

# Muğla Sıtkı Koçman University

Computer Engineering Department

CENG3511 : Artificial Intelligence

## Final Project Report

---

**Project Title:** Rock-Paper-Scissors AI Agent with Classification-Based Prediction

**Student:** Mandana Zooyousefin

**Student ID:** 180709711

**Advisor:** Prof.Dr. Bekir Taner DİNÇER

**GitHub Link:** <https://github.com/mandanazooyousefi/rock-paper-scissors-ai>

---

## 1. Project Description

This project focuses on the development of an intelligent agent that plays the traditional game of Rock-Paper-Scissors against a human user. The AI uses a machine learning classification model to learn the patterns of the player's past moves and makes predictions about future moves in order to win the game.

## 2. Game Design

- The user plays the game by selecting one of the three options: rock, paper, or scissors.
- The AI agent analyzes the player's most recent three moves.
- Using a Decision Tree Classifier, the AI predicts the user's likely next move.

- The agent then chooses the optimal counter move to maximize its chance of winning.
- A graphical user interface (GUI) was implemented using Tkinter for a better user experience.

### 3. AI Design

- **Method:** Supervised Machine Learning
- **Model:** DecisionTreeClassifier from scikit-learn
- **Training Data:** data.csv stores user moves over time
- **Features:** Previous 3 moves (e.g., [rock, paper, scissors])
- **Label:** Next move (target output)
- **Counter Strategy:** AI responds with the move that beats the predicted one

### 4. Development Tools

- Python 3.11
- scikit-learn, pandas, numpy
- Tkinter (Python built-in GUI)
- Visual Studio Code
- GitHub for version control

### 5. Evaluation and Results

- Initially, the AI selects random moves due to lack of training data.
- Over time, as more player data is collected, the AI improves its prediction accuracy.
- During tests, the AI achieved a higher win rate against repetitive or predictable player behavior.
- GUI version improves usability and allows live testing and score tracking.

## 6. Challenges and Improvements

- Challenge: Generating sufficient data for the model to learn meaningful patterns.
- Potential Improvement:
  - Use more complex models like Random Forest or Neural Networks
  - Include additional features such as player reaction time or game history depth
  - Save and reload trained models between sessions

## 7. Conclusion

This project demonstrates how even simple machine learning techniques can create an adaptive, interactive AI system in a game environment. The AI agent effectively learns from the user's behavior and makes strategic decisions, showing the practical power of classification models in real-time applications.

## 8. Files in the Project

- **main.py:** Terminal-based version of the Rock-Paper-Scissors game with data logging only.
- **gui.py:** Graphical user interface (GUI) version using Tkinter. Allows live gameplay.
- **ai\_model.py:** Contains machine learning logic: dataset preparation, model training, prediction.
- **data.csv:** Stores player move history used to train the AI model over time.
- **README.md:** Contains project overview, instructions, and technologies used.

- **AI\_Final\_Report.PDF:** Final project report summarizing objectives and results.