

Data Mining and Accounting Analytics -Introduction

1

Dr. Yi Long (Neal)

Most contents (text or images) of course slides are from the following textbook
Provost, Foster, and Tom Fawcett. Data Science for Business: What you need to
know about data mining and data-analytic thinking. " O'Reilly Media, Inc.", 2013

Outline

- Course Introduction
- Data Mining Applications Tasks
- Data Mining Tools (Python)

Big Data

40 ZETTABYTES

[43 TRILLION GIGABYTES]

of data will be created by 2020, an increase of 300 times from 2005

6 BILLION PEOPLE
have cell phones

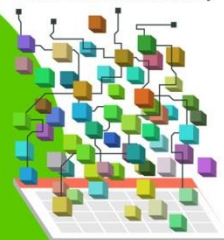


WORLD POPULATION: 7 BILLION

Volume SCALE OF DATA



It's estimated that
2.5 QUINTILLION BYTES
[2.3 TRILLION GIGABYTES]
of data are created each day



Most companies in the U.S. have at least
100 TERABYTES
[100,000 GIGABYTES]
of data stored



The FOUR V's of Big Data

From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can these massive amounts of data be used?

As a leader in the sector, IBM data scientists break big data into four dimensions: **Volume, Velocity, Variety and Veracity**

Depending on the industry and organization, big data encompasses information from multiple internal and external sources such as transactions, social media, enterprise content, sensors and mobile devices. Companies can leverage data to adapt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.

By 2015
4.4 MILLION IT JOBS
will be created globally to support big data,
with 1.9 million in the United States



As of 2011, the global size of data in healthcare was estimated to be

150 EXABYTES
[161 BILLION GIGABYTES]



**30 BILLION
PIECES OF CONTENT**
are shared on Facebook
every month



Variety DIFFERENT FORMS OF DATA



By 2014, it's anticipated there will be
**420 MILLION
WEARABLE, WIRELESS
HEALTH MONITORS**

**4 BILLION+
HOURS OF VIDEO**
are watched on
YouTube each month



400 MILLION TWEETS
are sent per day by about 200
million monthly active users



The New York Stock Exchange captures

**1 TB OF TRADE
INFORMATION**
during each trading session



Velocity ANALYSIS OF STREAMING DATA



Modern cars have close to
100 SENSORS
that monitor items such as
fuel level and tire pressure

By 2016, it is projected
there will be

**18.9 BILLION
NETWORK
CONNECTIONS**

— almost 2.5 connections
per person on earth



**1 IN 3 BUSINESS
LEADERS**

don't trust the information
they use to make decisions



Poor data quality costs the US
economy around
\$3.1 TRILLION A YEAR



**27% OF
RESPONDENTS**

in one survey were unsure of
how much of their data was
inaccurate

Veracity UNCERTAINTY OF DATA

Data is Valuable

Leaders

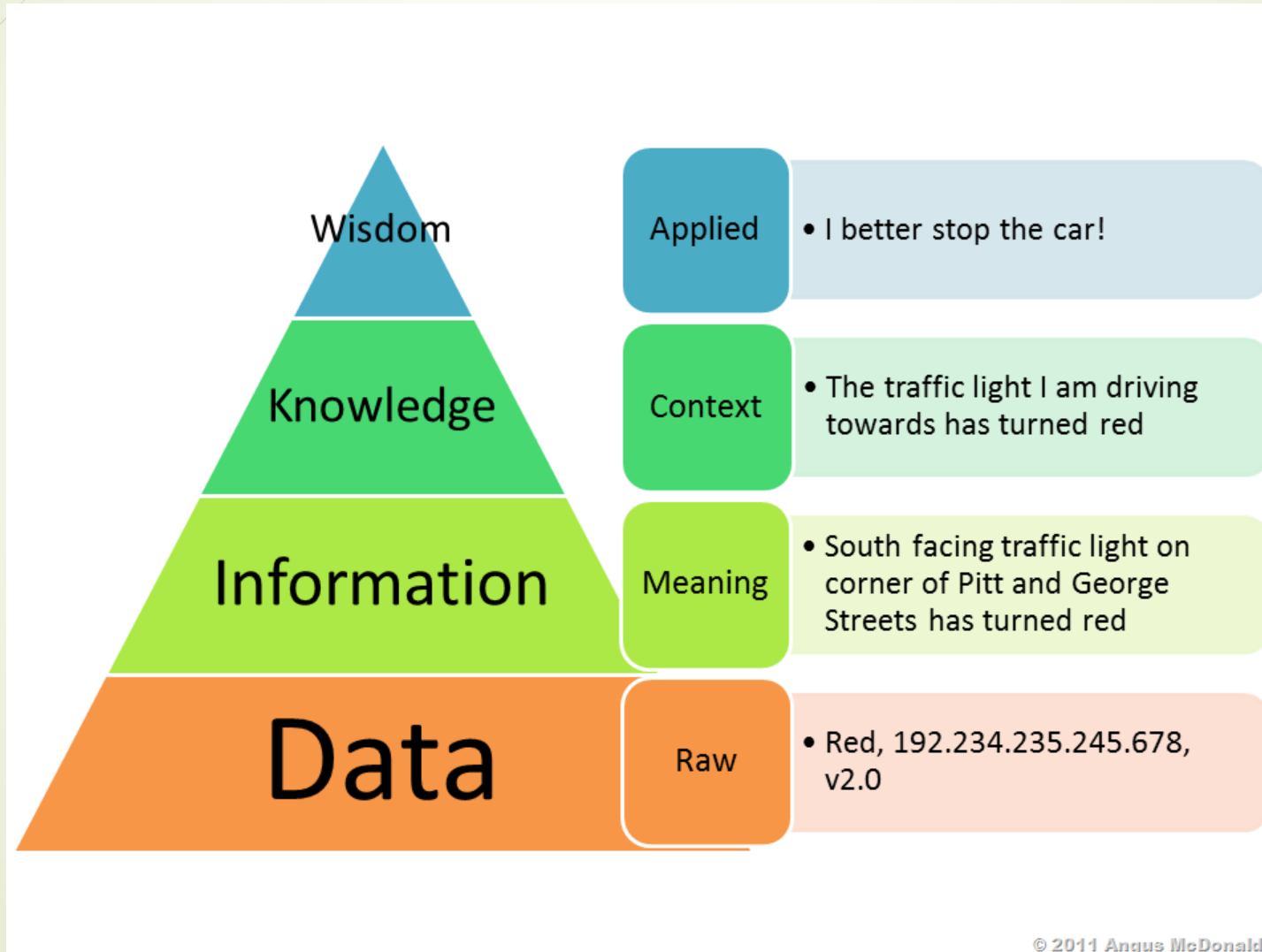
May 6th 2017 edition >

Regulating the internet giants

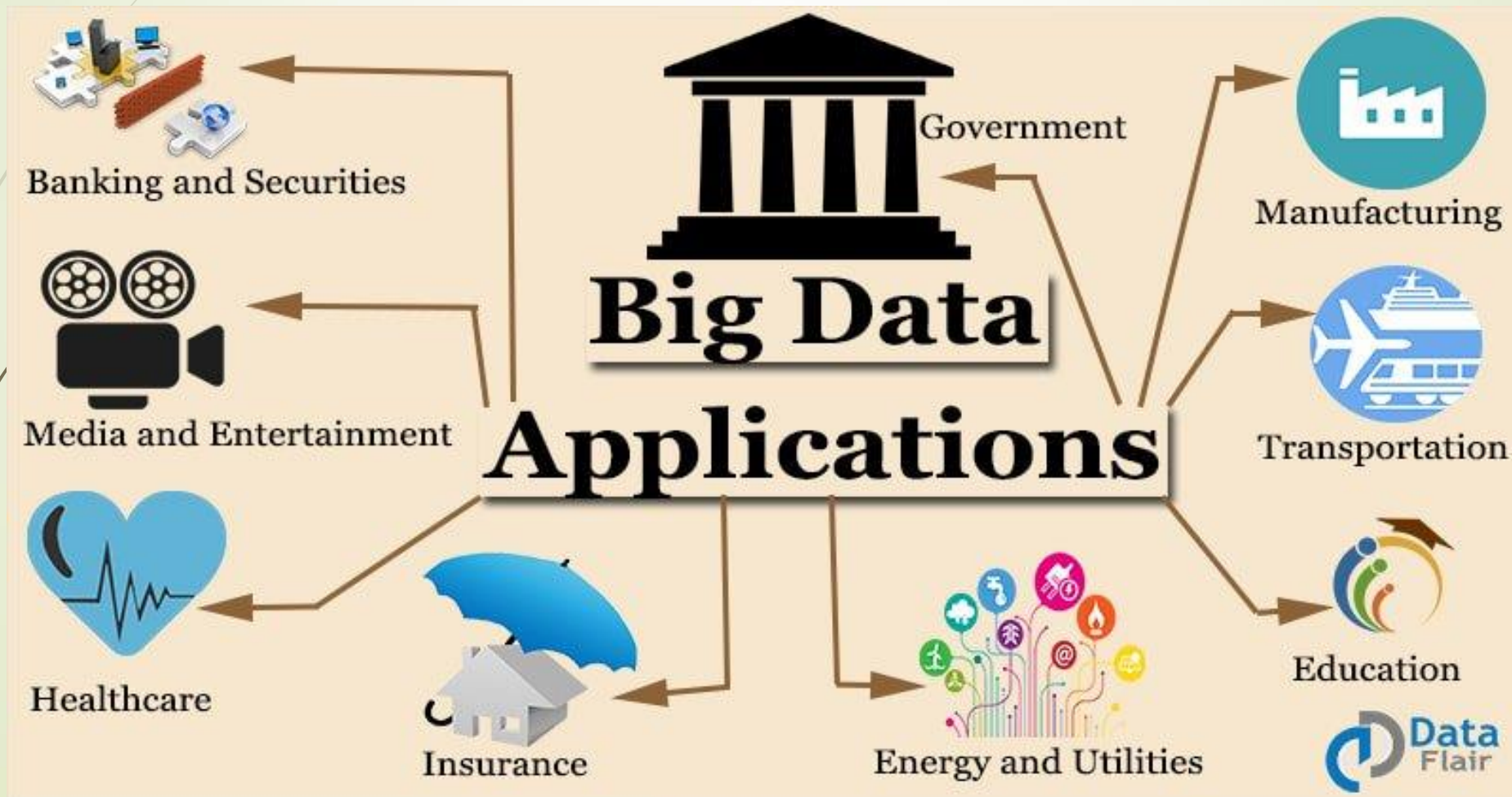
The world's most valuable resource is no longer oil, but data



Why Data Mining?

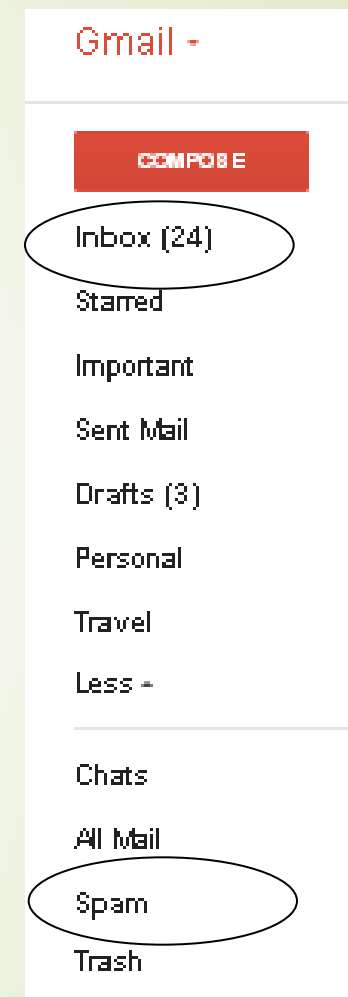


Applications of Big Data



Internet Companies

- Google service
 - ✓ Search engine
 - ✓ Spam filter
 - ✓ Advertising
- Alibaba
 - ✓ Advertising
 - ✓ Credit scoring



Other Applications

- Financial industry: banks and fund companies
 - ✓ Credit scoring
 - ✓ Fraud detection
 - ✓ Quantitative Trading
 - ✓ AML
 - ✓ KYC
- Weather forecast
- Smart healthcare
- ...

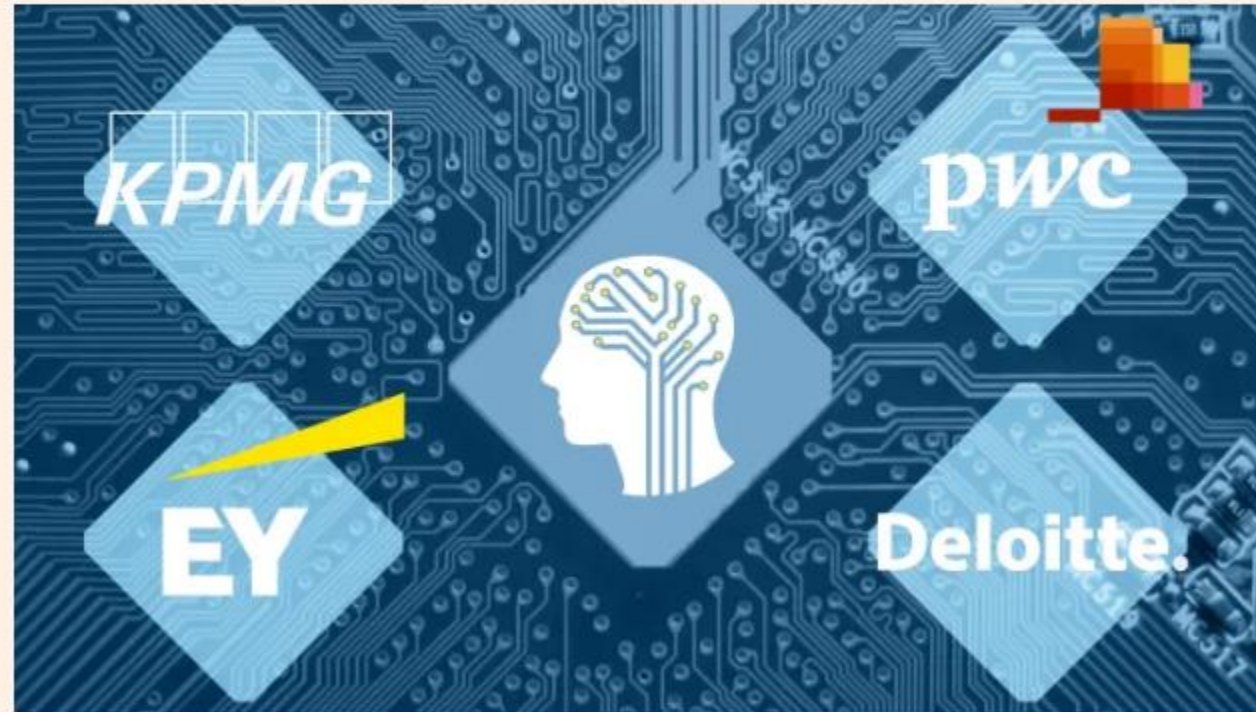
Why Now?

- The large-scale electronically recorded data
- The increasingly strong computational power: CPU, GPU
 - ✓ Processing power has been doubling every 1.5 years or so ((Moore's law)
- Developed algorithms/ technology:
 - ✓ Machine learning (deep learning), big data technology
- The high speed network infrastructures
 - ✓ Data can be transferred easily between collection, storage, and use
 - ✓ Cloud computing
- Cheap and reliable storage systems: HDD, SSD
 - ✓ A \$100 disk today has 1,000,000x more capacity than 30 years ago

Embrace The Future

Auditing to be less of a burden as accountants embrace AI

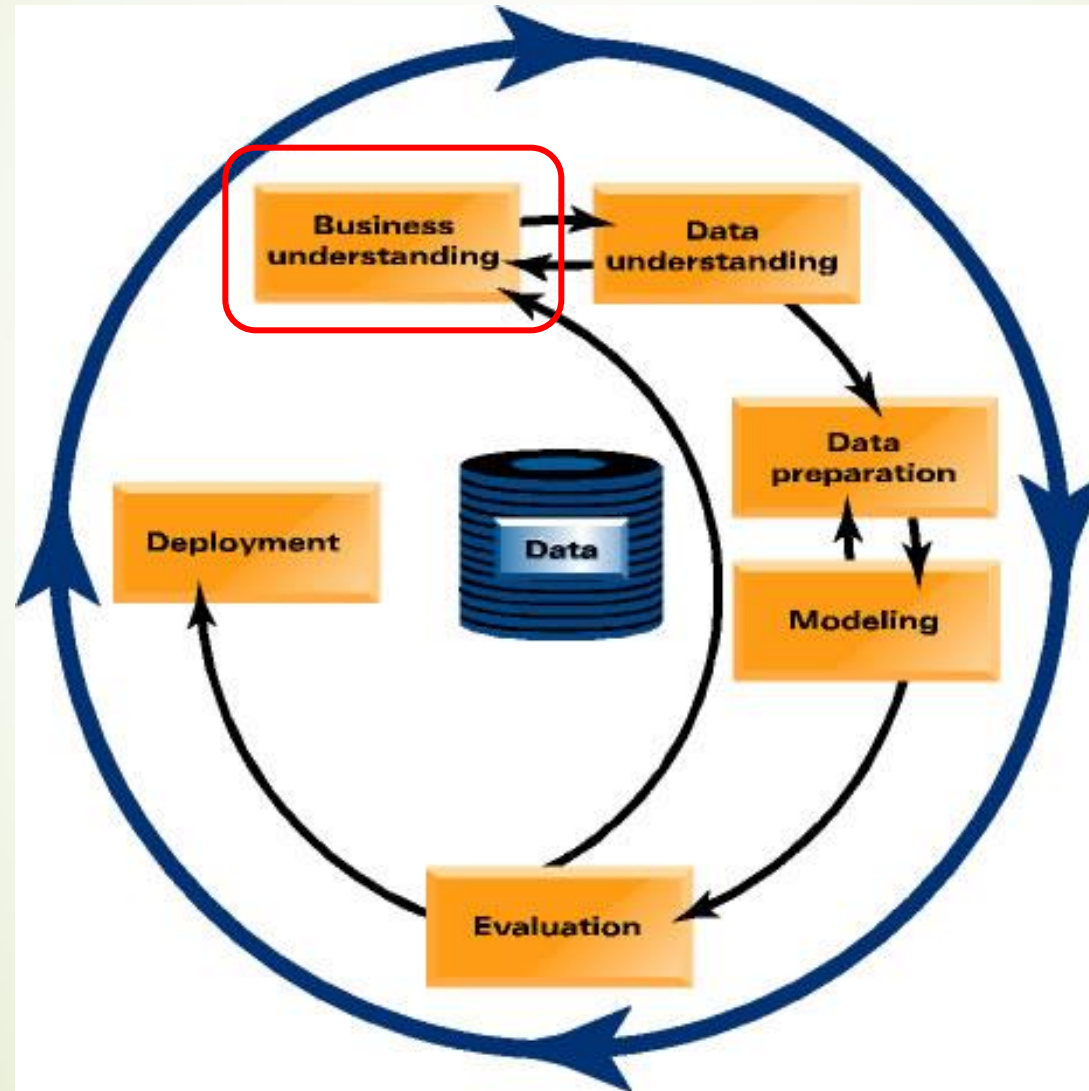
New technology helps firms automate mundane and inefficient processes



Better technology can improve the quality of audit work by carrying out tasks faster than a human could © FT montage

<https://www.ft.com/content/0898ce46-8d6a-11e7-a352-e46f43c5825d>

Cross Industry Standard Process

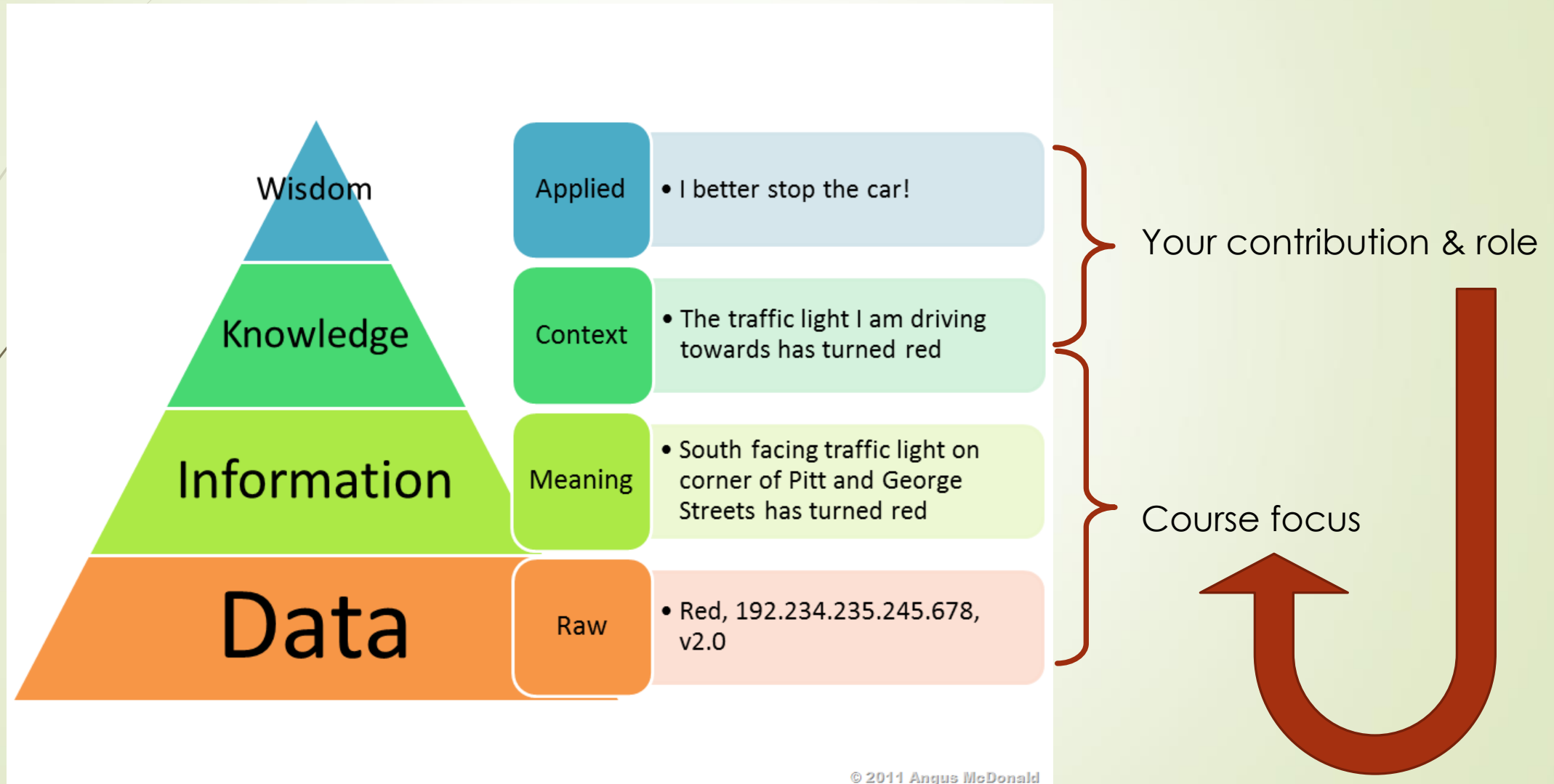


Solve Business Problems

3. **Don't expect stakeholders to always (or ever) be able to define the problem.** In my opinion, this is the number one most important skill for a Data Scientist above any technical expertise — the ability to clearly evaluate and define a problem. Most business stakeholders have problems but haven't thought about them long enough to be able to define the process behind them. This is the place where you will make Machine Learning and AI work for your organization — by deciphering the needs of the business into a process where Data Science can be applied effectively.

<https://towardsdatascience.com/dont-do-data-science-solve-business-problems-6b70c4ee0083>

Business Data Analytics



Learning Outcomes(3-fold)

1. Approach business problems data-analytically
 - ✓ Think carefully & systematically about whether & how data can improve performance
2. Be able to interact competently on the topic of data mining for business analytics
 - ✓ Know the basics of data mining processes, techniques, & concepts well enough
3. Receive hands-on experience on mining data
 - ✓ You should be able to follow up on ideas or opportunities

Final goal: compose data-analytic tools and business applications to provide data-analytic solutions or support decision making

Course Schedule(tentative)

Course	Content/ topic/ activity
Week 1	Course Introductions
Week 2	Python basics
Week 3	Data collection and process
Week 4	Introduction of accounting analytic
Week 5	Predictive models and decision tree
Week 6	SVM , LR and other predictive models
Week 7	Performance evaluation of supervised models
Week 8	Business analytical framework
Week 9	Unsupervised data mining and clustering
Week 10	Bayesian models and association rules (market basket analysis)
Week 11	Text analysis and applications
Week 12	Graph analysis and applications
Week 13	Other data mining tasks and models
Week 14	Course Introductions

Course Assessment / textbooks

Component/ method	% weight
Assignments (4 homework assignments +6 in-class labs)	50
Final Exam	20
Course Project (1 group +1 individual)	30

- Provost, Foster, and Tom Fawcett. *"Data Science for Business: What you need to know about data mining and data-analytic thinking."* O'Reilly Media, Inc.", 2013.
(Textbook)
- Jake VanderPlas. Python Data Science Handbook: Essential Tools for Working with Data .O'Reilly Media; 1 edition (December 10, 2016))
- Google Python Class. <https://developers.google.com/edu/python/>
- Learning Everything about Analytics. <https://www.analyticsvidhya.com/>

Data Analytics

ACT4321 Accounting Database and Data Visualization

Database Query

Information Management

Data analytic thinking

Data Analytics

- Traditional statistics
 - ✓ Test hypotheses and estimate the uncertainty of conclusions
- Data warehousing/storage
 - ✓ Data warehouses store data from across an enterprise
- Database Querying (SQL)
 - ✓ Very flexible interface to ask factual questions about data
- Regression Analysis
 - ✓ Dig into the reasons for a specific dataset
- OLAP – On-line Analytical Processing (GUI based manual search)

Data mining focuses on the **automated** search **for knowledge, patterns,** or **regularities** from data

ACT4311: Data Mining and Accounting Analytics



Data Analytics



ACT4321 Accounting Database and Data Visualization

Database Query



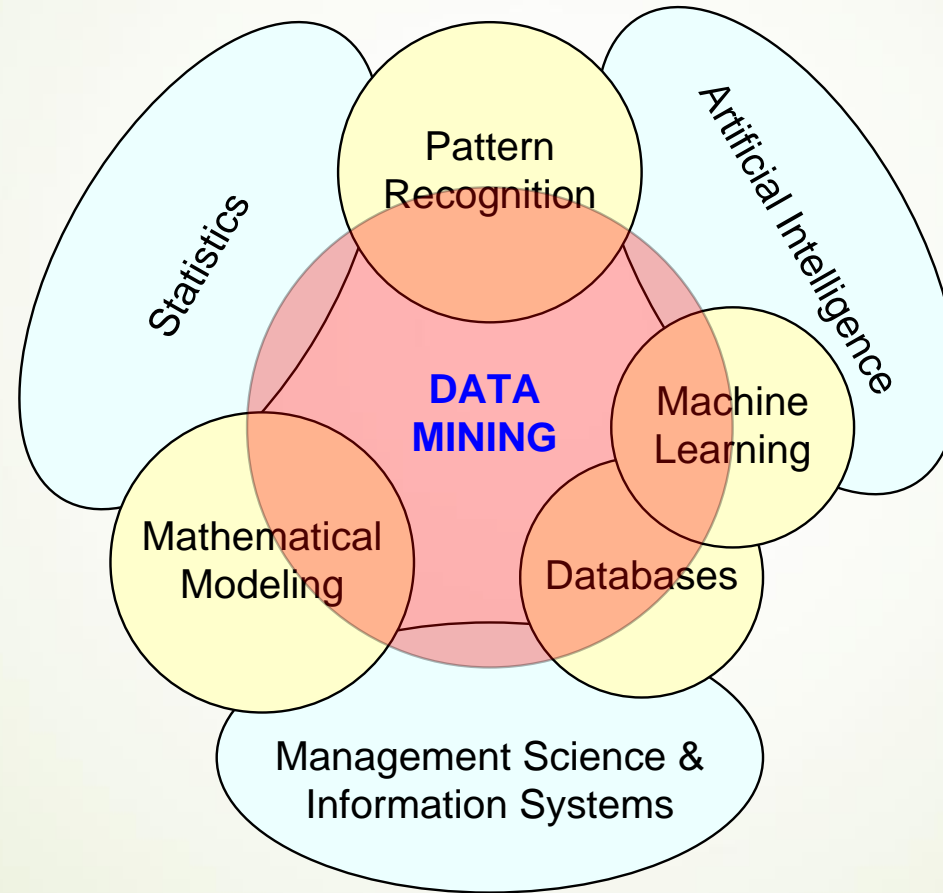
Information Management

Data analytic thinking

Choose Appropriate Techniques

- Who are the most profitable customers?
 - ✓ SQL query
- Is there really a difference between profitable customers and the average customer ?
 - ✓ Statistical hypothesis testing
- But who really are these customers? Can I characterize them?
 - ✓ OLAP + data mining
- Will some particular new customer be profitable? How much revenue should I expect this customer to generate?
 - ✓ Data mining

Data Mining



Formal definitions

Data Mining and Accounting Analytics



“Data mining is the computing process of discovering patterns in large data sets involving methods at the **intersection of** machine learning, statistics, and database systems.” – from Wikipedia

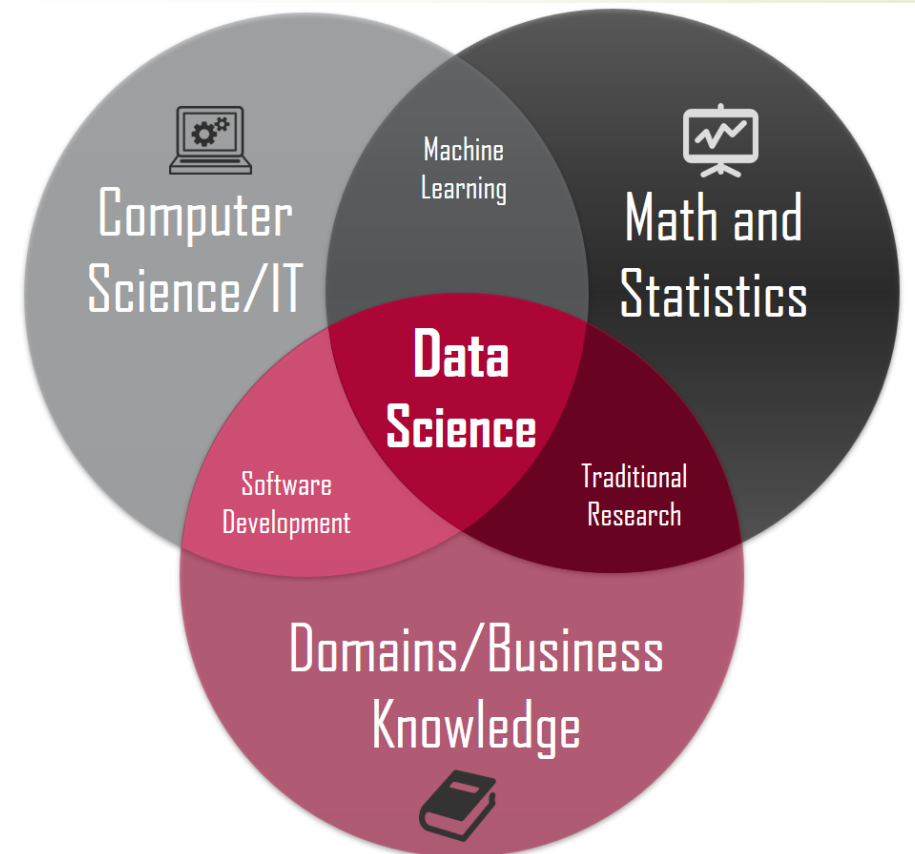


“Accounting analytics is the examination of Big Data using data science or data analytics tools to help answer accounting-related questions..” – from *social-metrics*

Data Science

Data Mining + Business/Accounting Analytics = Data Science

- The objective of Business Analytics is Data-driven decision-making (DDD) , which refers to the practice of basing decisions on the analysis of data, rather than purely on intuition.



Data Scientist:

The Sexiest Job of the 21st Century

**Meet the people who
can coax treasure out of
messy, unstructured data.**

*by Thomas H. Davenport
and D.J. Patil*

When Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join. But users weren't seeking out connections with the people who were already on the site at the rate executives had expected. Something was apparently missing in the social experience. As one LinkedIn manager put it, "It was like arriving at a conference reception and realizing you don't know anyone. So you just stand in the corner sipping your drink—and you probably leave early."

Outline

- Course Introduction
- Data Mining Applications Tasks
- Data Mining Tools (Python)

Wal-Mart: How to Arrange

- It is intuitive to sell baby-related products to new parents.
 - ✓ The arrival of a new baby in a family is one point where people do change their shopping habits significantly. In the Target analyst's word, "As soon as we get them buying diapers from us, they're going to start buying everything else too".
- "Men often bought beer at the same time they bought diapers." (very famous, known as market basket analysis)



MegaTelCo: Customer Churn

- Communications companies are now engaged in battles to attract each other's customers while retaining their own.
- Customers switching from one company to another is called **churn**,
- Attracting new customers is much more expensive than retaining existing ones, so a good deal of marketing budget is allocated to **prevent churn**.
- Marketing has already designed a special retention offer.
- Task for MegaTelCo: how the data science team should use MegaTelCo's vast data resources to decide which customers should be offered the special retention deal prior to the expiration of their contract?
- This example will be repeatedly used through the whole course.

Summary of Data Mining Tasks

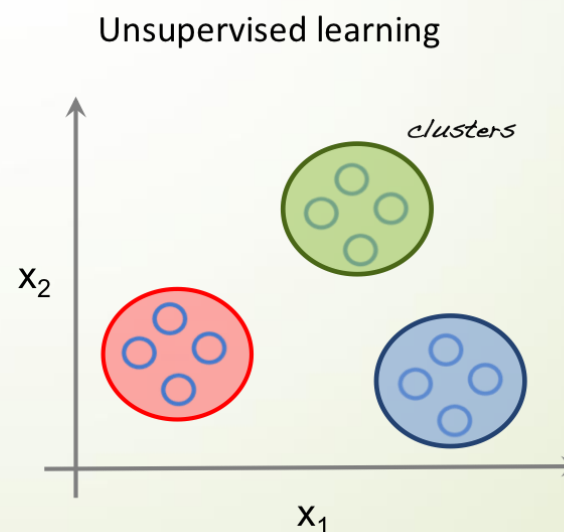
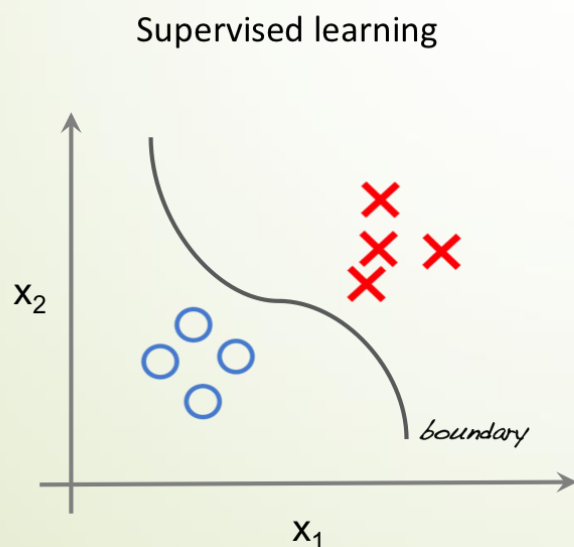
- **Affinity grouping** (a.k.a. “associations”, “market-basket analysis”)
 - ✓ What items are commonly purchased together?
 - **Similarity Matching**
 - ✓ What other companies are like our best small business customers?
 - **Description/Profiling/Feature Reduction**
 - ✓ What does “normal behavior” look like?
 - **Clustering**
 - ✓ Do my customers form natural groups?
-
- Predictive Modeling (including causal modeling & link prediction)
 - ✓ **Classification**: Will customer X churn next month/default on her loan?
 - ✓ **Regression**: How much would prospect X spend?
 - ✓ **Link prediction**: Who might be good “friends” on our social networking site?

Unsupervised

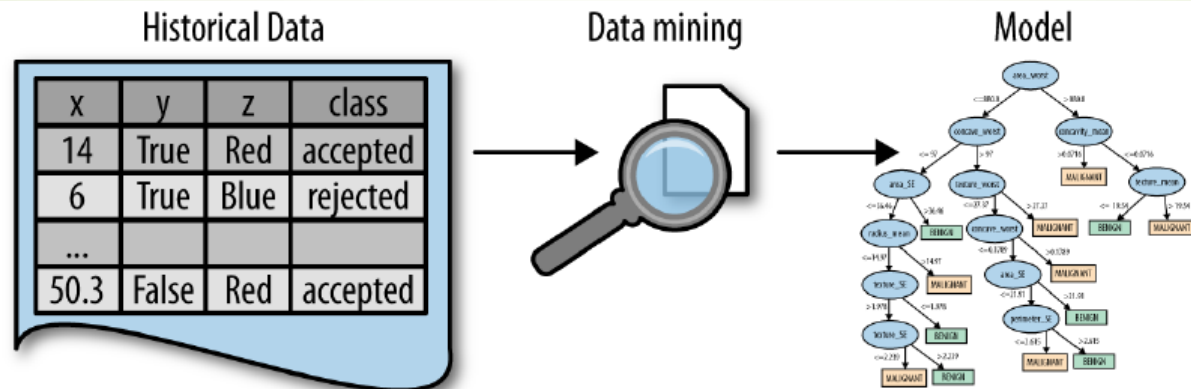
Supervised

Supervised v.s. Unsupervised learning tasks

- Supervised Learning: Data are some pairs of inputs/outputs(target), and the task is to predict an output when new inputs are presented
 - ✓ Among a set of pictures of duck/horse/cat faces, tell the specie of each picture
- Unsupervised Learning: Task can be done on data without expected output
 - ✓ Divide a set of pictures of animal faces into different groups



Results of Data Mining Tasks



Training data have all values specified

Model is deployed

Mining

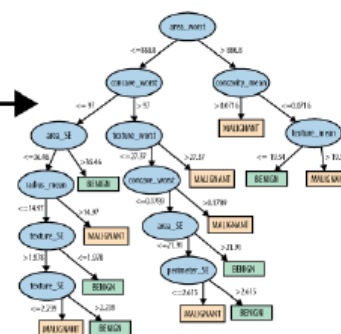
Use

New data item

x	y	z	class
30	false	Red	?

New data item has class value unknown (e.g. will customer accept?)

Model



Class: accepted,
Probability: 0.88

Business Understanding is Important

Marketing a standard product



Fund Raising



Churn Management



**Identify/ranking the potential customers, and then
take different actions**

Problem Definition

Marketing a standard product

- Fixed income
- Predict response probability



Fund Raising

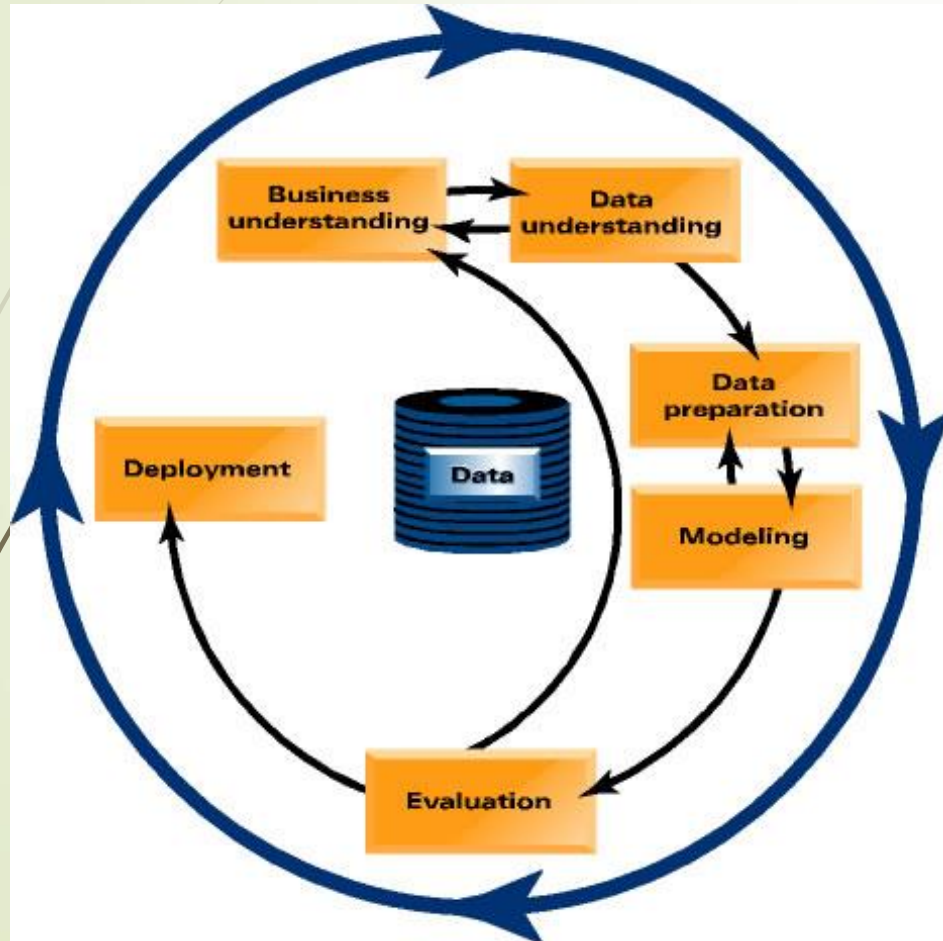


Churn Management

- Different income
- Predict response probability+ income

- Different income
- Some customers still stay
- Predict stay probability+ income if targeting
- Predict stay probability+ income if not targeting

Data Collection



What data might China Mobile mine to help with churn management?
E.g., Users' name, gender, age?

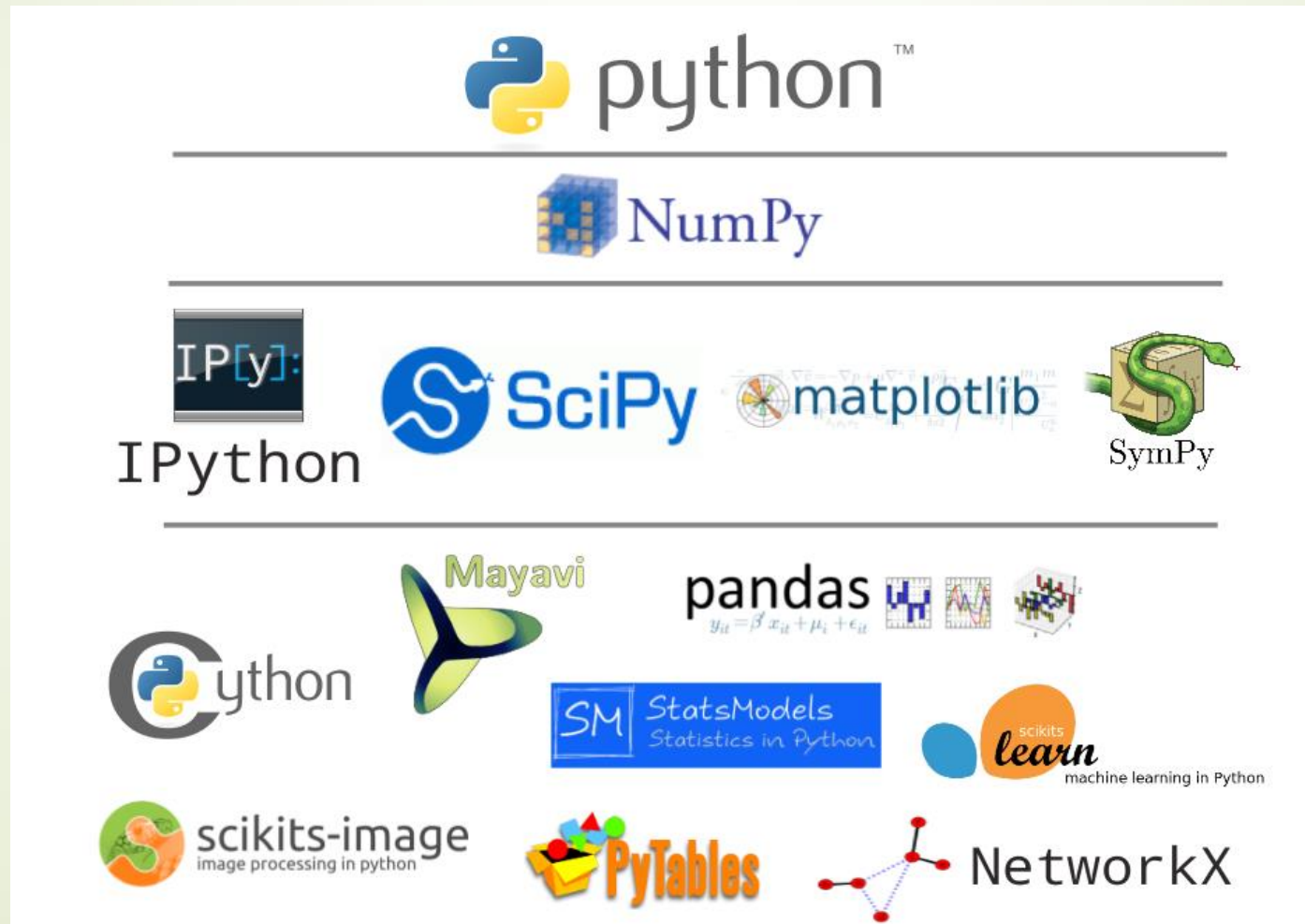
Outline

- Course Introduction
- Data Mining Applications Tasks
- Data Mining Tools (Python)

Data Mining Tools

- Big commercial software
 - ✓ SAS Enterprise Miner
 - ✓ IBM Cognos
 - ✓ SPSS/STATA
- Free software
 - ✓ RapidMiner (<https://rapidminer.com/>)
 - ✓ WEKA (<http://www.cs.waikato.ac.nz/ml/weka/>)
- Programming languages
 - ✓ R (strong in statistics)
 - ✓ **Python**

Why Python?



Python Is Promising (Deep learning)

- Theano
- **TensorFlow** (Google)
- **PyTorch** (Facebook)
- MXNet (Amazon)
- Caffe
- CNTK
- Keras

Useful packages

➤ Scipy+Numpy

- ✓ Useful data structure: ndarray, matrix, array
- ✓ Algorithms: statistics, scientific computation

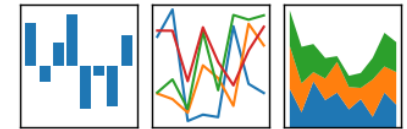


➤ Pandas

- ✓ Processing table -like dataset (such as Excel)

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



➤ Scikit-learn

- ✓ Various data mining/machine learning functions



➤ NLTK (Natural Language Toolkit)

➤ Visualization

- ✓ Explore data with informative figures



Python Versions

► Python 2.7

- ✓ More popular
- ✓ Old
- ✓ Some future changes can be imported
- ✓ Larger library base

► Python 3.7+

- ✓ Better performance
- ✓ Future orientated
- ✓ Some design flaws have been corrected
- ✓ Library base is big enough

Anaconda

- Anaconda is a leading open data science platform powered by Python
- Integrated with a collection of over 720 open source packages
- Compatible on Windows, MacOS or Linux
- Easy to install

		
 KAISER PERMANENTE®	J.P.Morgan	 macy's
 BOEING		 Los Alamos NATIONAL LABORATORY EST. 1943
GEICO ®	Bank of America 	 appnexus
 CISCO ™	SIEMENS	

Install Anaconda

➤ Install guide

- ✓ <https://docs.continuum.io/anaconda/install/>

➤ You can get

- ✓ Python
- ✓ Python Integrated Development Environment (IDE): Spyder, Jupyter
- ✓ Useful integrated packages : Numpy, Scikit-learn, Matplotlib....
- ✓ Package manager: conda: `conda install <package name>`

```
C:\Users\admin>conda install scrapy
Fetching package metadata: ..
Solving package specifications: .
Package plan for installation in environment d:\software\Anaconda:

The following packages will be downloaded:


```

package	build	
conda-3.9.0	py27_0	206 KB
conda-env-2.1.3	py27_0	54 KB



File Edit Search Source Run Tools View ?

Editor - C:\Users\Steve\xy\resnel\resnel.py

```

272 T = (s_data['T'] + p_data['T']) / 2.
273 return {'R': R, 'T': T}
274
275 def position_resolved(layer, dist, fresnel_data):
276     """
277     Starting with output of fresnel_main(), calculate the Poynting vector
278     and absorbed energy density a distance "dist" into layer number "layer"
279     """
280     vw = fresnel_data['vw_list'][layer]
281     kz = fresnel_data['kz_list'][layer]
282     th = fresnel_data['th_list'][layer]
283     n = fresnel_data['n_list'][layer]
284     n_0 = fresnel_data['n_list'][0]
285     th_0 = fresnel_data['th_0']
286     pol = fresnel_data['pol']
287
288     #amplitude of forward-moving wave is Ef, backwards is Eb
289     Ef = vw[0] * exp(1j * kz * dist)
290     Eb = vw[1] * exp(-1j * kz * dist)
291
292     #Poynting vector
293     if(pol=='s'):
294         poyn = ((n*cos(th)*conj(Ef+Eb)*(Ef-Eb)).real) / (n_0*cos(th_0)).real
295     elif(pol=='p'):
296         poyn = (((n*conj(cos(th))*(Ef+Eb)*conj(Ef-Eb)).real)
297                 / (n_0*conj(cos(th_0))).real)
298
299     #absorbed energy density
300     if(pol=='s'):
301         absor = (n*cos(th)*kz*abs(Ef+Eb)**2).imag / (n_0*cos(th_0)).real
302     elif(pol=='p'):
303         absor = (n*conj(cos(th))*
304                 (kz*abs(Ef-Eb)**2-conj(kz)*abs(Ef+Eb)**2)
305                 ).imag / (n_0*conj(cos(th_0))).real
306     return({'poyn':poyn, 'absor':absor})
307
308 def find_in_structure(d_list,dist):
309     """
310     d_list is list of thicknesses of layers, all of which are finite.
311
312     dist is the distance from the front of the whole multilayer structure
313     (i.e., from the start of layer 0.)
314

```

Object inspector

Source Console Object fresnel.fresnel_main

fresnel_main(pol, n_list, d_list, th_0, lam_vac)

Function of fresnel module

Main fresnel calc. Given parameters of a stack, calculates everything you could ever want to know about how light propagates in it. (If performance is an issue, you can delete some of the calculations without affecting the rest.)

pol is light polarization, "s" or "p".

n_list is the list of refractive indices, in the order that the light would pass through them. The 0'th element of the list should be the semi-infinite medium from which the light enters, the last element should be the semi-infinite medium to which the light exits (if any exits).

th_0 is the angle of incidence 0 for normal, pi/2 for glancing. Remember, for a dissipative incoming medium (n_list[0] is not real), th_0 should be complex so that n_0 sin(th_0) is real (intensity is constant as a function of lateral position).

d_list is the list of layer thicknesses (front to back). Should correspond one-to-one with elements of n_list. First and last elements should be "inf".

lam_vac is vacuum wavelength of the light.

Object inspector Variable explorer File explorer

Console

IPython 1

00:16:44

```

In [8]: pv_sim.testtt()
ISC = 4.103 mA/cm2
EQE for 400-800nm = (4.103 mA/cm2) / (25.923 mA/cm2) = 15.8%
Reflection into air = 16.2 mA/cm2 = 62.5%
Absorption in mirror = 0.96 mA/cm2
Thin-layer thicknesses in nm = [ 150.    70.    20.    20.    20.    0.34]
Absorption in thin layers = [ 1.18  0.51  4.64  1.91  0.    0.52]
(for, respectively, [ITO,PEDOT,SubPC,C60,TPBi,graphene])
C60 IQE = (1.49 mA/cm2) / (1.91 mA/cm2) = 77.8%
SubPC IQE = (2.61 mA/cm2) / (4.64 mA/cm2) = 56.3%
Out[8]: 4.1029296077801174

In [9]: 1.18 + 0.51 + 4.64 + 1.91 + 0.52
Out[9]: 8.76

In [10]: 1.18 + 0.51 + 4.64 + 1.91 + 0.52 + 16.2 + 0.96
Out[10]: 25.92

In [11]:

```

Console History log

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 289 Column: 28

Programing Basics

- **code** or **source code**: The sequence of instructions in a program.
- **syntax**: The set of legal structures and commands that can be used in a particular programming language (Thank you. Not thanks you)
- **output**: The messages printed to the user by a program.
- **console**: The text box onto which output is printed.

Programing Basics(Spyder)

Variable/Help docs

Source
code

The screenshot displays the Spyder IDE interface with three main panels:

- Source code:** The left panel shows the Python code for the `fresnel` module. It includes functions like `position_resolved` and `find_in_structure`. A red circle highlights the `position_resolved` function and its associated comments.
- Object inspector:** The top-right panel shows the `fresnel.fresnel_main` object. It contains a description of the `fresnel_main` function and its parameters: `pol` (light polarization), `n_list` (refractive indices), `th_0` (angle of incidence), `d_list` (layer thicknesses), and `lam_vac` (vacuum wavelength).
- Console:** The bottom-right panel shows the output of the code execution. It displays the results of the `pv_sim.testt()` function, including calculated values for ISC, EQE, and absorption in various layers.

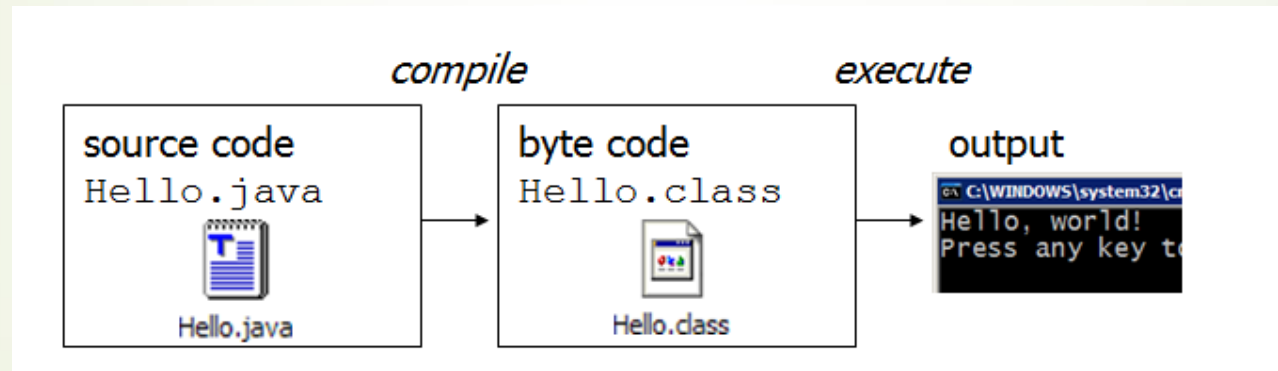
Red arrows point from the labels to their respective panels: "Source code" to the left panel, "Variable/Help docs" to the top-right panel, and "Console" to the bottom-right panel. A red circle highlights the `position_resolved` function in the source code. A red box highlights the console output for the `pv_sim.testt()` function.

Console

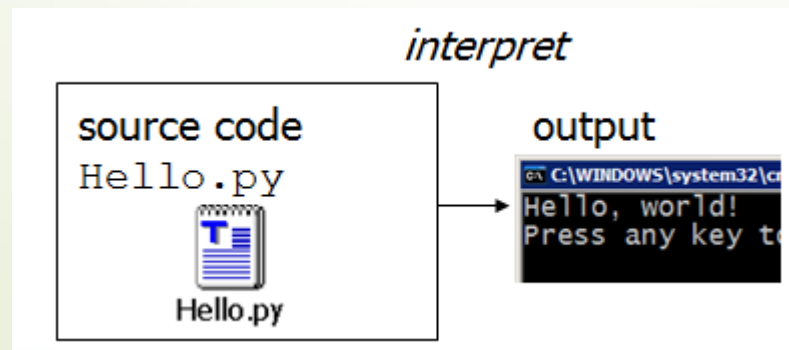
Output

Interpreted Language

- Many languages require you to *compile* (translate) your program into a form that the machine understands.

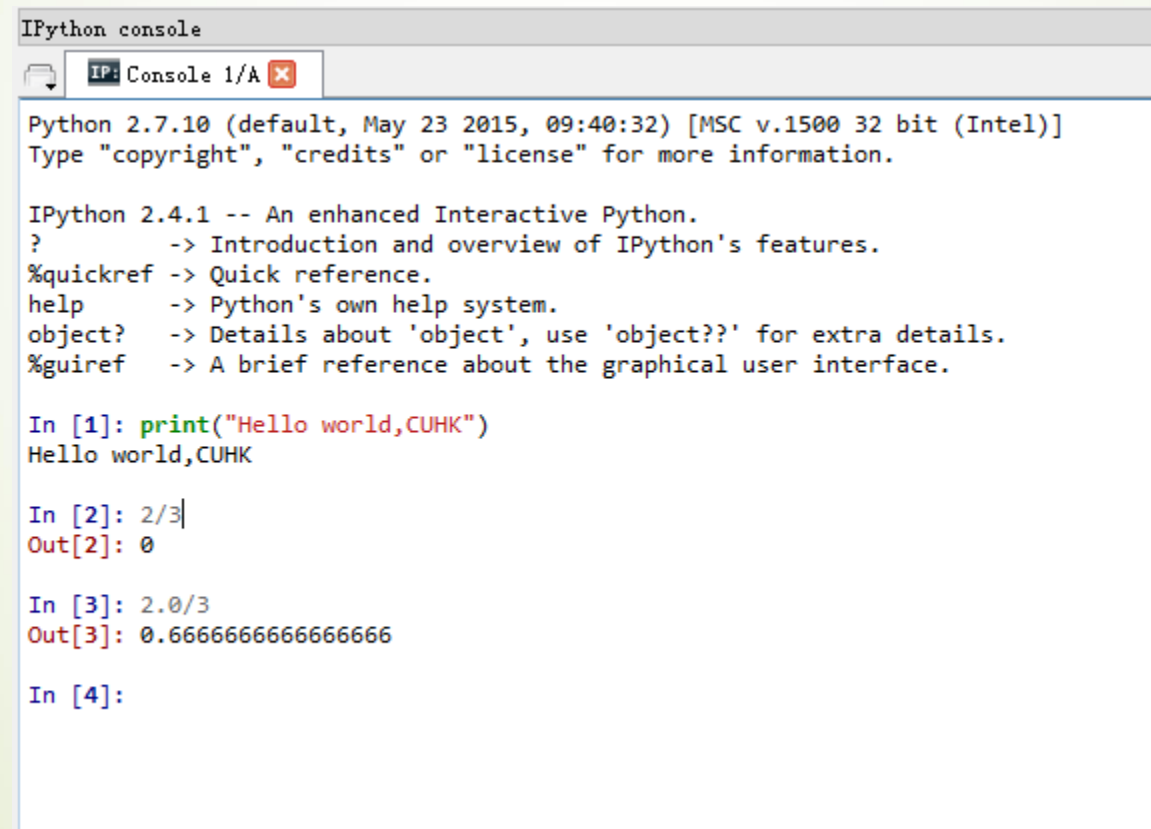


- Python is instead directly *interpreted* into machine instructions.



Interpreted Language(Spyder)

- You can write and execute Python in the console line by line

A screenshot of the IPython console window. The title bar says "IPython console". Below it is a tab labeled "IP: Console 1/A". The console text shows the Python version (2.7.10), the IPython version (2.4.1), and various help messages. It then shows three lines of code being executed: a print statement, a division, and a floating-point division. The output for each line is displayed below the input.

```
Python 2.7.10 (default, May 23 2015, 09:40:32) [MSC v.1500 32 bit (Intel)]
Type "copyright", "credits" or "license" for more information.

IPython 2.4.1 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.
%gui?           -> A brief reference about the graphical user interface.

In [1]: print("Hello world, CUHK")
Hello world, CUHK

In [2]: 2/3
Out[2]: 0

In [3]: 2.0/3
Out[3]: 0.6666666666666666

In [4]:
```

Python Basics (Overall)

- Values
- Operations on values
- Assignments
- Input/output operations
- Conditional actions
- Repeated actions

Values (Types)

➤ **Numbers** – not much of a surprise

- ✓ int: 30,-6000
- ✓ floating point: 2232.435

➤ **Boolean**: True, False

➤ **String**– a sequence of characters (in Python 3, always unicode)

- ✓ "I am a String", 'I am a String', """I am a String""", '''I am a String''
- ✓ '\ ' is a special control character in String, to escape some special characters
- ✓ \n, \t, \\, \', \"
- ✓ Useful functions: lower(), upper(), strip(),split(), replace(),endswith(), startswith()

➤ **None**

type(x) can view the type of a variable/value

Escape Characters in String

<code>\n</code>	newline	Advances the cursor to the next line for subsequent printing
<code>\t</code>	tab	Causes the cursor to skip over to the next tab stop
<code>\b</code>	backspace	Causes the cursor to back up, or move left, one position
<code>\r</code>	carriage return	Causes the cursor to go to the beginning of the current line, not the next line
<code>\\</code>	backslash	Causes a backslash to be printed
<code>\'</code>	single quote	Causes a single quotation mark to be printed
<code>\"</code>	double quote	Causes a double quotation mark to be printed

String Slices

- The "slice" syntax is a handy way to refer to sub-parts of sequences
- The slice `s[start:end]` is the elements beginning at start and extending up to but **not including** end

Hello

0 1 2 3 4

-5 -4 -3 -2 -1

- `s[1:4]` is 'ell' -- chars starting at index 1 and extending up to but not including index 4
- `s[1:]` is 'ello' -- omitting either index defaults to the start or end of the string
- `s[:]` is 'Hello' -- omitting both always gives us a copy of the whole thing (this is the pythonic way to copy a sequence like a string or list)
- `s[1:100]` is 'ello' -- an index that is too big is truncated down to the string length

String Slices (Cont'd)

- ▶ Python uses negative numbers to give easy access to the chars at the end of the string: `s[-1]` is the last char 'o', `s[-2]` is 'l' ...
 - ▶ Negative index numbers count back from the end of the string

Hello				
0	1	2	3	4
-5	-4	-3	-2	-1

- `s[-1]` is 'o' -- last char (1st from the end)
- `s[-4]` is 'e' -- 4th from the end
- `s[:-3]` is 'He' -- going up to but not including the last 3 chars.
- `s[-3:]` is 'llo' -- starting with the 3rd char from the end and extending to the end of the string.

- ▶ It is always true for any index `n`, `s[:n] + s[n:] == s`.

Other Operations on String

➔ https://www.tutorialspoint.com/python/python_strings.htm

Name	Purpose
<code>len(s)</code>	Calculate the length of the string <code>s</code>
<code>+</code>	Add two strings together
<code>*</code>	Repeat a string
<code>s.find(x)</code>	Find the first position of <code>x</code> in the string <code>s</code>
<code>s.count(x)</code>	Count the number of times <code>x</code> is in the string <code>s</code>
<code>s.upper()</code> <code>s.lower()</code>	Return a new string that is all uppercase or lowercase
<code>s.replace(x, y)</code>	Return a new string that has replaced the substring <code>x</code> with the new substring <code>y</code>
<code>s.strip()</code>	Return a new string with whitespace stripped from the ends
<code>s.format()</code>	Format a string's contents

Values-type casting

- Casting is when you convert a variable value from one type to another
 - ✓ `int(x)` to convert x from other type to int
 - ✓ `float(x)` to convert x from other type to float
 - ✓ `str(x)` to convert x from other type to string
- Type-casting should be done carefully
 - ✓ Will generate error if cannot be done
 - ✓ ~~`int('abc'), int('110.0')`~~
 - ✓ Will lose accuracy
 - ✓ `bool(0)` and `bool('')` and `bool(None)` are False

```
In [2]: a = "666"
```

```
In [3]: b=int(a)
```

```
In [4]: type(a)
```

```
Out[4]: str
```

```
In [5]: type(b)
```

```
Out[5]: int
```

Math Operations

➤ Math operators

- ✓ Addition: $5 + 6 = 11$
- ✓ Subtraction: $7.6 - 9 = -1.4$
- ✓ Multiplication: $5.7 * 3 = 17.1$
- ✓ Division: $7 / 5 = 1.4$; $7 // 5 = 1$
- ✓ Modulo: $7 \% 5 = 2$
- ✓ Power: $3 ** 2 = 9$

➤ String operators

- ✓ Addition: `"Hello" + " " + "World" = "Hello World"`
- ✓ Multiplication: `"Ha"*3 = "HaHaHa"`

```
In [4]: type(10)
```

```
Out[4]: int
```

```
In [5]: type(101)
```

```
Out[5]: long
```

```
In [6]: type(101+10)
```

```
Out[6]: long
```

```
In [7]: type(101+0.0)
```

```
Out[7]: float
```

float



~~long~~



int

Math Operations (in math package)

- After import math package by “from math import *”

Command name	Description
<code>abs(value)</code>	absolute value
<code>ceil(value)</code>	rounds up
<code>cos(value)</code>	cosine, in radians
<code>floor(value)</code>	rounds down
<code>log(value)</code>	logarithm, base e
<code>log10(value)</code>	logarithm, base 10
<code>max(value1, value2)</code>	larger of two values
<code>min(value1, value2)</code>	smaller of two values
<code>round(value)</code>	nearest whole number
<code>sin(value)</code>	sine, in radians
<code>sqrt(value)</code>	square root

Logic Operations

- Logic operations will generate Boolean value (True, False)

Operator	Meaning	Example	Result
==	equals	<code>1 + 1 == 2</code>	True
!=	does not equal	<code>3.2 != 2.5</code>	True
<	less than	<code>10 < 5</code>	False
>	greater than	<code>10 > 5</code>	True
<=	less than or equal to	<code>126 <= 100</code>	False
>=	greater than or equal to	<code>5.0 >= 5.0</code>	True

- Logic combination

Operator	Example	Result
and	<code>9 != 6 and 2 < 3</code>	True
or	<code>2 == 3 or -1 < 5</code>	True
not	<code>not 7 > 0</code>	False

Operators Precedence

- Use **()** to clearly indicate your intended execution sequence

Operator	Description
**	Exponentiation (raise to the power)
~ + -	Complement, unary plus and minus (method names for the last two are +@ and -@)
* / % //	Multiply, divide, modulo and floor division
+ -	Addition and subtraction
>> <<	Right and left bitwise shift
&	Bitwise 'AND'
^	Bitwise exclusive 'OR' and regular 'OR'
<= < > >=	Comparison operators
<> == !=	Equality operators
= %= /= //= -= += *= **=	Assignment operators

- ✓ $(4**2)/2$ better than $4**2/2$
- ✓ $A \text{ or } (B \text{ and } C)$ better than $A \text{ or } B \text{ and } C$



Variables and Assignments(1)

- **Variable:** A named piece of memory that can store(point to) a value
 - ✓ Compute an expression's result and then Store that result
 - ✓ Use that variable later and repeatedly in the program
- **Assignment statement :** Store/update a value into a variable
 - ✓ Variable_name = value expression
 - ✓ Example (for a rectangular)s:
height=3
width=4
perimeter = 2*(height+width)
area = height*width

Variables and Assignments(2)

➤ Do not use following key words for variable name

- ✓ False, class, finally, is, return, None, continue, for, lambda, try, type
- ✓ True, def, from, nonlocal, while, and, del
- ✓ as, elif, if, or, yield, assert, else, import, pass,
- ✓ break, except, in, raise, global, not, with

```
In [8]: type=3
```

```
In [9]: type(101+0.0)  
Traceback (most recent call last):
```

```
File "<ipython-input-9-9eafc29ab9ea>", line 1, in <module>  
    type(101+0.0)
```

```
TypeError: 'int' object is not callable
```

➤ Advanced assignment

- ✓ `result = 10, record="Hello"`
- ✓ `result = result + 10, then result=20` (add result by 10)
- ✓ `result += 10, then result =30`
- ✓ `record += " World", then record= "Hello World"`
- ✓ `a,b,c = 1,2,'123'`

Comments and Indentation

- **Comments** can help understand the code
 - ✓ `#This is a comment`
 - ✓ `""" This is a multi-line
Comment"""`
- **Indentation** is important for Python logics (not just for beautiful code)
 - ✓ Should be consistent within one file
 - ✓ Use 4 space for consistency (default in Spyder)
 - ✓ Google Python Coding Style:
http://zh-google-styleguide.readthedocs.io/en/latest/google-python-styleguide/python_style_rules/

Input and Output

- **input('xxxx')**: print XXX on the console, then read **keyboard input** and return the input line in the console as **String**

- ✓ Code:

```
age = input("How old are you? ")  
print "Your age is", age
```

- ✓ Output:

```
How old are you? 53
```

```
Your age is 53
```

- **print(item1,[item2], sep=' ')** :show the results on the console

- ✓ `print(1,2,3) → 1 2 3`

- ✓ `print(1,2,3, sep=',') → 1,2,3`

Exercise on String

<https://www.w3resource.com/python-exercises/string/>

- 1. Write a Python program to calculate the length of a string.
- 4. Write a Python program to get a string from a given string where all occurrences of its first char have been changed to '\$', except the first char itself.
- 5. Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.