Applied Analytics & Predictive Modeling

Spring 2021

Lecture-1

Lydia Manikonda

manikl@rpi.edu

Some of the slides adapted from Intro to Data Mining Tan et al. 2nd edition

Agenda

- Course logistics
- Instructor Info
- Syllabus

- Introduction to Data Mining
- Python basics – variables, data structures
- Colab – Jupyter notebook environment

Course Logistics

- Lectures: Every Mondays & Thursdays 6:55 pm to 8:15 pm
- Location: PITTS 4206
- Webex: https://rensselaer.webex.com/meet/manikl

- Website: https://predictivemodeling.github.io/
- Piazza for course-related discussions

Instructor

- Assistant Professor in Lally School of Management

- PhD in Computer Science
- AI & Machine Learning with a focus on Social Media Analytics

- Office hours: Tuesday 2 pm to 4 pm via webex
- Office Location: PITTS 1212

Syllabus

- Python basics
- Data cleaning and pre-processing
- Data analysis including dimensionality reduction
- Logistic regression
- Decision trees
- K-Nearest Neighbor algorithm
- Association rules, Market basket analysis
- Cluster analysis including NLP applications

…

Grading

MGMT 6160 (3 credits)

Component	Weight
Midterm exam	20%
Project	30%
Assignments	30%
Final Exam	10%
Class participation	5%
Research translation exercise	5%

MGMT 4963 (4 credits)

Component	Weight
Midterm exam	25%
Project	30%
Assignments	30%
Final Exam	10%
Class participation	5%

Tentative* A (93-100); A- (86-92); B+ (82-85); B (78-81); B- (74-77); C+ (70-73); C (66-69); C- (60-65); F (below 60)

About you…

- Please take the poll..

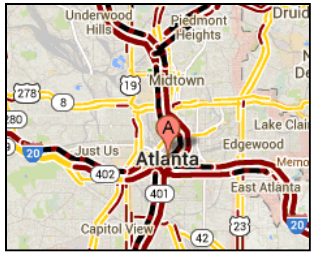
Overview of Data Mining

Large-Scale Data is Everywhere

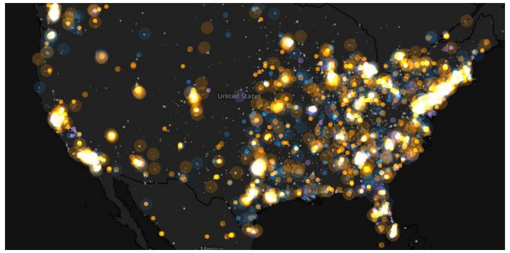
- There has been enormous data growth in both commercial and scientific databases due to advances in data generation and collection technologies
- New mantra
  - Gather whatever data you can whenever and wherever possible.
- Expectations
  - Gathered data will have value either for the purpose collected or for a purpose not envisioned.

Cyber Security

E-Commerce

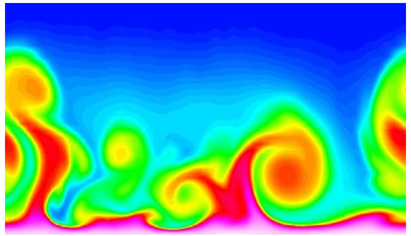


Traffic Patterns



Social Networking: Twitter

Sensor Networks



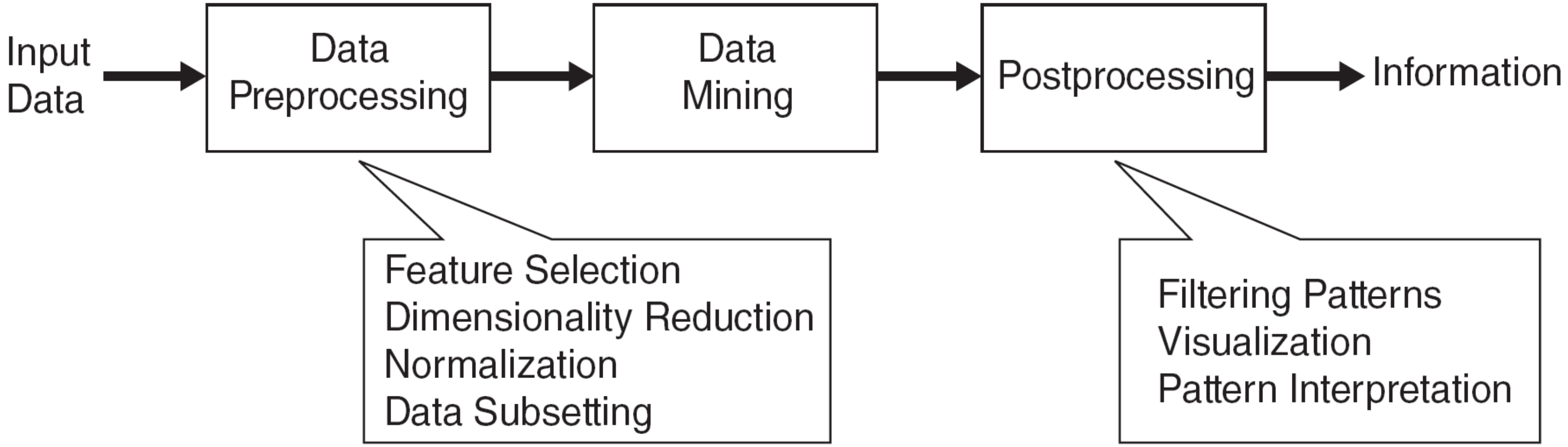
Computational Simulations

Why Data Mining? Commercial Viewpoint

- Lots of data is being collected and warehoused
  - Web data
    - Yahoo has Peta Bytes of web data
    - Facebook has billions of active users
  - purchases at department/ grocery stores, e-commerce
    - Amazon handles millions of visits/day
  - Bank/Credit Card transactions
- Computers have become cheaper and more powerful
- Competitive Pressure is Strong
  - Provide better, customized services for an edge (e.g. in Customer Relationship Management)

What is Data Mining?

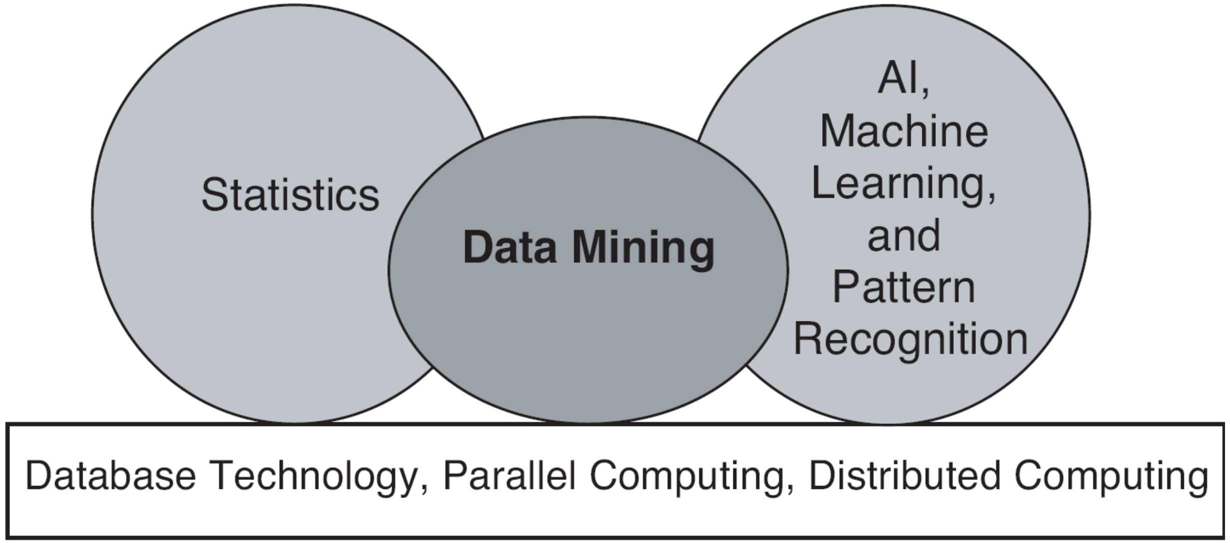
- Many Definitions
  - Non-trivial extraction of implicit, previously unknown and potentially useful information from data
  - Exploration & analysis, by automatic or semi-automatic means, of large quantities of data in order to discover
    meaningful patterns



Origins of Data Mining

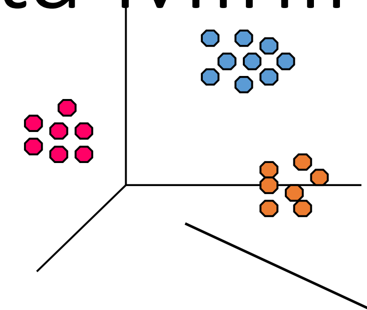
- Draws ideas from machine learning/AI, pattern recognition, statistics, and database systems

- Traditional techniques may be unsuitable due to data that is
  - Large-scale
  - High dimensional
  - Heterogeneous
  - Complex
  - Distributed


Statistics
Data Mining
AI, Machine Learning, and Pattern Recognition
Database Technology, Parallel Computing, Distributed Computing

- A key component of the emerging field of data science and data-driven discovery

Data Mining Tasks

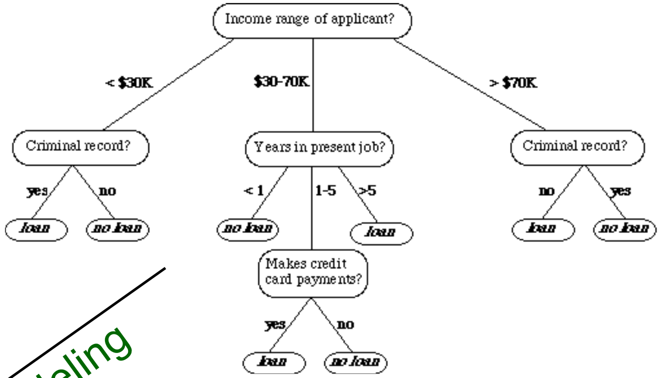


Clustering

Data

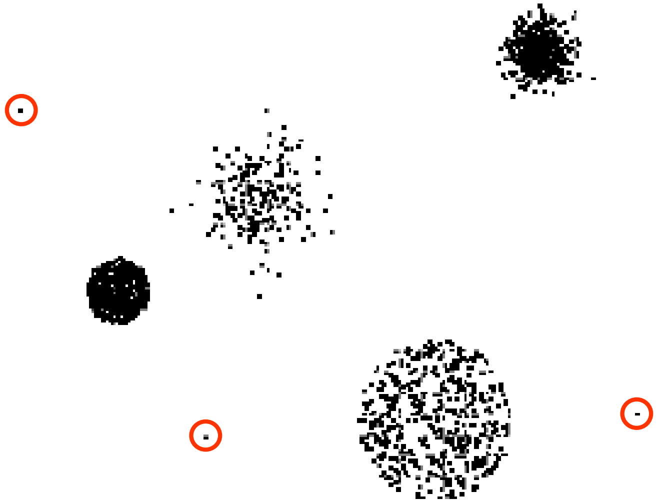
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes
11	No	Married	60K	No
12	Yes	Divorced	220K	No
13	No	Single	85K	Yes
14	No	Married	75K	No
15	No	Single	90K	Yes

Predictive Modeling



Association Rules

Anomaly Detection



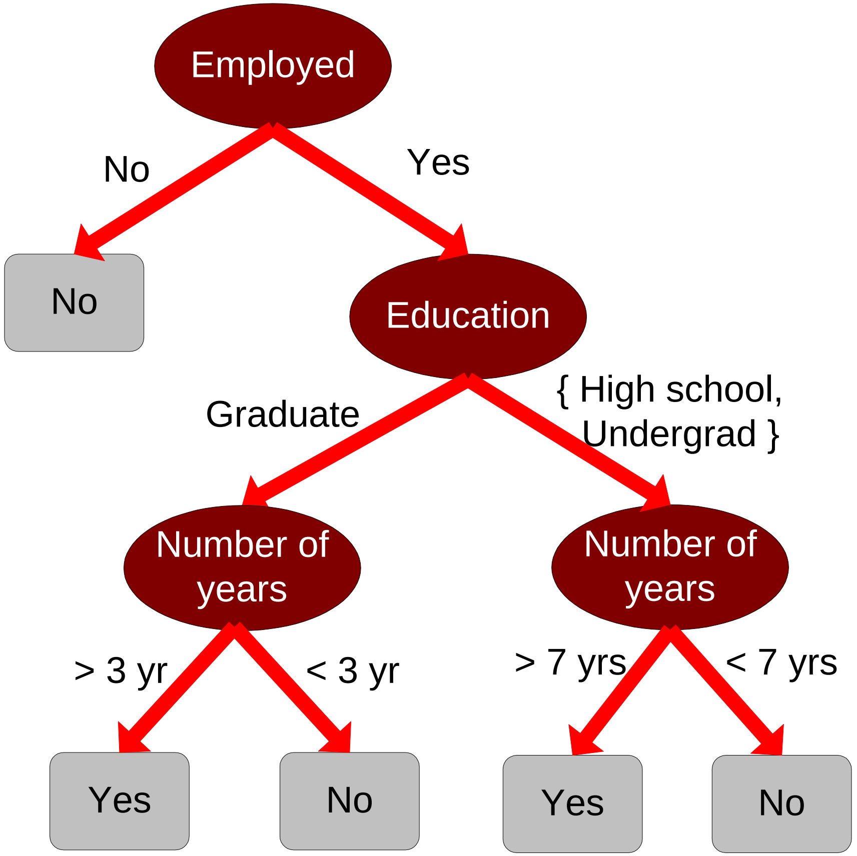
Predictive Modeling: Classification

- Find a model for class attribute as a function of the values of other attributes

Class

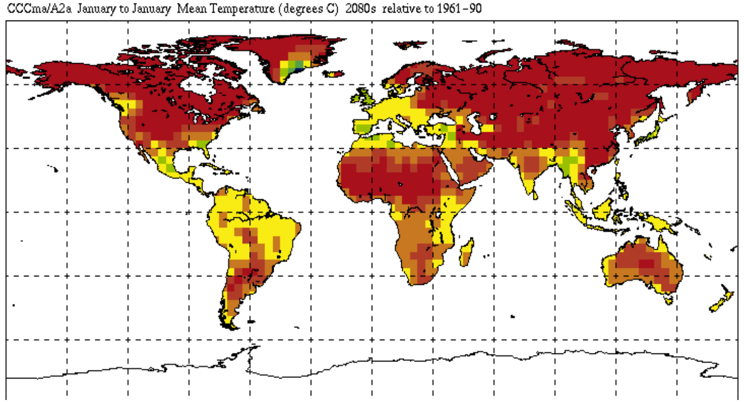
Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Graduate	5	Yes
2	Yes	High School	2	No
3	No	Undergrad	1	No
4	Yes	High School	10	Yes
…	…	…	…	…

Model for predicting credit worthiness



Great Opportunities to solve Society’s Major Problems

Improving health care and reducing costs


CCCma/A2a January to January Mean Temperature (degrees C) 2080s relative to 1961–90

Predicting the impact of climate change

Finding alternative/ green energy sources

Reducing hunger and poverty by
increasing agriculture production

What is NOT Data Mining?

- What is not Data Mining?
  - Look up phone number in phone directory
  - Query a Web search engine for information about “Amazon”

- What is Data Mining?
  - Certain names are more prevalent in certain US locations (O’Brien, O’Rourke, O’Reilly… in Boston area)
  - Group together similar documents returned by search engine according to their context (e.g., Amazon rainforest, Amazon.com)

Python fundamentals

Basics, loops, conditionals, functions, packages

Basics

Language introduction, setup, variables, data structures

Python Language Introduction

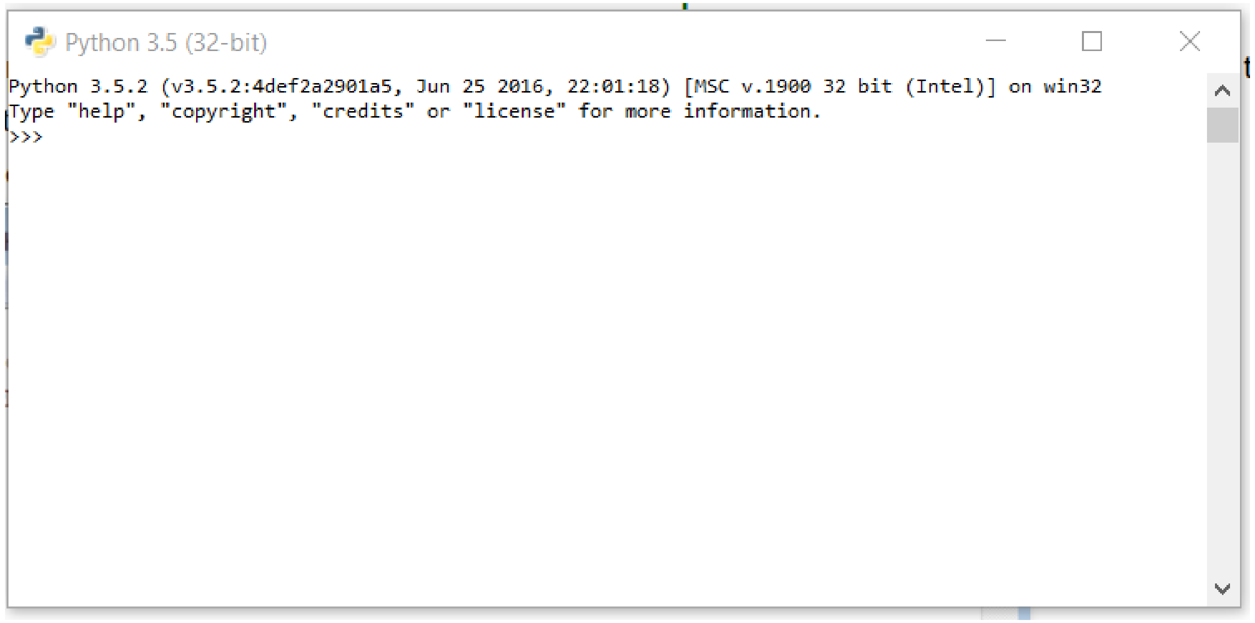
- General-purpose, high level programming language.
- Designed by Guido Van Rossum in 1991

- Main emphasis on
  - Code readability
  - Simple syntax

- 2 major versions – **Python 2** and **Python 3**

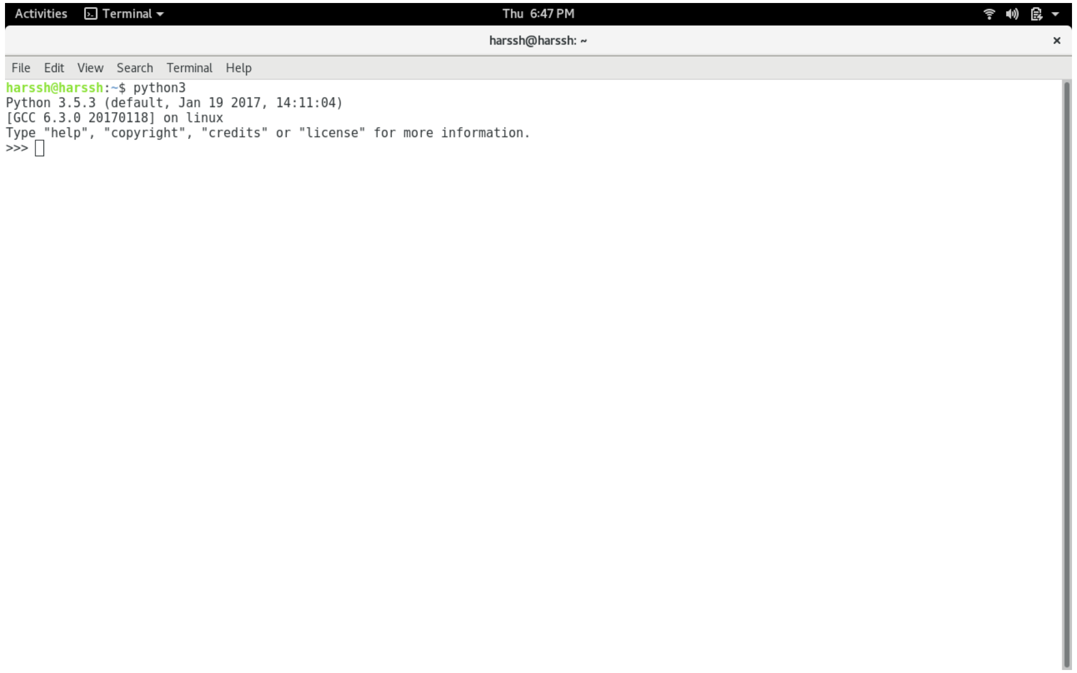
Finding an interpreter

- Windows



```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- Unix/Linux



```
harssh@harssh:~$ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170118] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

First program in Python

```
>> #Begins -- Comments
>> print(“Hello World”)
>> #Ends – Comments
```

# is used for single line comment in Python

""" this is a comment """ is used for multi line comments

Variables and Data Structures

- In programming languages such as C, C++ or C#, you need to declare the **type of variables** exclusively.
  - Data types can be int, float, char, String, etc.
- Python – take a variable and the value assigned to it automatically tells the data type.

```
>> myVar = 2 #int
>> print(myVar)

>> myVar2 = 2.5 #float
>> print(myVar2)

>> myVar3 = “Hello World!” #string
>> print(myVar3)
```

Data Structures

- Create a variable and assign any value you want!

- Python has 4 types of inbuilt data structures
- **List**
- **Dictionary**
- **Tuple**
- **Set**

List

- Most basic data structure in Python programming language.
- Mutable data structure
  - Elements of this list can be altered after creating the data structure

1. append() – used to add elements in the list
2. insert() – used to add elements in the list at a certain index till the last element

List

append()

```
>> #Create an empty list
>> list1=[]

>> #Append elements to the list
>> list1.append(2)
>> list1.append(4.5)
>> list1.append(“four”)

>> print(list1)
```

insert()

```
>> list1 = [1, 2, 3, 4, 5]
>> list1.insert(5, 10)
>> print(list1)


>> list1.insert(1,10)
>> list1.insert(8,20)
>> print(list1)
```

Example – Mixing append(), insert() and remove()

```
>> list1=[1,2,3,4,5]
>> list1.insert(5,12)
>> list1.insert(1,14)
>> print(list1) # [1, 14, 2, 3, 4, 5, 12]

>> list1.insert(8,20)
>> print(list1) # [1, 14, 2, 3, 4, 5, 12, 20]

>> list1.append(11)
>> print(list1) # [1, 14, 2, 3, 4, 5, 12, 20, 11]

>> list1.pop(5) #removes the element at index 5; if only pop() – removes the last element
>> print(list1)
```

List – Exercise

1. Create a list of size 5 containing 10,20,30,40,50 – one at a time by using the method insert().
2. Print the list.
3. Remove element from index ‘3’ and print the list.
4. Remove the last element and print the list.

Dictionary

- An unordered collection of data values in Python.
- It is used to store data values like a map.
- Unlike other Data Types that hold only single value as an element, Dictionary holds <key:value> pair.
- Dictionary values can be of any datatype – can be duplicated no repeated keys.

Dictionary

```
>> diction1={}
>> print(diction1)

>> diction1 = {1: ‘First’, 2: ‘Python’, 3: ‘Dictionary’}
>> print(diction1)

>> diction1 = {1: ‘First’, 2: [1,2,3,4]}
>> print(diction1)
```

Dictionary

```
>> diction1={}
>> diction1[0]=2
>> diction1[1]=4
>> diction1[2]=“Hello”
>> diction1[“3”]=“It is possible”
```

Dictionary – Exercise

1. Create a dictionary (d1) of size 10 where the keys are from 1 to 10 and their associated values are twice the key value.

2. For example, d1[3]=6 because the key is 3 and the value is twice the value of key which is 2*3.

Tuple

- Tuple is a collection of Python objects much like a list.
- The sequence of values stored in a tuple can be of any type, and they are indexed by integers.
- The important difference between a list and a tuple is that **tuples are immutable**.

Tuple

```
>> tuple1=()
>> print(tuple1)

>> tuple1=(1,2,3,4,5)
>> print(tuple1)

>> tuple1=(‘hello’, ‘world’)
>> print(tuple1)
```

Tuple

```
>> list1=[1,2,3,4,5]
>> list1[1]=3
>> print(list1)

>> list1=[7,6,5,4,3,2,1,0]
>> print(list1)

>> mytuple=(0,1,2,3,4,5,6,7)
>> print(mytuple)
>> mytuple[1]=3
```

Concatenate tuples

```
>> Tuple1 = (0, 1, 2, 3)
>> Tuple2 = (‘hello’, ‘world’)
>> Tuple3 = Tuple1 + Tuple2
>> print(Tuple3)
```

Tuple – Exercise

1. Create a tuple t1 that contains 1,2,3,4
2. Create a tuple t2 that contains ‘I’, ‘love’, ‘analytics’
3. Concatenate t1 and t2 to form t3 and print t3.

Set

- Set is an unordered collection of data type that is iterable, mutable and has no duplicate elements.
- Highly optimized method compared to list because it is very easy to check whether an element is present or not.

Set

```
>> set1 = set()
>> print(set1)

>> set1 = set(“Predictive”)
>> print(set1)

>> s1=“Predictive”
>> set1 = set(s1)
>> print(set1)

>> set1=set([“I”, “love”, “analytics”])
>> print(set1)
```

Set – Exercise

- S1=“Predictive”
- Create a set that has only one element which is S1. In other words, create a set that is {“Predictive”}.

Take input from the user

- input() function is used to take input from the user

```
>> # Python program to get input from user
>> name = input("Enter the course name: ")
>> # user entered the name ‘PredictiveModel'
>> print("I registered for ", name)
```

User input – Exercise

1. Taking 2 integers as input from the user and print their product.

```
>> num1 = int(input("Enter num1: "))
>> num2 = int(input("Enter num2: "))

>> num3 = num1 * num2
>> print("Product is: ", num3)
```

Revising all the concepts – Exercises

1. Given a list of keywords, create a dictionary of the keywords and print the keys and values.

   Input: Keywords = [‘hello’, ‘I’, ‘am’, ‘fine’]

   Dictionary: {‘hello’: 1, ‘I’:1, ‘am’:1, ‘fine’:1}

Today we learnt..

- Course logistics
- Definition of data mining and what is not data mining
- Python basics
  - Variables
  - Data structures
  - Exercises