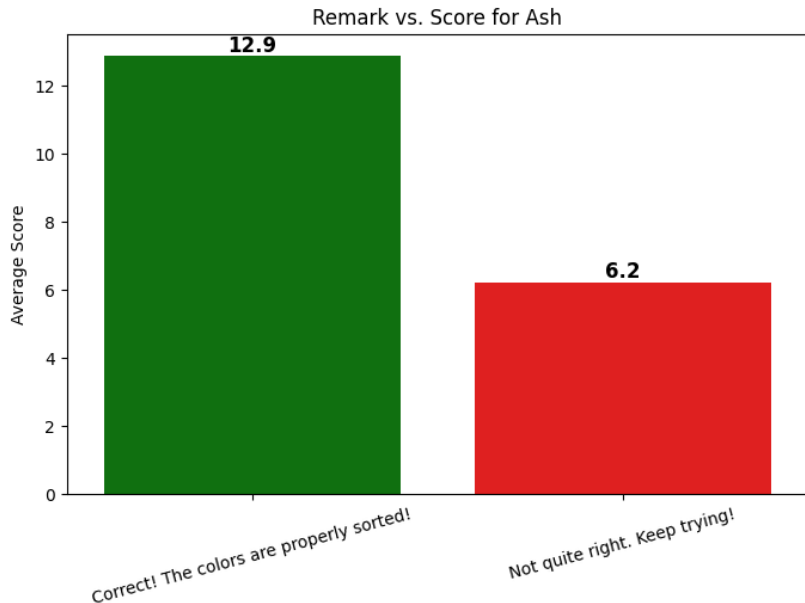


Predicted Scores For Next 10 Moves(dynamic) In DNF

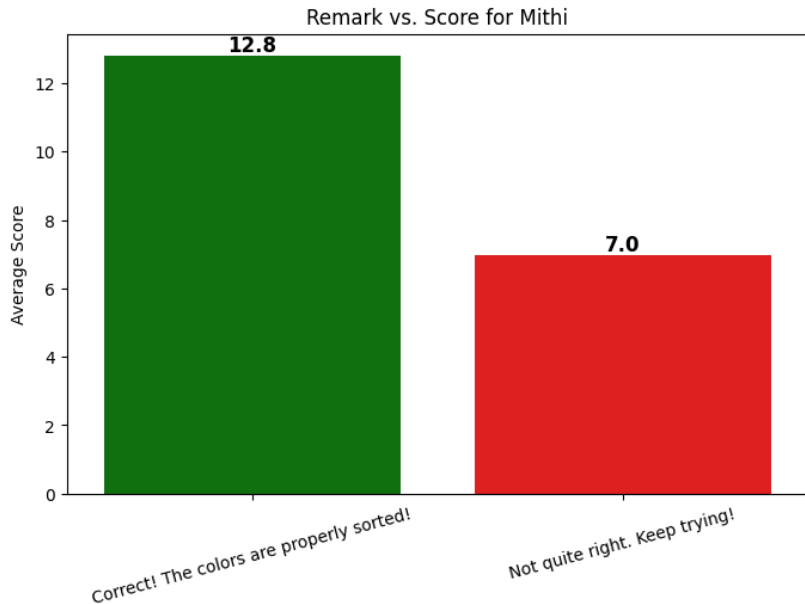
Dutch National Flag Algorithm - Metrics



Dutch National Flag Algorithm - Metrics



Dutch National Flag Algorithm - Metrics



Dutch National Flag Algorithm - Classification Report

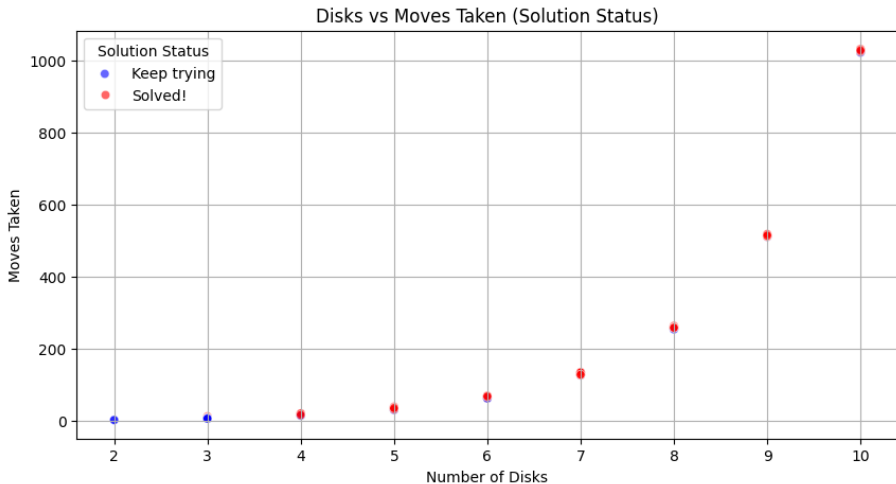
Metric	Value
Mean Absolute Error (MAE)	0.06
Mean Squared Error (MSE)	0.01
Root Mean Squared Error (RMSE)	0.12
R-squared (R^2 Score)	1.00

Table: Performance Metrics for Dutch National Flag Algorithm using Random Forest Regressor

Tower of Hanoi Algorithm

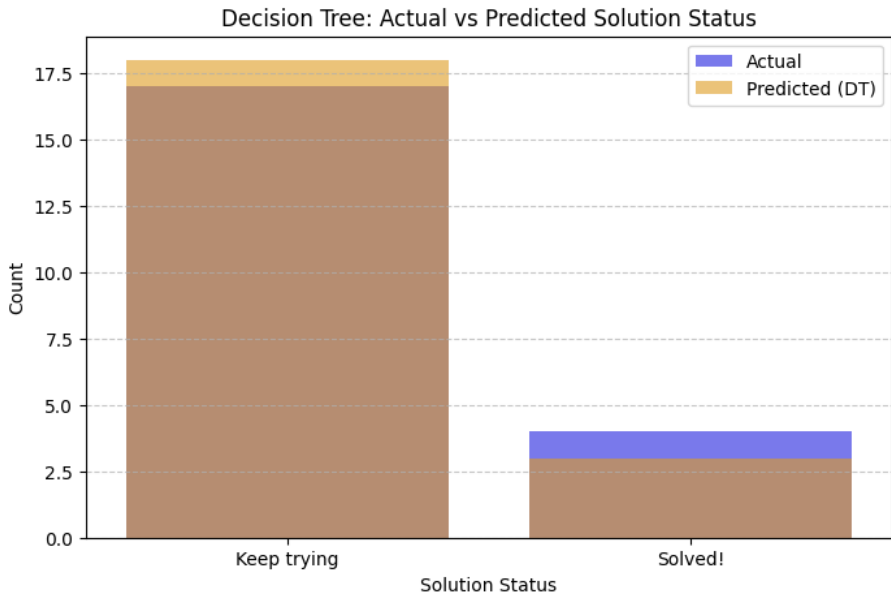
- The Tower of Hanoi model was evaluated using Decision Tree Classifier and Support Vector Machine (SVM) Classifier approaches.
- Decision Tree Classifier achieved high classification accuracy of 95%, correctly distinguishing between "Solved!" and "Keep trying" outcomes.
- Classification report indicates strong precision and recall for the "Keep trying" class, with slightly lower recall for "Solved!" cases.
- SVM Classifier demonstrated lower accuracy of 43%, with inconsistent performance, especially in identifying "Solved!" cases.
- Overall, Decision Tree Classifier outperformed SVM, providing reliable classifications of learner performance across varying disk counts.

Tower of Hanoi - Metrics

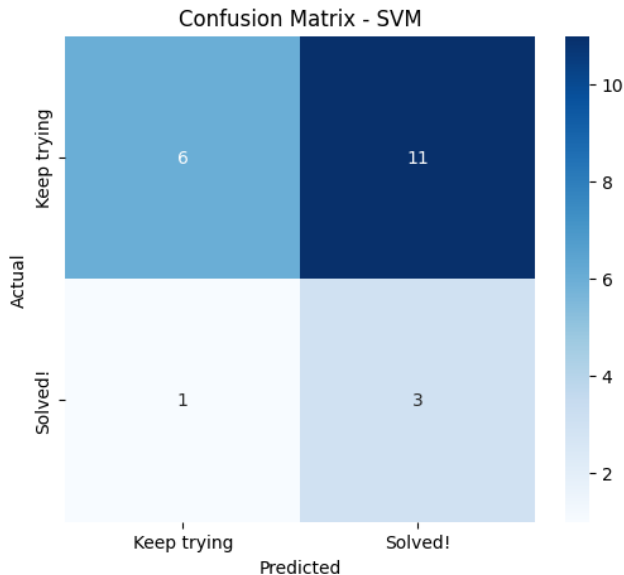


Disks vs Moves Taken In Towers Of Hanoi

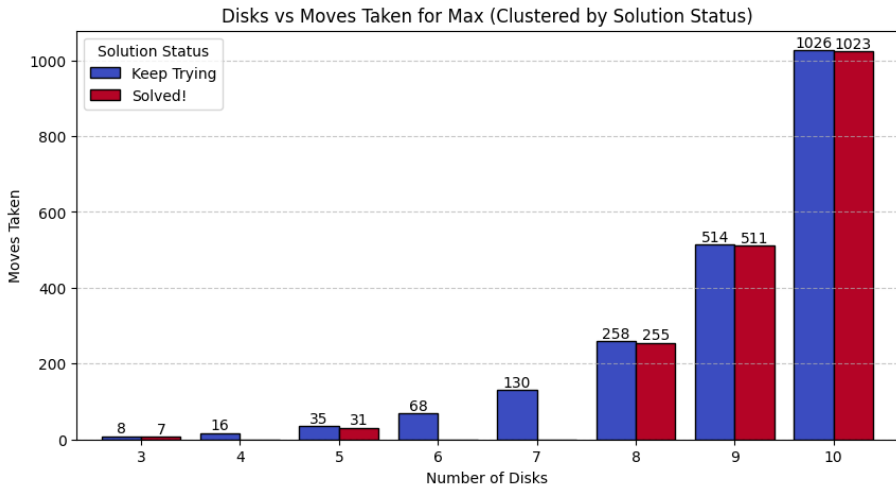
Tower of Hanoi - Metrics



Tower of Hanoi - Metrics

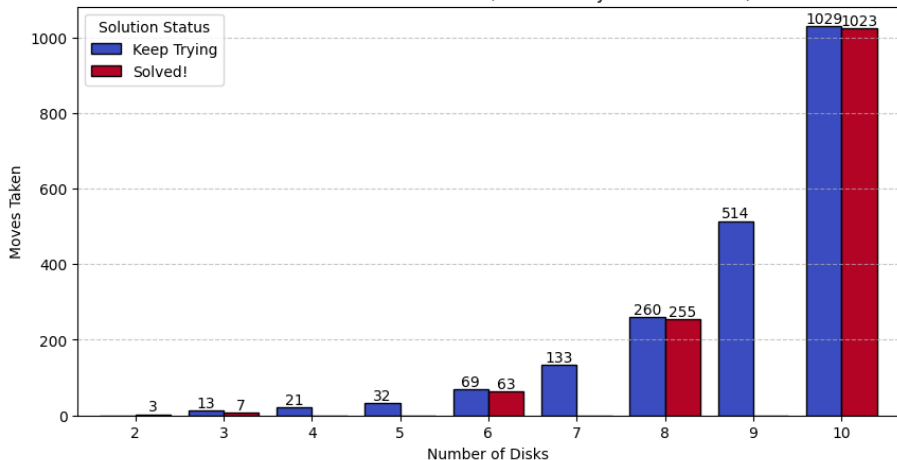


Tower of Hanoi - Metrics

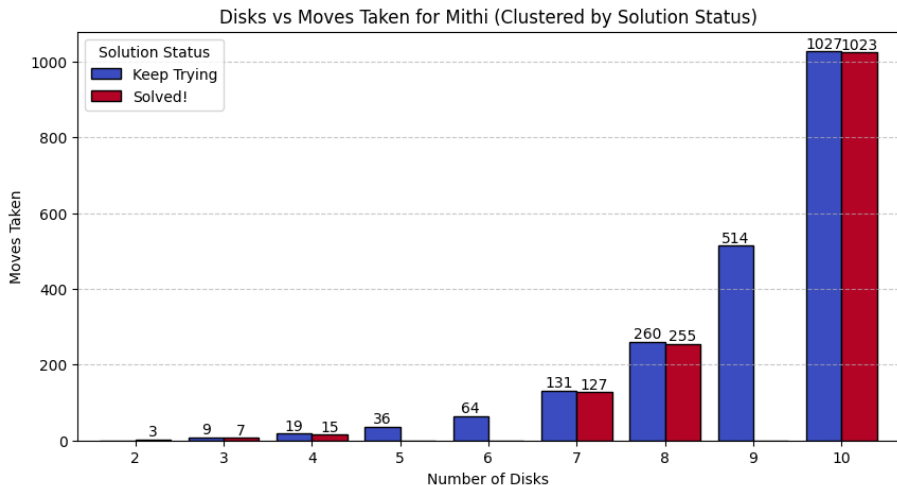


Tower of Hanoi - Metrics

Disks vs Moves Taken for Ash (Clustered by Solution Status)



Tower of Hanoi - Metrics



Tower of Hanoi - Decision Tree Classifier

Decision Tree Classifier				
Class	Precision	Recall	F1-Score	Support
Keep trying	0.94	1.00	0.97	17
Solved!	1.00	0.75	0.86	4
Accuracy	-	-	0.95	21
Macro avg	0.97	0.88	0.91	21
Weighted avg	0.96	0.95	0.95	21

Table: Classification Report for Tower of Hanoi using Decision Tree Classifier

Tower of Hanoi - SVM Classifier

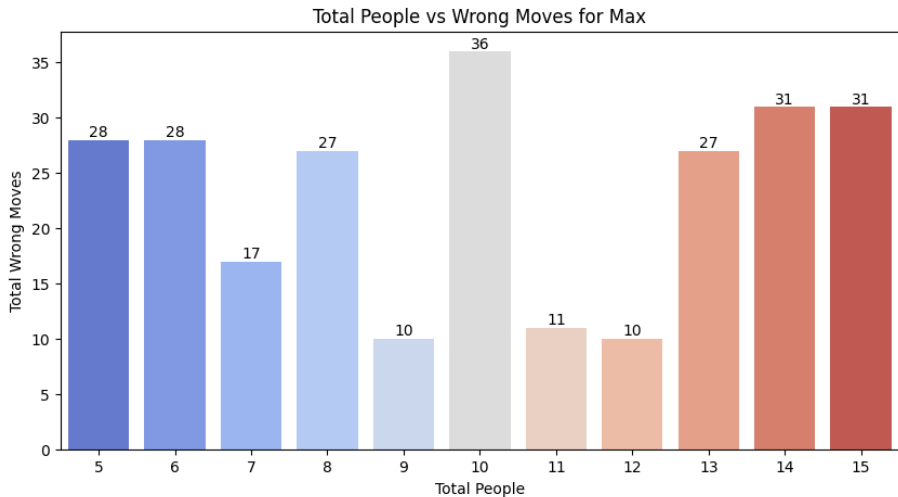
SVM Classifier				
Class	Precision	Recall	F1-Score	Support
Keep trying	0.86	0.35	0.50	17
Solved!	0.21	0.75	0.33	4
Accuracy	-	-	0.43	21
Macro avg	0.54	0.55	0.42	21
Weighted avg	0.73	0.43	0.47	21

Table: Classification Report for Tower of Hanoi using SVM Classifier

Josephus Problem Algorithm

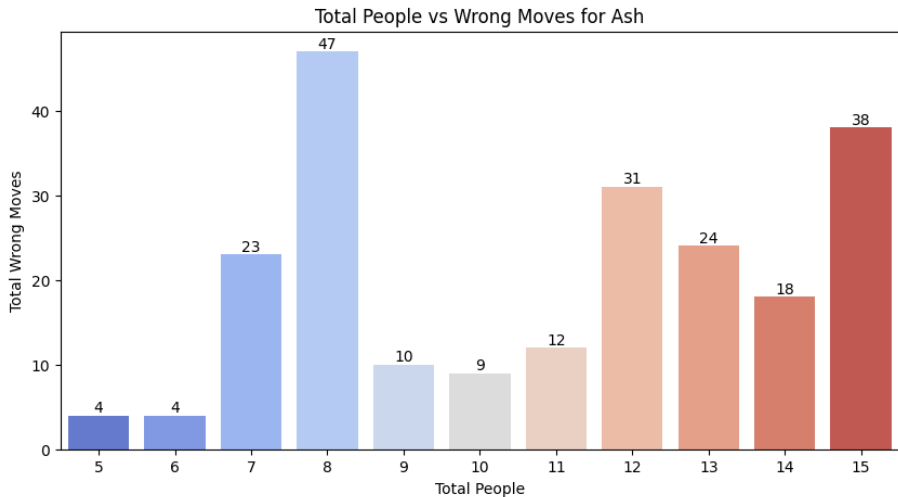
- Achieved high classification accuracy of 94% using Random Forest Regression.
- Strong precision and recall in predicting wrong moves, especially with higher support counts.
- Variability observed due to inherent complexity and randomness of the problem.
- Occasional misclassifications due to imbalanced data, especially with fewer wrong moves.
- Offers valuable insights into learner behavior and identifies error-prone areas effectively.

Josephus Problem - Metrics



Total People vs Wrong Moves For Max In Josephus Problem

Josephus Problem - Metrics



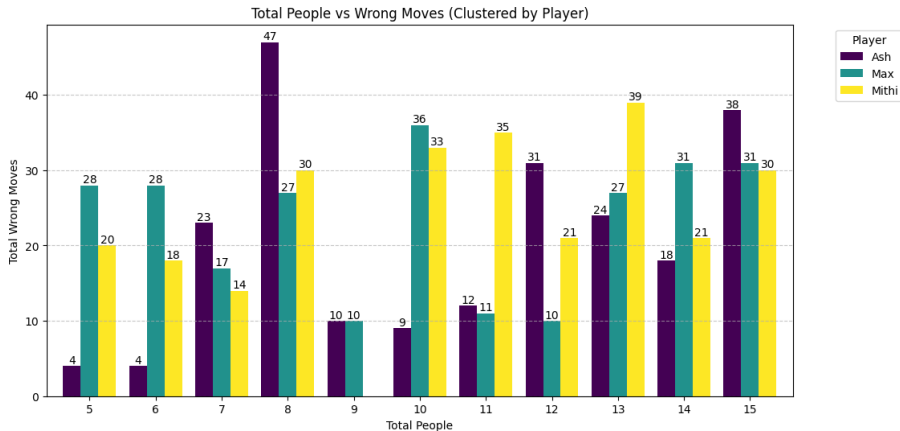
Total People vs Wrong Moves For Ash In Josephus Problem

Josephus Problem - Metrics



Total People vs Wrong Moves For Mithi In Josephus Problem

Josephus Problem - Metrics



Total People vs Wrong Moves For Every Player(Clustered) In Josephus Problem

Josephus Problem - Performance Metrics

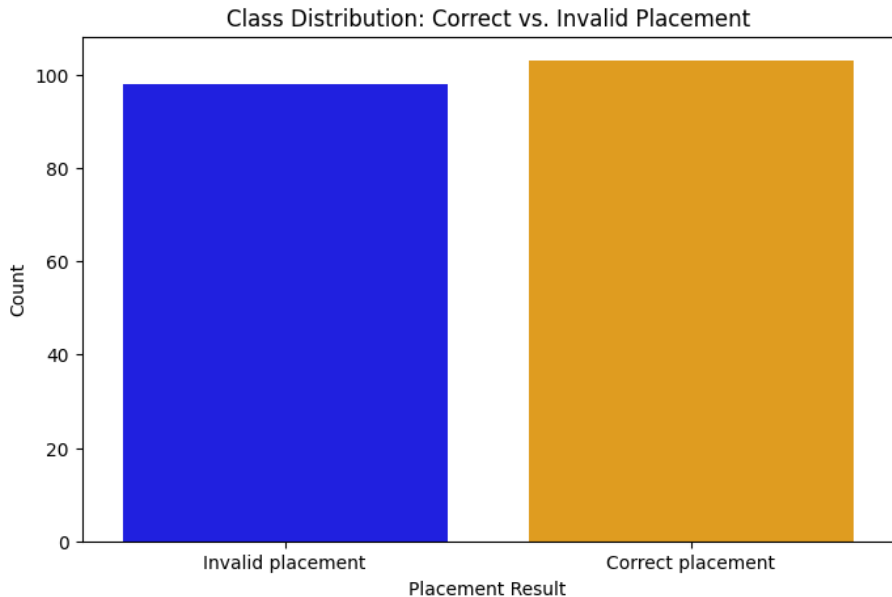
Metric	Precision	Recall	F1-Score	Support
Accuracy	-	-	0.94	31
Macro Avg	0.73	0.77	0.75	31
Weighted Avg	0.91	0.94	0.92	31

Table: Classification Report Summary for Josephus Problem using Random Forest Regression

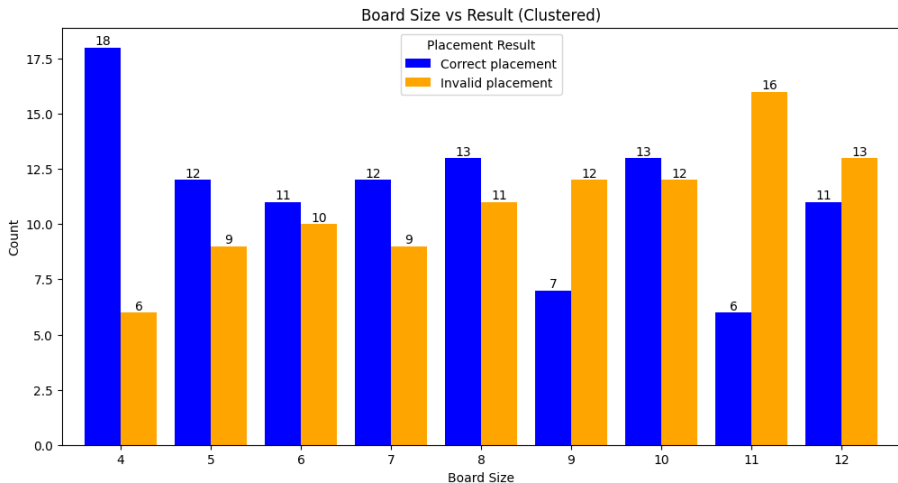
N-Queens Problem Algorithm

- Achieved moderate classification accuracy of 56% using Decision Tree Classifier.
- Effectively distinguishes between "Correct placement" and "Invalid placement" scenarios to some extent.
- Classification report shows balanced yet slightly lower precision and recall, especially for invalid placements due to data imbalance.
- Demonstrates reliable predictions for smaller board sizes, with variability increasing as board complexity rises.
- Player-specific analysis indicates predicted outcomes mostly align with actual placements, offering valuable feedback.
- Potential for improvement exists to enhance model consistency for larger boards.

N-Queens Problem - Metrics

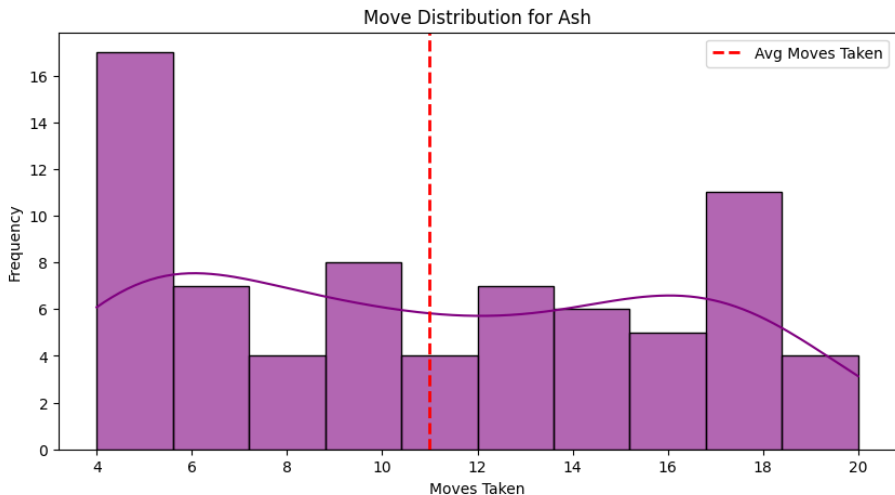


N-Queens Problem - Metrics



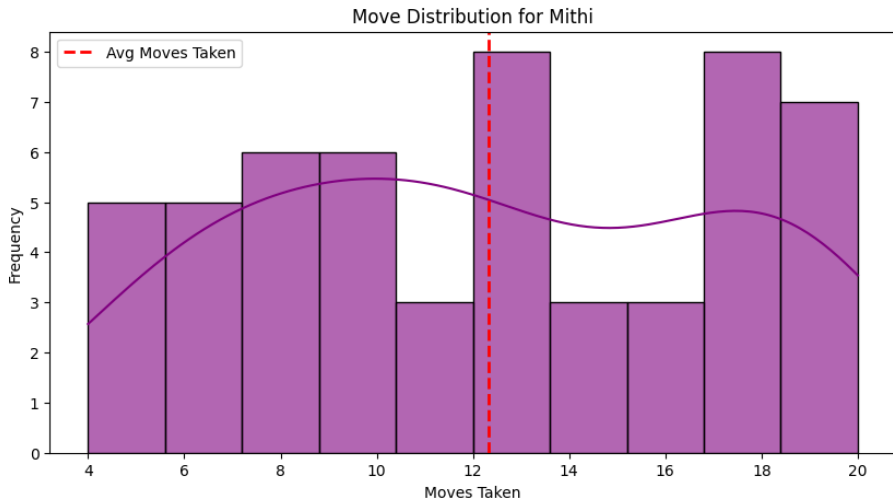
Board Size vs Result In N-Queens

N-Queens Problem - Metrics



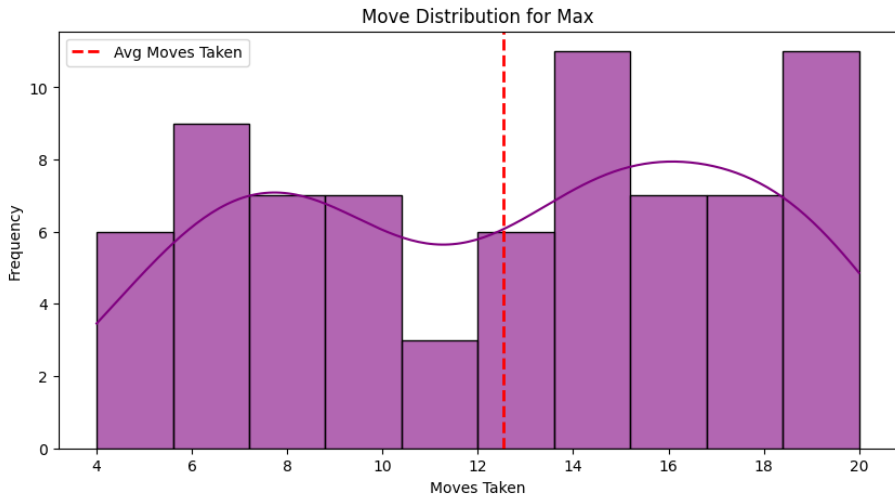
Move Distribution For Ash In N-Queens

N-Queens Problem - Metrics



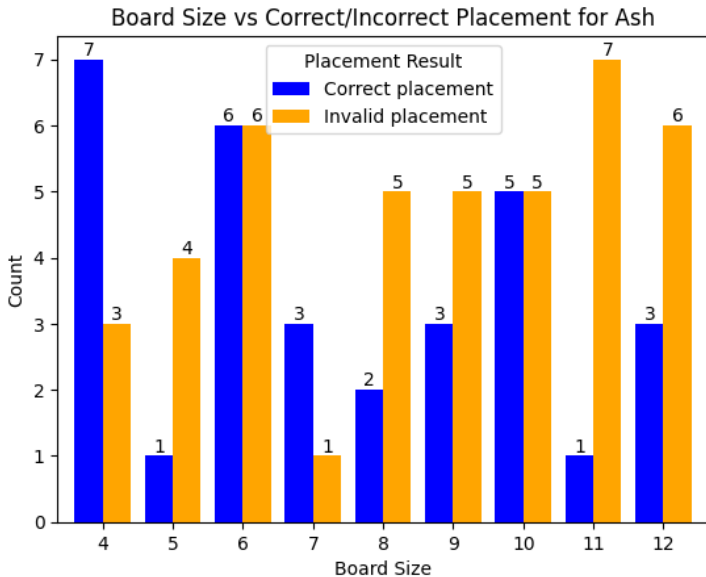
Move Distribution For Mithi In N-Queens

N-Queens Problem - Metrics

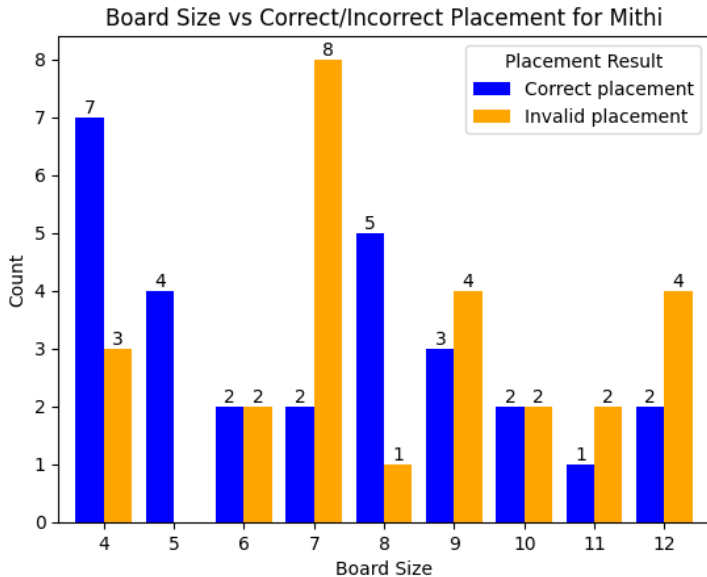


Move Distribution For Max In N-Queens

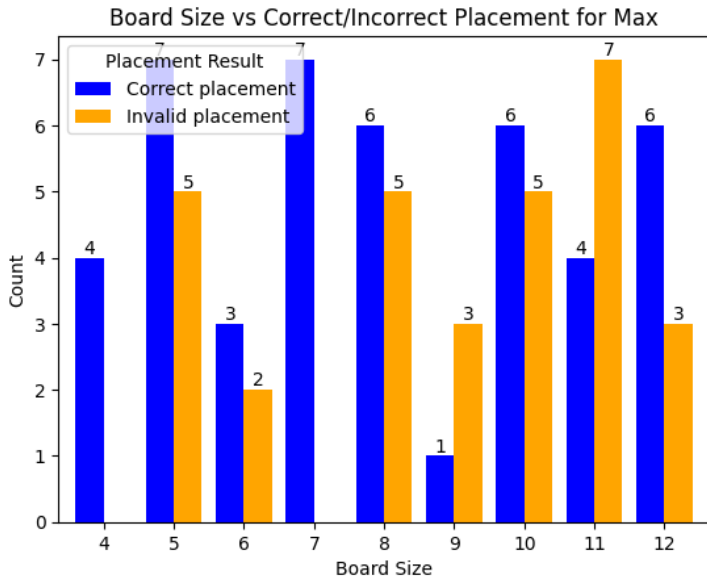
N-Queens Problem - Metrics



N-Queens Problem - Metrics



N-Queens Problem - Metrics



N-Queens Problem - Classification Report

Class	Precision	Recall	F1-Score	Support
0 (Correct Placement)	0.70	0.59	0.64	27
1 (Invalid Placement)	0.39	0.50	0.44	14
Accuracy	-	-	0.56	41
Macro Avg	0.54	0.55	0.54	41
Weighted Avg	0.59	0.56	0.57	41

Table: Classification Report for N-Queens Problem using Decision Tree Classifier

Impacts Pertaining to Society, Health, Safety, Legal, Environment, and Culture

- **Societal Impact:** Democratizes education by making interactive learning accessible to students from diverse backgrounds with minimal hardware requirements.
- **Health Impact:** Reduces cognitive fatigue through engaging, multisensory learning. Personalized feedback lowers stress and enhances retention.
- **Safety & Legal Impact:** Ensures data security, privacy, and compliance with intellectual property laws, fostering a safe digital learning ecosystem.
- **Environmental Impact:** Promotes sustainability by reducing reliance on paper-based materials and minimizing e-waste through low-energy requirements.
- **Cultural Impact:** Supports multilingual and customizable content, ensuring inclusivity for students from diverse linguistic and cultural backgrounds.

Plan for submitting Project Work

Month	Task
April	Paper writing
April end and May	Peer Review
May end	Submit paper to journals, conferences, etc.

Table: Task Submission

References I

-  Genady Kogan, Hadas Chassidim & Irina Rabaev.,2024. *The efficacy of animation and visualization in teaching data structures: a case study.*
-  LADISLAV VÉGH & Veronika Stoffová.,2017. *Algorithm Animations for Teaching and Learning the Main Ideas of Basic Sortings.*
-  Henry W. Robbins, Samuel C. Gutekunst, David B. Shmoys & David P. Williamson.,2023. *GILP: An Interactive Tool for Visualizing the Simplex Algorithm.*
-  Sarthak Goel, Vanshika Varshney, Shubham Dikshant, Akhil Sharma & Sherish Johri.,2023. *A Review of The Algorithm Visualization Field.*
-  Maximiliano Paredes-Velasco, J. Ángel Velázquez-Iturbide & Mónica Gómez-Ríos.,2022. *Augmented reality with algorithm animation and their effect on students' emotions.*

References II



Ken Perlin, Zhenyi He & Karl Rosenberg.,2018. *Chalktalk : A Visualization and Communication Language – As a Tool in the Domain of Computer Science Education.*



Andhika Kusheryanto, Anis Mirza, Ari Syaripudin & Deanna Durbin Hutagalung.,2024. *Development of the Story of Life: A Narrative and Educational Game Using the Godot Engine for Android.*



Mahesh Ranaweera Qusay & H. Mahmoud.,2024. *Deep Reinforcement Learning with Godot Game Engine.*







Martin Krajčovič, Gabriela Gabajová, Beáta Furmannová, Vladimír Vavřík, Martin GašoORCID & Marián Matys .,2021. *A Case Study of Educational Games in Virtual Reality as a Teaching Method of Lean Management.*





References III

-  Ariel Manzur & George Marques.,2018. *Godot Engine Game Development in 24 Hours, Sams Teach Yourself: The Official Guide to Godot 3.0.*
-  Maithili Dhule.,2022. *Beginning Game Development with Godot.*
-  John T. Stasko.,1997. *Using student-built algorithm animations as learning aids.*
-  Steven Hansen,N.Hari Narayanan & Mary Hegarty.,2002.*Designing Educationally Effective Algorithm Visualizations.*
-  Steven Hansen & N.Hari Narayanan .,2000. *On the Role of Animated Analogies in Algorithm Visualizations.*
-  Andreas Kerren.,2012. *Visualizations and Animations in Learning Systems.*

References IV

-  Bandura, A., and Sweller, J., 2020. *The Impact of Game-Based Learning on Student Motivation and Engagement*.
-  Carneiro, D., and Carvalho, M., 2022. *Algorithms Through Games*. Teaching Data Structures and Lecture Notes in Networks and Systems: Methodologies and Intelligent Systems for Technology Enhanced Learning, 12th International Conference, pp.3-12.
-  Cakiroglu, Y., and Ergun, M., 2016. *The Role of Interactive Animations in Enhancing Problem-Solving Skills*.
-  Jeetha, K.P.K., 2021. *Animation for Learning: Enhancement of Learning through Animation: A Review of Literature*. International Research Journal of Modernization in Engineering Technology and Science, Volume: 03/Issue: 01, January.

References V

-  Shute, V.J., and Ke, F., 2012. *Games, Learning, and Assessment: Progressive Design of Games to Facilitate Learning and Assessment*. In *Assessment in Game-Based Learning*, Springer, pp.43-58.
-  Su, S., Zhang, E., Denny, P., and Giacaman, N., 2021. *A Game-Based Approach for Teaching Algorithms and Data Structures Using Visualizations*. SIGCSE '21: Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, pp.1128-1134.
-  Wilson, A., and Martin, L., 2019. *Applications of Interactive Media in Education: Study on Adaptive Learning*.
-  Wu, B., and Richards, B., 2021. *An Overview of Gamification in E-Learning and Its Effects on Motivation and Learning*. *Computers in Human Behavior Reports*, 3, p.100086.

Thank You