**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **22/05/2020** | **Name:** | **Dhavala** |
| **Course:** | **TCSion** | **USN:** | **4AL17EC027** |
| **Topic:** | * **Understand Artificial Intelligence (AI) - Part 1** * **Understand Artificial Intelligence (AI) - Part 2** | **Semester & Section:** | **6TH SEM & A Section** |
| **Github Repository:** | **Dhavala27** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report**  Understand Artificial Intelligence (AI) - Part 1  In this course we are able to understand the role of basic knowledge representation, problem solving, learning methods in AI in engineering intelligent systems.  Objectives   * Develop intelligent systems by assembling solutions to concrete computational problems * Appreciate the role of problem solving, natural language processing and vision in understanding human intelligence from a computational perspective.   Goals   * Formulate problems as state space search problem and efficiently solve them * Write game playing programs * Use machine learning to find patterns in data * Build expert systems   What is AI?  Artificial intelligence is concerned with the design of intelligence in an artificial device.  Typical AI Problems  Intelligent entities (or “agents”) need to be able to do both “mundane” and “expert” tasks:  Mundane tasks   * Planning route, activity. * Recognizing (through vision) people, objects. * Communicating (through natural language) * Navigating round obstacles on the street   Expert tasks   * Medical diagnosis * Mathematical Problem solving   Intelligent behavior   * Perception * Reasoning * Learning * Understanding language * Solving problems   Applications   * Computer vision * Image recognition * Robotics * Language processing   Practical impact of AI  AI components are embedded in numerous device  AI systems are in every use   * Detecting credit card fraud * Configuring products * Aiding complex planning tasks * Advising physicians   Approaches to AI  Strong AI aims to build machine that can truly reason and solve problems which is self aware and whose overall intellectual ability is indistinguishable from that of a human being.   * Human like * Non-human like   Weal AI: Deals with the creation of some form of computer-based artificial intelligence that cannot truly reason and solve problems, but can act as if it were intelligent.  Weal AI holds that suitably programmed machine can simulate human cognition.  Strong AI maintains that suitably programmed machine is capable of cognitive metal states  Limits od AI today  Today’s successful AI systems   * Operate in well-defined domains * Employ narrow, specialized knowledge   Commonsense knowledge   * Understand unconstrained natural language * Needed in complex and open-ended world   What can AI system do   * Computer vision: face recognition * Robotics: autonomous automobile * Natural language processing: Simple machine translation * Expert system: medical diagnosis in a narrow domain * Spoken language: 1000 words continuous speech * Planning and scheduling: Hubble telescope experiments * Learning: Text categorization in to 1000 topics * Games: Grand master level in chess   What AI system can’t do yet   * Understand natural language robustly * Surf the web * Interpret an arbitrary visual scene * Learn a natural language * Construct plans in dynamic real time domains * Exhibit true autonomy and intelligence   Understand Artificial Intelligence (AI) - Part 2  In this module we are going to learn what is agent, intelligent agent, rational agent, bounded rationality, types of environment, different agent architectures  Agent   * Operate in an environment * Perceives its environment through sensor * Acts upon its environment through actuators/ effectors   An agent perceives its environment through sensor  It can change the environment through effectors  Performance measure: A subjective measure to characterize how successful an agent is.  Types of agent   * Softbots * Expert system * Autonomous spacecraft * Intelligent buildings   Fundamental faculties of intelligence   * Acting * Sensing * Understanding reasoning learning   Rationality   * Prefect rationality * Bounded Rationality * Rational Action |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date:** | **22/05/2020** | | **Name:** | **Dhavala** | |
| **Course:** | **Python** | | **USN:** | **4AL17EC027** | |
| **Topic:** | * **Application 2: Create Web maps with Python and Folium** | | **Semester & Section:** | **6TH SEM & A Section** | |
| **Github Repository:** | **Dhavala27** | |  |  | |
|  | |
|  | |
|  | |
| **AFTERNOON SESSION DETAILS** | | | | |
| **Image of session** | | | | |

|  |
| --- |
| **Report**  Install OpenCV:  1. Open the command line and type:  Pip install opencv-python  2. Open a Python session and try:  Import cv2  3. If you get no errors, you installed OpenCV successfully.  If any problem occurs  1. My OpenCV installation didn't work on Windows  Solution:   * Uninstall OpenCV with: * Pip uninstall opencv-python * 2. Download a wheel (.whl) file from [this link](http://www.lfd.uci.edu/~gohlke/pythonlibs/#opencv) and install the file with pip. Make sure you download the correct file for your Windows and your Python versions. For example, for Python 3.6 on Windows 64-bit you would do this: * Pip install opencv\_python-3.2.0-cp36m-win\_amd64.whl * 3. Try to import cv2 in Python again. If there's still an error, type the following again in the command line: * Pip install opencv-python 4. Try importing cv2 again. It should work at this point.   2. My OpenCV installation didn't work on Mac  Solution:  If pip install opencv-python didn't work, install OpenCV for Python 2 and use Python 2 to run the programs that contains cv2 code. Because Python 2 is installed by default on Mac, you don't need to install Python 2.  Here are the steps to correctly install OpenCV:   * Install *brew*. * To install *brew,*open your terminal and execute the following: * /usr/bin/ruby -e “$(curl-fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)” * OpenCV depends on GTK+, so install that dependency first with brew (always from the terminal): * Brew install gtk+ * Install OpenCV with brew: * Brew install opencv * Open Python 2 by typing: * python * Import cv2 in Python: * Import cv2 * If you get no errors, you installed OpenCV successfully.   3. My OpenCV installation didn't work on Linux   * Open your terminal and execute the following commands one by one: * sudo apt-get install libqt4-dev * cmake -D WITH\_QT=ON .. * make * sudo make install * If the above commands don't work, execute this: * sudo apt-get install libopencv-\* * Then, install OpenCV with pip: * Pip install opencv-python * Import cv2 in Python. If you get no errors, you installed OpenCV successfully.   HTML on Popups  Note that if you want to have stylized text (bold, different fonts, etc) in the popup window you can use HTML.  example:  import folium  import pandas  data = pandas.read\_csv("Volcanoes.txt")  lat = list(data["LAT"])  lon = list(data["LON"])  elev = list(data["ELEV"])   html = """<h4>Volcano information:</h4>  Height: %s m  """  map = folium.Map(location=[38.58, -99.09], zoom\_start=5, tiles="Mapbox Bright")  fg = folium.FeatureGroup(name = "My Map")   for lt, ln, el in zip(lat, lon, elev):  iframe = folium.IFrame(html=html % str(el), width=200, height=100)  fg.add\_child(folium.Marker(location=[lt, ln], popup=folium.Popup(iframe), icon = folium.Icon(color = "green")))   map.add\_child(fg)  map.save("Map\_html\_popup\_simple.html")  You can even put links in the popup window. For example, the code below will produce a popup window with the name of the volcano as a link which does a Google search for that particular volcano when clicked:  import folium  import pandas   data = pandas.read\_csv("Volcanoes.txt")  lat = list(data["LAT"])  lon = list(data["LON"])  elev = list(data["ELEV"])  name = list(data["NAME"])   html = """  Volcano name:<br>  <a href="https://www.google.com/search?q=%%22%s%%22" target="\_blank">%s</a><br>  Height: %s m  """  map = folium.Map(location=[38.58, -99.09], zoom\_start=5, tiles="Mapbox Bright")  fg = folium.FeatureGroup(name = "My Map")   for lt, ln, el, name in zip(lat, lon, elev, name):  iframe = folium.IFrame(html=html % (name, name, el), width=200, height=100)  fg.add\_child(folium.Marker(location=[lt, ln], popup=folium.Popup(iframe), icon = folium.Icon(color = "green")))   map.add\_child(fg)  map.save("Map\_html\_popup\_advanced.html") |