Sudoku Solver & Email Spam Detection

Prepared by

Saz Nawzad F190125
Sara Muayad F190246

Artificial Intelligence
summer 2024

August 2/ 2024

Introduction:

This project explains the process of both tasks which are sudoku puzzle and email spam detector. With an objective to demonstrate how we have accomplished our tasks.

Objective:

1. Sudoku solver:

   The objective in this task is to create the sudoku puzzle in a simple way and show a before solving of the puzzle and after solving of the puzzle next to it, making it easy to see the empty cells and what numbers have been put inside them.

2. Email spam Detection:

   The objective is to create a model that can show if a message is spam or ham(non spam), and based on the content of the text it can detect or predict if it is spam or non-spam.

Methods and Tools used:

1. Sudoku solver problem definition:
    - we have used numpy to create a numpy array for the board of the game
    - Tkinter to show before and after solving the sudoku puzzle. giving it a nice interface to see the comparison.
    - Functions to check empty cells, checking the validity of a number for the cell, and solving the board.

2. Email spam detection:
    - Preprocessing the data: The text is cleaned by taking off some unnecessary (/r, /n) lines in the dataset.
    - Splitting the data: the data is trained and tested by splitting them using train_test_split .
    - Feature extraction: converting the texts into matrix of token counts using CounteVectorizer to change it into numerical texts instead.
    - Training the model: in this case Multinomial is used in naive bayes.
    - Evaluating the model: in this case both confusion_matrix and classification_report is used.

Tools used:

- Libraries used for sudoku solver: ‘Numpy’, ‘tkinter’
- Libraries used for Spam email detection: ‘Numpy’, ‘Pandas’, ‘Matplotlib’, ‘sklearn’

Techniques used:

Sudoku puzzle:

Backtracking algorithm, validity check, solving board, finding empty cells, displaying board, drawing lines, Tkinter canvas display.

Email spam detection:

Confusion matrix, confusion report, Multinomial naive bayes, ConterVectorizer, Cleaning text, train_test_split.

Challenges :

1. In task one we had an issue with the way the result looked for the sudoku puzzle, therefore we decided to use canvas to show the result in a more simple and understanding way rather than show it on the terminal. Making it easier to compare the before and after with a nice interface by the tkinter.

2. The issues in this task that has been faced is visualization, which was intended to use splitting plot or heat map , therefore it did not work and we decided to use Pi chart to visualize the spam and non-spam to determine the spam messages.

Conclusion:

We have accomplished both task's goals. The sudoku puzzle being solved and displayed in a way to see before and after solving, giving a better understanding and comparison with a best accuracy of solving, while the email spamming model has achieved a position or a good high amount of accuracy to detect if a text is spam or not, as we used classification report and matrix to show the performance, therefore based on the text content the model detected the spam ones and visualized the percentage by using pi chart.