DeadLock Detection Using "Machine Learning"

Federal Institute Of Science Technology

by
Sabir Ebrahim
Richard TS
Sabarinath R (CSE -B)
on August 24,2020

» Introduction

- Process Scheduling is an OS task that schedules processes of different states like ready, waiting, and running. Process scheduling allows OS to allocate a time interval of CPU execution for each process
- Deadlock will prevent the proper process scheduling in an OS
- Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.

» Problem Statement

* To check weather a system is in safe state or not using Machine learning to avoid deadlock

» Design Methodolgy

- Created **Dataset** using Output of a python code executed using **Bankers Algorithm**
- Implemented **logistic regression** Machine Learning method to detect deadlock conditon
- **Logistic regression** is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis.

» Architecture

- Designing Algorithm for deadlock detection
- Implementing python code for deadlock detection using random resources
- saving the input and corresponding output into two files
- database created=> importing the database
- Training the machine using logistic regression
- Giving new data as input
- Predicting the output of a new data(resources) to detect deadline

» Algorithm 1(Deadlock Detection Algorithm)

1.Declare 1D Array *avail*

2.Declare 2D Arrary *maxm* and *allot*

3.Declare 2D Array need

4.need[i][j]=maxm[i][j]-allot[i][j]

5.work and finish be vectors of length 'm' and 'n' respectively.
Initialize: work = *avail* finish[i] = false; for i=1, 2, 3, 4....n

6.Find *a*n i such th*a*t both *a*) Finish[i] = f*a*lse b) need i <= work if no such i exists *g*oto step (8)

7.work = work + *a*llot[i], Finish[i] = true *g*oto step (6)

8.if finish [i] = true for *a*ll i then return 1,else return 0

» Implementation

* First we declared required data structures in the algorithm
* Then we inserted the resources required using random variables
* We implemented the bankers algorithm inside a function to check whether deadlock occurs or not
* All the input resources and their corresponding outputs were saved into files to create a dataset
* We implemented the program for a time period of 5 min

» Algorithm 2

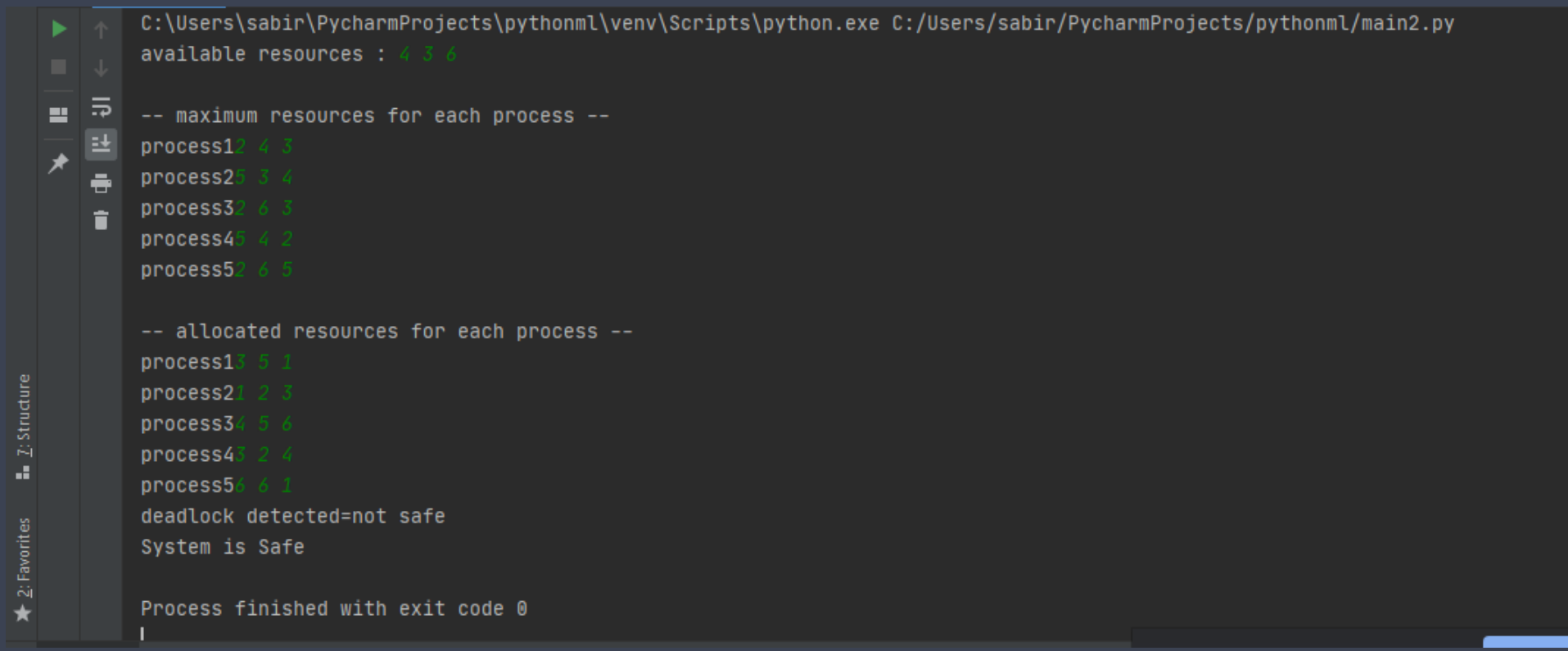
1.Input available and maximum and alloted resources for each process

2.Train the machine using logistic regression by using the created dataset

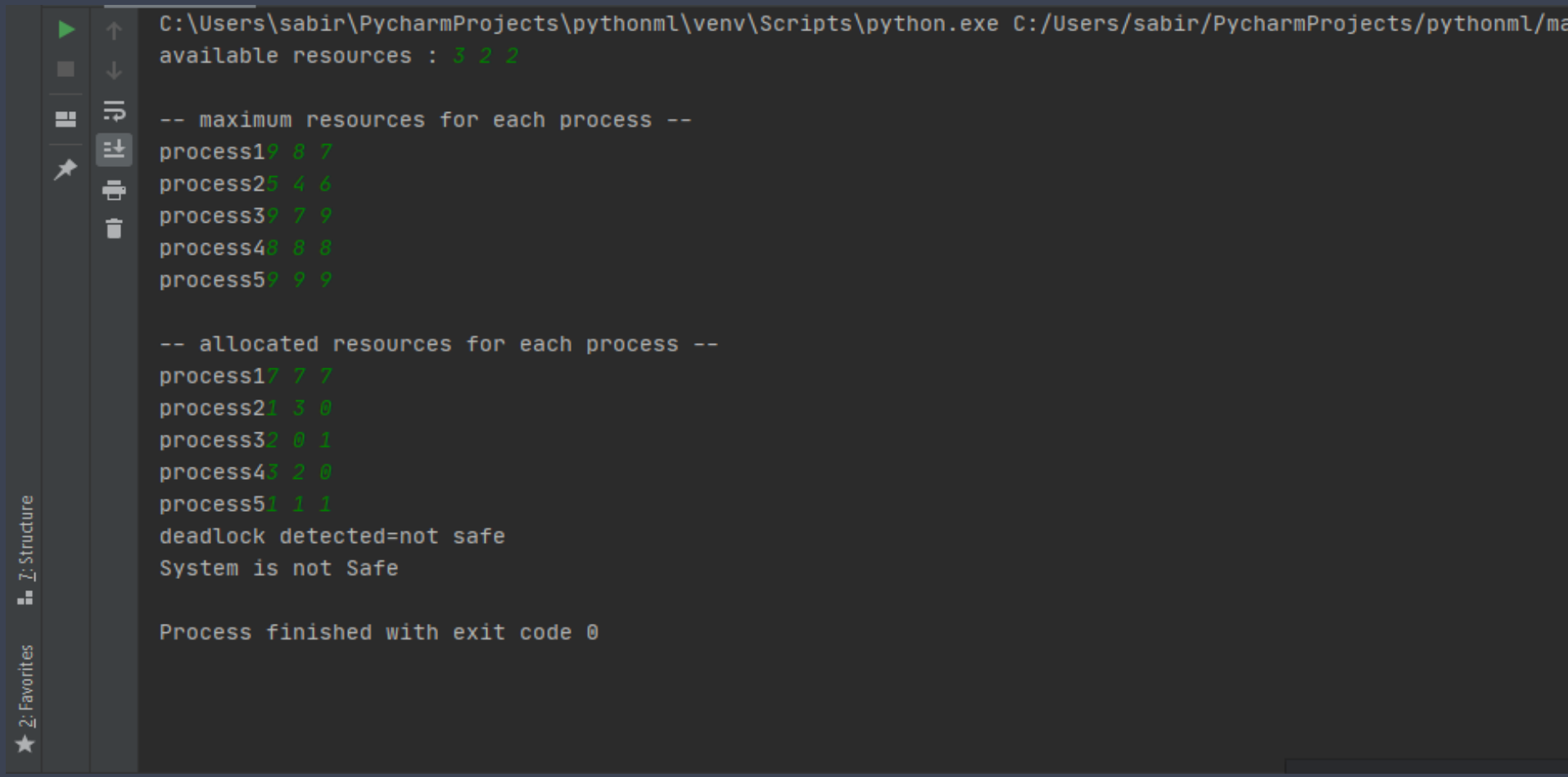
3.Test the output of the given resources by prediction

4.If prediction result=0 then print 'system is not safe',else print'system is safe'

» Output1


C:\Users\sabir\PycharmProjects\pythonml\venv\Scripts\python.exe C:/Users/sabir/PycharmProjects/pythonml/main2.py
available resources : 4 3 6
-- maximum resources for each process --
process1 2 4 3
process2 5 3 4
process3 2 6 3
process4 5 4 2
process5 2 6 5
-- allocated resources for each process --
process1 3 5 1
process2 1 2 3
process3 4 5 6
process4 3 2 4
process5 6 6 1
deadlock detected=not safe
System is Safe
Process finished with exit code 0
7: Structure
2: Favorites

» Output2


C:\Users\sabir\PycharmProjects\pythonml\venv\Scripts\python.exe C:/Users/sabir/PycharmProjects/pythonml/ma
available resources : 3 2 2
-- maximum resources for each process --
process1 9 8 7
process2 5 4 6
process3 9 7 9
process4 8 8 8
process5 9 9 9
-- allocated resources for each process --
process1 7 7 7
process2 1 3 0
process3 2 0 1
process4 3 2 0
process5 1 1 1
deadlock detected=not safe
System is not Safe
Process finished with exit code 0
2: Favorites
7: Structure

» Conclusion

The Machine was able to predict the occurrence of deadlock

References

- Introduction to Machine Learning , Dr Marco Gillies, University of London
- GeeksForGeeks.org
- www.datacamp.com
- wikipedia
- https://machinelearningmastery.com/logistic-regression-for-machine-learning/
- youtube.com
- stackoverflow.com