|  |  |  |  |
| --- | --- | --- | --- |
| **EDEI** | **Practice** | | |
| **DEPARTMENT** | Computing, Electronics and Mechatronics | **COURSE** | LRT-3082 Artificial Intelligence |
| **PROFESOR** | Dra. Alejandra Hernandez Sanchez | **TERM** | Spring 2024 |

***Theoretical part***

1. *Mention two practical applications of AI in everyday life.*
2. *What is the difference between supervised and unsupervised learning?*
3. *Define the concept of "overfitting" in the context of AI.*
4. *What are some of the rules and approaches that organizations such as Google and the European Union have defined for the development and use of AI?*
5. *Explain the importance of normalization in databases.*
6. *What is an imbalanced dataset and how can it affect the performance of a model?*
7. *Explain the concept of "dimensionality" in a dataset. (Open question)*
8. *What does the precision metric measure in a classification model?*
9. *What is the difference between uninformed search and informed search?*
10. *Name a practical problem where informed search can be applied.*

***Practical part***

***Excercise 1 Datasets***

* *Identify if the dataset is a Good dataset or need a technique of preprocessing. Justify the answer*
* *Process all the document of :*
  + *Iris*
  + *Iris1*
  + *Iris2*
  + *Iris3*

***Excercise 2 Metrics***

*Metrics obtained Example 1Credit Approval*

* ***A****ccuracy: 0.88*
* *Precision: 0.82*
* *Recall: 0.75*
* *F1-Score: 0.78*
* *Confusion Matrix:*
* *TP[[170 20]FN*
* ***FP[ 15 105]]TN***

*Example 2 Credit Approval*

* *Accuracy: 0.92*
* *Precision: 0.91*
* *Recall: 0.93*
* *F1-Score: 0.92*
* *Confusion Matrix:*
* *[[145 5]*
* *[ 7 143]]*

***Excercise 3 Searchs***

Excercise 1

A directed graph with nodes and weighted edges is given.

A B 1

A C 4

B D 2

C D 1

D E 3

Each node represents a state and each edge represents an action with an associated cost. The goal is to find the optimal path from a start node to a goal node using the uninformed search algorithms.

Implements the following uninformed search algorithms:

Breadth-First Search (BFS): Must explore nodes level by level.

Depth-First Search (DFS): Must explore nodes in depth, backtracking when necessary.

Uniform Cost Search (UCS): Must explore nodes in order of accumulated cost from the start node.

Code Requirements:

The code must be written in Python.

Must include comments explaining each part of the code.

Must correctly handle reading the input file and representing the graph.

Must print the path found and the total cost of the path for each algorithm.

Excercise 2

A graph representing a map of cities is provided. Each node in the graph is a city, and the edges represent connections between cities with an associated cost (distance in kilometers). The goal is to find the shortest route from a start city to a target city using the A\* algorithm.

In addition, a table with the straight-line distances (heuristic) from each city to the target city is provided. This heuristic will be used by the A\* algorithm to guide the search.

Implementation of Algorithm A:\*

Implement the A\* algorithm using the provided heuristic.

The algorithm must:

Compute the total cost f(n)=g(n)+h(n), where: g(n) is the cumulative cost from the start city to the current city.

h(n) is the heuristic (straight-line distance) from the current city to the target city.

Explore the cities in order of lowest f(n).

Return the optimal route and total cost.

Code Requirements:

The code must be written in Python.

It must include comments explaining each part of the code.

It must read the input files (graph and heuristics) and build the necessary data structures.

It must print the path found and the total cost.