AI MSE REPORT

Problem statement : Simple Game AI for Rock-Paper-Scissors

NAME : RAHUL PRAJAPATI

ROLL NO : 202401100400150

BRANCH : CSE (AIML)

INTRODUCTION PAGE

- The task is to develop a simple Artificial Intelligence (AI) system capable of playing the classic game **Rock-Paper-Scissors** against a human player.

- The AI must be able to learn from the user's previous moves and make predictions to counter them effectively.

- The primary objective is to create an AI model that can adapt to user inputs and improve its prediction accuracy over time.

- The program should be user-friendly, allowing the player to input moves quickly and providing an option to exit the game at any time.

METHODOLOGY

The method used in this code is a Frequency-Based Prediction Algorithm.

How it works:

- The AI tracks all user moves and counts the frequency of each move: Rock, Paper, and Scissors.

- The AI predicts the user's next move by finding the most commonly played move so far.

- It then chooses the counter-move to that most frequent move:
- If the most frequent move is 'Rock', the AI chooses 'Paper'

Why this method is used:

- It’s a simple yet effective approach for beginners to implement a learning mechanism.

- It is easy to code and provides decent performance against humans who have predictable patterns.

CODE:

```
import random
from collections import defaultdict


class RockPaperScissorsAI:
    def __init__(self):
        # Keep track of the user's moves and how often they pick each one
        self.user_moves = []
        self.move_counts = defaultdict(int)
        self.choices = {'r': 'rock', 'p': 'paper', 's': 'scissors'}

    def update_history(self, user_move):
        # Remember the user's move and update the count for it
        self.user_moves.append(user_move)
        self.move_counts[user_move] += 1

    def predict_move(self):
        # If it's the first round, pick randomly
        if not self.user_moves:
            return random.choice(list(self.choices.values()))
```

```
        # Find out which move the user picks most often
        most_common_move = max(self.move_counts,
key=self.move_counts.get)

        # Choose the move that beats the most common move
        if most_common_move == 'rock':
            return 'paper'
        elif most_common_move == 'paper':
            return 'scissors'
        else:
            return 'rock'

    def play(self, user_input):
        # Translate initial letter input to full move name
        if user_input in self.choices:
            user_move = self.choices[user_input]
        else:
            return "Oops! Please choose 'r' for rock, 'p' for paper, or 's' for
scissors."

        # AI predicts the move, then remembers what the user picked
        ai_move = self.predict_move()
        self.update_history(user_move)
```

```
        # Figure out who won
        result = self.determine_winner(user_move, ai_move)

        # Return the results as a dictionary
        return {
            'user_move': user_move,
            'ai_move': ai_move,
            'result': result
        }

    def determine_winner(self, user_move, ai_move):
        # Determine the winner according to game rules
        if user_move == ai_move:
            return "It's a tie!"
        elif (user_move == 'rock' and ai_move == 'scissors') or \
             (user_move == 'scissors' and ai_move == 'paper') or \
             (user_move == 'paper' and ai_move == 'rock'):
            return 'You win!'
        else:
            return 'AI wins!'

def main():
```

```
    ai = RockPaperScissorsAI()
    print("Welcome to Rock-Paper-Scissors! Type 'exit' to quit.")

    while True:
        user_input = input("Your move (r for rock, p for paper, s for scissors, or 'exit' to quit): ").lower()
        if user_input == 'exit':
            print("Thanks for playing! Goodbye.")
            break

        # Get the result of the round
        result = ai.play(user_input)

        # Print the results in a friendly way
        if isinstance(result, str):
            print(result)  # Display error message if invalid move
        else:
            print(f"You chose: {result['user_move']}")
            print(f"AI chose: {result['ai_move']}")
            print(f"Result: {result['result']}")

if __name__ == "__main__":
    main()
```

OUTPUTS:

```
Welcome to Rock-Paper-Scissors! Type 'exit' to quit.
Your move (r for rock, p for paper, s for scissors, or 'exit' to quit): r
You chose: rock
AI chose: paper
Result: AI wins!
Your move (r for rock, p for paper, s for scissors, or 'exit' to quit): p
You chose: paper
AI chose: paper
Result: It's a tie!
Your move (r for rock, p for paper, s for scissors, or 'exit' to quit): s
You chose: scissors
AI chose: paper
Result: You win!
Your move (r for rock, p for paper, s for scissors, or 'exit' to quit): exit
Thanks for playing! Goodbye.
```

REFERENCES:

- Concepts and inspiration derived from basic AI principles in decision-making systems.
- Python documentation: [Python Official Docs](https://docs.python.org/3/)
- Tutorials and guides on building AI systems for games from various online resources.