# [AI-PACKAGE]

Milestone 3

**Due Date: Thurs 9th of May**

## General instructions:

### Regarding your AI-Package:

1. Each **team member** should participate of **each** milestone, as he/she will be asked for a **modification** for **any** milestone.
2. Add your new milestone folder to your 'AI-Package'.
3. Only **One AI-Package** folder should be available on shared folder either by **updating** old package **or remove** the old and upload the **whole** package again (including previous milestones).
4. After you finish writing your code:
    a. Open the folder shared with your team on GoogleDrive (the one with your team number)
    b. Upload all the project files with the same hierarchy.
5. Submit **only running** code that you have tested before.
6. **Compressed** files (.zip/.rar) are **not allowed.**
7. The Submission of package is **only** through **your shared folder on google drive.**

### Regarding the submission [Milestone 3]:

1. Add a new folder named 'MachineLearning' in the 'AI-Package' project.
2. Add the shared template file named 'MachineLearning.py' to "MachineLearning' folder.
3. Your code should be written **only** in 'MachineLearning.py' file under 'MachineLearning' folder.
4. 4. Please, read code documentation carefully.
5. 5. This milestone will be **autograded**.

**This package is intended for team work contribution. Sharing ideas or part of the answers is considered plagiarism and will not be tolerated. All submissions will be checked for plagiarism automatically.**

## Milestone 3

- A dataset holds a diagnosis for the eyes of 24 patients.

- The diagnosis is based on the following features:
  1. Age: (0) young, (1) adult.
  2. Prescription: (0) myope, (1) hypermetrope.
  3. Astigmatic: (0) no, (1) yes.
  4. tear production rate: (0) normal, (1) reduced.

- The output classes are:
  1. Need contact lenses (1): the patient should be fitted with a special type of contact lenses.
  2. No contact lenses (0): the patient should not be fitted with a special type of contact lenses.

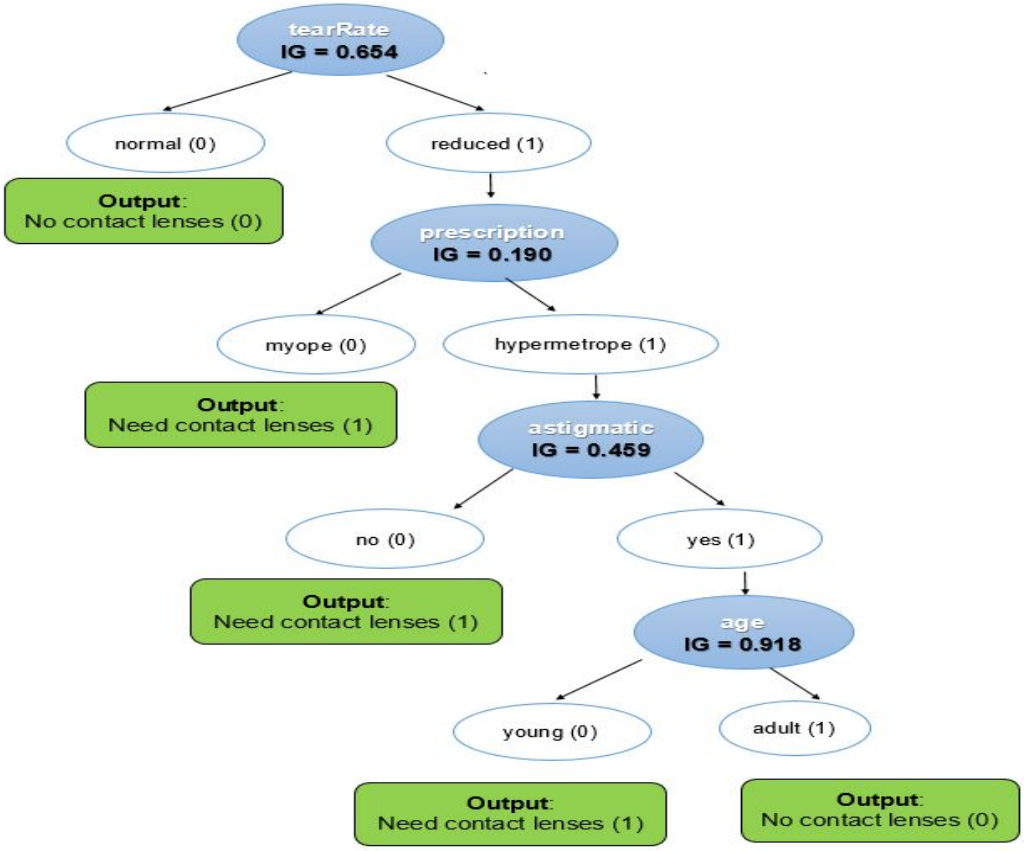
- Dataset Sample:

| Age | Prescription | Astigmatic | Tear Production Rate | Diagnosis |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |

- Task:
  1. Classification using ID3 algorithm.
  2. Each feature has only two attributes 0 or 1.
  3. Output classes are only two values => 0 (no special contact lenses) and 1 (need special contact lenses)

- Problem Solution:

  Note that you are not asked to render the tree, this is just a guide for you




- Sample Input & Output:

| | Sample Input | Sample Output |
|---|---|---|
| Test case 1 | [0, 0, 1, 1] | 1 |
| Test case 2 | [1, 1, 0, 0] | 0 |
| Test case 3 | [1, 1, 0, 1] | 1 |

- Template:

```
class item:
    def __init__(self, age, prescription, astigmatic, tearRate, needLense):
        self.age = age
        self.prescription = prescription
        self.astigmatic = astigmatic
        self.tearRate = tearRate
        self.needLense = needLense

def getDataset():
```

Class item:
Each object represents a row in the dataset.

```
class Feature:
    def __init__(self, name):
        self.name = name
        self.visited = -1
        self.infoGain = -1

class ID3:
    def __init__(self, features):
        self.features = features


    def classify(self, input):
        #takes an array for the features ex. [0, 0, 1, 1]
        #should return 0 or 1 based on the classification
        pass
```

Class feature:
Each object represents a feature (age, tear rate, prescription, and

Class ID3:
- Your code should appear here.
- Classify function takes input features as array and classify case to 0 or 1.

```
dataset = getDataset()
features = [Feature('age'),Feature('prescription'),Feature('astigmatic'),Feature('tearRate')]
id3 = ID3(features)
cls = id3.classify([0, 0, 1, 1]) # should print 1
print('testcase 1: ', cls)
cls = id3.classify([1, 1, 0, 0]) # should print 0
print('testcase 2: ', cls)
cls = id3.classify([1, 1, 1, 0]) # should print 0
print('testcase 3: ', cls)
cls = id3.classify([1, 1, 0, 1]) # should print 1
print('testcase 4: ', cls)
```

Main:
- Getting the dataset in dataset variable.
- Initializes features.
- Creating object from ID3 class.
- 4 test cases with the expected output.