
Introduction to **Decision Sciences** with Python

Vinayaka Gude, Ph.D.
Elon University

Agenda

01

Course Introduction & Schedule

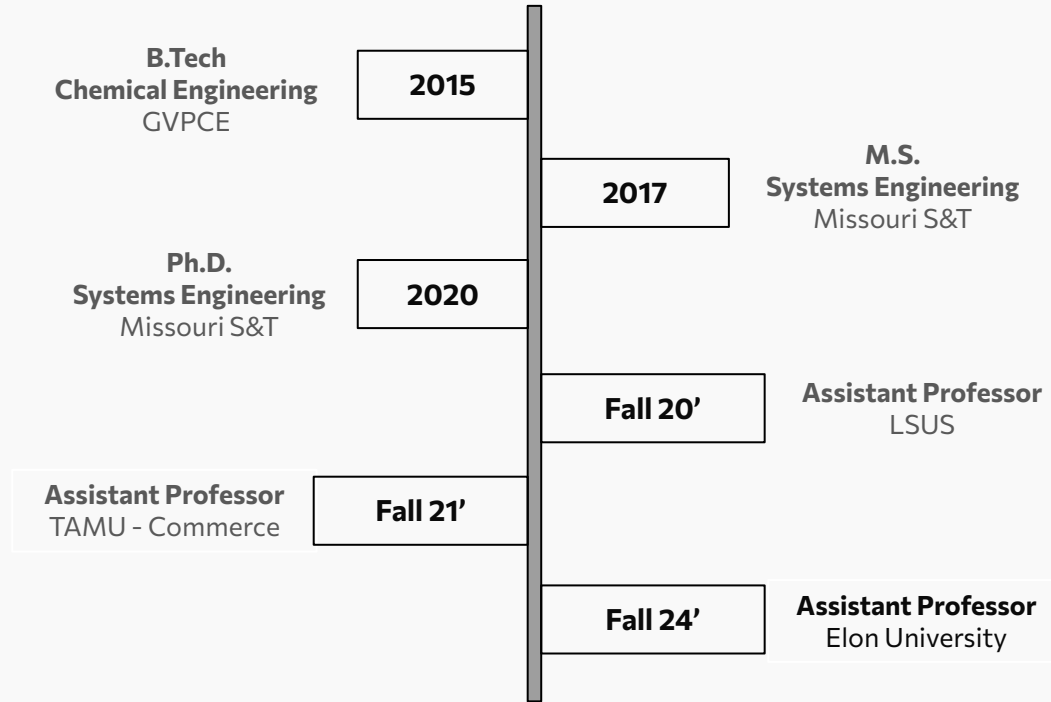
02

Data Analytics Introduction

03

Coding Demo: Data Structures

Personal Background



Class Schedule

Date	Lectures	Project	Deliverables
30-Dec	Introduction	Problem Identification	Assignment 1 (Due: 01/02)
31-Dec	Data Collection & Cleaning	Data Cleaning	
1-Jan	Descriptive Statistics	Descriptive Statistics & Visualization	
2-Jan	Machine Learning Overview	Machine Learning implementation	Assignment 2 (Due: 01/07)
3-Jan	Regression	Perform Regression	
6-Jan	Classification: Logistic Regression	Perform Classification	
7-Jan	Other Supervised Learning algorithms		Assignment 3 (Due: 01/09)
8-Jan	Reinforcement Learning	Practice	
9-Jan	Unsupervised Learning	Practice	
10-Jan	Deep Learning	Practice	

AI Policy

Use it with caution..



Make sure you can answer “yes” to these questions

Before you start

- ☐ My school and my professor allow the use of the AI tools I'm considering for this assignment
- ☐ I clearly understand when and how I can use AI for this assignment

Doing the work

- ☐ I am using my own thoughts, words and tone of voice
- ☐ I have checked sources generated by AI and properly cited any facts, statistics or quotes
- ☐ I have critically analyzed the AI output and identified any false, biased or harmful information
- ☐ I have documented where and how I used AI and cited that use according to my professor's expectations
- ☐ I have not used confidential, protected or copyrighted information

When the assignment is complete

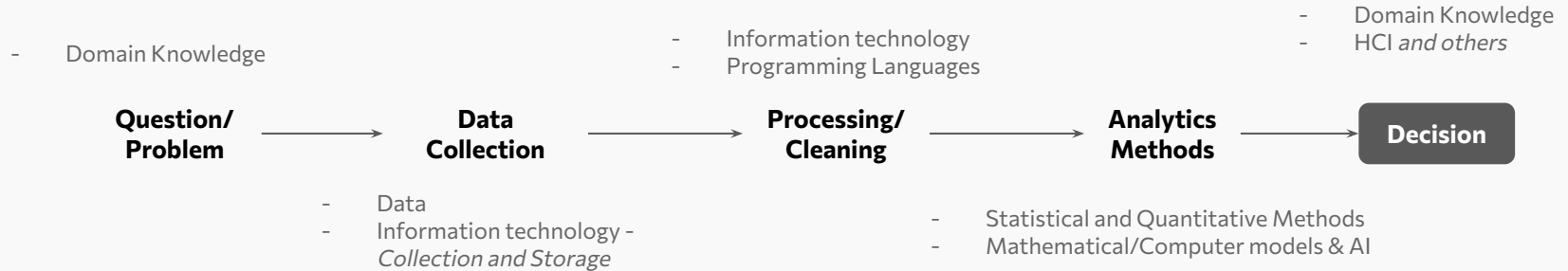
- ☐ I can explain my findings and demonstrate full understanding without the aid of AI
- ☐ I can prove what sources I used and how I verified the information

Student Guide to AI - available on Moodle

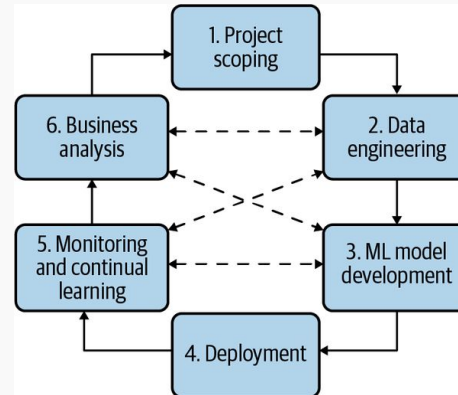
Introductions

- Name and major
 - What's your experience with the world of analytics and data science?
 - What was the memorable thing you did this year?
 - Be honest—how many hours did your phone keep you company yesterday?
 - What kind of problems are you interested in solving?
-

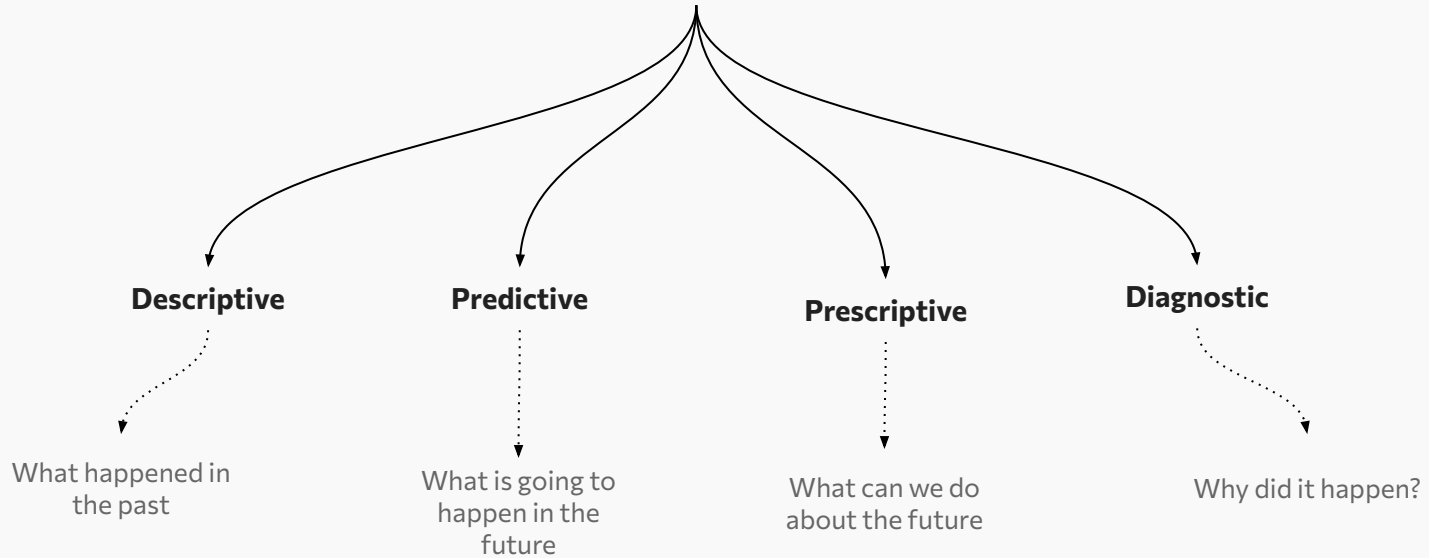
Data Analytics Process



Process of **transforming data into actionable insights** by identifying meaningful trends, making predictions and other techniques.



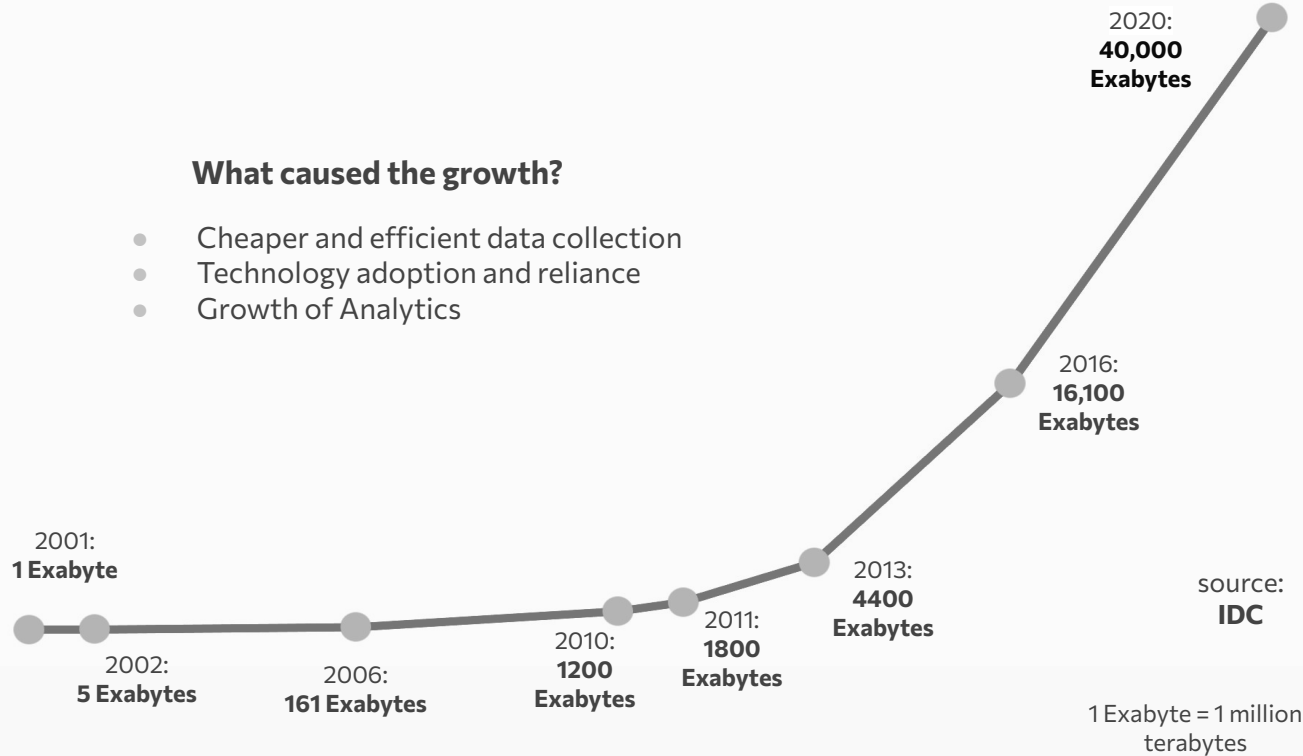
Types of Analytics



Growth in Analytics

What caused the growth?

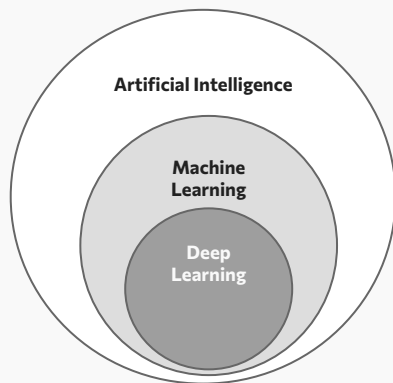
- Cheaper and efficient data collection
- Technology adoption and reliance
- Growth of Analytics



Data Roles

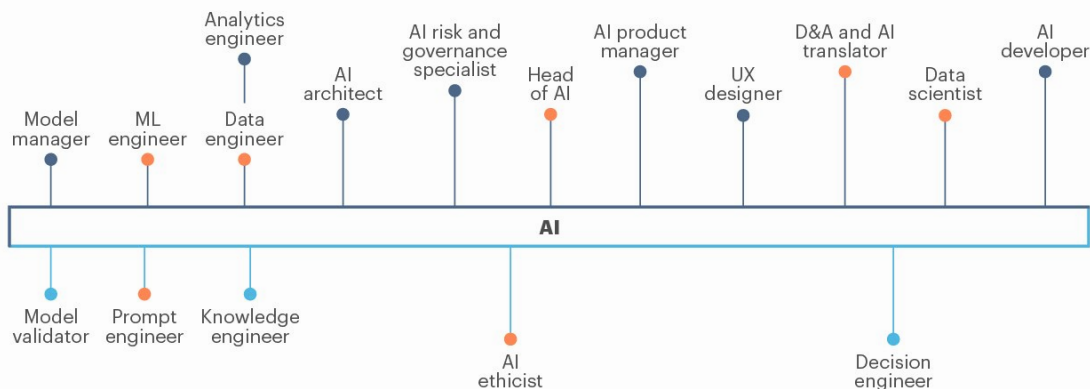
	Examine	Execute	Experiment	Explore	Explain
	<i>ETL Validation</i>	<i>Data Wrangling</i>	<i>Hypotheses Evaluation</i>	<i>Analysis</i>	<i>Communication</i>
Data Engineer					
Data Scientist					
Data Analyst					

Careers in Analytics after AI



Roles for AI

● AI roles ● Emerging AI roles ● Must-have AI roles



Source: Gartner
806919_C

Gartner

Let's Code!

Data Structures

[**https://github.com/vinayaka-elon/GITAM**](https://github.com/vinayaka-elon/GITAM)

Lists

Collection of objects enclosed in **[square brackets]**

allows us to put many values in a **single “variable”**

`students = ['Sam', 'Ram', 'Lily']`

elements in a list are separated by ,

Lists can include letters, special characters, spaces, digits and even **other lists**

`data = [4, [5,6], 7]`

Heterogeneous data types

Lists can also be empty

`data = []`

Lists Indexing

0	1	2	4	5
'Sam'	'Ram'	'Lily'	'Dan'	'Glen'

```
students[1] → "Ram"
```

```
students[-1] → "Dan" .....> students[4]
```

```
students[2:4] → ['Lily', 'Dan', 'Glen']
```

```
students[:3] → ['Sam', 'Ram', 'Lily']
```

Lists functionality

```
students = [ 'Sam', 'Ram', 'Lily', 'Dan', 'Glen' ]
```

```
len(students) → 5
```

Length

```
students + ["Ted"] →  
[ 'Sam', 'Ram', 'Lily', 'Dan', 'Glen', 'Ted' ]
```

Concatenation

```
[1,2]*5 → "[1,2,1,2,1,2,1,2,1,2]"
```

Repeat

```
"Ram" in students → True
```

Sublist test

List Methods

append()	Adds an element at the end of the list
len()	Returns the length of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the first item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

range(5) → [0,1,2,3,4]

range(0,10,2) → [0,2,4,6,8]

Lists are **mutable**

Arrays

Like lists, **arrays** are **ordered**, **mutable**, enclosed in **square brackets**, and able to store **non-unique** items.

Numpy → supports multiple data types

Difference between an array and a list?

- Arrays **need to be declared**. Lists don't
 - Arrays are **memory efficient**
 - Arrays are great for **numerical operations**
-

Creating Arrays

```
students = [ 'Sam', 'Ram', 'Lily', 'Dan', 'Glen' ]
```

```
import numpy  
students_array = np.array(students)
```

```
Zeros = np.zeros(2,1)
```

```
[[0]  
 [0]]
```

```
constants = np.full((2,1),10)
```

```
[[10]  
 [10]]
```

```
Ones = np.ones(2,1)
```

```
[[1]  
 [1]]
```

```
randoms = np.random.rand((2,1))
```

```
[[0.14]  
 [0.23]]
```

```
eye = np.eye(2)
```

```
[[1 0]  
 [0 1]]
```

More an arrays

Use list methods and then convert into an array

Indexing → similar to a list

Mathematical operations> **A+B; A-B; A*B; A/B**

Mathematical functions

<https://numpy.org/doc/stable/reference/routines.math.html>

Dictionaries

store a **mapping** between a **set of keys** and a **set of values**

(Can be any data type)



Immutable → integers, string, tuple

```
students = { 'Sam': 'A', 'Ram': 'A', 'Lily': 'C' }
```

A single dictionary can store **multiple data types**

can **define, modify, view, lookup** or **delete** the
key-value pairs in the dictionary

Also known as **hash tables** and **associative arrays**

Working with Dictionaries

```
students = { 'Sam': 'A', 'Ram': 'A', 'Lily': 'C' }
```

Accessing a
dictionary

```
students['Sam'] → 'A'
```

```
students['A'] → KeyError
```

Updating a dictionary

.....>

```
students['Sam'] = 'B'
```

Keys must be **unique**

.....> Assigning to an existing key replaces its value

Dictionaries are **unordered**

Dictionary Methods

```
del students['Sam']
```

```
students.clear()
```

```
students.keys() → ['Ram', 'Lily']
```

```
students.values() → ['A', 'C']
```

```
students.items() → [('Ram', 'A'), ('Lily', 'C')]
```

Practice Problems

1. Given the list `fruits = ["apple", "banana", "cherry", "date", "elderberry"]`, retrieve the first element.
 2. Using the list `numbers = [10, 20, 30, 40, 50, 60, 70]`, create a new list containing only the first three elements.
 3. Combine the lists `a = [1, 2, 3]` and `b = [4, 5, 6]` into a single list.
 4. Reverse the list `colors = ["red", "blue", "green", "yellow"]` without using loops or functions.
 5. Replace the second element of the list `names = ["John", "Jane", "Alice"]` with "Bob".
 6. Create a NumPy array `arr` with the elements `[1, 2, 3, 4, 5]`.
 7. Using the array `arr = np.array([10, 20, 30, 40, 50])`, retrieve the element at index 3.
 8. Change the last element of the array `arr = np.array([5, 10, 15, 20])` to 25.
 9. Given the dictionary `person = {"name": "Alice", "age": 30, "city": "New York"}`, access the value associated with the key "age".
 10. Add a new key-value pair `"occupation": "Engineer"` to the dictionary `person = {"name": "Alice", "age": 30}`.
-

Identify a Problem

Each group brainstorm and identify a business problem they want to address.

- Identify the stakeholders
- Define the use cases and possible impact

Resume suggestion

ATS Evaluation

<https://skillsyncer.com/>
<https://www.onetonline.org/>

Thank you!

Any questions?

gude.vinayaka@outlook.com