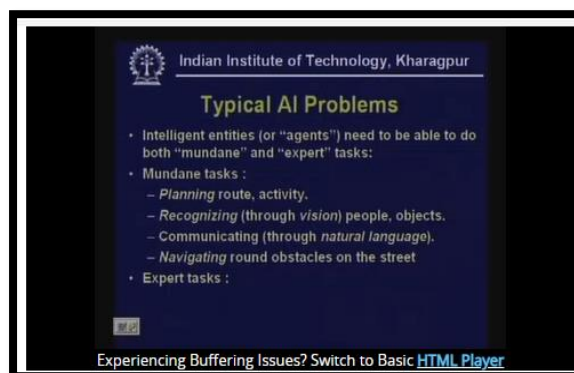
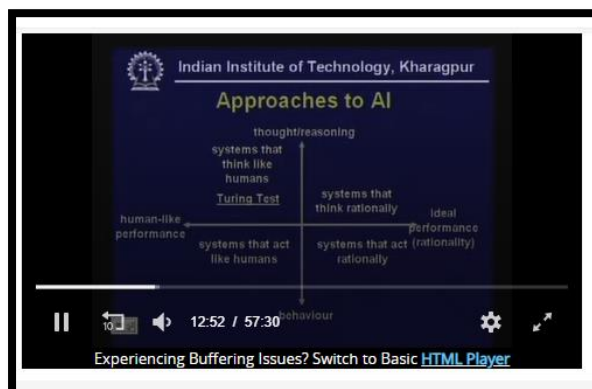


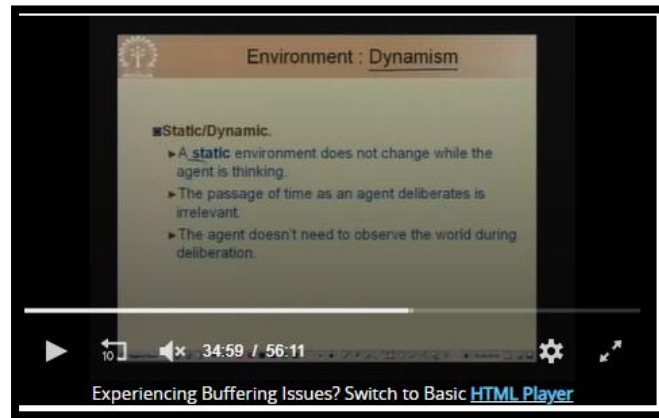
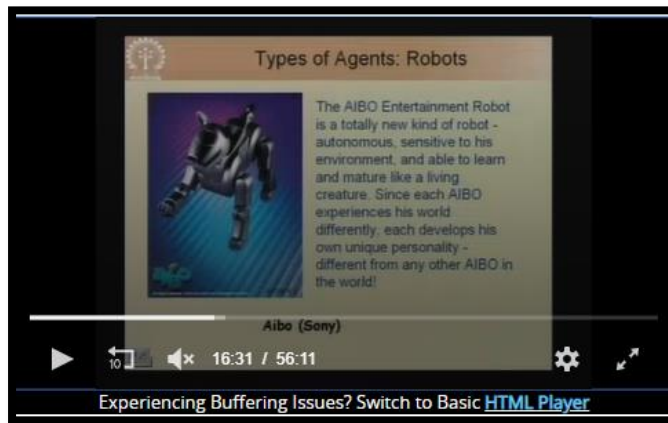
## DAILY ASSESSMENT FORMAT

Date:	22/05/2020	Name:	Davis S. Patel
Course:	TC ION	USN:	4AL16EC045
Topic:	Understand Artificial Intelligence (AI) - Part 1 Understand Artificial Intelligence (AI) - Part 2 Assessment	Semester & Section:	8 <sup>th</sup> - A
GitHub Repository:	Davis		

### FORENOON SESSION DETAILS

Image of session





Total Marks	Pass Marks	Attempts Taken	Duration	Start Time	View Assessment	Already cleared
30.0	18.0	01	30 Mins	17 May 2020 12:00 AM TO 16 Jul 2020 12:00 AM	Analysis At the End of Assessment	assessment.
My Attempts						
Attempted On	Attempted Duration (Submission Time)		Marks Obtained	Status	Action	
22 May 2020 01:55 PM	0:26:17 Hrs(02:21 PM)		20.0/30.0	Pass	-	

## **Report –**

### **Understand Artificial Intelligence (AI) - Part 1 & Part 2**

Artificial intelligence applies to computing systems designed to perform tasks usually reserved for human intelligence using logic, if-then rules, and decision trees. AI recognizes patterns from vast amounts of quality data providing insights, predicting outcomes, and making complex decisions. AI or artificial intelligence is not natural, and it is a simulation of how the human mind works. Much like a machine, the human brain controls all function of an organism. For example, how the mind managed active functions and another is autonomous responses that are basically knee-jerk responses. More technical know-how is available to give machines an iota of sentience, like self-driving cars and interactive robots that can manage without human assistance.

We have computers that are made up of electronic components, to make computer approximate what is human thinking. For example, every biological component has an electronic part that approximates how machine work in the real world. AI constructs have sensors that are similar to sight, hearing, touch, and feeling that is almost similar. Human operate on the same idea, but men are biological machines based on carbon, and AI is steel and electronics.

One requirement to make AI this responsive and smart is using lots of data, the human mind has the most data in nature. Teaching machines with AI, allow them to do tasks smartly, and better than humans. Here are several components of AI that are integral to making it work, as a simulacrum of the human mind. These six aspects are important to guide how artificial intelligence functions.

### **History of AI -**

Artificial intelligence was founded as an academic discipline in 1955, and in the years since has experienced several waves of optimism,<sup>[10][11]</sup> followed by disappointment and the loss of funding (known as an "AI winter"), followed by new approaches, success and renewed funding. For most of its history, AI research has been divided into sub-fields that often fail to communicate with each other.<sup>[15]</sup> These sub-fields are based on technical considerations, such as particular goals (e.g. "robotics" or "machine learning"), the use of particular tools ("logic" or artificial neural networks), or deep philosophical differences.

The main branches of Artificial Intelligence are:

- **Perception** - understanding images, audio, etc.
- **Reasoning** - answering questions from data
- **Planning** - inferring the required steps to reach a goal
- **Motion** - moving a robot in an environment
- **Natural language processing** - understanding human language

### **Types of AI**

Based on the functionality of AI-based systems, AI can be categorized into the following types:

1. Reactive Machines AI
2. Limited Memory AI
3. Theory Of Mind AI
4. Self-aware AI

### **Applications of Artificial Intelligence**


The applications for artificial intelligence are endless. The technology can be applied to many different sectors and industries. AI is being tested and used in the healthcare industry for dosing drugs and different treatment in patients, and for surgical procedures in the operating room.

Artificial intelligence also has applications in the financial industry, where it is used to detect and flag activity in banking and finance such as unusual debit card usage and large account deposits - all of which help a bank's fraud department. Applications for AI are also being used to help streamline and make trading easier. This is done by making supply, demand, and pricing of securities easier to estimate.

## Conclusion

Artificial Intelligence and Machine Learning are products of both science and myth. The idea that machines could think and perform tasks just as humans do is thousands of years old. The cognitive truths expressed in AI and Machine Learning systems are not new either. It may be better to view these technologies as the implementation of powerful and long-established cognitive principles through engineering.

We should accept that there is a tendency to approach all important innovations as a Rorschach test upon which we impose anxieties and hopes about what constitutes a good or happy world. But the potential of AI and machine intelligence for good does not lie exclusively, or even primarily, within its technologies. It lies mainly in its users.

 **Digital Learning Hub**  
Learn, Share, Collaborate

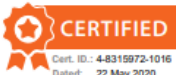
TATA CONSULTANCY SERVICES


*This is to certify that*  
**Davis Shashikant Patel**  
*has successfully completed*  
**Career Edge - Knockdown the Lockdown**  
*online course offered by TCS iON*

Start Date: 17 May 2020 | End Date: 22 May 2020

Topics:

- Communication Skills ■ Presentation Skills ■ Soft Skills ■ Career Guidance Framework ■ Resume Writing
- Group Discussion Skills ■ Interview Skills ■ Business Etiquette ■ Effective Email Writing ■ Telephone Etiquette
- Accounting Fundamentals ■ IT Foundational Skills ■ Overview of Artificial Intelligence\* (Source: NPTEL)

  
Cert. ID: 4-8315972-1016  
Dated: 22 May 2020

  
**Mehul Mehta**  
Global Delivery Head, TCS iON

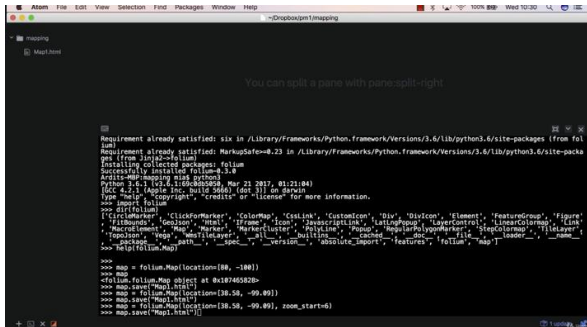
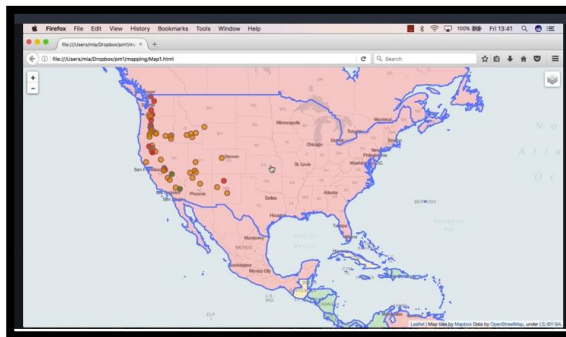
Act  
Got

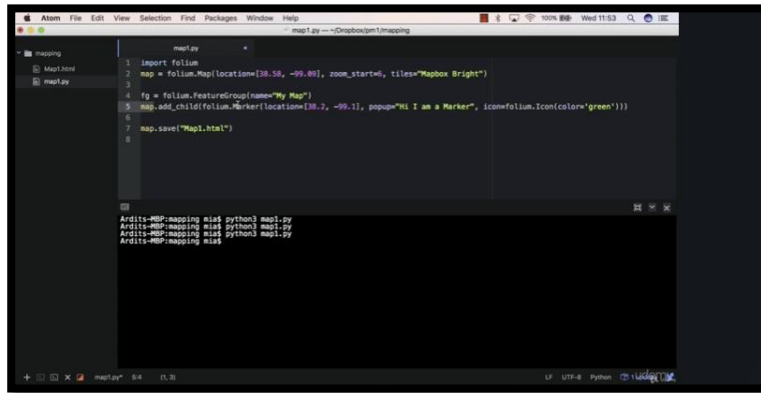
## DAILY ASSESSMENT FORMAT

<b>Date:</b>	<b>22/05/2020</b>	<b>Name:</b>	<b>Davis S. Patel</b>
<b>Course:</b>	<b>Python Course</b>	<b>USN:</b>	<b>4AL16EC045</b>
<b>Topic:</b>	<b>Application 2: Create Web maps with Python and Folium</b>	<b>Semester &amp; Section:</b>	<b>8<sup>th</sup> - A</b>
<b>GitHub Repository:</b>	<b>Davis</b>		

## AFTERNOON SESSION DETAILS

### Image of Session –

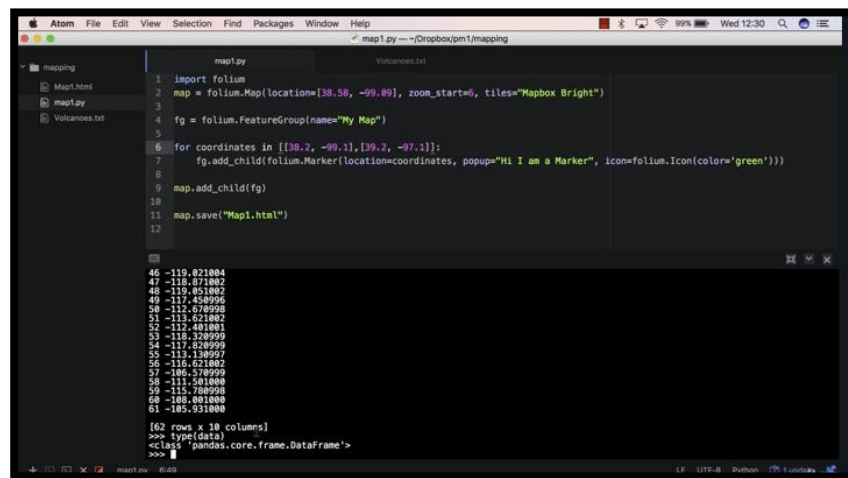




The screenshot shows the Atom editor with a file named `map1.py` open. The file contains the following Python code:

```
1 import folium
2 map = folium.Map(location=[38.58, -99.09], zoom_start=6, tiles="Mapbox Bright")
3
4 fg = folium.FeatureGroup(name="My Map")
5 map.add_child(folium.Marker(location=[38.2, -99.1], popup="Hi I am a Marker", icon=folium.Icon(color='green')))
6
7 map.save("Map1.html")
8
```

Below the editor, a terminal window is open, showing the command `python map1.py` being executed multiple times, with the output `Arduio-MP-mapping mis python map1.py` appearing repeatedly.



The screenshot shows the Atom editor with a file named `map1.py` open. The file contains the following Python code:

```
1 import folium
2 map = folium.Map(location=[38.58, -99.09], zoom_start=6, tiles="Mapbox Bright")
3
4 fg = folium.FeatureGroup(name="My Map")
5
6 for coordinates in [[38.2, -99.1], [39.2, -97.1]]:
7     fg.add_child(folium.Marker(location=coordinates, popup="Hi I am a Marker", icon=folium.Icon(color='green')))
8
9 map.add_child(fg)
10
11 map.save("Map1.html")
12
```

Below the editor, a terminal window is open, displaying a list of coordinates (latitude and longitude) for 62 rows. The output is as follows:

```
46 -119.071804
47 -118.871802
48 -119.071802
49 -117.458996
50 -112.678998
51 -113.621802
52 -112.481801
53 -118.328999
54 -117.828999
55 -114.18997
56 -116.621802
57 -186.578999
58 -111.581800
59 -115.788998
60 -185.831800
61 -185.931800
```

Below the list of coordinates, the terminal shows the command `[62 rows x 10 columns]` and the output `>>> type(data)` and `<class 'pandas.core.frame.DataFrame'>`.

## **REPORT –**

### **Application 2: Create Web maps with python and Folium**

Folium is a Python package built to bridge the data wrangling muscle of Python with Leaflet's easy-to-use JavaScript library for creating attractive, interactive web maps. The open source Leaflet is a highly popular web mapping tool due to its flexibility, with a healthy number of community-developed plug-ins further expanding its native capabilities. While Python is a robust programming language, with many packages contributing to geospatial analysis—Pandas, GeoPandas, Fiona, Shapely, Matplotlib, and Descartes to name a few—Folium differentiates itself through ease of use and the interactive potential of the final product. After some experimentation with the library, it did not take very long to produce a functional, albeit simple, web map with clustered point data, accompanied by popup windows. However, it was obvious that there is more to explore with Folium, as it plays well with many types of geospatial data, includes built-in functions and methods for producing choropleths, temporal visualizations, and allows for the marriage of the best of Python and Leaflet.

The code and resulting maps show a straightforward exercise in extracting the geographic coordinates (already matching Leaflet's default web-Mercator projection) and a few attribute values corresponding to warehouse/distribution centers in Pennsylvania's Lehigh Valley from an excel spreadsheet. The Pandas library was used to read the excel document and convert the desired information to a data frame. Folium was used to initialize a Leaflet map, add records as points with some stylization applied. This is brief code that could easily be added at the end of a more intensive spatial analysis using Python. It can provide a quick way to publish results in an interactive format without necessitating the use of JavaScript/html/CSS, or could serve as a jump start on more elaborate styling.



**Mapbox Bright and Stament Terrain are both types of base maps.**

```
tiles = "Mapbox Bright"
```

```
tiles = "Stamen Terrain"
```

## **HTML on Popups**

### **Code –**

```
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=%22s%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")
```

## **GeoJason Polygon Layer**

GeoJSON is a format for encoding a variety of geographic data structures [...]. A GeoJSON object may represent a region of space (a Geometry), a spatially bounded entity (a Feature), or a list of Features (a Feature Collection). GeoJSON supports the following geometry types: Point, Line String, Polygon, MultiPoint, Multiline String, MultiPolygon, and Geometry Collection. Features in GeoJSON contain a Geometry object and additional properties, and a Feature Collection contains a list of Features.

## **Choropleth Maps in Python**

A Choropleth Map is a map composed of colored polygons. It is used to represent spatial variations of a quantity. Plotly figures made with `px.scatter_geo`, `px.line_geo` or `px.choropleth` functions or containing `go.Choropleth` or `go.Scattergeo` graph objects have a `go.layout.Geo` object which can be used to control the appearance of the base map onto which data is plotted.

