DAILY ASSESSMENT FORMAT

Date:	22-05-2020	Name:	BINDUSHRI
Course:	TCSion	USN:	4AL17EC011
Topic:	1.Artificial intelligence part 1 & 2 2.certificate	Semester & Section:	6th A
Github Repository:	Bindushri		

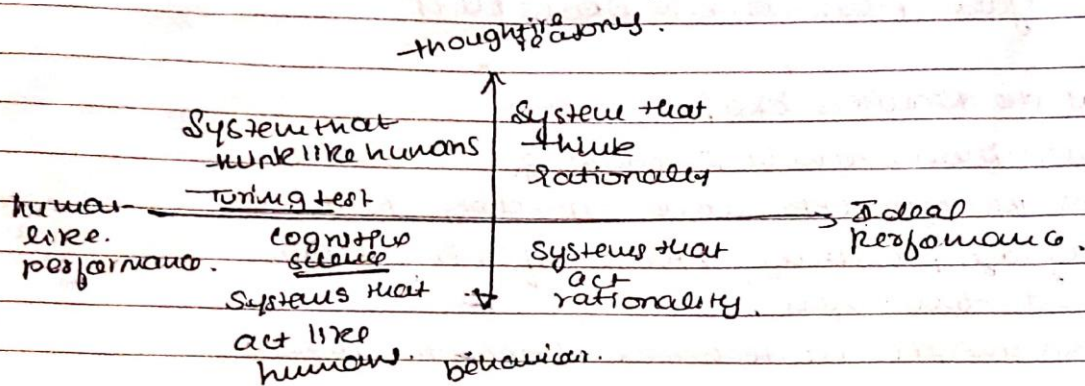| FORENOON SESSION DETAILS |
|---|

22-05-2020.

Day13: Artificial Intelligence - Part 1

understand the role of basic
- knowledge of representation
- Problem solving

* Assess the applicability, strength & weakness
*
Artificial Intelligence :- is concerned with the design
of intelligence. in an artificial device

thought processing.

System that think like humans	System that think rationally
Turing test	
human-like performance	Ideal performance
cognitive science	Systems that act rationally.
Systems that act like humans behaviour.	

Basic Intelligence behaviour
1) perception   2) reasning   3) learning 4) solving
Problem   5) understand the language.

Application ~~testing~~ Robotics, Image Recognization

Day14 : Artificial Intelligence - Part2

* <u>Agent</u>:- operates in an environment. recieves
percepts from environment and its
action can change the environment.

~~Agent~~ Agent has actuators or effectors
to take the actions
also have the sensor to sense the change Environment

* Autonomous agent decede automaticostey which
measure Pts performance is maximum

* task mapping:

Types of agent :- 1) Softbot 2) Expert systems
3) Autonomous Spacecraft 4) Intelligent
Building

Determinism Environment

1) Deterministic. 2) Stochastic 3) Strategic
3) Episodic / Sequential.

Table based agent :- Simpler way to specify
a mapping from percepts to action.
* tables may become very large.
* mapping is impericitly defined by program
1) Rule Based 2) neural networks 3) algorithm

State-based agent :- Information comes from
Sensors - percepts
changes the agents current to the state of world

Summary : An agent program maps from percepts
to action & updates its external state

Date:22may2020	**Name:Bindushri**
Course: python	**USN:4AL17EC011**
Topic:	**Sem&Sec:6th A**
Basics:sec16-17	
AFTERNOON SESSION DETAILS	
Image of session	

Course content

- 123. Convert Images to Numpy Arrays
  6min · Resources

- 124. Indexing, Slicing, and Iterating Numpy Arrays
  4min

- 125. Stacking and Splitting Numpy Arrays
  6min

Section 17: Application 2: Create Webmaps with Python and Folium
0 / 16 | 1hr 20min

Section 18: Fixing Programming Errors
0 / 6 | 39min

Section 19: Application 3: Build a Website Blocker
0 / 10 | 1hr 20min

Section 20: Application 4: Build a Personal

Overview   Q&A   Notes   Announcements

22-05-2020.

Section 16:- numpy

* to Install numpy
→ Precompiled Extension package of python
→ numpy-1.11.0+mkl-cp35-cp35m-win-amd64.wh.

* → In[]: import numpy.
  →   n=numpy.arange (27).
  In [12] : type (n)
  Out[12]: numpy.
  In[12]: print [n]

  O/P → [0 1 2 . . . . 26]

→ In order to get 2'dimensional array.

  In [13] : n.reshape [3,9]

  O/P. array ([0 , 1, - - - - - 6],
              [9 - - - - - .17],
              [18 - - - - - - 26])

→ In order to get 3'dimensional array.

  In [15]: n.reshape (3,3,3)

              array ((([0, 1, 2],
                       [3, 4, 5],
                       [6,7,8],

                      [[9, 10, 11],
                       [12,13,14],
                       [15,16,17]]

                      [[18, 19,20]
                       [21, 22,23]
                       [24, 25 ,26]]))

Installing opencv :- pip install opencv - Python

* converting image to numpy - Arrays

In[21]: Import cv2
In[ ]: img*cv2.imread ["Smallgray.png",0)
        img.

O|p : array ([[187, 158, 104, 121, 143],
              [198, 125, 255, 255, 147]
              [209, 134, 255, 97, 182]], dtype=uint8)

In[ ]: cv2. imwrite (" newsmallgray.png," img)
o|p → true

→ Indexing, slicing and iterating numpy Arrays

In [39]: img [0:2 , 2:4]
out [39]: array ([[104, 121]
                   [255, 255]], dtype =uint8)

‡0 to Run iterate value by value

In[47]: for i in img.flat:
            print [i]

o|p   187
      158

      164
      121
      .
      .
      .
      .
      182

Section 17

Application: Create webmaps with python and
folium.

✗ map has built using python library)

✓ to build map in python (map1.html).

Methods:
[Create folder → mapping → and open
atom. VS code. → open terminal
→ pip install folium → pip install pandas]

≫ python
≫ import folium
≫ map = folium.map (location= [80, -100])
≫ map
≫ map. save ("map1.html")

c/o :- (this creates the map.);

Note:- tiles = "MapboxBright"
→ tiles = "stamen terrain"

thes are two types of basemaps, but
mapbox bright doesn't work anymore
stamen terrain works great

✳ Adding points to map

✓ map1. py → folder mapping

1. import folium

2. map = folium.Map (location = [38.58, -99.09], zoom _ start = 6, tiles = "Mapbox Bright")

3. Map.save ("Map1.html")

to add elements to the map then

→ map.add_child (folium.Marker(location = [38.2, -99.1], popup = "Hi I am a marker", icon = folium.Icon (color = 'green')))

O/P :- [check popup the ~~on your feet~~ map window there the ~~etp~~ location of your place with "Hi am a marker" is printed.]

1. import folium.

2. map = folium.Map (location = [38.58, -99.09], zoom start = 6, tiles = "Mapbox Bright")

3. fg = folium. Featuregroup (name = "my map")

4. fg.add_child (folium. Marker(location = [38.2, -99.1], popup = "Hi I am a Marker", icon = folium. Icon (color = 'green'))

6. map.add_child (fg)

7. map.save ("map1.html")

[can also execute this line multiple time with changing location values]

※ Adding multiple points

this is adding a for loop before the line 4.

→ Extract data.

Mapplug ——→ (volcano.txt) under (mapl.py)

↓

adding multiple points from the file.

>>> pip install pandas

>> import pandas
>>> data = pandas.read_csv ("volcanoes.txt")
>> data.
          → (produces the o/p)

>>> data. columns
>> lat = list(data ["LON"])
>> lat

code

1. import folium
2. import pandas.

3. data = pandas. read_csv ("volcanoes.txt")
4. lat = list (data ["LAT"])
5. lon = list (data ["LON"])
7. elev = list (data ["ELEV"])
8. map = folium. Map (location = [38, 58, -99.09].
          zoom_Start = 6, tiles = "Mapbox Brignt")
9
10. fg = folium _feature group (name = "my map")

11. for lt, ln,eln zip (lat, lon,elev):
          fg. add_child (folium _marker (location=
          [lt, ln], propup= str (el)+ "m"
          folium, Iccen= (ccolor = 'green')))
                    folium. Icon

12. map. add _child (fg)
13. map. Save ("map.html")

Colorpoints

```python
import folium
import pandas

data = pandas.read_csv("volcanoes.txt")
lat = list(data["LAT"])
lon = list(data["LON"])
elev = list(data["ELEV"])

def color_producer(elevation):
    if elevation < 1000:
        return 'green'
    elif 1000 <= elevation < 3000:
        return 'orange'
    else:
        return 'red'

map = folium.map(location=[38,58,-99.09],
        zoom_start=6, tiles="Mapbox Bright")

fg = folium.FeatureGroup(name="My map")

for lt, ln, el in zip(lat, lon, elev):
    fg.add_child(folium.marker(location=
        [lt, ln], popup=str(el)+"m", icon=
        folium.Icon(color_producer(el))))

map.add_child(fg)
map.save("map1.html")
```