

INTRODUCTION TO AI



PROJECT REPORT

PROBLEM STATEMENT: - AI-Based Number Guessing Game

NAME: - JAYANT – SINGH

BRANCH: - CSE(AIML)

SECTION: - B

ROLL NO.: - 202401100400102

INTRODUCTION

AI-Based Number Guessing Game – Problem Statement

Objective: Create a Python-based game where an AI guesses a number that the user secretly thinks of, using logical strategies to minimize attempts.

How It Works:

1. The **user picks a number** within a given range (e.g., 1 to 100).
2. The **AI makes a guess** and asks for feedback:
 - a. **Too high** → AI adjusts to a lower number.
 - b. **Too low** → AI adjusts to a higher number.
 - c. **Correct** → AI wins, and the game ends.
3. The AI continues guessing until it finds the correct number.

AI Strategy:

- **Binary Search:** AI halves the range after each guess, reducing attempts significantly.
- **Future Enhancements:** The game can later include machine learning for better predictions.

Goal: Make the AI guess the number efficiently while keeping the game fun and interactive.

METHODOLOGY

The code works in the following manner: -

1. **Random Target:** The computer generates a random integer within a specified range.
2. **User Guess:** The player inputs a numerical guess.
3. **Comparison & Feedback:**
 - a. If the guess is too low, the computer indicates "Higher".
 - b. If the guess is too high, the computer indicates "Lower".
 - c. If the guess is correct, the game ends.
4. **Iteration:** Steps 2 and 3 repeat until the correct guess is made.
5. **Guess Count:** The number of guesses is tracked and displayed upon a correct guess.
6. **Input Validation:** The program handles non-numerical input to prevent errors.

CODE

```
def human_guess_number(max_range=100):
    """
    A number guessing game where the human guesses the AI's number.

    Args:
        max_range: The upper limit of the range (inclusive).

    Returns:
        None. Prints the number of guesses and the final guess.
    """

    secret_number = random.randint(1, max_range)
    guess_count = 0

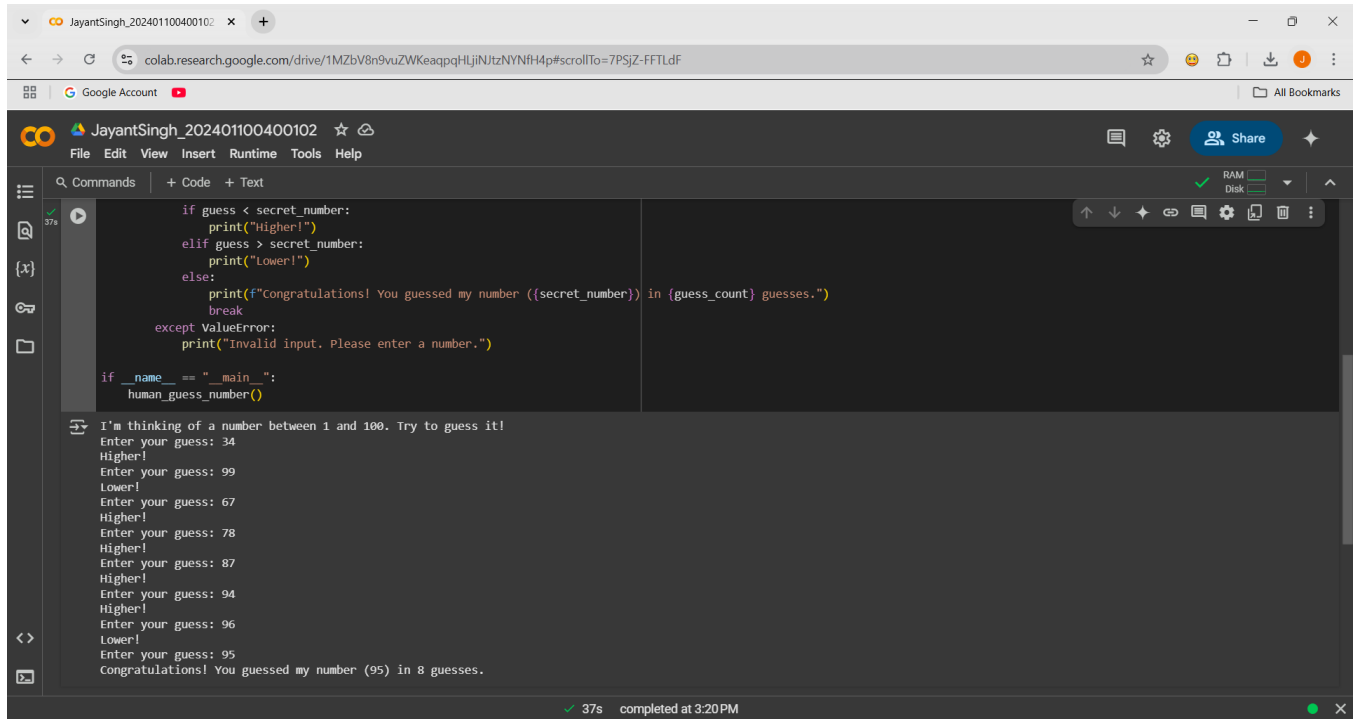
    print(f"I'm thinking of a number between 1 and {max_range}. Try to guess it!")

    while True:
        try:
            guess = int(input("Enter your guess: "))
            guess_count += 1

            if guess < secret_number:
                print("Higher!")
            elif guess > secret_number:
                print("Lower!")
            else:
                print(f"Congratulations! You guessed my number ({secret_number}) in {guess_count} guesses.")
                break
        except ValueError:
            print("Invalid input. Please enter a number.")

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

OUTPUT



The screenshot displays a Google Colab notebook interface. The browser address bar shows the URL: `colab.research.google.com/drive/1MZbV8n9vuZWKeaqpqHljiN/tzNYNfh4p#scrollTo=7PSjZ-FFTLdF`. The notebook is titled "JayantSingh_202401100400102". The code editor contains a Python script for a number guessing game. The output area shows the game's execution, including prompts for guesses and feedback messages.

```
if guess < secret_number:
    print("Higher!")
elif guess > secret_number:
    print("Lower!")
else:
    print(f"Congratulations! You guessed my number ({secret_number}) in {guess_count} guesses.")
    break
except ValueError:
    print("Invalid input. Please enter a number.")

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

I'm thinking of a number between 1 and 100. Try to guess it!

Enter your guess: 34
Higher!

Enter your guess: 99
Lower!

Enter your guess: 67
Higher!

Enter your guess: 78
Higher!

Enter your guess: 87
Higher!

Enter your guess: 94
Higher!

Enter your guess: 96
Lower!

Enter your guess: 95
Congratulations! You guessed my number (95) in 8 guesses.

37s completed at 3:20 PM