*Tong Zhou, Farzana Yasmin and Christoph F. Eick*

COSC 4368 Fall 2025

Problem Set 2

**Task 3: Predicting Heart Disease using Decision Trees**

Individual Task

****

**Submission deadline:** Sa.., October 11 end of the day

**Weight:** 7% of the points that are allocated to the Problem Set Tasks

Responsible TA: Farzana

Last updated: Sept. 5, 9a

**Learning Objectives**:

1. Understand how decision trees are used for classification tasks.
2. Learn to tune hyperparameters in decision trees to avoid overfitting.
3. Gain proficiency in using recall, precision, and accuracy as metrics for evaluating classification model performance.
4. Learn to use cross-validation techniques to assess model performance.

In this assignment, you will work with a dataset containing medical data used for predicting heart disease. Your goal is to build, evaluate and compare different decision tree models for predicting whether a patient has a heart disease or not

**Overview:**
Heart disease refers to a range of conditions that affect the heart, often linked to blockages in the heart's arteries or irregular heart function. Risk factors include age, gender, high cholesterol, high blood pressure, and lifestyle factors such as exercise habits. The dataset provided includes various medical attributes that help in predicting heart disease. These factors, such as chest pain type, cholesterol levels, blood pressure, and exercise responses, are crucial in assessing an individual's risk. By analyzing these variables, machine learning models like decision trees and support vector machines (SVMs) can help predict the likelihood of heart disease and support preventive measures.

**Dataset:**

*Optional*  **Download the “heart-disease2.csv” dataset** [**here**](https://uofh-my.sharepoint.com/%3Ax%3A/g/personal/jyanagli_cougarnet_uh_edu/EfBsGgyHzN5AuWxvvHXh0SsBXkfQBoMWy2mvVn5RivYm_w?e=CTSwbe)**.**

This dataset contains medical data used for predicting heart disease. The data includes various attributes such as age, sex, chest pain type (cp), resting blood pressure (trestbps), cholesterol (chol), fasting blood sugar (fbs), resting electrocardiographic results (restecg), maximum heart rate achieved (thalach), exercise-induced angina (exang), and ST depression induced by exercise relative to rest (oldpeak). Attributes “chest pain type (cp)” and “resting electrocardiographic results (restecg)” have been one-hot encoded because those attributes are categorical, not nominal-the distinction is especially essential for SVMs.

The data was collected from patients during regular medical check-ups and heart disease screenings. Standard medical procedures and equipment were used to record the various attributes. Data was anonymized to protect patient privacy.

More details about the attributes of this dataset are listed below:

1. age: Age of the patient (in years) - Domain [29, 77],
2. sex: Sex of the patient (1 = male, 0 = female),
3. trestbps: Resting blood pressure (in mm Hg on admission to the hospital) – Domain [94, 200],
4. chol: Serum cholesterol in mg/dl – Domain [126, 564],
5. fbs: Fasting blood sugar > 120 mg/dl (1 = true; 0 = false),
6. thalach: Maximum heart rate achieved – Domain [71, 202],
7. exang: Exercise-induced angina (1 = yes; 0 = no),
8. oldpeak: ST depression induced by exercise relative to rest – Domain [0, 6.2],
9. slope: Domain [0, 2],
10. ca: Domain [0, 4],
11. thal: Domain [0, 3],
12. cp: Chest pain type (0 = typical angina, 1 = atypical angina, 2 = non-anginal pain, 3 = asymptomatic),
13. restecg: Resting electrocardiographic results (0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy) and
14. target: (Heart Disease Status: 1 = presence of heart disease, 0 = absence of heart disease; class attribute that needs to be predicted using SVM and Decision Trees).

The first 3 examples of the dataset are listed below:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| age | sex | trestbps | chol | fbs | thalach | exang | oldpeak | slope |
| 63 | 1 | 145 | 233 | 1 | 150 | 0 | 2.3 | 0 |
| 37 | 1 | 130 | 250 | 0 | 187 | 0 | 3.5 | 0 |
| 41 | 0 | 130 | 204 | 0 | 172 | 0 | 1.4 | 2 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ca | thal | cp\_0 | cp\_1 | cp\_2 | cp\_3 | restecg\_0 | restecg\_1 | restecg\_2 | target |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 2 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

**Tasks**

1. A. Using all attributes, **build a Decision Tree model** to predict whether a patient has a heart disease or not: Train the Decision Tree model using the given maximum depths: 3, 7, 11, 15.  **8 points**
2. Perform 5-fold cross-validation for each of the 5 max depths and compute accuracy (mean of validation scores), precision and recall. Generate a table, as given below, for the obtained results. **5 points**

|  |
| --- |
| **Decision Tree Experiments** |
| Max Depths | Accuracy | Precision | Recall |
| 2 |  |  |  |
| 3 |  |  |  |
| 5 |  |  |  |
| 8 |  |  |  |
| 12 |  |  |  |

1. Explain how the tree size/depth affects model performance in the context of overfitting/underfitting. **3 points**

1. Explain the meaning of the difference in accuracy, precision and recall scores in relation to the task; only if there is a significant difference. **2 points**

## **Deliverables:**

1. A python/R file with your code and analysis.
2. Report discussing your findings for tasks B-D

Note: If you use a Python notebook like Jupyter Notebook, you can include your conclusions in the notebook right after each question (no word document required), submit the python notebook file and submit a pdf copy of the notebook.

**Submission Guidelines[[1]](#footnote-1): needs to be updated**

1. Name your python/R files to **COSC4368F24-PS1T2-Firstname-Lastname.ipynb** or any other appropriate format.
2. Name the pdf copy of your report **COSC4368F24-PS1T2-Report-Firstname-Lastname.pdf**
3. Create a folder and name it **COSC3337F24-PS1T2-Firstname-Lastname**.The folder should contain both python/R file and pdf copy of your report named correctly. Compress (zip) the folder and submit it to MS TEAMS.
4. Submit on the Assignment tab in MS Teams.

**Guideline for Code Grading:**

1. Code documentation and readability: Use of comments, proper indentation, clear notations, and simplicity.
2. Code completeness: Working code with no errors including instructions for running your code. Indicate any module(s) to be downloaded before code is run.

**References:**

Python:

1. <https://www.datacamp.com/tutorial/decision-tree-classification-python>

R:

1. <https://www.datacamp.com/tutorial/decision-trees-R>

Task4

*Using Reinforcement Learning*

*To Discover Paths in a 2-Agent Transportation World*

Group Task (4(5) Students per Group)

First Draft



Responsible TA: Tong

Weight: 24% of the available problem set points.

Last updated: August 30, 2025

Expected Start Date: September 24, 2025

Deadlines: Submit 1-page Status report by October 13; Submit project report and source code by Nov. 7 end of the day; likely there will be demos scheduled in the Nov. 10 week.

In this project we will use reinforcement to learn and adapt “promising paths” in 2-agent setting. Learning objectives of Task4 include:

* Understanding basic reinforcement learning concepts such as utilities, policies, learning rates, discount rates and their interactions.
* Obtain experience in designing agent-based systems that explore and learn in initially unknown environment and which are capable to adapt to changes.
* Learning how to conduct experiments that evaluate the performance of reinforcement learning systems and learning to interpret such results.
* Development of visualization techniques summarizing how the agents move, how the world and the q-table changes, and the system performance.
* Development of path visualization and analysis techniques to interpret and evaluate the behavior of agent-based path-learning systems.
* Develop and learn coordination strategies for collaborating agents
* Learning to develop AI software in a team.



Figure 1: Visualization of the PD-World



Figure 2: An Urban Grid World.

In particular in this project you will use *Q-learning/SARSA[[2]](#footnote-2) for the PD-Word assuming a 2 agent setting (*[*http://www2.cs.uh.edu/~ceick/ai/2025-World.pptx*](http://www2.cs.uh.edu/~ceick/ai/2025-World.pptx) *)*, conducting four experiments using different parameters and policies, and summarize and interpret the experimental results. Moreover, you will develop path visualization techniques that are capable to shed light on what paths the learning system actually has learnt from obtained Q-Tables—we call such paths *attractive paths* in the remainder of this document. Moreover, you will analyze if the two agents collaborated well by avoiding blockage that occurs if the two agents work on the same path.

Two agent named ‘M’ (male) and ‘F (female) are solving the block transportation problem jointly. Agent alternate applying operators to the PD-World, with the female agent acting first. Moreover, both agents cannot be in the same position at the same time; consequently, there is a blockage problem, limiting agent mobility and ultimately efficiency in case that both agents work on the same path at the same time. There are two approaches to choose from to implement 2-agent reinforcement learning.

1. Each agent uses his/her own reinforcement learning strategy and Q-Table. However, we assume that the position the other agent occupies is visible to each agent, and can therefore can be part of the chosen reinforcement learning state space.
2. A single reinforcement learning strategy and Q-Table is used which moves both agents, selecting an operator for each agent and then executing the selected two operators.

Extra credit is given to groups who devise and implement both 2-agent learning approaches and compare their results for experiments 2 and 3 (see below)

In experiments we assume that q values are initialized with 0 at the beginning of the experiment. The following 3 policies will be used in the experiments:

* **PRANDOM**: If pickup and dropoff is applicable, choose this operator; otherwise, choose an applicable operator randomly.
* **PEXPLOIT**: If pickup and dropoff is applicable, choose this operator; otherwise, apply the applicable operator with the highest q-value (break ties by rolling a dice for operators with the same q-value) with probability 0.8 and choose a different applicable operator randomly with probability 0.2.
* **PGREEDY**: If pickup and dropoff is applicable, choose this operator; otherwise, apply the applicable operator with the highest q-value (break ties by rolling a dice for operators with the same q-value).



Figure 3: Visualization of an Attractive Path for a Search Problem

Objectives of the Experimental Evaluation: Besides analyzing the performance of various variations with respect to the bank account/how quickly the transportation problem was solved, the experimental evaluation should also additionally analyze:

1. Agent coordination: Do the two agents get in their ways blocking each other or do they do a good job in dividing the transportation task intelligently between each other. Agent coordination could, for example, be measured by computing the average Manhattan distance between the two agents during the run of a specific experiment.
2. Paths learned: Does the particular approach do a good job in learning paths between block sources and block destinations; is the learnt path the shortest path or close to the shortest path between the source and the destination.

Please conduct the following four experiments:

1. In Experiment 1 you use α=0.3 and γ=0.5, and run the traditional Q-learning algorithm for 8000 steps; initially you run the policy PRANDOM for 500 steps, then
	1. Continue running PRANDOM for 7500 more steps[[3]](#footnote-3)
	2. Run PGREEDY for the remaining 7500 steps
	3. Run PEXPLOIT for the remaining 7500 steps

Summarize and interpret the different results you obtain by running these three strategies. Also report one of the final Q-Table of experiment 1.c. Also assess the quality of the coordination between the two agents for experiments 1b and 1c.

1. Experiment 2 is the same as experiment 1.c except you run the SARSA q-learning variation for 8000 steps. When analyzing Experiment 2 center on comparing the performance of Q-learning and SARSA. Also report one of the final Q-tables of this experiment. Also assess the quality of agent coordination,
2. In Experiment 3 you rerun either[[4]](#footnote-4) Experiment 1.c or 2 but with learning rates α=0.15 and α=0.45. When interpreting the results focus on analyzing the effects of using the 3 different learning rates on the system performance.
3. Experiment 4 is the somewhat similar to Experiment 1c or 2; you use α=0.3 and γ=0.5 in conjunction with either Q-learning or SARSA[[5]](#footnote-5) as follows: you run PRANDOM for the first 500 steps; next, you run PEXPLOIT; however, after a terminal state is reached the third time, change the two pickup locations to: (1,2) and (4,5); the drop off locations and the Q-table remain unchanged; finally, you continue running PEXPLOIT with the “new” pickup locations until the agent reaches a terminal state the sixth time. When interpreting the results of this experiment center on analyzing on how well the learning strategy was able to adapt to the change of the pickup locations and to which extend it was able to learn “new” paths and unlearn “old” paths which became obsolete.

For all experiments, if a terminal state is reached, restart the experiment by resetting the PD world to the initial state, but do not reset the Q-table. Run each experiment twice, and report[[6]](#footnote-6) and interpret the results; e.g., utilities computed, rewards obtained in various stages of each experiment.

Assess which experiment obtained the best results[[7]](#footnote-7). Next, analyze the various q-tables you created and try identify attractive paths[[8]](#footnote-8) in the obtained q-tables, if there are any. Moreover, briefly assess if your system gets better after it solved a few PD-world problems—reached the terminal state at least once. Briefly analyze to which extend the results of the two different runs agree and disagree in the 4 experiments. Analyze agent coordination for experiments 1.c and 4. Finally, analyze how well the approach adapted to change in the fourth experiment.

Moreover,

* Make sure that you use different random generator seeds in different runs of the same experiment to obtain different results—having identical results for the 2 runs of the same experiment is unacceptable. It is okay just to report and interpret the Q-tables for the better of the two runs for each experiment, but you should report the performance variables for all eight runs.
* You should use the traditional Q-learning and SARSA algorithm in the project and not any other Q-learning variations or reinforcement learning algorithms!
* Never update the q-values of operators are not applicable in a Q-Tables!
* Groups who develop good methods for visualizing q-tables and good visualizations for the analysis of attractive paths obtain extra credit.
* Groups who develop sophisticated methods to analyze agent coordination receive a small amount of extra credit.
* Allow in your software design that the positions of dropoff and pickup positions might be changing before and during a run; otherwise, you will need to write a lot of additional software for experiment4.
* Evidence of the running of your system has to be provided using screen shots that will be delivered in a separate document.
* Groups that develop a very well designed and visually appealing visualization component will receive extra credit for this part of the 4368 Group Project!

Write a 8-12 pages report that summarizes the findings of the project. Be aware of the fact that at least 15-20% of the points available for this project are allocated to the interpretation of the experimental results. Finally, submit the source code of the software you wrote in addition to your project report and be ready to demo the system you developed. **needs to be updated**

Project Links

<http://courses.cs.washington.edu/courses/cse473/15sp/assignments/project3/project3.html>

<http://ai.berkeley.edu/project_overview.html>

<https://github.com/kristofvanmoffaert/Gridworld>

<http://cs.stanford.edu/people/karpathy/reinforcejs/gridworld_td.html>

<https://mediatum.ub.tum.de/doc/1238753/1238753.pdf>

<http://www2.econ.iastate.edu/tesfatsi/RLUsersGuide.ICAC2005.pdf>

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.113.7978&rep=rep1&type=pdf>

[1911.10635.pdf (arxiv.org)](https://arxiv.org/pdf/1911.10635.pdf)

1. Updates for Task3 if any will be added to this specification by Sept. 30, 2025, the latest. [↑](#footnote-ref-1)
2. SARSA is a variation of Q-learning that uses the q-value of the actually chosen action and not the q-value of the best action! [↑](#footnote-ref-2)
3. A step is the application of an operator by one of the two agents. [↑](#footnote-ref-3)
4. You have a choice here! [↑](#footnote-ref-4)
5. Again you have a choice here. [↑](#footnote-ref-5)
6. Additionally, report the following Q-tables for Experiments 2 (or Experiment 3 if you prefer that, in this case you will only need to report the final Q-Table of Experiment 2) in your report a) when the first drop-off location is filled (the fifth block has been delivered to it) and b) when a terminal state is reached and c) the final Q-table of each experiment.

The Q-table in the screenshot should be presented as a matrix, with *s* rows (states) and *t* columns (operators).  Thus, the Q-table for State Space 0, in the World 2022 pptx slides, has 25 x 2 rows and 6 columns; however, the q-values for the drop-off and pickup operators do not need to be reported. [↑](#footnote-ref-6)
7. Provide graphs that show, how the algorithm’s performance variables changed over the duration of the experiment in the three experiments. [↑](#footnote-ref-7)
8. A path going from (i,j) to (i’,j’) is attractive if, the q-values of the motion operators are high in comparison to other directions. Be aware of the fact that different paths are attractive for agents who hold a block and for agents how do not hold a block. [↑](#footnote-ref-8)