Exam: Artificial Intelligence – Algorithms and Application

Module Exam

Summer 2024 Date: 06.09.2024

TECHNISCHE UNIVERSITÄT DARMSTADT

WIRTSCHAFTS INFORMATIK

Important Information

- Please check your exam copy for completeness.
  It covers **20 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. .....

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

Repeater:
❑ yes ❑ no

Section	Max. Points	Achieved Points
1	30	
2	30	
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: ……....

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

1 Basic Concepts and Algorithms (30 Points)

1.1 Please provide the **definition of artificial intelligence** that we have discussed in the lecture and **name two key participants** of the **Dartmouth conference**. (3 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) Is ***feature scaling*** in **general required** after a ***normalization*** has been applied?
Please **briefly explain** your **decision**. (2 P)

1.3 Please **explain** the problem of ***"Model Autophagy Disorder (MAD)"*** in the **context** of **LLM/ChatGPT**. (3 P)

1.4 Please briefly **explain** the **difference** between ***random sampling*** and ***random walk*** in search algorithms by comparing both concepts. (2 P)

1.5 Please **define** ***supervised*** and ***unsupervised learning*** and **explain** the **difference** between the two approaches. (3 P)

1.6 Please briefly **explain** how ***CAPTCHAs*** function as a **reverse Turing Test**. (4 P)

1.7 Please briefly **explain** what ***neural networks*** are and **how** they **relate** to **deep learning**. (2 P)

1.8 Please briefly **discuss two ethical considerations** and their **potential on society** when using ***Large Language Models (LLMs)*.** (5 P)

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

2 Application of Machine Learning Algorithms (30 Points)

Consider Table 1. Table 1 represents various **customers** of a **financial services company** that **assesses creditworthiness**. Table 1 includes **two features**: "**SCORE_A**" and "**SCORE_B**". Each row in Table 1 also has a **class label** that is either "**TRUE**" or "**FALSE**" and is stored in the third **column** called "**CREDIT**".

Table 1. Customer Data.

ID	SCORE_A	SCORE_B	CREDIT
1	40	20	FALSE
2	50	50	TRUE
3	60	90	TRUE
4	10	25	FALSE
5	70	70	TRUE
6	60	10	FALSE
7	25	80	TRUE

2.1 The company has asked you to prepare a management presentation. Please **visualize** the data captured in Table 1 in a two-dimensional scatterplot. You can use symbols to visualize the CREDIT label. (5 P)

2.2 The company then asks you to **predict** the **creditworthiness** of the following new **customer** "$\mathbf{c_1}$".

$$c_1 = \{\text{SCORE\_A: } 20, \text{ SCORE\_B: } 35, \text{ CREDIT: } ?\}$$

Please **use** the **KNN algorithm** with ***k = 5*** and the ***Euclidean distance*** to **predict** the **creditworthiness** of the **above customer** based on the **data** in **Table 1**. Please **explain** your **calculations**. (5 P)

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

2.3 Next, you apply a pre-trained classification tree to the data in Table 1 to predict each customer's CREDIT label again. The predictions produced by the classification tree are shown in the following Table 2.

Table 2. Predicted CREDIT Labels.

ID	Predicted CREDIT
1	FALSE
2	FALSE
3	TRUE
4	TRUE
5	TRUE
6	FALSE
7	FALSE

To evaluate the tree's prediction performance based on the predicted labels in Table 2, please **compute** a **confusion matrix** and the following measures: **Accuracy**, **Precision**, **Recall**. (10 P)

2.4 Please **explain**: **Why** is it **not** a **good idea** to **evaluate** your **classification model** on **training data** like in task 2.3? (2 P)

2.5 As a next step, you want to improve your classification model. **How many models** will be **built** and **tested** when you use **grid search**, assuming that you consider the following **three hyperparameters?**

- *max_depth* = {2, 3, 5, 10, 20}
- *measure* = {gini, entropy}
- *min_samples_leaf* = {5, 10, 20, 50}

Please also **explain** your **calculations**. (2 P)

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

2.6 Please **fill** the six **missing parts** of the following **Python code** to **run** the **grid search** mentioned in the previous task. (3 P)

```
GridSearchCV(
    cv=_____________________,
    estimator=DecisionTreeClassifier(random_state=_____________),
    n_jobs=_____________________,
    param_grid={
        'criterion': [_____________________],
        'max_depth': [_____________________],
        'min_samples_leaf': [_____________________]},
    scoring='accuracy', verbose=1)
```

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

2.7 Please **name one method** other than grid search that can be used for **parameter tuning**. (1 P)

2.8 Please **explain *Wolpert's free lunch theorem*** using the **classification problem** of this section as an example. (2 P)

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

3 Data Preprocessing with Python (30 Points)

3.1 Please write a **Python function** named ***load_data*** that **accepts** a ***filename***, **reads** a ***CSV file*** into a ***Pandas DataFrame***, and **prints** the ***first five rows***. (2 P)

3.2 Please **write** a **new function** named ***handle_missing_values*** that does the same as ***load_data*** but **fills** in **missing values** with the **mean** of their **columns** in the **created DataFrame** before **printing** its ***first five rows***. (2 P)

3.3 Please **write** a **new function** named ***save_clean_data*** that does the same as ***handle_missing_values*** but also **saves** the **cleaned DataFrame** to a **new CSV file** and **adds** a **print statement** when the data is successfully saved. (2 P)

3.4 Please **write** a **short documentation** using **Python documentation** to **explain** the ***purpose*** and ***parameters*** of your function ***save_clean_data***. (2 P)

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

3.5 Please consider the **following code snippet** that is designed to **predict fraud** in a **car insurance dataset** using a simple **logistic regression** model. The code snippet **contains 6** ***errors or logical mistakes*** that will **result in** a ***runtime error*** **or** ***incorrect results*** from your model. Please **identify** the **6** ***errors or logical mistakes*** and **explain how** to **fix them**. (12 P)

```
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler

# Load data
data = pd.read_csv('car_insurance.csv')

# Prepare data
features = data[['age_string', 'total_claims']]
labels = data['fraud]

# Split data
X_train, X_test, y_train, y_test = train_test_split(features, labels,
test_size=0.99, random_state=42)

# Scale features
scaler = StandardScaler()
X_train_scaled = X_train
X_test_scaled = scaler.transform(X_train)

# Initialize and train model
model = LogisticRegression()
model.fit(X_train_scaled, y_train)

# Predict and evaluate
predictions = model.predict(X_test_scaled)
accuracy = accuracy_score(y_train, predictions)
print(f' accuracy ')
```

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

3.6 Please assume you have a Pandas dataframe that contains the following columns:

`Car_Model, Car_Year, Failure_Type, Repair_Cost`

Please provide **Python code** to **answer** the **following questions** about the dataframe. (10 P)

Question	Python Code
How can you **extract all records** where the **Car_Model** is **'Panamera'**?	
How would you **calculate** the **average Repair_Cost** for **each Failure_Type**?	
How can you **sort** the **DataFrame** by the **Car_Year** in **descending** order?	
How do you **find all entries** where the **Car_Year** is **before 2015** and the **Repair_Cost** is **greater than $1000**?	

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

How can you **summarize** the **total Repair_Cost** per **car_model**?	