**DAY 5 ASSIGNMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **22-05-2020** | **Name:** | **Ashish Shanbhag** |
| **Course:** | **TCS-ION** | **USN:** | **4AL16EC008** |
| **Topic:** | **1. Understand Artificial Intelligence (AI) - Part 1**  **2. Understand Artificial Intelligence (AI) – Part 2**  **3. Assessment** | **Semester & Section:** | **8th A** |
| **Github Repository:** | **Ashish Shanbhag** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Artificial Intelligence**  Artificial Intelligence is concerned with the design of intelligence in an artificial device. The term was coined by McCarthy in 1956. There are two ideas in the definition and they are Intelligence and artificial device.  What is intelligence? – Is it that which characterize humans? Or is there an absolute standard of judgement? – Accordingly there are two possibilities: A system with intelligence is expected to behave as intelligently as a human and A system with intelligence is expected to behave in the best possible manner – Secondly what type of behavior are we talking about? – Are we looking at the thought process or reasoning ability of the system? – Or are we only interested in the final manifestations of the system in terms of its actions?  **Typical AI problems**  While studying the typical range of tasks that we might expect an “intelligent entity” to perform, we need to consider both “common-place” tasks as well as expert tasks. Examples of common-place tasks include  – Recognizing people, objects.  – Communicating (through natural language).  – Navigating around obstacles on the streets.  These tasks are done matter of factly and routinely by people and some other animals.  Expert tasks include:  • Medical diagnosis.  • Mathematical problem solving  • Playing games like chess  These tasks cannot be done by all people, and can only be performed by skilled specialists.  **Intelligent behaviour**  This discussion brings us back to the question of what constitutes intelligent behaviour. Some of these tasks and applications are:    * Perception involving image recognition and computer vision  * Reasoning  * Learning  * Understanding language involving natural language processing, speech processing  * Solving problems  * Robotics   **Practical Impact of AI**  AI components are embedded in numerous devices e.g. in copy machines for automatic correction of operation for copy quality improvement. AI systems are in everyday use for identifying credit card fraud, for advising doctors, for recognizing speech and in helping complex planning tasks. Then there are intelligent tutoring systems that provide students with personalized attention. Thus AI has increased understanding of the nature of intelligence and found many applications. It has helped in the understanding of human reasoning, and of the nature of intelligence. It has also helped us understand the complexity of modeling human reasoning.  **Approaches to AI**  **Strong AI** aims to build machines that can truly reason and solve problems. These machines should be self aware and their overall intellectual ability needs to be indistinguishable from that of a human being. Excessive optimism in the 1950s and 1960s concerning strong AI has given way to an appreciation of the extreme difficulty of the problem. Strong AI maintains that suitably programmed machines are capable of cognitive mental states.  **Weak 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. Weak AI holds that suitably programmed machines can simulate human cognition.  **Applied AI**: aims to produce commercially viable "smart" systems such as, for example, a security system that is able to recognise the faces of people who are permitted to enter a particular building. Applied AI has already enjoyed considerable success.  **Cognitive AI**: computers are used to test theories about how the human mind works--for example, theories about how we recognise faces and other objects, or about how we solve abstract problems.  **Limits of AI**  Today Today’s successful AI systems operate in well-defined domains and employ narrow, specialized knowledge. Common sense knowledge is needed to function in complex, open-ended worlds. Such a system also needs to understand unconstrained natural language. However these capabilities are not yet fully present in today’s intelligent systems.  **What can AI systems do**  Today’s AI systems have been able to achieve limited success in some of these tasks.  • In Computer vision, the systems are capable of face recognition  • In Robotics, we have been able to make vehicles that are mostly autonomous.  • In Natural language processing, we have systems that are capable of simple machine translation.  • Today’s Expert systems can carry out medical diagnosis in a narrow domain  • Speech understanding systems are capable of recognizing several thousand words continuous speech  • Planning and scheduling systems had been employed in scheduling experiments with the Hubble Telescope.  • The Learning systems are capable of doing text categorization into about a 1000 topics  • In Games, AI systems can play at the Grand Master level in chess (world champion), checkers, etc.  **What can AI systems NOT do yet?**  • Understand natural language robustly (e.g., read and understand articles in a newspaper)  • Surf the web  • Interpret an arbitrary visual scene  • Learn a natural language  • Construct plans in dynamic real-time domains  • Exhibit true autonomy and intelligence |

**DAY 5 ASSIGNMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **22-05-2020** | **Name:** | **Ashish Shanbhag** |
| **Course:** | **PYTHON** | **USN:** | **4AL16EC008** |
| **Topic:** | **Python** | **Semester & Section:** | **8th A** |
| **Github Repository:** | **Ashish Shanbhag** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS**      **Project Exercise with Python and MySQL: Interactive English Dictionary**  **Introduction to the App & Making of the App –**  **Code:**    **Output –**    In the example we used the following SQL statement in our Python code:  query = cursor.execute ("SELECT \* FROM Dictionary WHERE Expression = 'rain'")  This statement retrieved all the rows of the Dictionary table where the value of the column Expression was rain. The string inside cursor.execute () is SQL code that Python sends to the database. That kind of language is understood by the database.  **Data Analysis with Pandas–**  Installing Pandas  Make sure you have pandas installed. You can install it with pip:  pip install pandasorpip3 install pandas  Also, in the next lecture, we will use an enhanced Python interactive shell called IPython.IPython is just like the normal shell you get when you run python, but IPython provides better printing for large text. This ability makes IPython suitable for data analysis because the program prints data in a well-structured format. You can install IPython with pip:  pip install ipython orpip3 install ipython  **Loading JSON Files**  In the previous lecture you learned that you can load a CSV file with this code:   1. import pandas 2. df1 = pandas.read\_csv("supermarkets.csv")   Try loading the supermarkets.json file for this exercise using read\_json instead of read\_csv.  The supermarkets.json file can be found inside the supermarkets.zip file attached in the previous lecture*.*  The code for loading the supermarkets.json file in Python with pandas would be this:   1. import pandas 2. df2 = pandas.read\_json("supermarkets.json")   The df2 data frame should contain this data:  https://i.udemycdn.com/redactor/raw/2019-03-04_22-08-20-408555f66e663f35bb73f1a9c8b43778.png  **Loading Excel Files**  To load Excel (.xlsx) files in Python with pandas. Pandas may require the xlrd library as a dependency. If you get an error such as ModuleNotFoundError: No module named 'xlrd', you can fix the error by installing xlrd:pip install xlrd orpip3 install xlrd  **Geopy -**  Geopy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources.  Geopy includes geocoder classes for the [OpenStreetMap Nominatim](https://nominatim.org/), [Google Geocoding API (V3)](https://developers.google.com/maps/documentation/geocoding/), and many other geocoding services. The full list is available on the [Geocoders doc section](https://geopy.readthedocs.io/en/latest/" \l "geocoders). Geocoder classes are located in [geopy. Geocoders](https://github.com/geopy/geopy/tree/master/geopy/geocoders). |