Exam: Artificial Intelligence – Algorithms and Application

Module Exam

Summer 2023   Date: 01.09.2023

Important Information

TECHNISCHE UNIVERSITÄT DARMSTADT

WIRTSCHAFTS INFORMATIK

- Please check your exam copy for completeness.
  It covers **19 pages** (cover sheet included).
- Fill out the cover sheet immediately after receiving the exam.
- Use only the examination paper to solve the tasks. If you do not have enough space, you can receive additional paper during the examination. Additional papers must also be marked with your name and matriculation number.
- Please leave a **correction margin of 3 cm**.
- You have a total of **90 minutes** to complete the exam.
- Except for a **non-programmable calculator, no other aids** are allowed in the exam.

We wish you much success!

Please fill out clearly in block letters.

First Name ....   Last Name ....   Seat No. ....

❑ Master
Matr. No. ....   Course of Study ....   ❑ Diplom

Repeater:
❑ yes   ❑ no

Section	Max. Points	Achieved Points
1	36	
2	24	
3	30	
Sum	90	

Exam Review („Klausureinsicht"):

(do not fill out before the review)

I have reviewed the corrected exam:

☐   There are no complaints about the correction.

☐   Complaints about the correction exist (see additional sheet).

Date: …………………………………..

Signature: …………………………......

1   Basic Concepts and Algorithms (36 Points)

1.1   The researcher **McCarthy** played a **crucial role** for the **field of artificial intelligence**. **Why**? (1 P)
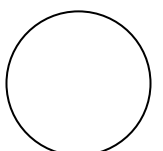
1.2   Please briefly **explain** the **concept** of an **agent** in **artificial intelligence** based on the **definition** of **Russell & Norvig**. Please **draw** the **architecture** of a "**reflex agent**" and briefly **explain** it by **comparing** it to the **general agent model**. (6 P)
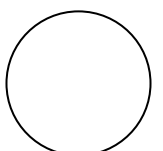
1.3   **Please** briefly **explain** what kind of problem the *traveling salesman problem* represents using the **national park example** from the lecture. (2 P)

1.4   Please provide an example of a "*CAPTCHA*". Please also **explain how** a "*CAPTCHA"* is **related** to **Artificial Intelligence** and **name three ability domains** of **Artificial Intelligence** that are **needed** to **solve it.** (3 P)
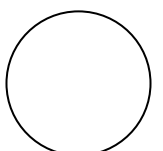
1.5   Why are **biases** in AI-based systems **not** just a **simple machine learning problem**? Please **explain** using an **example** that we did <u>**not**</u> **discuss** in the **lecture**. (3 P)

1.6   **Yandex's Alice chatbot** was taken **offline** after a short time and is considered a **"failed"** AI project. Please **explain why** this was the case. (1 P)

1.7   Please **explain**: What **role** did the **research** of **Professor Fei-Fei Li** from the Stanford University **play** in **neural networks**? **What in particular** has **she** become **known for**? (1 P)
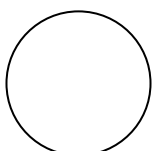
1.8   Please **explain** in your own words the **benefit** of the **bias** in **perceptron modeling**. (1 P)

1.9   Please **illustrate** the **difference** between **feedback** and **feedforward ANNs** with a **sketch**.

How do the **connections among neurons** in the <u>**same**</u> **layers** look like in each case? (2 P)
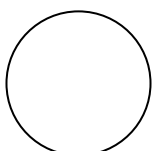
1.10 One of your friends is building a machine learning model using a **logistic regression**. He wants to predict if a car has a specific problem (classification). The data has five different problem types. However, as soon as he runs his prediction, he only gets error messages. Please briefly **explain**: What is the **most probable reason** for those **errors**? (2 P)

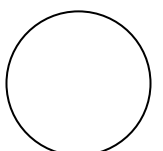**1.11** Please **define**: **What is** a **model** in machine learning? (2 P)

1.12 Please **explain**: **What** is the **difference** between the **agent's program** and **function** in our **cleaning bot example** from chapter 2? (1 P)

1.13 Please **explain**: **What** was the **overarching goal** of this year's **capstone project** and **why** was it **suitable** to be **solved** with **machine learning algorithms**? (2 P)
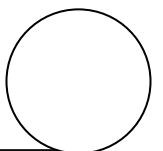
1.14 Please briefly **define it** and **explain its structure**: What is a *data frame* in Python? (3 P)

First Name........................   Last Name ....   Matr. No...............................

1.15 Please briefly **explain** the **difference** between a *bar chart* and a *histogram*. **What** are they usually **used for**? (3 P)

1.16 Please **define** the two steps *mutation* and *crossover* in the context of **genetic algorithms**. **What** is the **difference** between the **two steps**? (3 P)

2 Application of Search Algorithms (24 Points)
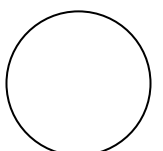
Consider the following **AI problem**:

Your local DIY-store plans to improve the shopping experience with automated agents. The agents have a language interface and a route-finding module. Customers can ask the agent where to find specific tools and materials and the agent guides them through the store. Your job is to help develop the agent. For that purpose, you received the following floor plan:

---------------- = 2 meters
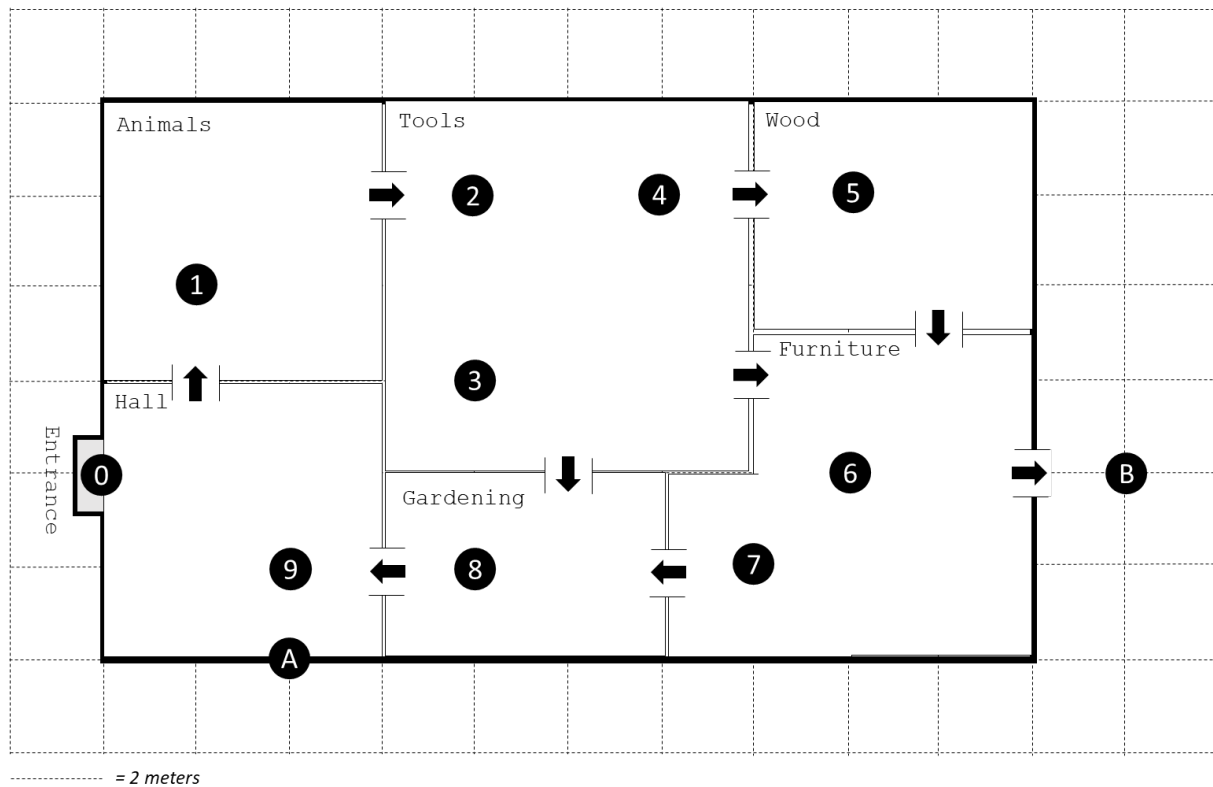
On the floor plan, you can see that each section (e.g., "Tools") has subsections that are indicated by a number, and that certain sections are connected with one another via doors. To simplify the development, you can assume that the agent travels in Manhattan distance, and that it travels along the walls. Of course, the agent can only enter a room through a door. As the store management wants to avoid collisions, the agent is only allowed to move towards increasing numbers and always only drives towards one subsection in each room it passes (e.g., to exclusively one subsection from number 2, 3 to 4 in the tools section). On the map, there are two exits, an official one near the checkout (A) and one in the outdoor area of the store (B).

2.1 Please **classify** the **agent's task environment** with the **PEAS** framework. (4 P)

2.2 To implement a first proof-of-concept agent, you **model** the **store** with **simple search trees**. For that purpose, you perform the following two tasks (a and b):

  a) Please **draw** the **subset** of a possible **search tree** of the **DYI store** to the official exit in state with a *depth-first search*. **How many nodes** do you have to **visit** until you **reach** the **exit** in this case? (4 P)

   (**Note**: You can decide with which logic your algorithm always chooses between multiple next nodes (e.g., it always chooses the most left one) when searching the tree. Please indicate your assumption.)
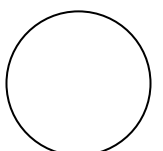
b) Please **draw** again the **subset** of a possible **search tree** of the **DYI store** until it reaches the official exit but this time you use the ***greedy best first strategy***. Please use the **Euclidian distance** to the **official exit** from each state's nearest section door as a heuristic. Please **explain** your **results**. (8 P)
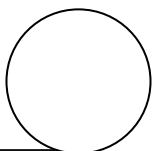
2.3 As a next step, you are asked to implement an emergency program. You decide to implement an *A* search algorithm* to find the **fastest path** to one of the **exists** (A or B). By doing so, please assume that **you** are in **state 2** in the **tools section** and the fire alarm starts. **Which path** is the **best** for *each* **exit**? Use the same **heuristic** as in task 2.2 b) for the A*-algorithm. (8 P)
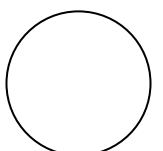
3 Programming with Python (30 Points)

One of your colleagues prepared the following Python notebook for an analysis of car failures.

Please consider the following code when answering the next questions:

```python
from pandas import read_excelll
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score

dataset = read_excel(open("car_maintenance.xlsx", "rb"))
```

Out[3]:

	PART_1023	PART_99	PART_02	OIL	CHECK_STATUS	FOLLOW-UP
0	1	2	1	0.25	acc	yes
1	2	1	1	0.15	unacc	no
2	1	2	1	0.25	acc	yes
3	1	1	1	0.15	good	yes
4	2	1	1	0.15	vgood	no
5	1	1	1	0.50	acc	yes

```python
dataset.describe()
```

Out[5]:

	PART_1023	PART_99	PART_02	OIL
count	31.000000	31.000000	31.0	31.000000
mean	1.580645	33.387097	1.0	0.180645
std	0.672022	179.210617	0.0	0.160040
min	1.000000	1.000000	1.0	0.050000
25%	1.000000	1.000000	1.0	0.100000
50%	1.000000	1.000000	1.0	0.150000
75%	2.000000	1.000000	1.0	0.200000
max	3.000000	999.000000	1.0	0.800000

Note: The notebook continues on the next page.

```
1    # remove outliers
2    # thr = dataset["PART_99"].quantile(0.999)
3    thr = 3
4    dataset = dataset[dataset["PART_99"] < thr]
5    dataset.describe()
6
7
8
9    # analytics
10   X = dataset.loc[:, "PART_1023":"CHECK_STATUS"]
11   Y = dataset.loc[:, "FOLLOW-UP"]
12   X_train, X_test, Y_train, Y_test = train_test_split(…)
13
14   clf = DecisionTreeClassifier()
15   clf = clf.fit(X_train, Y_train)
16
17   prediction = clf.predict(X_test)
18
19   confusion_matrix(Y_test, prediction)
20
21
```
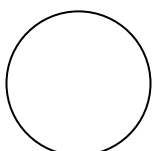
Python Code

3.1 Which machine learning model is trained in the above code? (1 P)

3.2   Please **explain what** the **"rb"** in the **open() function**, which is part of the read_excel() command, **means**. (1 P)

**3.3**   Please **explain** the **result** of the **dataset.describe() command** displayed by **OUT[5]**. (4 P)

3.4 Unfortunately, some part of the code is missing in the third code cell. Please complete the code by **adding** a **command** to **delete column `"PART_02"`** in **line 7**. (2 P)

Note: You can write the code directly above in line 7 or in the space below this task (assuming it would be written in line 7).

3.5 Please **write** the **code** for the **missing parameters** of the **`train_test_split()` function** in line 12 of the third code cell (indicated by "…") to **split** your **dataset** with a **`test_size`** of **0.3.** (2 P)
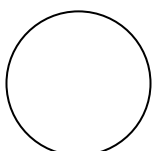
3.6 When you run the above code cells, you get an `ImportError`. Please **explain** or **provide corrected code** on how you can fix the error in the above code cells. (2 P)

3.7 Please **write code** to **implement another classification model** of your choice (e.g., from `scikit` learn). Your **code** should **include** the code to **create** an **object**, **train** it, and **use** the **predict()** function on it to **predict** the **class variable**. Please feel free to use the model you used in the lectorial. (6 P)
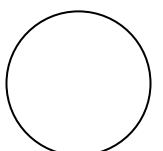
3.8 Please **explain** the **purpose** of `confusion_matrix()` in the **code**. **Why** do you **need** it and **what** does **it tell you**? (4 P)

3.9 Please **explain**: **What** could be a **possible step** to **improve** the **performance** of the above **classification model**? Please **provide** some **Python/pseudo code** to **implement** this **step**. (2 P)