

Due: 04 April 2024, 23:00, to [Submit](#)

ASSIGNMENT1

AIN422



Instructors: Dr. Cemil ZALLUHOĞLU

TA: Hayriye ÇELİKBİLEK

Course: AIN422 Spring 2024

Subject: Basic Machine Learning and Artificial Intelligence

Given: 15.03.2023

Due: 04.04.2023, 23:00, to [Submit](#)

Introduction

The [Submit](#) system is our department's assignment management web application for students and instructors.

Since we handle student and instructor needed requirements privately using our services, rather than having the requirements only for developing some software, we will collect the following assignments (excluding this assignment) using the [Submit](#) webpage.

Policy

- ❖ Be sure to complete the submission deadline. The deadline is 23:00, and you see 23:59 on the live because of the advance compensation of potential problems. **Last-minute excuses will not be tolerated.**
- ❖ Save all your work until the assignment is graded.
- ❖ You can ask your questions via [Piazza](#), and you are supposed to be aware of everything discussed on Piazza.
- ❖ You must submit your work using the file hierarchy as stated below.
- ❖ No other submission methods will be accepted (mail)

Academic Integrity

All work on assignments must be done **individually** unless stated otherwise. You are encouraged to discuss the given assignments with your classmates, but these discussions should be carried out **abstractly**. Discussions about a particular solution to a problem (either in actual code or pseudocode) will not be tolerated. In short, you are turning in someone else's work (from the internet), in whole or part, as your own will be considered a **violation of academic integrity**. The former condition also holds for the material on the web, as everything on the web has been written by someone else.

References for the Academic Integrity (AI):

https://academicintegrity.ucsd.edu/AI-Handbook-for-UCSD_2019.pdf,
academicintegrity.org/resources/facts-and-statistics

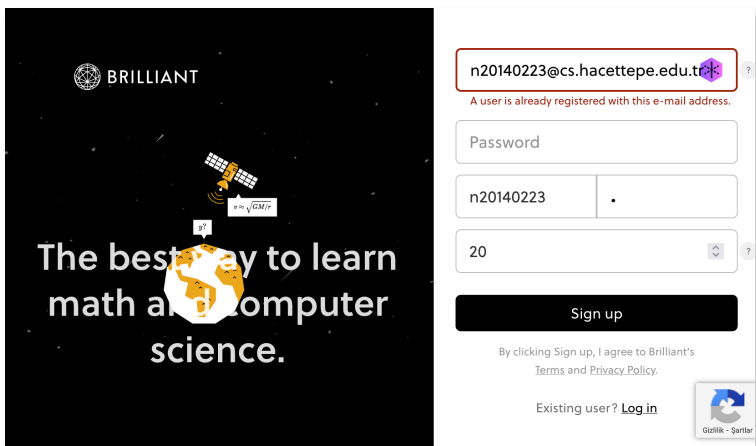
Joining Brilliant

Brilliant is an online learning platform that focuses on STEM-related topics. The 60+ courses on Science, Technology, Engineering and Math topics all offer an **interactive and hands-on learning experience**.

It is mandatory to use your **CS e-mail address** as your account, your **student ID b2XXXXXX** as your **name** and the text **'.'** (dot) as your **surname** while signing up (see Figure 1).

It is a must to use the **invitation link** to let course instructors see your solutions and time on the platform.

<https://brilliant.org/classroom/join/j72pbg/>



BRILLIANT

The best way to learn math and computer science.

n20140223@cs.hacettepe.edu.tr

A user is already registered with this e-mail address.

Password

n20140223 .

20

Sign up

By clicking Sign up, I agree to Brilliant's Terms and Privacy Policy.

Existing user? [Log in](#)

Gizlik - Şartlar

Figure 1. How to join brilliant.org

Work To Be Done

You are responsible for solving 10 of your selected lessons out of 24 in the Brilliant Classroom (see Figure 2).



Figure 2. The overall display of the lessons.

If you cannot solve some of these examples correctly, do not worry; the interactive problems include the solutions if you make any mistakes. There will be no point deduction about the spent time and correctness in Assignment 1; we value your effort. Take your effort wisely.

These **lessons should take 5 to 15 minutes each**, and the problems are fundamental and similar to your AIN422 slide examples, so you do not have to worry.

Solving the whole lesson is recommended but optional to ensure you are comfortable with the first assignment.

The Data

Heart Attack information of some patients with previous heart attack records and without any heart issues is given in the "**medical_heart.csv**" file. The data consists of 304 rows and 14 columns. One of the rows is the header, and one of the columns is the output of your prediction project. Splitting this data into training and test chunks is your responsibility. However, do not waste some of the data within chunks. Think about chunk dependencies you create in your model that differ from real-world scenarios. **It is obligatory to use balanced splitting** in order to avoid data bias. **Using cross-validation or nested cross-validation (some sort of cross-validation) is also obligatory.** Nevertheless, please do not use it because I said it without understanding. Show your understanding in your report.

The Report

The report "bXXXXXXX-Report.pdf" has to consist of the following sections. Each section and your overall report should have the ideal size you prefer. **Do not make it too short or too long please.** Make it a manageable length.

- ❖ **Cover:** Your name, student number, course name, assignment name, and the delivery date.
- ❖ **Introduction:** Define the project in your own words. What is the problem description? Project goals?
- ❖ **Data:** What are your materials? Define the data, its columns, units, and statistics.
- ❖ **Method:** What are your methods for solving the problem? Which classifiers have you selected? Why? Explain your method.
- ❖ **Development**
 - ❖ **Plan:** What is the project plan? Your requirements. (For example, your hardware requirements might affect the number of selected hyperparameters)
 - ❖ **Analysis:** Analyze the features of data. Pairwise and individual graphics related to them. Which features have you selected? Why?
 - ❖ **Design:** Which hyperparameters have you selected to tune? Have you used cross-validation or nested cross-validation?
 - ❖ **Implementation:** The flow of the implementation. Explain your code parts, such as functions, selected methods, and Jupyter blocks.
- ❖ **Programmer Catalog:** Write down the time you spend on analysis, design, implementation, testing, and

reporting. How can a programmer reuse your code for other purposes / other data? Is it reusable for a programmer? Why? Prepare the programmer manual of the program (should use unit/function explanations and usage examples) if your program is reusable. You may use the ".py" version of your project here.

❖ **User Catalog:** How can one reuse your code for other purposes / other data? Is it reusable for a regular non-programmer person? Why? Prepare the program's user manual (you might use screenshots) if your program is reusable. State the restrictions of your project. You may use the ".py" version of your project here.

❖ **Results:** Your results. Which metrics (select balanced metrics only) have you used? Why? Visual statements are preferable. Performance evaluation, comments, insights, and future work might be listed.

❖ **References:** Resources you have used to prepare the project and the report. Use numbers that are indicated in the text. Realize the possible academic integrity issues.

❖ Please be aware that your points will be deducted if you miss some of the parts mentioned here or skip them too quickly without a clear sight in your report.

❖ Please be aware that the explanation of each part given here is an example to clarify its purpose so that you can understand and prepare it. Do not get too stuck to the examples here.

❖ Including at least one or, ideally, more **flowcharts** of your algorithm or project flow is **obligatory**.

The Code

The code part should consist of several sections and have to be a "bXXXXXXXX-Supervised.ipynb" notebook file:

- ❖ Please do not forget to write your name and student number at the beginning of your code.
- ❖ Using **random seed 42** for all your random/normal operations is **obligatory throughout your project**, such as splitting, training and testing. Random seeding is necessary for the applicability and grading of your project. Please use random state variables wherever possible (i.e., in classifiers and splitters of sci-kit-learn).
- ❖ You must investigate each feature column/type by its statistical features, meaning and correlation to other feature columns. **Visual demonstrations are necessary and obligatory**. You can take advantage of feature deduction techniques if necessary, which means you need to be able to explain it in your report, or you should exclude it. There is no single answer or "correct way to do it". The only correct way is for you to be able to reason what you are doing. You may take groups of columns rather than individuals. This part will be demonstrated in your report; it is a must.
- ❖ Using the **sci-kit-learn** library to have a common language/framework is **obligatory**. You may select/tune multiple classifiers, regressors or any other Neural Network approach. **It is obligatory to stay in general machine learning or artificial intelligence approaches. Do not use deep learning please.**

- ❖ Using the [matplotlib](#) library for your visual project parts, such as data pre-processing and evaluation of your models, is [recommended](#).
- ❖ It is [obligatory to use some cross-validation](#) while training. Explain your cross-validation and why it is that way.
- ❖ You should tune 2-10 parameters. Do not go like 50 parameters; be reasonable with your problem scale and data.
- ❖ Your [evaluation metric](#) has to be one/some of the [balanced metrics](#). Do your search; do not come up with something basic and unbalanced, like accuracy. You may search the official metrics site of the sci-kit-learn library.
- ❖ Lastly, you will convert your ".ipynb" file onto a ".py" file, or you will have a [split](#) among them that has [fundamental modules inside your ".py" file](#). Do not forget to handle your design in your report.

CS Submit System

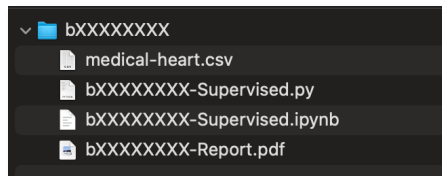


Figure 3. Showing your overall project hierarchy.

Finally, submit your **b2XXXXXXXX.zip** folder with your data, notebook, code, and report inside (see Figure 3) to the [Submit](#) web page once completed (see Figure 4). These four files and their naming are obligatory to have.

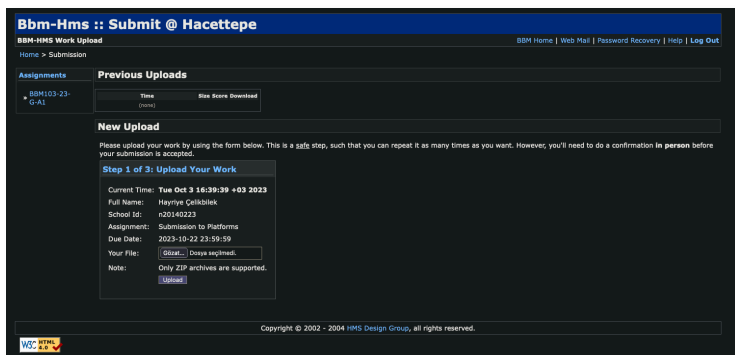


Figure 4. The overall vision of the Submit system.


Grading

In evaluating the assignment, the scoring is as follows:

Evaluation	Points	Evaluate Yourself
Solving 10 lessons from Brilliant	20	
Solving the whole 24 lessons from Brilliant	10 (Bonus)	
Wrong Folder, File names and hierarchy	-100	
Solving Brilliant without enrolling with the classroom link	-100	
No student ID signup naming in Brilliant	-100	
No student ID in code.	-100	
No student ID in report.	-100	
Missing Submit submission	-100	
Missing random seed	-100	
Lack of visual demonstrations	-30	

Evaluation	Points	Evaluate Yourself
Chosing basic metrics to evaluate your model	-40	
Report - Cover	2	
Report - Introduction, Data, Results	5	
Report - Method	8	
Report - Development Plan, Analysis	5	
Report - Development Design, Implementation	10	
Report - Programmer and User Catalogs	10	
Code	40	
Total	110	

You may include your evaluation guess as a file in your .zip folder named "**self-evaluation-table.pdf**" (or any other extension). This self-evaluation table is **optional** for feedback about your expectations, self-awareness, and effort.

 **GOOD LUCK
CONQUERERS OF THE
DECISION WORLD!**

At the end of this assignment, the best homework owner will be invited for tea time in my room (Lab. 5). We will take memory photos to remember the good times and share the photos with the class as a reward for the winner and to boost morale for the class. Better keep things tight from the beginning!

Appendix

Algorithmic Thinking, Visualization and Flowcharts:

- Python Code Debugger: visualizing code execution: pythontutor.com
- Creating flowcharts from text (pseudocode): chartmage.com/index.html
- Automatically create flowcharts from Python code:
<https://github.com/cdfmlr/pyflowchart>
<http://flowchart.js.org/>
- Creating flowcharts from code-like text: app.code2flow.com
- Creating flowcharts manually: draw.io or app.diagrams.net

Visualizing your mind and code while developing to debug or for reporting purposes is essential!

Formatting and Presentation Checklist for the Report:

- ☑ The target reader is a general high school graduate with a basic understanding of programming.
- ☑ Include a table of contents at the beginning of the report for easy navigation if your report is longer than 5-7 pages.
- ☑ Use a standard font and font size (e.g. Times New Roman, 12 pt) for consistency and readability.
- ☑ To maintain a professional and organised appearance, use a consistent formatting style throughout the report, including margins, line spacing, line breaks, and indentations.
- ☑ Use a clear, easy-to-read layout with sufficient white space to avoid clutter and make the report visually appealing. Aim for a balance between text and white space.

- ☑ Use a consistent and visually appealing style for headings and subheadings, such as bold text, italics, or larger font sizes. Ensure the hierarchy is clear and each section is easily distinguishable from the others.
- ☑ Use a clear and concise writing style, avoiding overly complex or technical language unless necessary. Use plain language wherever possible, and explain any technical terms or concepts unfamiliar to the reader.
- ☑ Break up long paragraphs into shorter ones for easier reading. Aiming for no more than five sentences per paragraph is a good guideline.
- ☑ Use (sub)headings to organize the content and guide the reader. This helps create a logical flow of information and allows the reader to find the information they need quickly.
- ☑ Use bullet points to highlight important information. This makes the information stand out and easier to read and digest.
- ☑ Use diagrams, charts or tables to explain complex concepts whenever possible. Visual aids can help to simplify complex information and make it easier for the reader to understand (such as flowcharts).
- ☑ Use consistent and appropriate terminology and abbreviations throughout the report.
- ☑ Include code snippets to illustrate specific examples or concepts and format them clearly for readability. Use a monospace font (e.g. Courier) and separate code from text using a different background colour, indentation, or border. Also, test and verify the code examples to check whether they work correctly before including them in your documentation.
- ☑ Use proper grammar, spelling and punctuation, and proofread carefully for errors to make your report well-organized. Use a

spell checker and grammar checker, but read the report out loud or have someone else proofread it to catch any mistakes.

<https://www.scribens.com/> or https://www.paperrater.com/free_paper_grader

- ☑ Provide a clear and concise conclusion summarizing the main findings or conclusions.
- ☑ Use appropriate referencing (if needed), following a recognized citation style consistently and accurately.