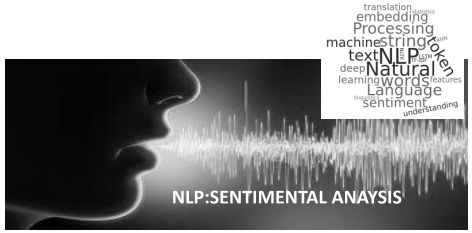
Area of interests



Course outline

Course learning outcomes (CLOs): Students are able to
1 Explain the process in NLP and techniques.
2. Use the appropriate models and metrics tools for the right problem.

Course Description:
NLP overview, word tokenization and text preprocessing, text extraction methods, machine-learning models in NLP, Deep-learning models in NLP, Transformer, model evaluation and explain-ability, evaluation metrics, NLP-based systems, and case studies

兴趣领域

课程大纲

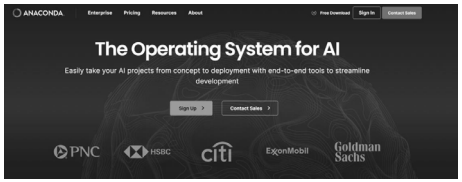
课程学习成果 （CLO）： 学生能够 1 解释 NLP 和技术的过程。

2. 针对正确的问题使用适当的模型和指标工具。

课程描述：
NLP 概述、单词标记化和文本预处理、文本提取方法、NLP 中的机器学习模型、NLP 中的深度学习模型、Transformer 、模型评估和可解释性、评估指标、基于 NLP 的系统和案例研究

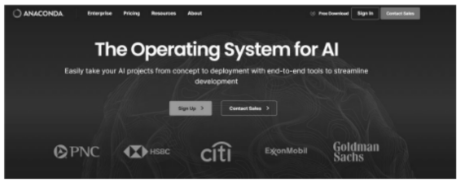
Anaconda/Miniconda

- Pre-install libraries
- Good package management
- Easy to install, maintain, and export
- Cross platforms
- https://www.anaconda.com/



蟒蛇/迷你康达

- 预安装库
- 良好的包管理
- 易于安装、维护和导出
- 跨平台
- https ://www .anaconda .com /



Dev tools for the course

Asst. Prof. Pree Thiengburanathum

课程的开发工具

Kaggle

Colab

Github

Kaggle Colab Github

Introduction (cont.)

- Data continues to grow exponentially
  - Estimated to be 2.5 MTB a day
  - Grow to 40 BTB by 2020 (50 * of 2010)
- Approx. 80% of data is estimated to be unstructured/text-rich data
  - >4.5 billion web pages
  - >40 million articles (5 million in English)
  - >500 million tweets a day, 200 billion a year
  - >1.5 trillion queries on Google a year

引言（续）

- 数据继续呈指数级增长
  - 预计每天 2.5 MTB
  - 到 2020 年增长到 40 BTB （2010 年的 50 *）
- 据估计，大约 80％ 的数据是非结构化/文本丰富的数据
  - > 45 亿个网页
  - > 4000 万篇文章（英文500 万篇）
  - 每天> 5亿条推文，每年2000 亿条
  - Google 每年有 > 1.5 万亿次查询

Machine Versus Men (cont.)

- Can machine beat the best of man in what man is supposed to be the best at?
- https://www.youtube.com/watch?v=YgYSv2KSyWg
- Watson, which is called DeepQA
- Watson had access to 200 million pages of structured and unstructured content consuming four terabytes of disk storage.

机器与人（续）

- 机器能在人类最擅长的事情上击败最好的人吗?
- https ://www .youtube .com /watch ?v= YgYSv2KSyWg
- Watson ，称为 DeepQA
- Watson 可以访问 2 亿页结构化和非结构化内容，占用 4 TB 的磁盘存储。

DeepQA overall architecture

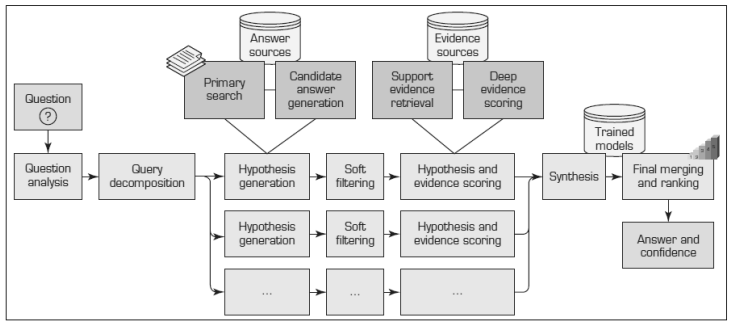


FIGURE 7.1 A High-Level Depiction of DeepQA Architecture.

DeepQA 整体架构

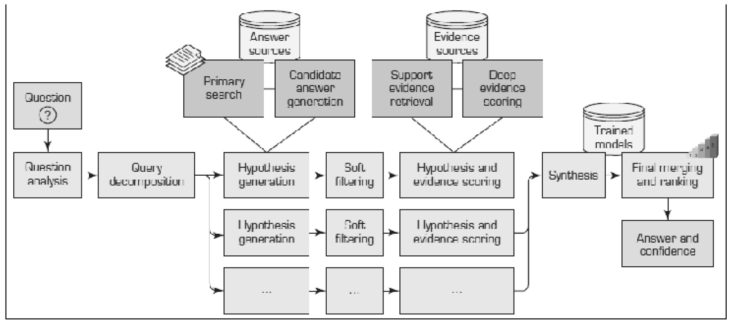
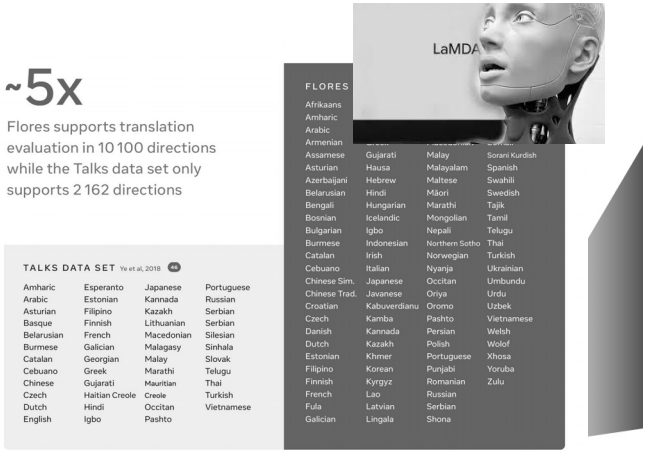


FIGURE 7.1 A High-Level Depiction of DeepQA Architecture.

Meta NLP 2022

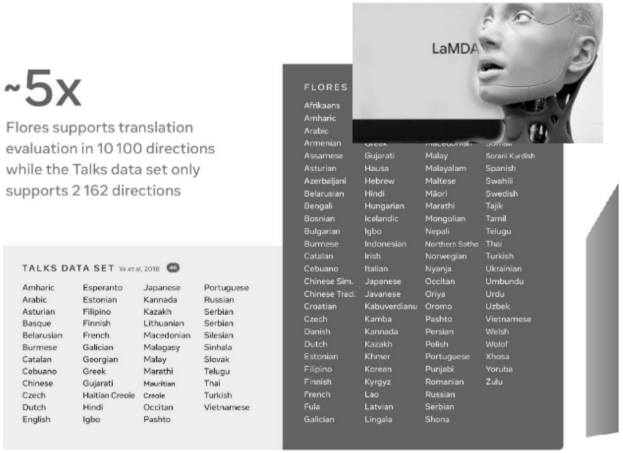
- LaMDA – conversation robot, BERT+GPT-3
- FLORES-101 – Largest language transition dataset



元自然语言处理

2022

- LaMDA – 对话机器人，BERT + GPT - 3
- FLORES - 101 – 最大语言转换数据



Meta NLP 2023

- GPT-3 4,096 and 2,049 tokens
- GPT-4 8,192 and 32,768 tokens
- Introduced “Multimodal” (e.g., can also understand image)
- Better and solve math problems
- Even more languages (with low-resources)
- Able to include reference and source of the text generated

元 NLP 2023

- GPT-3 4,096 和 2,049 个代币
- GPT-4 8,192 和 32,768 个代币
- 引入了“ 多模态” （例如，也可以理解图像）
- 更好地解决数学问题
- 更多语言（资源匮乏）
- 能够包含所生成文本的参考和来源

Structured vs Unstructured data

	Indicator ID	Dimension List	Timeframe	Numeric Value	Missing Value Flag	Confidence Int
2	214390830	Total (Age-adjusted)	2008	74.6%		73.8%
3	214390833	Aged 18-44 years	2008	59.4%		58.0%
4	214390831	Aged 18-24 years	2008	37.4%		34.6%
5	214390832	Aged 25-44 years	2008	66.9%		65.5%
6	214390836	Aged 45-64 years	2008	88.6%		87.7%
7	214390834	Aged 45-54 years	2008	86.3%		85.1%
8	214390835	Aged 55-64 years	2008	91.5%		90.4%
9	214390840	Aged 65 years and over	2008	94.6%		93.8%
10	214390837	Aged 65-74 years	2008	93.6%		92.4%
11	214390838	Aged 75-84 years	2008	95.6%		94.4%
12	214390839	Aged 85 years and over	2008	96.0%		94.0%
13	214390841	Male (Age-adjusted)	2008	72.2%		71.1%
14	214390842	Female (Age-adjusted)	2008	76.8%		75.9%
15	214390843	White only (Age-adjusted)	2008	77.6%		75.0%
16	214390844	Black or African American only (Age-adjusted)	2008	66.5%		57.1%
17	214390845	American Indian or Alaska Native only (Age-adjusted)	2008	80.5%		77.7%
18	214390846	Asian only (Age-adjusted)	2008	DSU		
19	214390847	Native Hawaiian or Other Pacific Islander only (Age-adjusted)	2008			
20	214390848	2 or more races (Age-adjusted)	2008	73.6%		69.6%

Figure 1.1 An Excel table is an example of structured data.

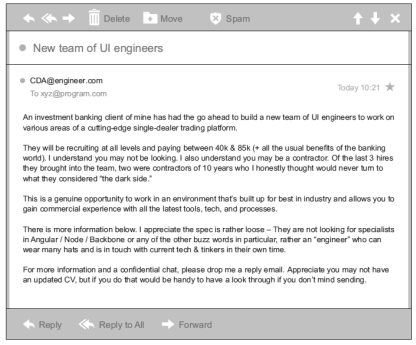


Figure 1.2 Email is simultaneously an example of unstructured data and natural language data.

Images source: Course

结构化数据与非结构化数据

	Indicator ID	Dimension List	Timeframe	Numeric Value	Missing Value Flag	Confidence Int
2	214390830	Total (Age-adjusted)	2008	74.6%		73.8%
3	214390833	Aged 18-44 years	2008	59.4%		58.0%
4	214390831	Aged 18-24 years	2008	37.4%		34.6%
5	214390832	Aged 25-44 years	2008	66.9%		65.5%
6	214390836	Aged 45-64 years	2008	88.6%		87.7%
7	214390834	Aged 45-54 years	2008	86.3%		85.1%
8	214390835	Aged 55-64 years	2008	91.5%		90.4%
9	214390840	Aged 65 years and over	2008	94.6%		93.8%
10	214390837	Aged 65-74 years	2008	93.6%		92.4%
11	214390838	Aged 75-84 years	2008	95.6%		94.4%
12	214390839	Aged 85 years and over	2008	96.0%		94.0%
13	214390841	Male (Age-adjusted)	2008	72.2%		71.1%
14	214390842	Female (Age-adjusted)	2008	76.8%		75.9%
15	214390843	White only (Age-adjusted)	2008	73.6%		72.9%
16	214390844	Black or African American only (Age-adjusted)	2008	77.6%		75.0%
17	214390845	American Indian or Alaska Native only (Age-adjusted)	2008	66.5%		57.1%
18	214390846	Asian only (Age-adjusted)	2008	80.5%		77.7%
19	214390847	Native Hawaiian or Other Pacific Islander only (Age-adjusted)	2008	DSU		
20	214390848	2 or more races (Age-adjusted)	2008	73.6%		69.6%

Figure 1.1 An Excel table is an example of structured data.

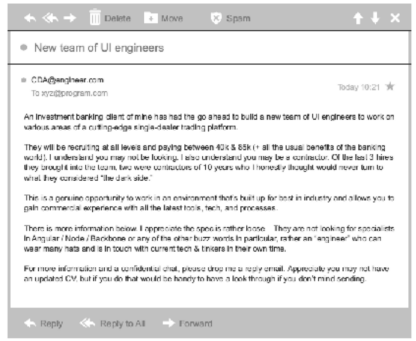


Figure 1.2 Email is simultaneously an example of unstructured data and natural language data.

图片来源： Course t

Text Analytics

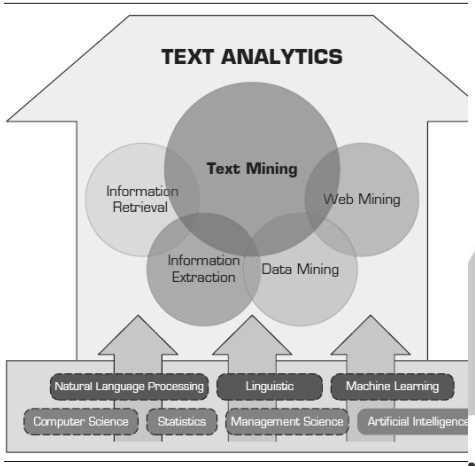
- The vast majority of business data is stored in text documents that are virtually unstructured.
- Text analytics is a broader concept that includes information retrieval (e.g., searching and identifying relevant documents for a given set of key terms) as well as information extraction, data mining, and Web mining,
- Whereas text mining is primarily focused on discovering new and useful knowledge from the textual data sources.

文本分析

- 绝大多数业务数据存储在文本文档中，这些文档几乎是非结构化的。
- 文本分析是一个更广泛的概念，包括信息检索（例如，搜索和识别一组给定关键术语的相关文档）以及信息提取、数据挖掘和 Web 挖掘。
- 而文本挖掘主要侧重于从文本数据源中发现新的和有用的知识。

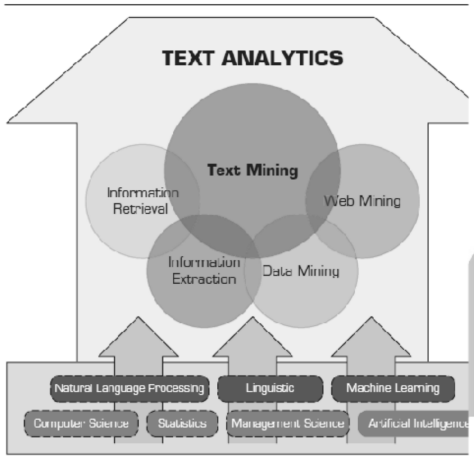
Text Analytics (cont.)

- Text Mining is a derivative of Data Mining.
- Sentimental Analysis is a derivative of Text Mining.



文本分析（帐户）

- 文本挖掘是数据挖掘的衍生物。
- 情感分析是一个文本挖掘的衍生物。



Text Mining

- AKA. Text data mining/knowledge discovery in textual database
- Large amount of unstructured data
- Word, PDF, XML, etc.

- Benefit domains:
  - In the areas where very large amount of textual data are being generated.
  - Could you give some example?

文本挖掘

- 即。文本数据库中的文本数据挖掘/知识发现
- 海量非结构化数据
- Word 、 PDF 、 XML 等

- 福利领域：
  - 在生成大量文本数据的区域。
  - 你能举一些例子吗？

Text Mining (cont.)

- Law – court orders
- Academic research – research article
- Finance – quarterly report
- Medicine – discharge summaries
- Biology- molecular interactions
- Marketing – customer comments
- Social Science – Web board, Twitter , etc.
- Technology – e-mail platforms? Is Gmail the smartest?

文本挖掘（续）

- 法律 – 法院命令
- 学术研究 – 研究文章
- 财务 – 季度报告
- 药物 – 出院总结
- 生物学-分子相互作用
- 营销 – 客户评论
- 社会科学 – Web board 、 Twitter 等。
- 技术 – 电子邮件平台？Gmail 是最聪明的吗？

Text Mining applications

- **Information extraction** – identify the key phrases and relationship within text
- **Topic tracking** – predict/recommend other document of interest to user.
- **Summarization** – summarizing a document to save time of the reader.
- **Categorization** – identify the main themes of a document and put to the right themes
- **Clustering** – group similar documents without having a predefined set of categories
- **Concept linking** – connects related documents by identify their shared concepts.
- **Question answering** – find the best answer to a given question

文本挖掘应用程序

- 信息提取 – 识别文本中的关键短语和关系
- 主题跟踪 – 预测/推荐用户感兴趣的其他文档。
- 摘要 – 总结文档以节省读者的时间。
- 分类 – 确定文档的主要主题并放置正确的主题
- 聚类 – 对相似的文档进行分组，而无需一组预定义的类别
- 概念链接 – 通过识别相关文档的共享概念来连接相关文档。
- 问题解答 – 找到给定问题的最佳答案

Text Mining applications (cont.)

- **Intention mining/recognition/detection** – discover user intention based on comments, reviews, tweets, blogs
- **Concept mining** – extract idea and concept from large static social media
- **Sentiment Analysis** – categorize text to sentiment polarity (pos, neg, neu)
- Topic modeling – unconver the topical structure of a large collection of docs.

文本挖掘应用程序（续）

- 意图挖掘/识别/检测——发现用户意图
  基于评论、评论、推文、博客
- 概念挖掘 – 从大型静态社交媒体中提取想法和概念
- 情绪分析 – 根据情绪极性（pos 、 neg 、 neu ）对文本进行分类
- 主题建模 – 解开大量文档的主题结构。

NLP applies in Software Engineering

- **Code Generation and Understanding:** NLP techniques can be used to convert natural language commands into code and to help developers understand complex codebases.
- **Automated Documentation:** NLP can be used to generate and maintain technical documentation based on code changes.
- **Bug Tracking and Analysis:** It can assist in categorizing and prioritizing bugs by analyzing bug reports.
- **Customer Support:** NLP can power chatbots and support systems that interact with users to solve technical problems.
- **Code Reviews:** NLP can automate some aspects of code reviews by summarizing changes and identifying potential issues.

NLP 在软件工程中的应用

- 代码生成和理解：NLP 技术可用于将自然语言命令转换为代码，并帮助开发人员理解复杂的代码库。
- 自动化文档：NLP 可用于根据代码更改生成和维护技术文档。
- 错误跟踪和分析：它可以通过分析错误报告来帮助对错误进行分类和优先级排序。
- 客户支持：NLP 可以为聊天机器人和支持系统提供动力，这些系统可以与用户交互以解决技术问题。
- 代码审查：NLP 可以通过总结更改和识别潜在问题来自动化代码审查的某些方面。

Level of difficulty

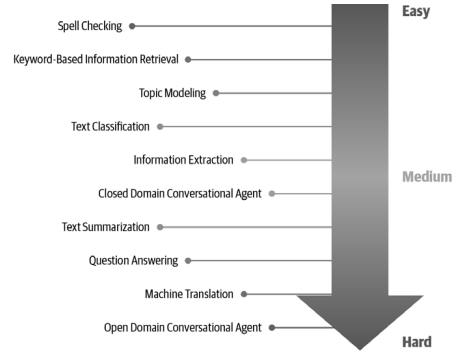


Figure 1-2. NLP tasks organized according to their relative difficulty

难度级别

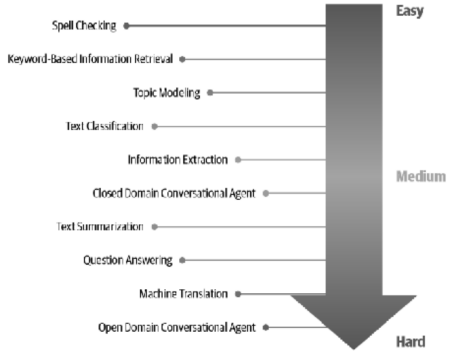


Figure 1-2. NLP tasks organized according to their relative difficulty

Intro Natural Language Processing (NLP)

- Natural language vs Programming language
  - NL - human share information with human
  - PL – human tells machines what to do
- NLP – Machine can now process natural language (i.e., interpreter)

自然语言处理 （NLP ） 简介

- 自然语言与编程语言
  - NL - 人类与人类共享信息
  - PL – 人类告诉机器该做什么
- NLP – 机器现在可以处理自然语言（即解释器）

Intro NLP (cont.)

- Subfield of AI and Computation Intelligent/linguistics.
- Try to understanding the natural human language.
- Moving forward to syntax-drive (word counting) to true understanding and processing of NLP
- Considering grammatical, semantic constraint and context.

介绍 NLP （续）

- 人工智能与计算智能/语言学的子领域。
- 尝试理解自然的人类语言。
- 向语法驱动（字数统计）迈进，以真正理解和处理 NLP
- 考虑语法、语义约束和上下文。

NLP practical applications

- **Editing** – spelling, grammar, style
- **Dialog** – Chatbot, assistant, scheduling
- **Email** – spam filter, classification, prioritization
- **Text mining** – Summarization, knowledge extraction
- **News** – event detection, fact checking, fake news detection
- **Attribution** – plagiarism detection, literacy forensics,
- **Creative writing** – Movie scripts, poetry, song lyrics.
- **Search**- web, documents, autocomplete
- **Chatbot** – Ambiguous commands, Q/A, scheduling

NLP 实际应用

- 编辑 – 拼写、语法、风格
- 对话框 – 聊天机器人、助手、日程安排
- 电子邮件 – 垃圾邮件过滤器、分类、优先级
- 文本挖掘 – 摘要、知识提取
- 新闻 – 事件检测、事实核查、假新闻检测
- 署名 – 剽窃检测、识字取证、
- 创意写作 – 电影剧本、诗歌、歌词。
- 搜索 - Web 、文档、自动完成
- 聊天机器人 – 模棱两可的命令、问答、日程安排

Intro Natural Language Processing (NLP)

- Bag-of-words (classical method)
  - Text, sentences, paragraph, or document -> words
  - Classification model <spam/legitimate>
  - One bag is filled with words found in spam messages (Viagra, stock, buy)
  - Another bag is filed with words related to user's friend or workplace.
- Human do not use words without some order or structure
  - Semantic and syntactic structure
- Text mining need to look for ways beyond the bag-of-words.

自然语言处理 （NLP ） 简介

- 词袋（经典方法）
  - 文本、句子、段落或文档 - > 个单词
  - 分类模型< 垃圾邮件/合法>
  - 一个袋子里装满了垃圾邮件中的单词（伟哥、股票、购买）
  - 另一个袋子上写着与用户的朋友或工作场所有关的文字。
- 人类不会在没有某种顺序或结构的情况下使用单词
  - 语义和句法结构
- 文本挖掘需要寻找超越文字袋的方法。

Challenge in NLP (non-practical)

- **Part of speech tagging (POS-tagging**)- identify Adverb verb, noun in the sentence.
- **Text segmentation** - Chinese/Thai/Other languages.
- **Word sense disambiguation** – a word may has more than one meaning.
- **Syntactic ambiguity** – grammar is ambiguous
- **Imperfect or irregular input** – typos , grammar errors

NLP 中的挑战（非实用）

- 词性标记（POS -tagging ）- 识别句子中的副词动词、名词。
- 文本分割 - 中文/泰语/其他语言。
- 词义消歧 – 一个词可能有多个含义。
- 句法歧义 – 语法模棱两可
- 输入不完美或不规则 – 错别字、语法错误

Text mining terminologies

- ***Unstructured data (versus structured data).*** *– human readable*
- ***Corpus*** *= dataset*
- ***Terms*** *– single word or multiword pharse from corpus*
- ***Concepts*** *– features generated from documents.*
- ***Stemming*** *– process of reducing inflected word to their root*
- ***Stop words-*** *words that are filtered out after processed.*
- ***Synonyms and polysemes*** – syntactically different/identical words
- ***Tokenizing*** *– block of text*
- ***Word frequency*** *- #time that word occur in the document*

文本挖掘术语

- 非结构化数据（与结构化数据相比）。– 人类可读
- 语料库 = 数据集
- 术语 – 语料库中的单字或多字音
- 概念 – 从文档生成的功能。
- 词干提取 – 将屈折词减少到词根的过程
- 停用词 - 处理后过滤掉的词。
- 同义词和多义词 – 句法不同/相同的词
- 标记化 – 文本块
- 单词频率 - 该单词在文档中出现 *# time*

Text mining terminologies (cont.)

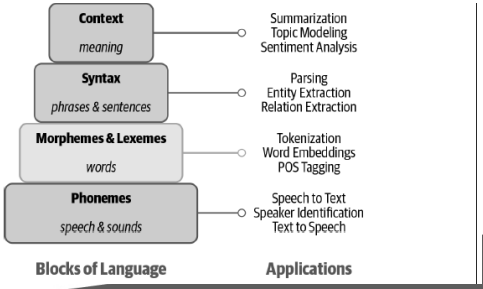
- ***Morphology*** *– form and formation of words in a language*
- ***Word frequency*** *– the number of times a word is found*

文本挖掘术语（续）

- 形态学 – 语言中单词的形式和形成
- 词频 – 找到一个词的次数

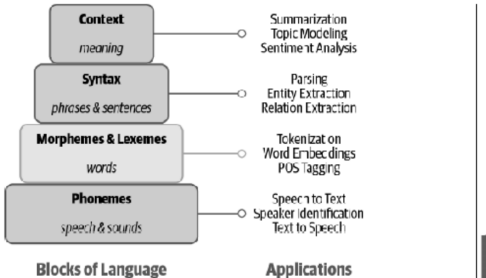
Language

- “Language is a structured system of communication that involves complex combinations of its constituent components, such as characters, words, sentences, etc.”



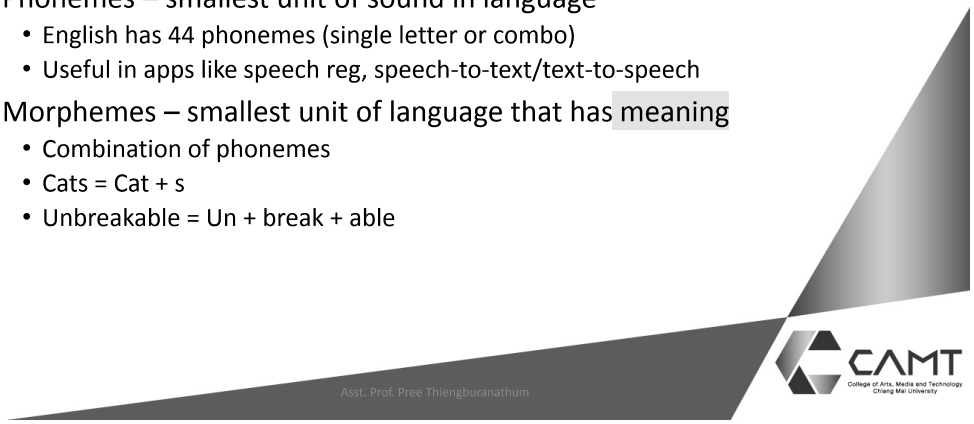
语言

- “语言是一种结构化的交流系统，涉及其组成成分的复杂组合，如字符、单词、句子等。



Language (cont.)

- Phonemes – smallest unit of sound in language
  - English has 44 phonemes (single letter or combo)
  - Useful in apps like speech reg, speech-to-text/text-to-speech
- Morphemes – smallest unit of language that has meaning
  - Combination of phonemes
  - Cats = Cat + s
  - Unbreakable = Un + break + able

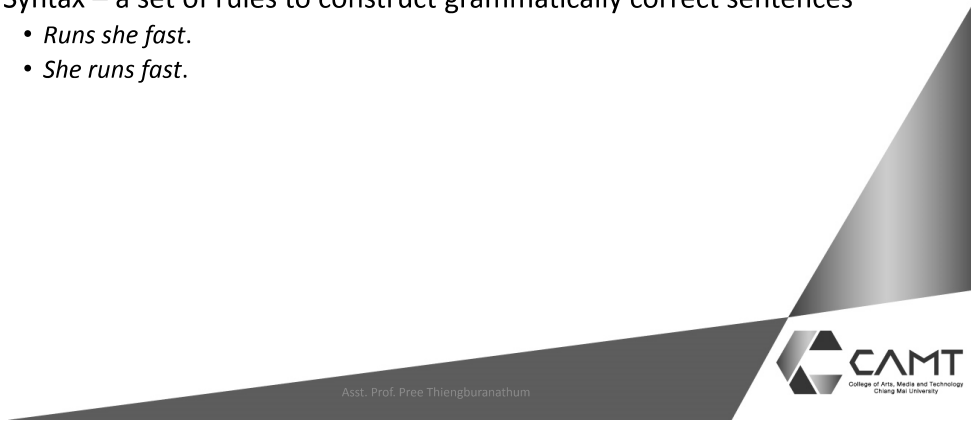


语言（续）

- 音素 – 语言中声音的最小单位
  - 英语有 44 个音素（单个字母或组合）
  - 在语音注册、语音转文本/文本转语音等应用中很有用
- 语素 – 具有意义的最小语言单位
  - 音素组合
  - 猫 = 猫 + s
  - 牢不可破 = Un + break + able

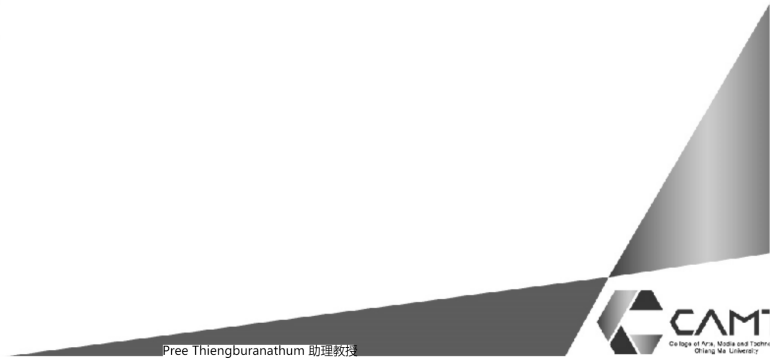
Language (cont.)

- Syntax – a set of rules to construct grammatically correct sentences
  - *Runs she fast*.
  - *She runs fast*.



语言（续）

- 语法 – 一组用于构造语法正确的句子的规则
  - 她跑得很快。
  - 她跑得很快。



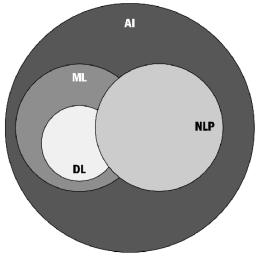
Why NLP is challenges?

- The **ambiguity** and **creativity** of human language
- Ambiguity - uncertainty of meaning. Most human languages are inherently ambiguous
  - “I made her duck.”
  - “Call me a taxi.”
  - “The teacher said the test would be difficult tomorrow.”
- Creativity – language is not a rule-based driven.
  - Various styles, dialects
  - Poem is a great example.

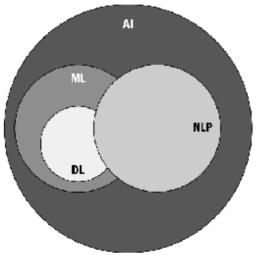
为什么NLP是挑战?

- 人类语言的模糊性和创造性
- 歧义 - 含义的不确定性。大多数人类语言本质上是模棱两可的
  - “我让她鸭子。”
  - “叫我打车。”
  - “老师说明天的考试会很困难。”
- 创造力 – 语言不是基于规则的驱动。
  - 各种风格、方言
  - 诗歌就是一个很好的例子。

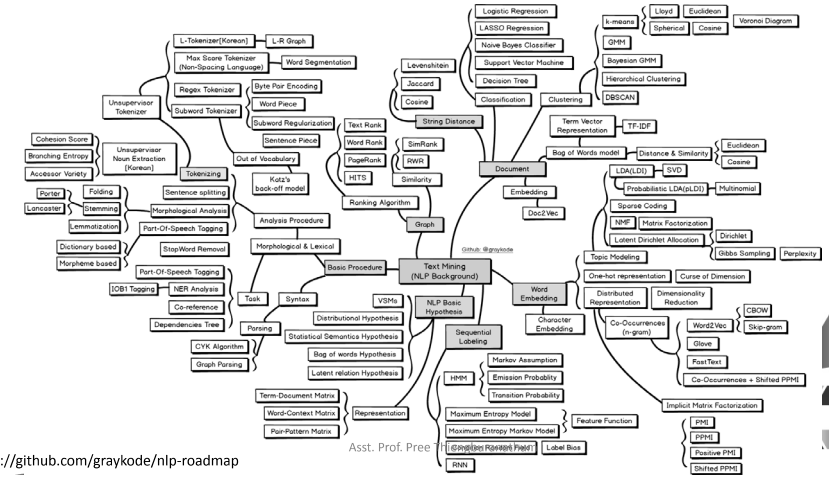
NLP related fields of studies.



NLP相关研究领域。



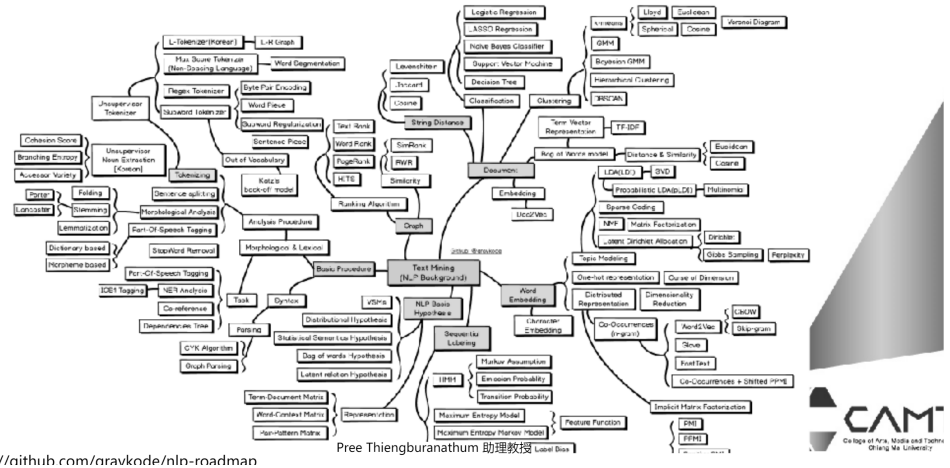
NLP road map



Source:https://github.com/graykode/nlp-roadmap

Asst. Prof. Pree Thiengburanathum

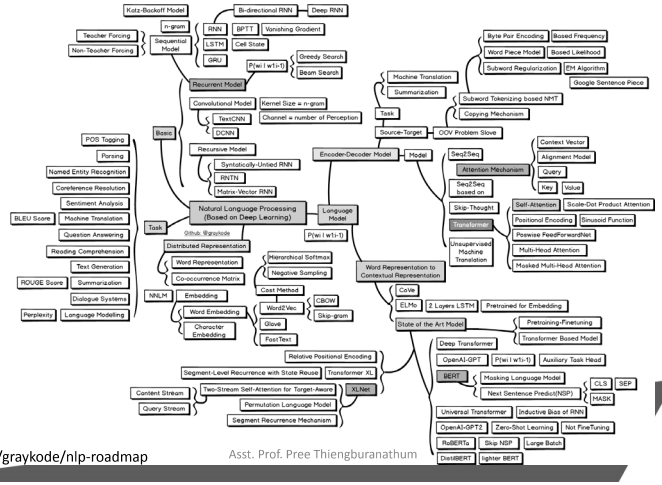
NLP 路线图



Source:https://github.com/graykode/nlp-roadmap

Pree Thiengburanathum 助理教授

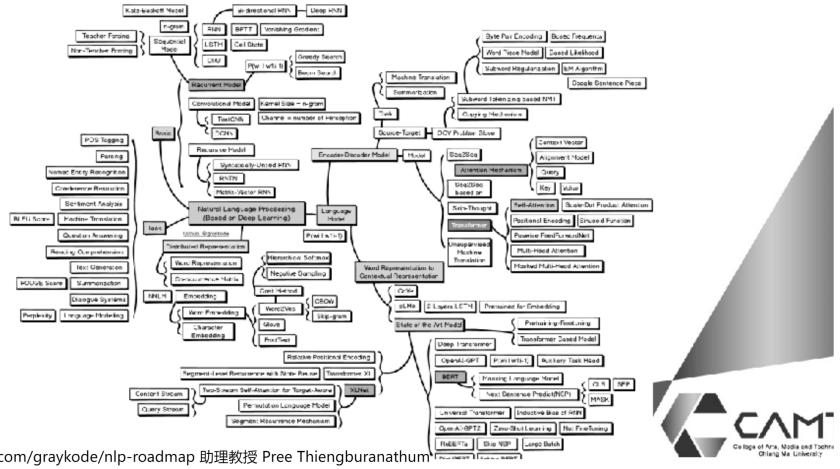
NLP road map (cont.)



Source: https://github.com/graykode/nlp-roadmap

Asst. Prof. Pree Thiengburanathum

NLP路线图（续）



资料来源：https://github.com/graykode/nlp-roadmap 助理教授 Pree Thiengburanathum

Case study 2021 / NLP application in Action

- AI chat bot with Stress detection
- Read more at:
  - https://arxiv.org/abs/1911.00133

案例研究 2021 / NLP 应用在行动

- 具有压力检测功能的 AI 聊天机器人
- 更多信息，请访问：
  - https://arxiv.org/abs/1911.00133

Conversational Agents

- AKA. Dialogue System, Dialogue Agents, Chatbots
- Personal Assistants on phones or other platforms
  - Alexa, SIRI, Google Assistant, Cortana
- Playing music, setting times and clock
- Chatting for fun
- Booking, scheduling reservation
- Clinical uses for mental heath (like my project)

对话代理

- 即。对话系统、对话代理、聊天机器人
- 手机或其他平台上的个人助理
  - Alexa、SIRI、Google Assistant、Cortana
- 播放音乐、设置时间和时钟
- 聊天的乐趣
- 预订、安排预订
- 精神健康的临床用途（如我的项目）

Conversational Agents (cont.)

- Chatbots
  - Mimic informal human chatting
  - For fun, even for therapy
- Task-based
  - Personal assistant
  - Book seat in restaurant, movie theater, flights

会话代理（续）

- 聊天机器人
  - 模仿非正式的人类聊天
  - 为了好玩，甚至为了治疗
- 基于任务
  - 私人助理
  - 预订餐厅、电影院、航班座位

Chatbot Arch.

- Rule-based
  - Pattern-action rules (ELIZA)
- Corpus-based(data-driven)
  - Information Retrieval
  - Model-based (My chatbot)

聊天机器人拱门。

- 基于规则
  - 模式操作规则 （ELIZA）
- 基于语料库（数据驱动）
  - 信息检索
  - 基于模型（我的聊天机器人）

GPT-Baker

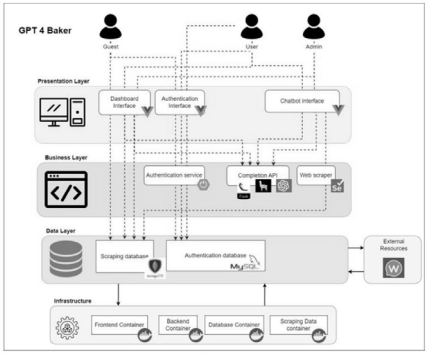


Figure 1: GPT 4 Baker System Architecture

GPT-贝克

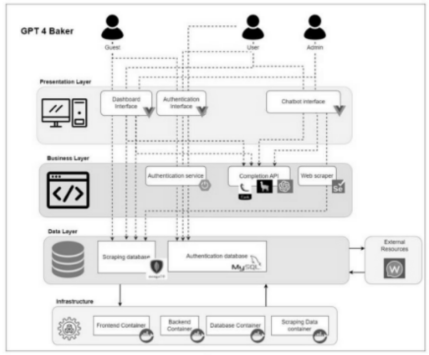


Figure 1: GPT 4 Baker System Architecture

Class activity (if we have time)

- https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

- Answer the following question:
- How many rows/sample?
- What is the longest sample?
- How many word?
- What is the average word length?

课堂活动（如果我们有时间）

- https://www.kaggle.com/datasets/uciml/sms-spam-collectiondataset

- 回答以下问题：
- 每个样本有多少行?
- 最长的样本是多少?
- 多少字?
- 平均字长是多少?

Natural Language Processing for SE
66/2
NLP Overview (2)

Asst. Prof. Pree Thiengburanathum

面向 SE 的自然语言处理

66/2

NLP概览 （2）

Pree Thiengburanathum 助理教授

Where we are now

我们现在所处的位置

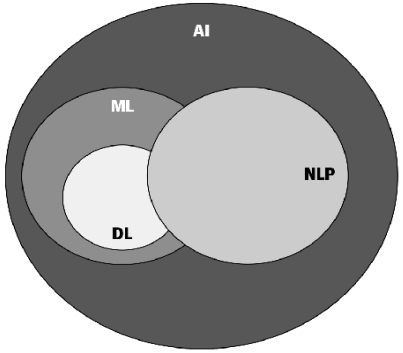
Today's Agenda

- Use of Information Technology in Business
- DS methodology
- NLP-based project proposal

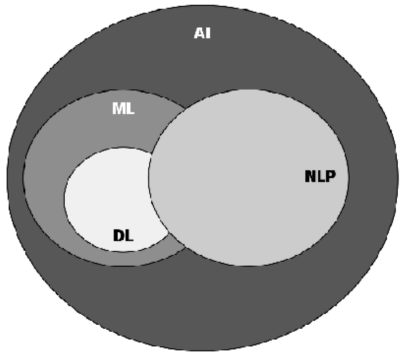
今日议程

- 资讯科技在商业中的应用
- DS 方法论
- 基于NLP的项目建议书

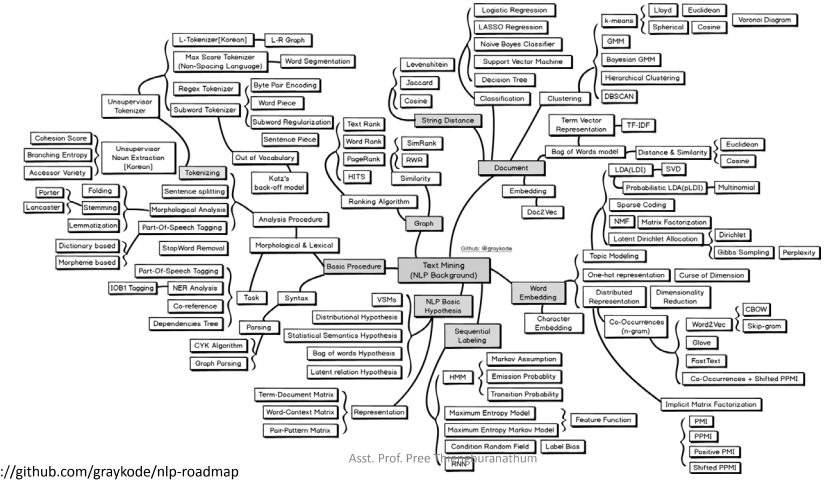
NLP related fields of studies.



NLP相关研究领域。



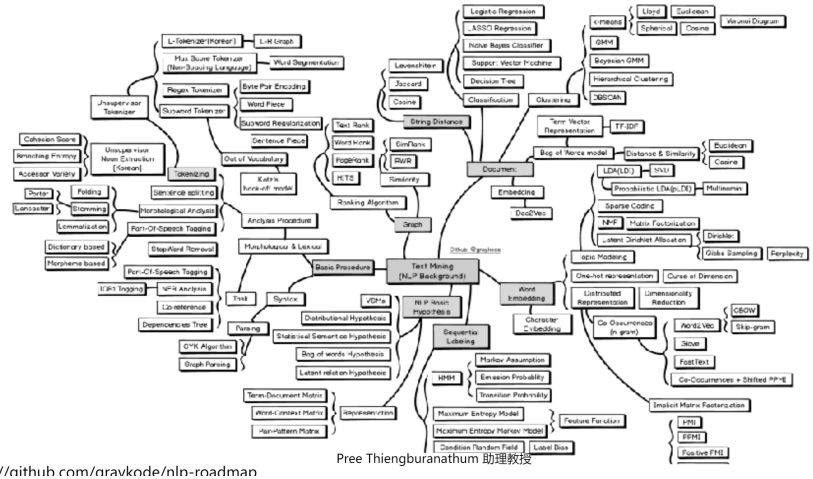
NLP road map



Source:https://github.com/graykode/nlp-roadmap

Asst. Prof. Pree Thiengburanathum

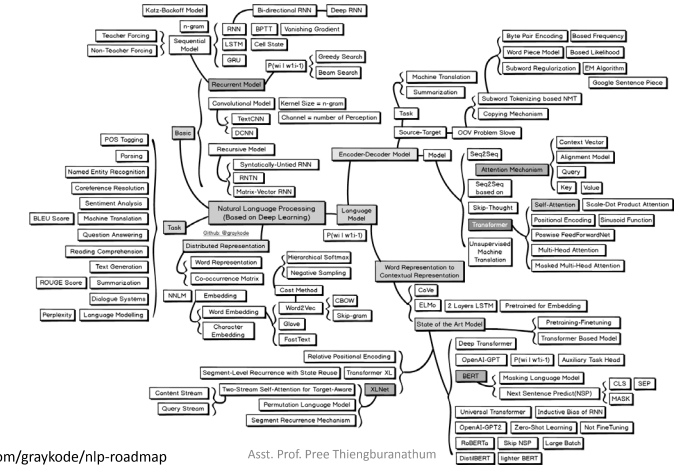
NLP 路线图



Source:https://github.com/graykode/nlp-roadmap

Pree Thiengburanathum 助理教授

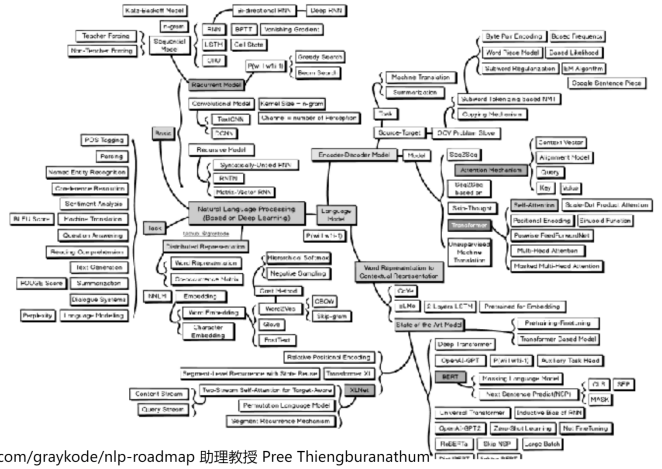
NLP road map (cont.)



Source: https://github.com/graykode/nlp-roadmap

Asst. Prof. Pree Thiengburanathum

NLP路线图（续）



资料来源：https://github.com/graykode/nlp-roadmap 助理教授 Pree Thiengburanathum

Why NLP is challenges?

- The **ambiguity** and **creativity** of human language
- Ambiguity - uncertainty of meaning. Most human languages are inherently ambiguous
  - “I made her duck.”
  - “Call me a taxi.”
  - “The teacher said the test would be difficult tomorrow.”
- Creativity – language is not a rule-based driven.
  - Various styles, dialects
  - Poem is a great example.

为什么NLP是挑战?

- 人类语言的模糊性和创造性
- 歧义 - 含义的不确定性。大多数人类语言本质上是模棱两可的
  - “我让她鸭子。”
  - “叫我打车。”
  - “老师说明天的考试会很困难。”
- 创造力 – 语言不是基于规则的驱动。
  - 各种风格、方言
  - 诗歌就是一个很好的例子。

Challenge in NLP (non-practical)

- **Part of speech tagging (POS-tagging**)- identify Adverb verb, noun in the sentence.
- **Text segmentation** - Chinese/Thai/Other languages.
- **Word sense disambiguation –** a word may has more than one meaning.
- **Syntactic ambiguity** – grammar is ambiguous
- **Imperfect or irregular input** – typos , grammar errors

NLP中的挑战（非实用）

- 词性标记（POS-tagging）- 识别句子中的副词动词、名词。
- 文本分割 - 中文/泰语/其他语言。
- 词义消歧 – 一个词可能有多个含义。
- 句法歧义 – 语法模棱两可
- 输入不完美或不规则 – 错别字、语法错误

Developer vs Data scientist

- Somewhat common (good at designing and building complex system, with tools and frameworks)
- Software dev. -> well-defined components
- Data Science -> work on component isn't well defined (i.e., data pre-processing, analysis)
- Data Science -> create system that rely on statically results

开发人员与数据科学家

- 有点普通（擅长设计和构建复杂的系统，使用工具和框架）
- 软件开发 ->定义明确的组件
- 数据科学 - >组件上的工作没有很好地定义（即，数据预处理、分析）
- 数据科学 - >创建依赖于静态结果的系统

Developer vs Data scientist (cont.)

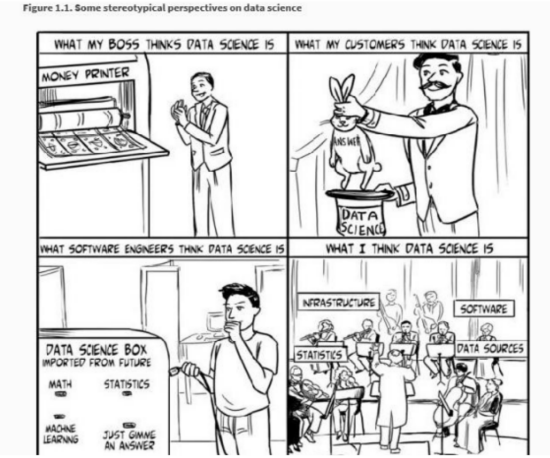
Dealing with uncertainty is often what separates the role of a data scientist from that of a software developer.

开发人员 vs 数据科学家（续）

处理不确定性通常是将数据科学家的角色与软件开发人员的角色区分开来的原因。

The role of data scientist

数据科学家的角色



Goal of Data science

- “Find a patterns” Kenny Cheung
- “Turn **data** to data product”

数据科学的目标

- 《寻找模式》 Kenny Cheung
- “将数据转化为数据产品”

Think like data science

If a human expert can easily create a pattern in his or her own mind, it is generally not worth the time and effort of using data science to “discover” it.

像数据科学一样思考

If a human expert can easily create a pattern in his or her own mind, it is generally not worth the time and effort of using data science to “discover” it.

Think like data science

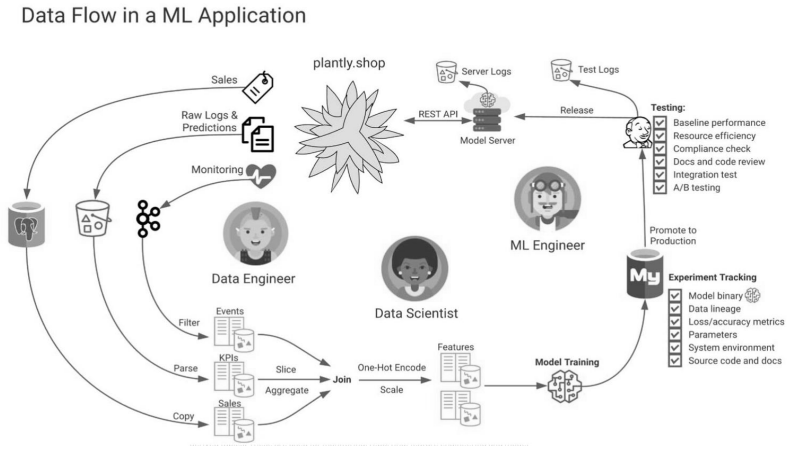
The key to success is getting the right data and finding the right attributes.

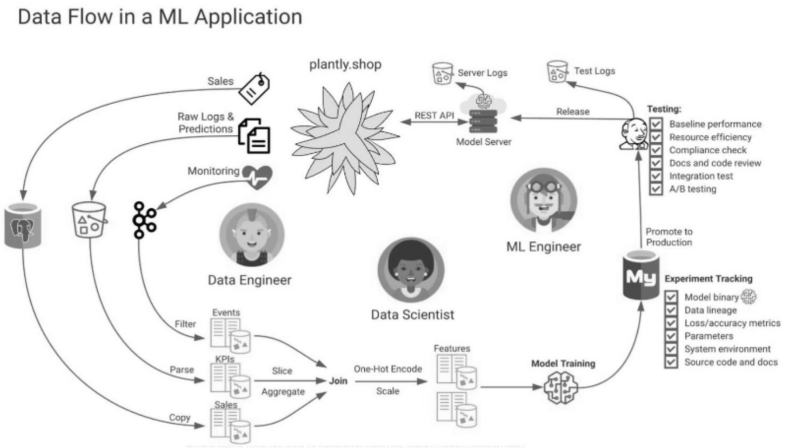
像数据科学一样思考

The key to success is getting the right data and finding the right attributes.

Example of Data Flow in process (back-end)

进程中的数据流示例（后端）

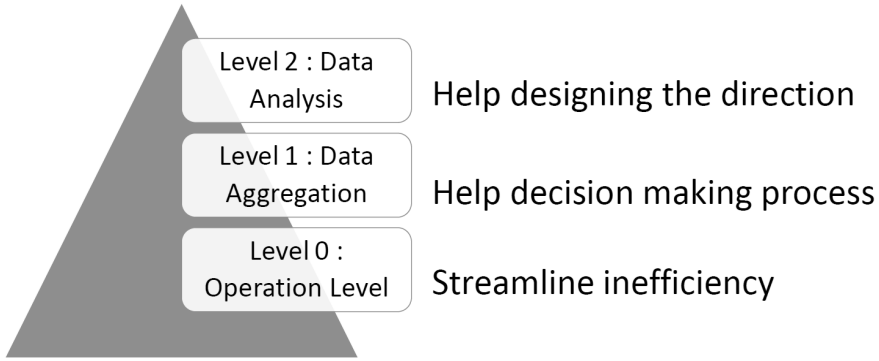




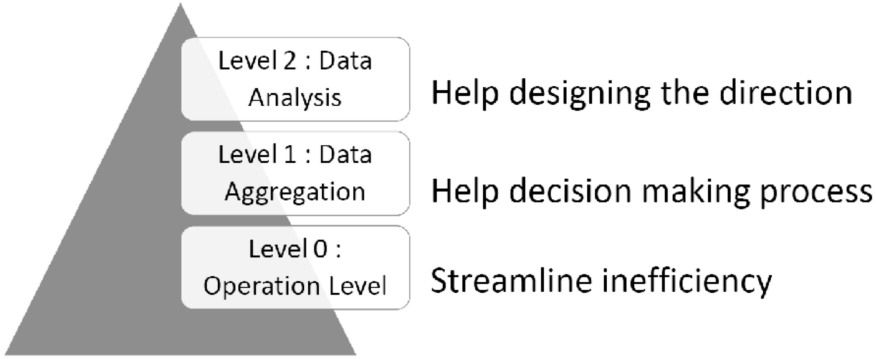
Data Science Usage and Application

数据科学的使用和应用

Level of Information technology usage

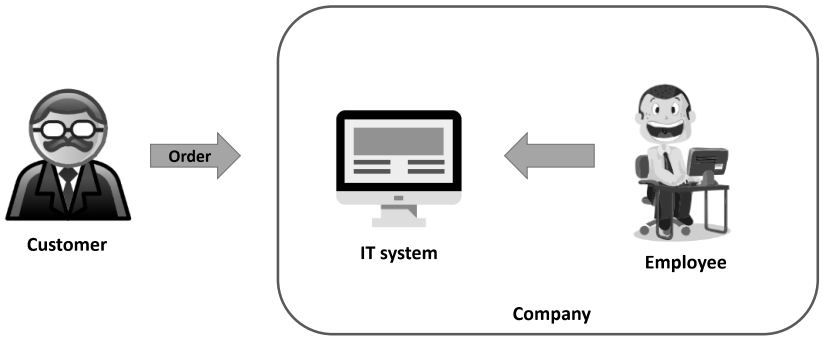


信息技术使用水平



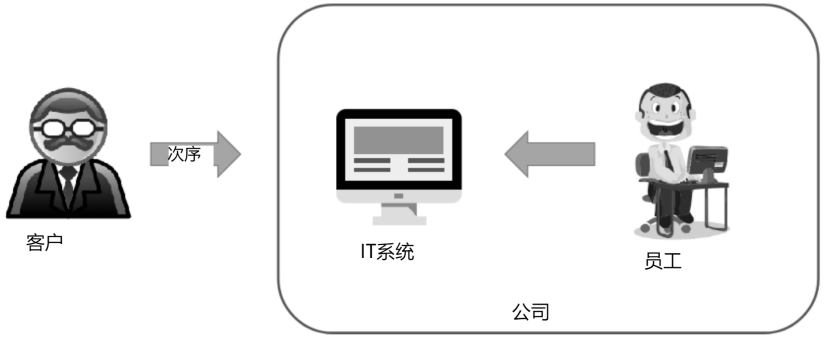
Level of Information Technology Usage (Where most SE project are)

Level 0: Operation level



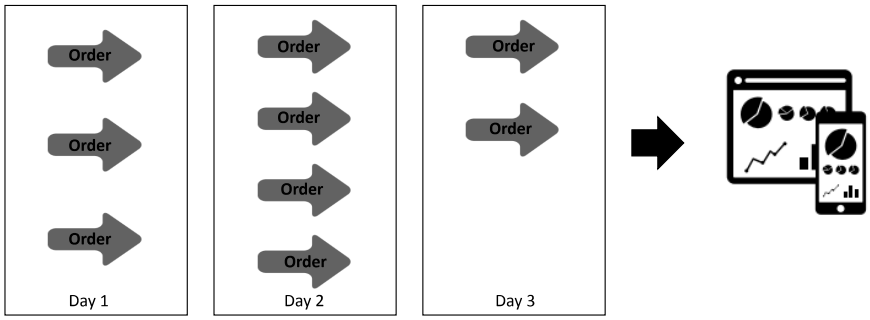
信息技术使用水平（大多数 SE 项目所在位置）

级别 0：操作级别



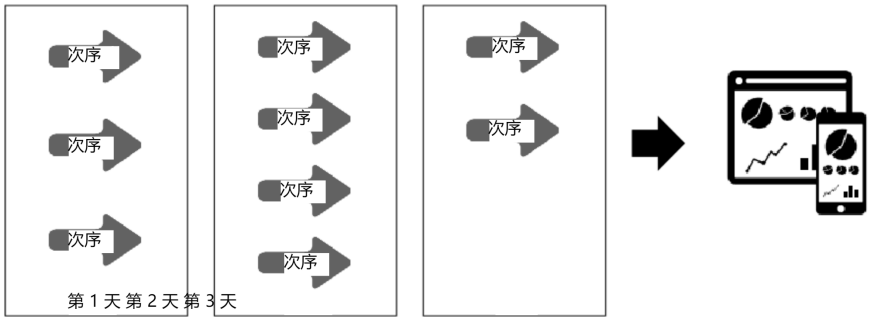
Level of Information Technology Usage

Level 1: Data aggregation



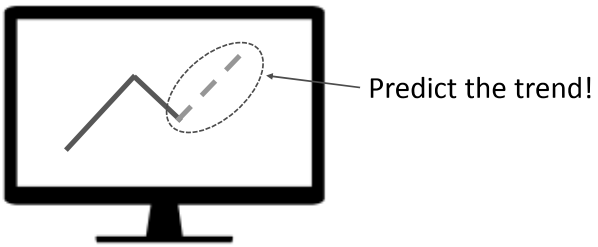
资讯科技使用水平

级别 1：数据聚合



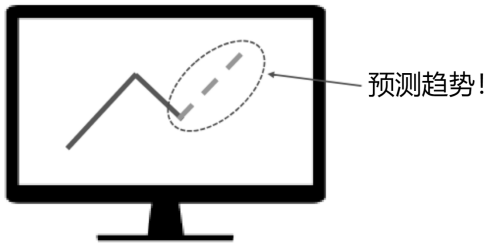
Level of Information Technology Usage

Level 2: Data analysis



资讯科技使用水平

第 2 级：数据分析



Benefit of the DS tools for **Business** tools

DS 的优势

工具

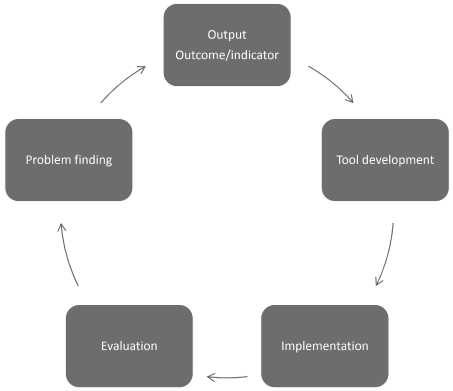
业务工具

对于企业

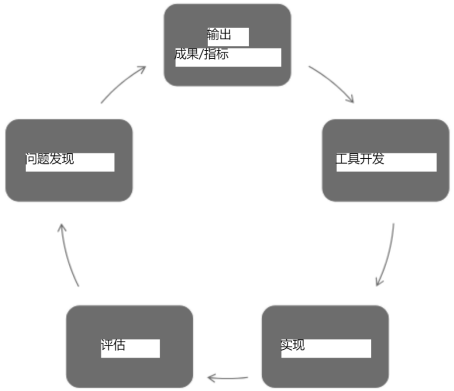
Use of Information Technology in Business (Data Science)

资讯科技在商业中的应用（数据科学）

Usage of Information Technology in Business



资讯科技在商业中的应用



Problem identification

- Review the environment or contexts of the problem
- Significant academic vs. practical impact
- Talk to your stakeholders

问题识别

- 查看问题的环境或上下文
- 重大的学术与实践影响
- 与您的利益相关者交谈

Outputs, Outcomes and Indicators identification

产出、成果和指标的确定

Outputs

- Outputs are a quantitative summary of an activity
- I.e., the activity is ‘we provide training’ and the output is ‘we trained 50 people to NVQ level 3’. An output tells you an activity has taken place.

Activity	Output
CV checking drop ins	Number of people getting support with their CV
Parenting skills classes	Number of people attending parenting skills classes
Cardio vascular health checks	Number of health checks conducted

Source: https://www.kent.gov.uk/__data/assets/pdf_file/0009/41499/Community-Mental-Health-and-Wellbeing-Service-Market-Engagement-event-Julia-Slay-presentation.pdf

输出

- 产出是活动的定量摘要
- 即，活动是“我们提供培训”，输出是“我们培训了 50 人达到 NVQ 3 级”。输出告诉您已执行活动
地方。

Activity	Output
CV checking drop ins	Number of people getting support with their CV
Parenting skills classes	Number of people attending parenting skills classes
Cardio vascular health checks	Number of health checks conducted

来源：https://www.kent.gov.uk/__data/assets/pdf_file/0009/41499/Community-Mental-Health-and-Wellbeing-Service-Market-Engagementevent-Julia-Slay-presentation.pdf

Outcome

- The **change that occurs** as a result of an activity (e.g., improved well-being of training participants)
- Outcome : change direction + target component
- Need to be cleared
- Sometimes it takes years for outcomes to take place

结果

- 由于活动而发生的变化（例如，改善培训参与者的幸福感）
- 结果：改变方向+目标分量
- 需要清除
- 有时需要数年时间才能取得成果

Example of outcomes:

- Reduce labor cost in organization
- Reduce computation time during training model
- Increase predictive accuracy power of the model.
- Increase usability and user experience of the recommendation system

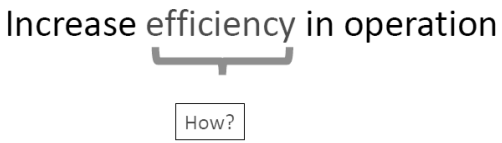
结果示例：

- 降低组织中的人工成本
- 减少训练模型期间的计算时间
- 提高模型的预测准确性。
- 提高推荐系统的可用性和用户体验

Outcome (cont.)

- Good outcome

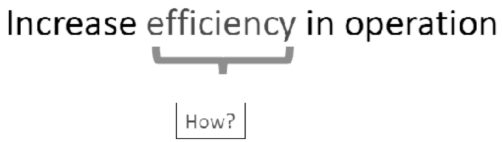
- Poor outcome



结果（续）

- 良好的结果

- 不良预后



Indicators identification

- To identify the desirable outcome in term of processes or results (i.e., to measure something)
- Usually present in in number of percentage (ratio of, percentage of)
- Indicators can be shared: reduced school drop-out rates = graduation rate
- Good indicators must be simple, reliable and valid.
- Stakeholders are often the best people to help you identify indicators, so ask them how they know that change has happened for them

指标识别

- 根据过程或结果（即衡量某事）确定理想的结果
- 通常以百分比数（ratio of，百分比）的形式出现
- 可以共享的指标：降低的辍学率=毕业率
- 好的指标必须简单、可靠和有效。
- 利益相关者通常是帮助您确定指标的最佳人选，因此请询问他们如何知道已经发生了变化

Example of indicators

Outcome	Indicator
Increased infant breastfeeding	Number & percentage of mothers who are exclusively breastfeeding up to six months of age.
Improved work attendance by District Officials	Number of work days attended per year by District Officials
Less grade repetition	Pass rate
Beneficiaries access financial support for tertiary education	Number and percentage of beneficiaries that have bursaries and student loans

指标示例

Outcome	Indicator
Increased infant breastfeeding	Number & percentage of mothers who are exclusively breastfeeding up to six months of age.
Improved work attendance by District Officials	Number of work days attended per year by District Officials
Less grade repetition	Pass rate
Beneficiaries access financial support for tertiary education	Number and percentage of beneficiaries that have bursaries and student loans

Solution Development

- The objective of this step is to develop a tool to solve the problem.
- The first step is to develop the **strategy**.
- The problem solving strategy is a conceptual framework to solve the problem.
- This step does not include the specification of the solution.
- The second step is to develop the **solution**.
- 2 types of solution : develop by yourself or use the existing solution.

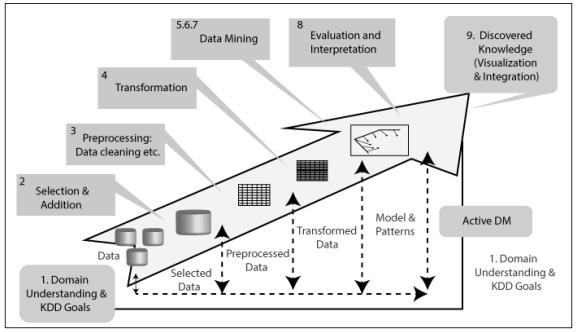
解决方案开发

- 此步骤的目的是开发一种解决问题的工具。
- 第一步是制定战略。
- 问题解决策略是解决问题的概念框架。

- 此步骤不包括解决方案的规范。
- 第二步是开发解决方案。
- 2种解决方案：自行开发或使用现有解决方案。

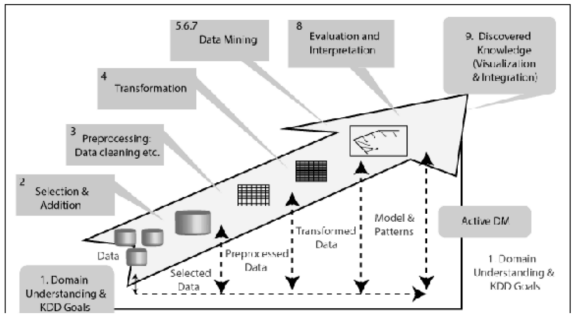
Knowledge Discovery in Databases Process (KDD)

- is the process of finding valid, novel, useful and understandable patterns in data, to verify hypothesis of the user or to describe/predict the future behavior of some event

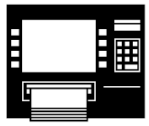


数据库过程中的知识发现 （KDD）

- 是在数据中寻找有效、新颖、有用和可理解的模式的过程，以验证用户的假设或描述/预测某些事件的未来行为



Problem identification



Problem: The institute rents the building and the labor cost is the second highest cost.

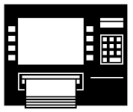
Activities	Outputs
Deployed ATM across the region.	Number of ATM machines being deployed Number of people have used
Online banking	Number of transaction

问题识别

问题：研究所租用建筑物，劳动力成本是第二高的成本。

活动成果	
部署了跨 地区。 使用过网上银行的人数 交易数量	ATM机数量 部署

Outcomes and indicators



Outcomes	Indicators
Reduce the cost of labors	Percentage of cost of labors / months
Reduce the cost of renting the building	Percentage of cost of renting spending / months

成果和指标

成果指标	
降低人工成本 人工成本百分比 /	月份
降低建筑物的租赁成本	租房支出成本的百分比/月

Solution Development

Problem

The institute rents the building and the labor cost is the second highest cost.

Strategy

Develop a novel approach which does not need to rent and use less employee.

解决方案开发

Problem

The institute rents the building and the labor cost is the second highest cost.

Strategy

Develop a novel approach which does not need to rent and use less employee.

Data science simple process in 1997

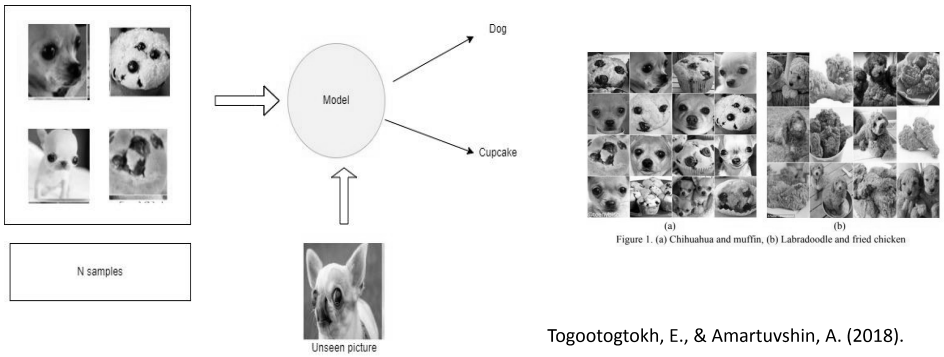
In 1997, University of Michigan statistics professor C.F. Jeff Wu

1997 年的数据科学简单过程

1997年，密歇根大学统计学教授C.F. Jeff Wu

Supervise learning

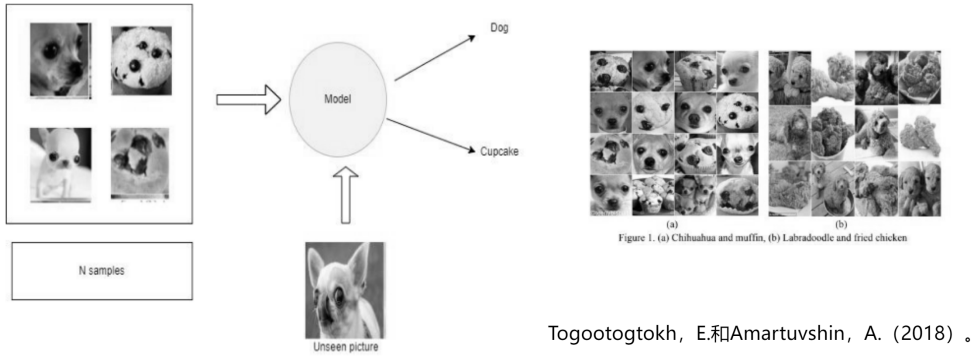
- **human intervene (help labeling)**



Togootogtokh, E., & Amartuvshin, A. (2018).

监督学习

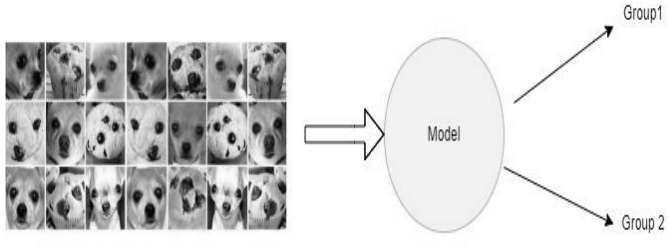
- 人工干预（帮助标记）



Togootogtokh，E.和Amartuvshin，A.（2018）。

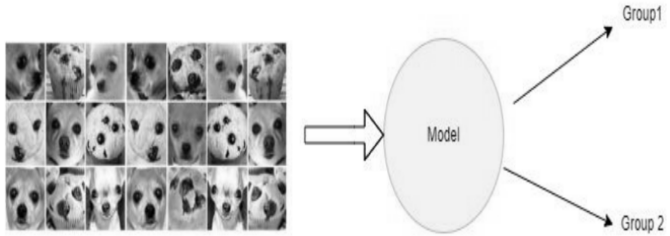
Unsupervised learning

- let the model work on its own (deal with un-labelled data)
- find similarities and differences between data points



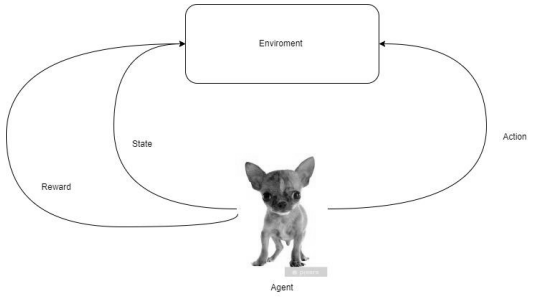
无监督学习

- 让模型独立工作（处理未标记的数据）
- 查找数据点之间的相似性和差异性



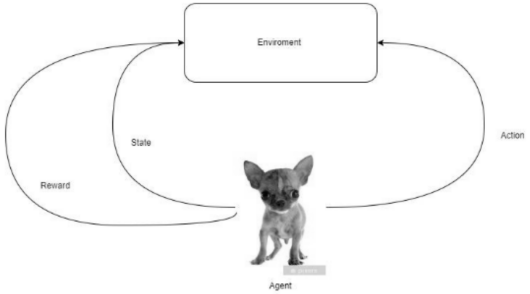
Re-enforcement learning

- Learn from mistakes
- Rewards and punishments, max(total reward of the agent)



再执法学习

- 从错误中吸取教训
- 奖惩，max（代理总奖励）

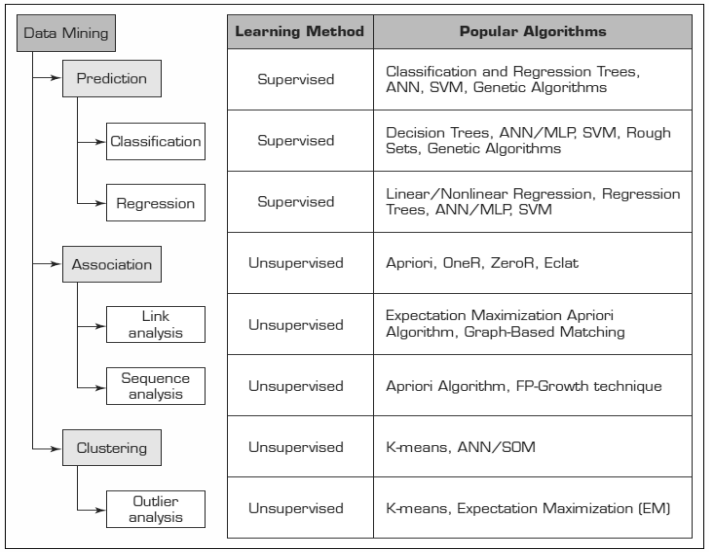


Data science -The methods

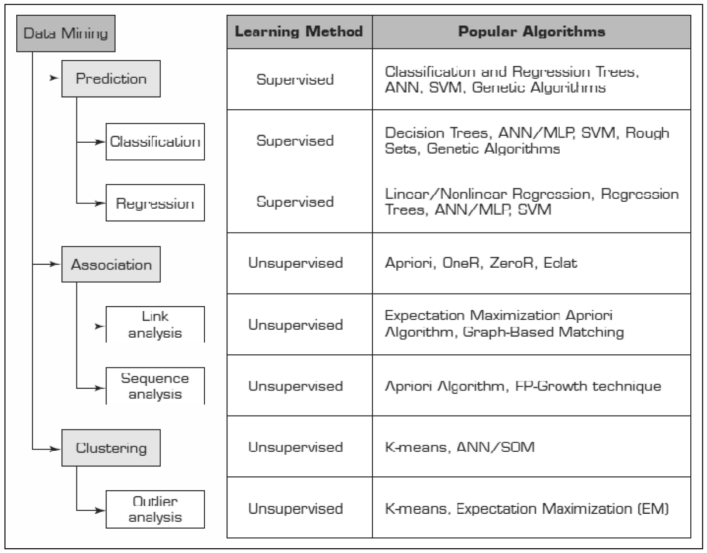
- **Predictions**- predict the winner of football match
- **Associations** – find the commonly co-occurring group of things (beer and chips in shelf)
- **Clusters** – identify natural grouping of things based their own attributes.
- **Sequential relationships** – discover time-order event. (banking customer has c-account will open open i-account with in a period)

数据科学 - 方法

- 预测 - 预测足球比赛的获胜者
- 关联 – 找到通常同时出现的一组事物（货架上的啤酒和薯条）
- 聚类 – 根据事物自身的属性识别事物的自然分组。
- 顺序关系 – 发现时间顺序事件。（银行客户拥有C-Account，将在一段时间内开立I-Account）



Learning Method	Popular Algorithms
Supervised	Classification and Regression Trees, ANN, SVM, Genetic Algorithms
Supervised	Decision Trees, ANN/MLP, SVM, Rough Sets, Genetic Algorithms
Supervised	Linear/Nonlinear Regression, Regression Trees, ANN/MLP, SVM
Unsupervised	Apriori, OneR, ZeroR, Eclat
Unsupervised	Expectation Maximization Apriori Algorithm, Graph-Based Matching
Unsupervised	Apriori Algorithm, FP-Growth technique
Unsupervised	K-means, ANN/SOM
Unsupervised	K-means, Expectation Maximization (EM)



Learning Method	Popular Algorithms
Supervised	Classification and Regression Trees, ANN, SVM, Genetic Algorithms
Supervised	Decision Trees, ANN/MLP, SVM, Rough Sets, Genetic Algorithms
Supervised	Linear/Nonlinear Regression, Regression Trees, ANN/MLP, SVM
Unsupervised	Apriori, OneR, ZeroR, Eclat
Unsupervised	Expectation Maximization Apriori Algorithm, Graph-Based Matching
Unsupervised	Apriori Algorithm, FP-Growth technique
Unsupervised	K-means, ANN/SOM
Unsupervised	K-means, Expectation Maximization (EM)

Predictions

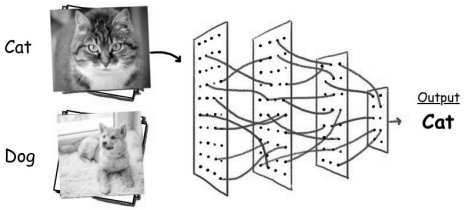
- Guessing, predicting, forecasting, and recommending
- Tell the nature of future occurrences of certain events based on what has happened in the past
- I.e. forecasting the absolute temperature of a day.

预测

- 猜测、预测、预测和推荐
- 根据过去发生的事情，告诉某些事件未来发生的性质
- 即预测一天的绝对温度。

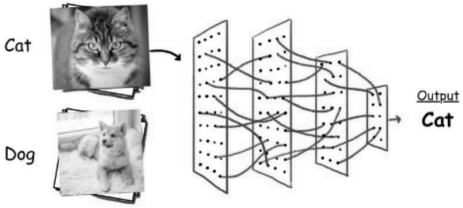
Classification

- Most **common** of all data mining tasks
- A **forced choices** or **known choices**.
- Analyze historical data and generate a **predictive model**.
- Hope that the model can be used to predict the future unclassified records
- Common classification algorithms – NN, DT, Logistic regression



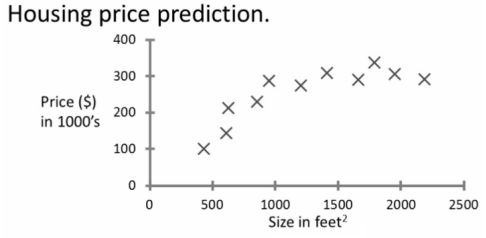
分类

- 所有数据挖掘任务中最常见的
- 被迫选择或已知选择。
- 分析历史数据和生成预测模型。
- 希望该模型可用于预测未来的未分类记录
- 常见分类算法 – NN、DT、Logistic 回归



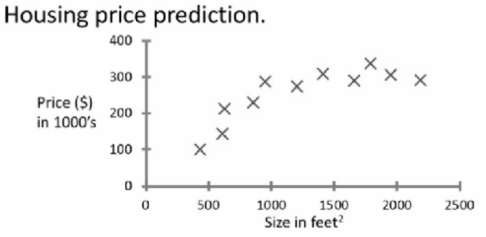
Regression

- To predict value of dependent variable, based on its relationship with values of at least one independent variable.
- Explain the impact of changes in an independent variable on the dependent variable.



回归

- 根据因变量与至少一个自变量的值的关系来预测因变量的值。
- 解释自变量变化对因变量的影响。



Clusters

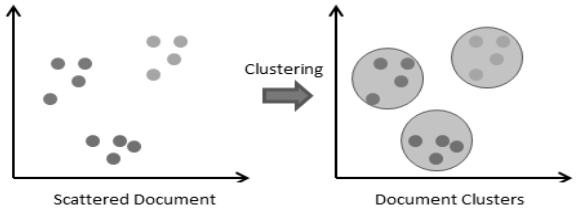
- Identify natural groupings of things based on their known characteristics,
- I.e. assigning customers in different segments based on their demographics and past purchase behaviors.

集群

- 根据事物的已知特征识别事物的自然分组，
- 即根据客户的人口统计数据和过去的购买行为将客户分配到不同的细分市场。

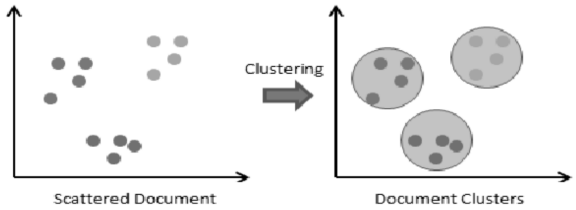
Clustering

- Partitions a collection of things
- E.g. objects, events, etc.
- Class label are unknown



聚类

- 对事物的集合进行分区
- 例如，对象、事件等。
- 类标签未知



Associations

- Find association among your problem attributes or variables
- E.g. Find relations such as a patient with high-blood-pressure is more likely to have heart-attack disease.
- E.g. Find a products that customers usually purchased together.

协会

- 查找问题属性或变量之间的关联
- 例如，找到诸如高血压患者更有可能患有心脏病发作疾病之类的关系。

- 例如，查找客户通常一起购买的产品。

Association Rules

- Also known as **market basket analysis**
- Association rules helps uncover relationship between items from large databases
- C1 – {Milk, Eggs, Sugar, Bread}
- C2 – {Milk, Eggs, Cereal, Bread{
- C3 – {Eggs, Sugar}
- Find associations/correlation between the different items that customers place in their basket? Which product are bought together?
- *Apriori* algorithm method
  - Frequent itemset
    - Itemset construction
    - Support count
  - Associate rules

关联规则

- 也称为市场篮分析
- 关联规则有助于发现大型数据库中项目之间的关系
- C1 – {牛奶、鸡蛋、糖、面包}
- C2 – {牛奶、鸡蛋、麦片、面包{
- C3 – {鸡蛋、糖}
- 寻找客户放入购物篮的不同商品之间的关联/相关性？哪些产品是一起购买的？
- Apriori算法方法
  - 常用项集
    - 项目集构造
    - 支持计数
  - 关联规则

Sequence analysis

- Discover time-ordered events.
- i.e. predicting that an existing banking customer who already has a checking account will open a savings account followed by an investment account within a year.
- Gene prediction
- Protein structure prediction
- Heath Informatics

序列分析

- 发现按时间排序的事件。
- 即预测已经拥有支票账户的现有银行客户将在一年内开设储蓄账户，然后开设投资账户。
- 基因预测
- 蛋白质结构预测
- 希思信息学

Assessment

- **Predictive accuracy** – with unseen data how well the model perform in terms of %
- **Speed** – computation cost when constructing and using the model
- **Robustness** – Giving noisy data, can the model still make reasonable prediction
- **Scalability** – how's about with larger data?
- **Interpretability** – level of understanding

评估

- 预测准确性 – 使用看不见的数据，模型的性能（以百分比表示）
- 速度 – 构建和使用模型时的计算成本
- 鲁棒性 – 给出嘈杂的数据，模型是否仍能做出合理的预测
- 可扩展性 – 如何处理更大的数据?
- 可解释性 – 理解水平

Estimating the true accuracy of models

$$(True\ Classification\ Rate)_i = \frac{(True\ Classification)_i}{\sum_{i=1}^{n}(False\ Classification)_i}$$

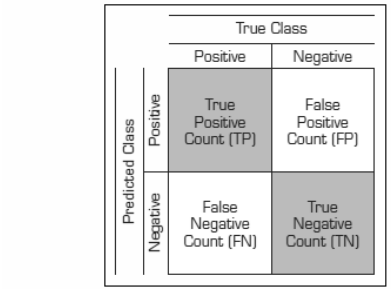
$$(Overall\ Classifier\ Accuracy)_i = \frac{\sum_{i=1}^{n}(True\ Classification)_i}{Total\ Number\ of\ Cases}$$

估计模型的真实准确性

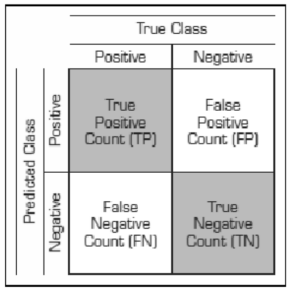
$$(True\ Classification\ Rate)_i = \frac{(True\ Classification)_i}{\sum_{i=1}^{n}(False\ Classification)_i}$$

$$(Overall\ Classifier\ Accuracy)_i = \frac{\sum_{i=1}^{n}(True\ Classification)_i}{Total\ Number\ of\ Cases}$$

Confusion matrix (getting more insight)



混淆矩阵（获得更多见解）



Estimating the error of regression models

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

Where,

$\hat{y}$ – *predicted value of y*

$\bar{y}$ – *mean value of y*

估计回归模型的误差

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

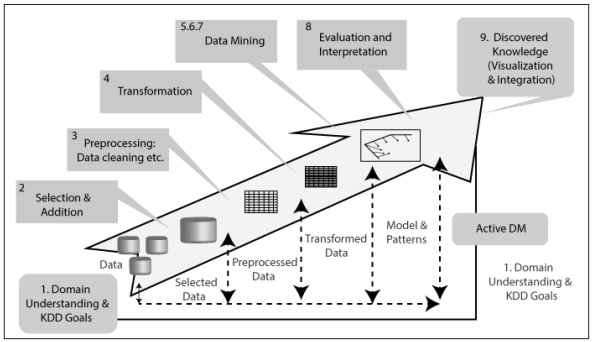
$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

Where,

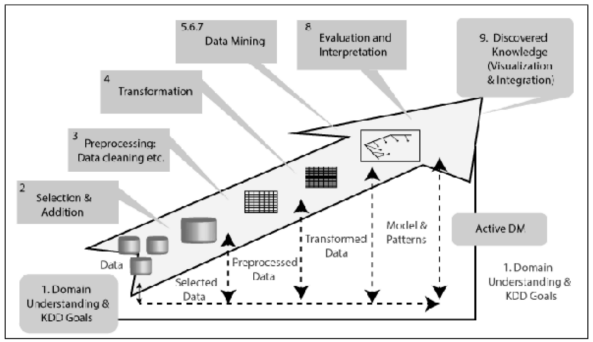
$\hat{y}$ – *predicted value of y*

$\bar{y}$ – *mean value of y*

Your toy dataset (SMS dataset)



您的玩具数据集（短信数据集）



Your toy dataset (SMS dataset)

In [2]:
```
df= pd.read_csv("/kaggle/input/sms-spam-collection-dataset/spam.csv",encoding='ISO-8859-1')
df
```

Out[2]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN
...	...	...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

5572 rows × 5 columns

您的玩具数据集（短信数据集）

In [2]:
```
df= pd.read_csv("/kaggle/input/sms-spam-collection-dataset/spam.csv",encoding='ISO-8859-1')
df
```

Out[2]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN
...	...	...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

5572 rows × 5 columns

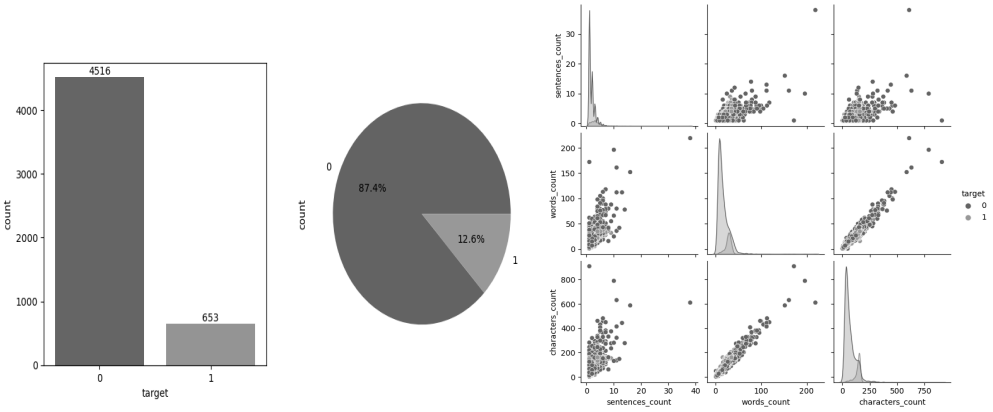
Your toy dataset (SMS dataset)

```
In [3]: # Drop unnecessary columns from the DataFrame

        columns_to_drop = ["Unnamed: 2", "Unnamed: 3", "Unnamed: 4"]
        df.drop(columns=columns_to_drop, inplace=True)

In [4]: # Rename the columns
        df.columns = ['label', 'message']

In [5]: df.shape

Out[5]: (5572, 2)
```

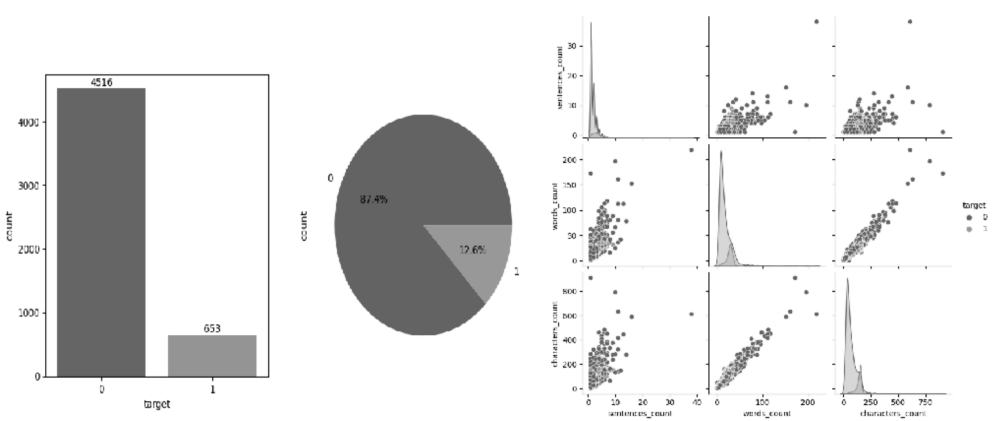
您的玩具数据集（短信数据集）

```
In [3]: # Drop unnecessary columns from the DataFrame

        columns_to_drop = ["Unnamed: 2", "Unnamed: 3", "Unnamed: 4"]
        df.drop(columns=columns_to_drop, inplace=True)

In [4]: # Rename the columns
        df.columns = ['label', 'message']

In [5]: df.shape

Out[5]: (5572, 2)
```

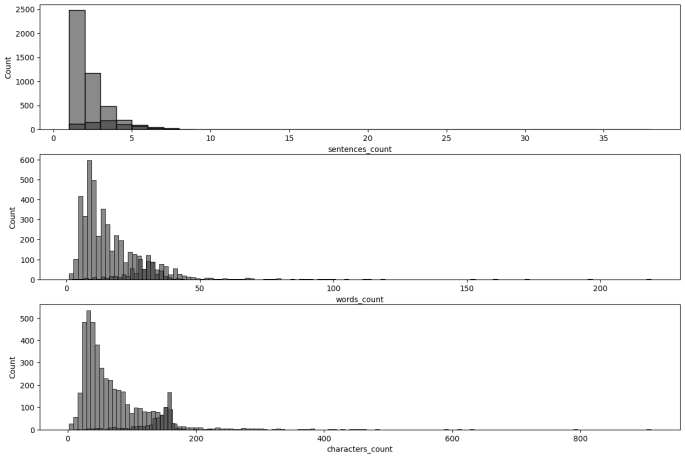
Your toy dataset (SMS dataset)EDA



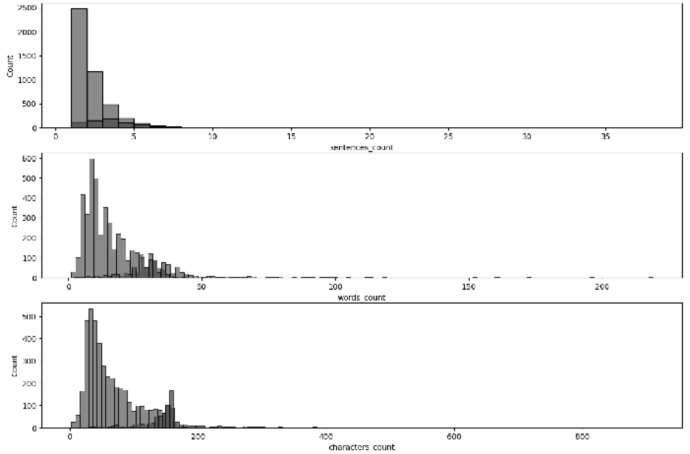
您的玩具数据集（SMS数据集）EDA



Your toy dataset (SMS dataset)EDA



您的玩具数据集（SMS数据集）EDA



Domain/Data Understanding

- How many features?
- How many sample?
- What are they? How are they related? correlate?
- What DS task shall we perform?
- How do we do it?

领域/数据理解

- 有多少功能?
- 多少个样品?
- 它们是什么？它们有什么关系？相关?
- 我们应该执行什么 DS 任务?
- 我们是怎么做到的?

Workshop

- Write 1 page essay in English for one of the three case of your choice.
- You may discuss with your classmates, but you need to write on your own.
- Your essay must include the follow topics

车间

- 用英语写 1 页的文章，用于您选择的三个案例之一。
- 你可以和你的同学讨论，但你需要自己写。

- 您的论文必须包括以下主题

Workshop 1

- **Business objectives, define problem (no more than one problem)**
- **Identify activities, outputs, outcomes, and indicators identification**
- **Identify Stakeholders** (who involves)
- **Identify Data source** (where can you get?, How does it look like?)
- **Identify level of IT usage (level0, 1 or 2)**
- **Identify DM technique** (which Data science/ Datamining technique you might use?)
- **Data as data product (solution)** (what should be your output product to the users or stakeholders?)

工作坊 1

- 业务目标，定义问题（不超过一个问题）
- 确定活动、产出、成果和指标识别
- 确定利益相关者（参与）
- 识别数据源（从哪里可以得到？，它是什么样子的？）
- 确定 IT 使用级别（级别 0、1 或 2）
- 识别 DM 技术（您可能使用哪种数据科学/数据挖掘技术?

- 数据作为数据产品（解决方案）（向用户或利益干系人输出的产品应该是什么?

Case study 1 : Road regulation in university campus (Beginner)

Identify who can enter or helmet detection for those who ride a motorcycle

案例研究1：大学校园的道路管制（初级）

识别谁可以进入或为骑摩托车的人进行头盔检测

Case study 2: A university campus public transport (Intermediate, further away)

Electric bus vs Mobus

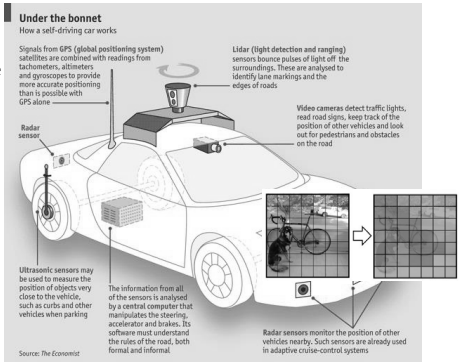
案例研究 2：大学校园公共交通（中级，更远）

电动巴士与Mobus

Case Study 3:

The 2020 United States presidential election (Advanced) or Any problem you think that it should belong to the group.

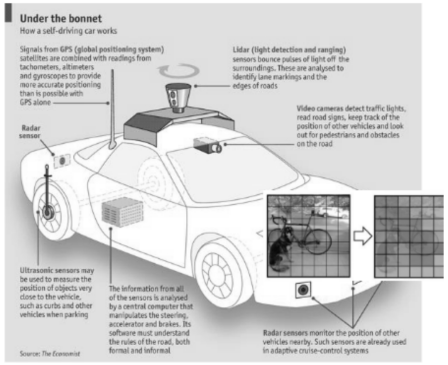
Sarcasm or Irony sentence detection?
How did you get here? Did someone leave your cage open?



SE 953482 Natural Language Processing for SE
66/2
Text Extractions

案例研究 3：2020 年美国总统大选（高级）或您认为它应该属于该组的任何问题。

讽刺还是讽刺句子检测？
你是怎么来到这里的？有人把你的笼子打开吗？



SE 953482 自然语言

SE 的处理

66/2

文本提取

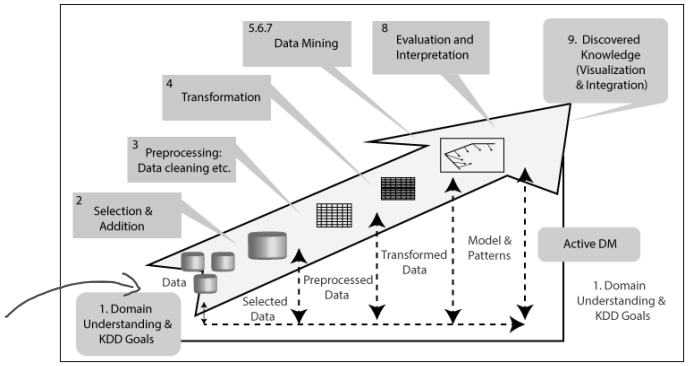
Where we are now

1.	NLP Overview	3
2.	Data science methodology	3
3.	Word Tokenization, Text preprocessing	3
3.	Text extraction methods	6
4.	Machine-learning models in NLP	6
5.	Deep-learning models in NLP	6
6.	Transformers	3
7.	Evaluation metrics and explain-ability	3
8.	NLP-based Systems	3
9.	Case studies and Project	9

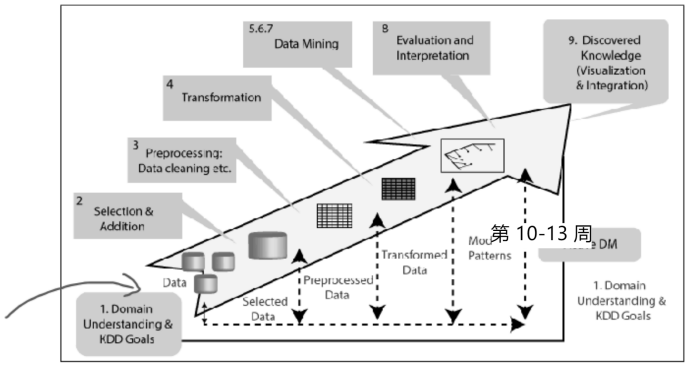
我们现在所处的位置

1. NLP概述3
2. 数据科学方法论 3
3. 文字标记化、文本预处理 3
3. 文本提取方法 6
4. NLP 6 中的机器学习模型
5. NLP 6 中的深度学习模型
6. 变形金刚 3
7. 评估指标和可解释性 3
8. 基于 NLP 的系统 3
9. 案例研究和项目 9 Pree Thiengburanathum 助理教授

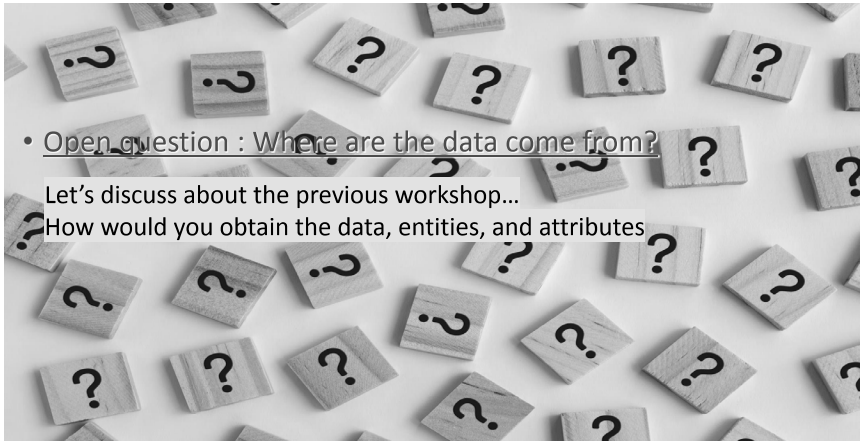
Knowledge Discovery in Databases Process



数据库中的知识发现过程



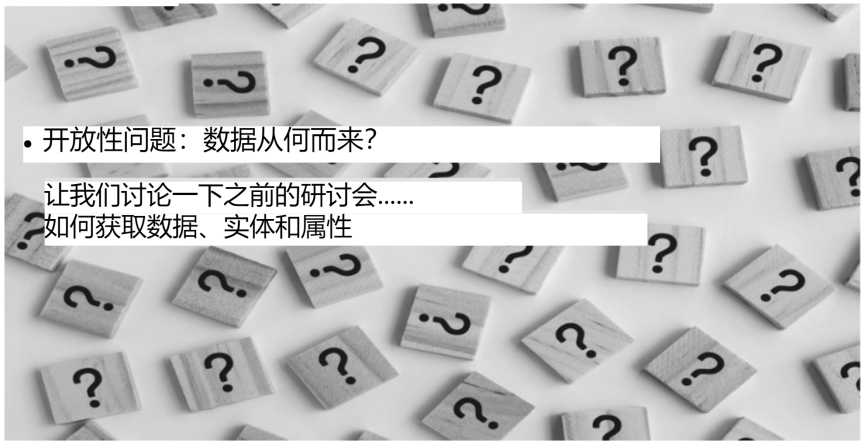
Definition



- Open question : Where are the data come from?

Let's discuss about the previous workshop…
How would you obtain the data, entities, and attributes

定义



- 开放性问题：数据从何而来?

让我们讨论一下之前的研讨会……
如何获取数据、实体和属性

Primary data vs Secondary data

- Primary data – firsthand data or raw data
  - Expensive
  - Various methods (e.g., surveys, interview, focus groups, case studies, etc.).
- Secondary data – been collected by someone else (published data)
  - Easily available (Kaggle, githib, reddit, UCI repo, data.go.th
  - Irrelevance, redundant, and less accuracy
  - Books, reports, censuses, government publications, etc.

主要数据与次要数据

- 原始数据 – 第一手数据或原始数据
  - 贵
  - 各种方法（例如，调查、访谈、焦点小组、案例研究等）。
- 次要数据 – 由其他人收集（已发布数据）
  - 容易获得（Kaggle、githib、reddit、UCI repo、data.go.th
  - 无关紧要、冗余且准确性较低
  - 书籍、报告、人口普查、政府出版物等。

Example

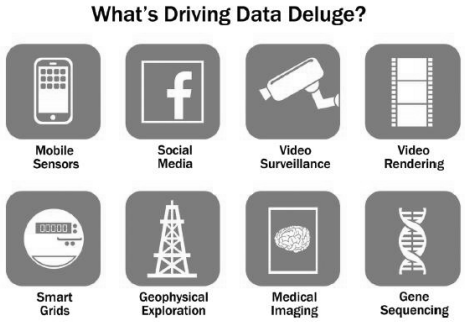
- John’s experiment used data from a book
- Marry conducted her experiments through questionnaire by surveying his organization

例

- 约翰的实验使用了一本书中的数据
- Marry通过调查他的组织，通过问卷调查进行实验

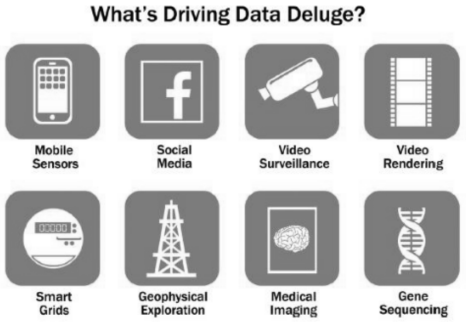
Where the data come from?

- Massive of digital information
- Credit cards
- Social networking
- Capturing traffic flow
- Measuring pollution
- Interviews, Surveys, Questionnaires
- Etc.



数据从何而来?

- 海量数字信息
- 信用卡
- 社交
- 捕获流量
- 测量污染
- 访谈、调查、问卷
- Etc.



Comparison

Metrics	Primary	Secondary
Accuracy	High	Low
Control	High	Low
Relevancy	High	Low
Ownership	?	?
Accessibility	?	?
Bias	?	?
Up-to-dated	?	?

比较

计量指标 小学 中学		
准确率 高 低 控制 高 低 相关性 高 低 所有权？？		
可及性？？		
偏见？？		
最新？？		

Basic methods to access the data

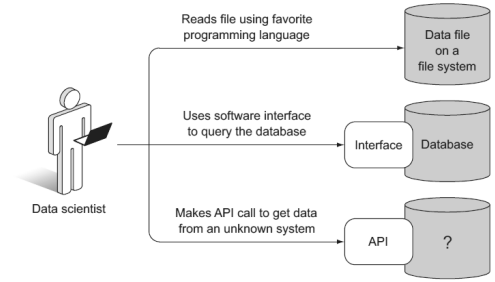


Figure 3.3 Three ways a data scientist might access data: from a file system, database, or API

访问数据的基本方法

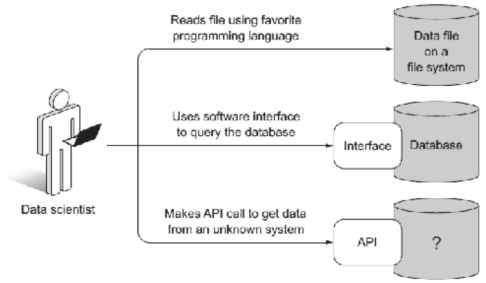
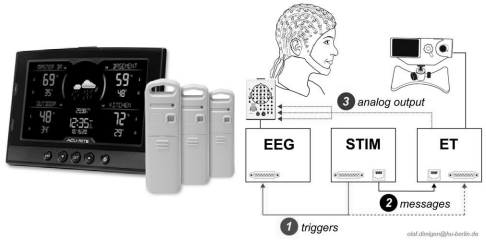


Figure 3.3 Three ways a data scientist might access data: from a file system, database, or API

Data acquisition (Time-series)

- Hardware, Software, questionnaire, interview
  - To allow us to measure something in the real world.
  - Weather station (Temp. and humidity)
  - EYE-EEG (operate between 500-1000mhz)



Open question: There should be way to collected data, could you explore more method?

数据采集（时间序列）

- 硬件、软件、问卷调查、访谈
  - 让我们测量现实世界中的东西。
  - 气象站（温度和湿度）
  - EYE-EEG（在500-1000MHz）

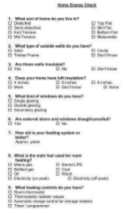
开放性问题：应该有收集数据的方法，您能探索更多方法吗？

Surveys/ Questionnaires (Multivariance)

- Usually deployed by utility companies, building management companies, energy analysis companies or government.
- **Field Interviewer** from the Energy Information Admiration (EIA)
  - Use questionnaire to collect data from selected housing uniting.
  - Data includes
    - Building characteristic
    - Energy consumption and expense
    - Household demographics
- Data from interview + data from energy suppliers
  - Estimate energy costs
  - Usage for heating and cooling

调查/问卷（多方差）

- 通常由公用事业公司、建筑管理公司、能源分析公司或政府部署。
- 能源信息钦佩 （EIA） 的现场采访者
  - 使用问卷从选定的住房单元收集数据。
  - 数据包括
    - 建筑特色
    - 能源消耗和费用
    - 家庭人口统计
- 访谈数据+能源供应商数据
  - 估算能源成本
  - 用于加热和冷却



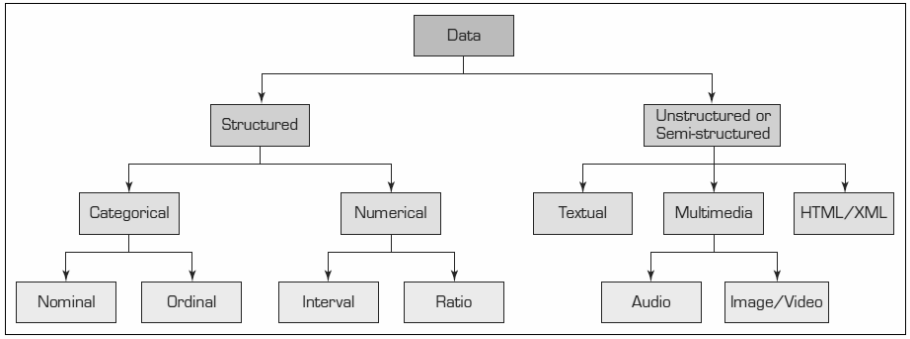
Social media (Multivariance + Textual)

- Twetter
- Facebook
- Pantip
- Ecommerce website like JD.com, Wongnai, etc
- WebCrawler /Scraper

社交媒体（多方差 + 文本）

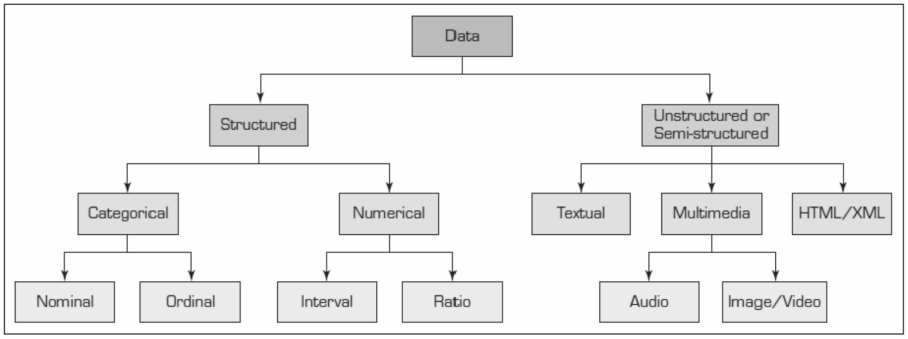
- 滴水机
- 脸书
- 潘蒂普
- 电子商务网站，如 JD.com，Wongnai等
- WebCrawler /爬虫

Data Mining – Taxonomy of Data in DM



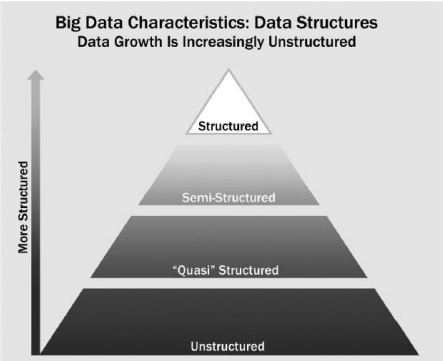
(Turban, Sharda, & Delen, 2014)

数据挖掘 – DM 中的数据分类

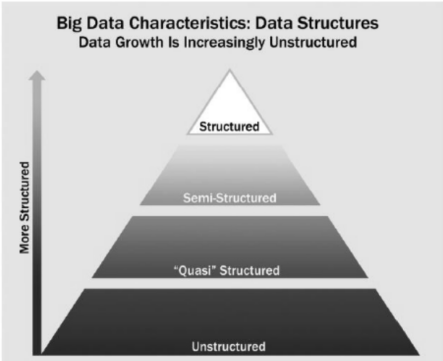


（Turban，Sharda和Delen，2014）

Big data growth increase in unstructure



大数据增长在非结构化中增加



Structure data

- Data containing a defined data type, format, and structure
- E.g. Transaction data, online analytical processing [OLAP] data cubes, traditional RDBMS, CSV files, and even simple spreadsheets).

SUMMER FOOD SERVICE PROGRAM 1]				
(Data as of August 01, 2011)				
Fiscal Year	Number of Sites	Peak (July) Participation	Meals Served	Total Federal Expenditures 2]
	----------Thousands----------		--Mil.--	---Million $---
1969	1.2	99	2.2	0.3
1970	1.9	227	8.2	1.8
1971	3.2	569	29.0	8.2
1972	6.5	1,080	73.5	21.9
1973	11.2	1,437	65.4	26.6
1974	10.6	1,403	63.6	33.6
1975	12.0	1,785	84.3	50.3
1976	16.0	2,453	104.8	73.4
TQ 3]	22.4	3,455	198.0	88.9
1977	23.7	2,791	170.4	114.4
1978	22.4	2,333	120.3	100.3
1979	23.0	2,126	121.8	108.6
1980	21.6	1,922	108.2	110.1
1981	20.6	1,726	90.3	105.9
1982	14.4	1,397	68.2	87.1
1983	14.9	1,401	71.3	93.4
1984	15.1	1,422	73.8	96.2
1985	16.0	1,462	77.2	111.5
1986	16.1	1,509	77.1	114.7
1987	16.9	1,560	79.9	129.3
1988	17.2	1,577	80.3	133.3
1989	18.5	1,652	86.0	143.8
1990	19.2	1,692	91.2	163.3

结构数据

- 包含已定义数据数据类型、格式和结构
- 例如，在线交易数据分析处理 [OLAP] 数据多维数据集，传统 RDBMS、CSV 文件，甚至是简单的电子表格）。

SUMMER FOOD SERVICE PROGRAM 1]				
(Data as of August 01, 2011)				
Fiscal Year	Number of Sites	Peak (July) Participation	Meals Served	Total Federal Expenditures 2]
	----------Thousands----------		--Mil.--	---Million $---
1969	1.2	99	2.2	0.3
1970	1.9	227	8.2	1.8
1971	3.2	569	29.0	8.2
1972	6.5	1,080	73.5	21.9
1973	11.2	1,437	65.4	26.6
1974	10.6	1,403	63.6	33.6
1975	12.0	1,785	84.3	50.3
1976	16.0	2,453	104.8	73.4
TQ 3]	22.4	3,455	198.0	88.9
1977	23.7	2,791	170.4	114.4
1978	22.4	2,333	120.3	100.3
1979	23.0	2,126	121.8	108.6
1980	21.6	1,922	108.2	110.1
1981	20.6	1,726	90.3	105.9
1982	14.4	1,397	68.2	87.1
1983	14.9	1,401	71.3	93.4
1984	15.1	1,422	73.8	96.2
1985	16.0	1,462	77.2	111.5
1986	16.1	1,509	77.1	114.7
1987	16.9	1,560	79.9	129.3
1988	17.2	1,577	80.3	133.3
1989	18.5	1,652	86.0	143.8
1990	19.2	1,692	91.2	163.3

Data Type - Qualitative

- Extract from field notes, interview transcripts
- Data can be expressed in discrete (i.e. categorical, enumerated) as follows:
  - **Nominal**- variable with no inherent order or ranking sequence (e.g. gender, nationality)
  - **Ordinal** (socio-economic status)

Open question Qualitative vs Quantitative?

数据类型 - 定性

- 摘自现场笔记、采访记录
- 数据可以用离散（即分类、枚举）表示如下：
  - 名义变量，没有固有的顺序或排名顺序（例如性别、国籍）
  - 序数（社会经济地位）

悬而未决的问题：定性与定量？

Qualitative vs Quantitative data

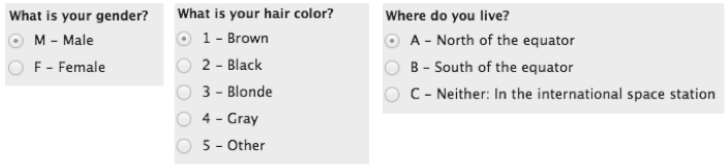
Qualitative	Quantitative
Origin = SC	Origin = NS
Sample size = Small	Sample size = Large
Cost = Low-High	Cost = Low-High
Style = personal voice, literary	Style = formal, scientific
Type = Description	Type = numerical
Source = Interviews	Source = Instruments
2+3 more..	

定性数据与定量数据

定性定量	
起源 = SC 起源 = NS 样本量 = 小样本量 = 大 成本 = 低-高 成本 = 低-高 风格 = 个人声音，文学 风格 = 正式、科学 类型 = 描述 类型 = 数字 来源 = 访谈 来源 = 仪器 2+3 更多..	

Data type - Nominal data

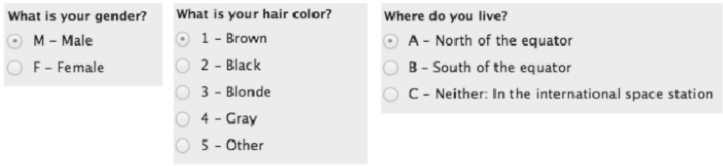
Also known as categorical
Finite set, qualitative
No order



Source: https://www.mymarketresearchmethods.com/types-of-data-nominal-ordinal-interval-ratio/ (Accessed August, 2018)

数据类型 - 名义数据

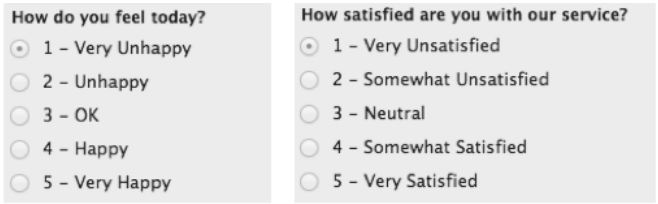
也称为分类
有限集，定性 无序



来源：https://www.mymarketresearchmethods.com/types-of-data-nominal-ordinal-interval-ratio/（2018年8月访问）

Data type - Ordinal data

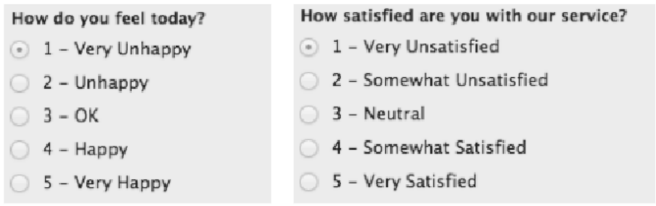
Similar with nominal but with rank order



Source: https://www.mymarketresearchmethods.com/types-of-data-nominal-ordinal-interval-ratio/ (Accessed August, 2018)

数据类型 - 序数数据

与名义相似，但有排名顺序

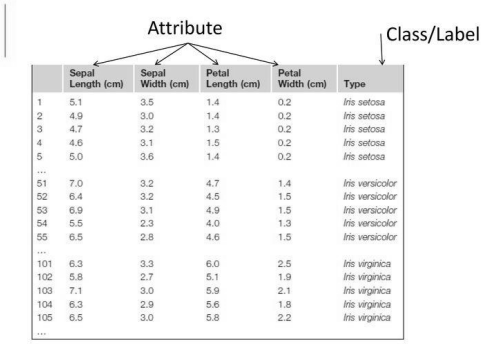


2018）来源：https://www.mymarketresearchmethods.com/types-of-data-nominal-ordinal-interval-ratio/（2018 年 8 月访问）

Dataset with attribute and class

具有属性和类的数据集

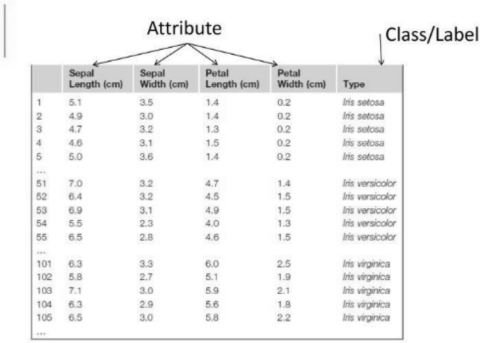
DATASET WITH ATTRIBUTE AND CLASS



	Sepal Length (cm)	Sepal Width (cm)	Petal Length (cm)	Petal Width (cm)	Type
1	5.1	3.5	1.4	0.2	*Iris setosa*
2	4.9	3.0	1.4	0.2	*Iris setosa*
3	4.7	3.2	1.3	0.2	*Iris setosa*
4	4.6	3.1	1.5	0.2	*Iris setosa*
5	5.0	3.6	1.4	0.2	*Iris setosa*
...					
51	7.0	3.2	4.7	1.4	*Iris versicolor*
52	6.4	3.2	4.5	1.5	*Iris versicolor*
53	6.9	3.1	4.9	1.5	*Iris versicolor*
54	5.5	2.3	4.0	1.3	*Iris versicolor*
55	6.5	2.8	4.6	1.5	*Iris versicolor*
...					
101	6.3	3.3	6.0	2.5	*Iris virginica*
102	5.8	2.7	5.1	1.9	*Iris virginica*
103	7.1	3.0	5.9	2.1	*Iris virginica*
104	6.3	2.9	5.6	1.8	*Iris virginica*
105	6.5	3.0	5.8	2.2	*Iris virginica*
...					

The data type of an attribute (numeric, ordinal, nominal) affects the methods we can use to analyze and understand the data.

DATASET WITH ATTRIBUTE AND CLASS



	Sepal Length (cm)	Sepal Width (cm)	Petal Length (cm)	Petal Width (cm)	Type
1	5.1	3.5	1.4	0.2	*Iris setosa*
2	4.9	3.0	1.4	0.2	*Iris setosa*
3	4.7	3.2	1.3	0.2	*Iris setosa*
4	4.6	3.1	1.5	0.2	*Iris setosa*
5	5.0	3.6	1.4	0.2	*Iris setosa*
...					
51	7.0	3.2	4.7	1.4	*Iris versicolor*
52	6.4	3.2	4.5	1.5	*Iris versicolor*
53	6.9	3.1	4.9	1.5	*Iris versicolor*
54	5.5	2.3	4.0	1.3	*Iris versicolor*
55	6.5	2.8	4.6	1.5	*Iris versicolor*
...					
101	6.3	3.3	6.0	2.5	*Iris virginica*
102	5.8	2.7	5.1	1.9	*Iris virginica*
103	7.1	3.0	5.9	2.1	*Iris virginica*
104	6.3	2.9	5.6	1.8	*Iris virginica*
105	6.5	3.0	5.8	2.2	*Iris virginica*
...					

The data type of an attribute (numeric, ordinal, nominal) affects the methods we can use to analyze and understand the data.

Scaling for numerical input

- To range (-1, +1)
- To speed up the convergence

[0] 30 male 38000.00 urban democrat
[1] 36 female 42000.00 suburban republican
[2] 52 male 40000.00 rural independent
[3] 42 female 44000.00 suburban other

数字输入的缩放比例

- 范围 （-1， +1）

加快收敛 [0] 30 男 38000.00 市区民主人士 [1] 36 女 42000.00 郊区共和派 [2] 52 男 40000.00 农村独立 [3] 42 女 44000.00 郊区其他

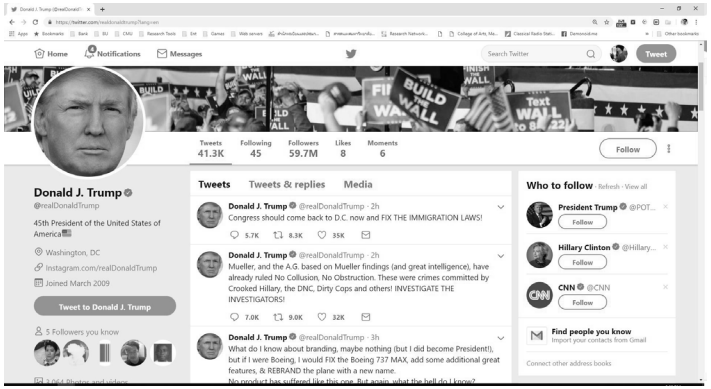
Scaling for numerical input (cont.)

- [0] -1.23 -1.0 -1.34 (0.0 1.0) (0.0 0.0 0.0 1.0)
- [1] -0.49 1.0 0.45 (1.0 0.0) (0.0 0.0 1.0 0.0)
- [2] 1.48 -1.0 -0.45 (-1.0 -1.0) (0.0 1.0 0.0 0.0)
- [3] 0.25 1.0 1.34 (1.0 0.0) (1.0 0.0 0.0 0.0)

数值输入的缩放比例（续）

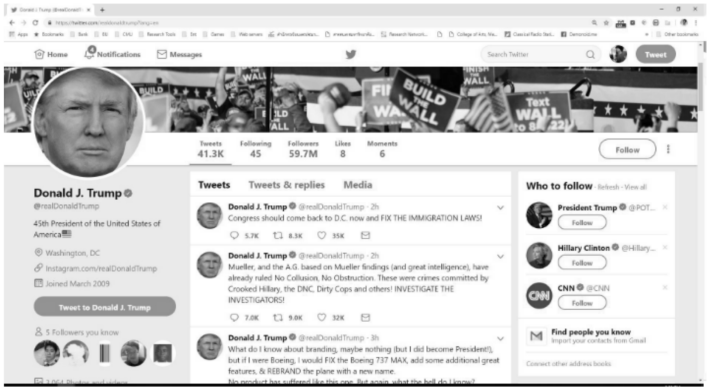
- [0] -1.23 -1.0 -1.34 (0.0 1.0) (0.0 0.0 0.0 1.0)
- [1] -0.49 1.0 0.45 (1.0 0.0) (0.0 0.0 1.0 0.0)
- [2] 1.48 -1.0 -0.45 (-1.0 -1.0) (0.0 1.0 0.0 0.0)
- [3] 0.25 1.0 1.34 (1.0 0.0) (1.0 0.0 0.0 0.0)

Data hidden in **text**



Asst. Prof. Pree Thiengburanathum

隐藏在文本中的数据



Pree Thiengburanathum 助理教授

WebCrawler / scraper

- https://www.twitter.com
- Beautiful soup
- Scrapy
- Selenium

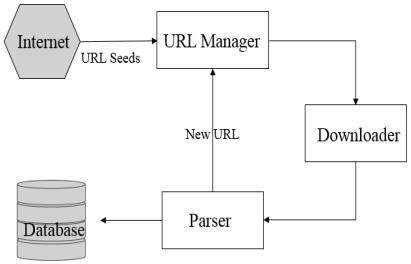
WebCrawler / 刮板

- https://www.twitter.com
- 美丽的汤
- 刮擦
- 硒

Architecture of Web Crawler

The web crawlers can continuously download content of web pages and search for new URLs.

A crawler framework mainly includes the URL managers, downloaders, and parsers



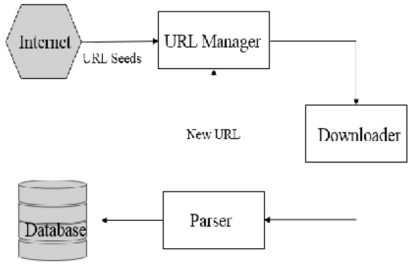
Architecture of Web crawler

(Yang, Thiengburanathum2020)

网络爬虫的架构

网络爬虫可以不断下载网页内容并搜索新的 URL。

爬虫框架主要包括 URL 管理器、下载器和解析器



网络爬虫的架构

（杨，Thiengburanathum2020）

Handling Text in Python

```
>>> txt1 = 'The Radical Left Democrats will never be satisfied with anything we give them. They will always Resist and Obstruct!'
>>> len(txt1)
116
>>> txt2 = txt1.split(' ');
>>> len(txt2);
19
>>> txt2
['The', 'Radical', 'Left', 'Democrats', 'will', 'never', 'be', 'satisfied', 'with', 'anything', 'we', 'give', 'them.', 'They', 'will', 'always', 'Resist', 'and', 'Obstruct!']
>>>
```

Asst. Prof. Pree Thiengburanathum

在 Python 中处理文本

```
>>> txt1 = '激进的左翼民主党人永远不会对我们给予的任何东西感到满意他们。他们将永远反抗和阻挠!
仅限>>>（txt1）
116
>>> txt2 = txt1.split（' '）；
仅限>>>（txt2）；
19
>>> txt2 ['The', 'Radical', 'Left', 'Democrats', 'will', 'never', 'be', 'satisfied', 'with', 'anything', 'we', 'give', 'them.', 'they', 'will', 'always', 'Resist', 'and', 'Obstruct！'] >>>
```

Pree Thiengburanathum 助理教授

Handling Text in Python (cont.)

- **Finding long words: e.g. words that has more than 3 letters.**

>>> [w for w in txt2 if len(w) > 3]

['Radical', 'Left', 'Democrats', 'will', 'never', 'satisfied', 'with', 'anything', 'give', 'them.', 'They', 'will', 'always', 'Resist', 'Obstruct!']

- **Finding capitalized words:**

>>> [w for w in txt2 if w.istitle()]

['The', 'Radical', 'Left', 'Democrats', 'They', 'Resist', 'Obstruct!']

在 Python 中处理文本（续）

- 查找长单词：例如，包含超过 3 个字母的单词。

>>> [w for w in txt2 if len（w） > 3] ['激进'， '左派'， '民主党人'，'将'， '从不'， '满意'， '与'， '任何事情'，

'给予'， '他们'， '将'， '总是'， '抵制'， '阻挠！

- 查找大写单词：

>>> [w for w in txt2 if w.istitle（）] ['The'， 'Radical'， 'Left'，'Democrats'， 'They'， 'Resist'， 'Obstruct！']

Handling Text in Python (cont.)

Find words which ends with ‘s’:

>>> [w for w in txt2 if w.endswith('s')]

['Democrats', 'always’]

Find unique words: using set():

>>> txt3 = 'To be or not to be’

>>> txt4 = txt3.split(' ‘)

>>> len(txt4)

6

>>> len(set(txt4))

5

>>> set(txt4)

{'be', 'not', 'to', 'To', 'or'}

在 Python 中处理文本（续）

查找以“s”结尾的单词：

>>> [w for w in txt2 if w.endswith（'s'）]

['民主党人'， '总是']

查找唯一单词：使用 set（）：

>>> txt3 = 'To be or not to be' >>> txt4 = txt3.split（' '） >>> len（txt4） 6

>>> len（set（txt4）） 5

>>> set（txt4） {'be'，'not'， 'to'， 'to'， 'or'}

Handling Text in Python (cont.)

```
>>> set([w.lower() for w in txt4])
{'be', 'not', 'or', 'to'}
>>>
```


在 Python 中处理文本（续）

```
>>> set（[w.lower（） for w in
txt4]） {'be'， 'not'， 'or'， 'to'}
>>>
```


Handling Text in Python (cont.)

- **Word comparation in Python**
- s.startswith(t)
- s.endswith(t)
- t in s
- s.istitle()
- s.islower()
- s.isupper()
- s.isdigit(), s.isalnum(), s.isalpha()

在 Python 中处理文本（续）

- Python 中的单词比较
- s.startswith（t）
- s.endswith（t）
- t 在 s 中
- s.istitle（）
- s.islower（）
- s.isupper（）
- s.isdigit（），s.isalnum（），s.isalpha（）

String operations

- s.lower(); s.upper(); s.titlecase()
- s.splits()
- s.splitlines()
- s.join(t) //
- s.strip() // take out all the white space
- s.find(t)// find the specific substring
- s.replace(u, v)// u will be replace by v

字符串操作

- s.lower（）;s.upper（）;s.titlecase（）
- s.splits（）
- s.splitlines（）
- s.join（t） //
- s.strip（） // 去掉所有的空白
- s.find（t）// 查找具体的子字符串
- s.replace（u， v）// 您将被 v 替换

Words to characters

```
>>> txt5 = 'ouagadougou'
>>> txt6 = txt5.split('ou')
>>> txt6
['', 'agad', 'g', ‘’]
>>> 'ou'.join(txt6)
'ouagadougou'
```


单词到字符

```
>>> txt5 = '瓦加杜古'
>>> txt6 = txt5.split（'ou'）
>>> txt6
['， '立即'， 'g'， '']
>>> 'ou'.join（txt6）
 “瓦加杜古”
```

Words to characters (cont.)

```
>>> txt5.split('')
Traceback (most recent call last):
 File "<pyshell#7>", line 1, in <module>
   txt5.split('')
ValueError: empty separator
>>> list(txt5)
>> [c for in c txt5]
['o', 'u', 'a', 'g', 'a', 'd', 'o', 'u', 'g', 'o', 'u']
```


单词到字符（续）

```
>>> txt5.split（''） 回溯（最近一次调用
最后一次）： 文件 “” ，第 1 行，在
txt5.split（''） ValueError：空分隔符
>>> list（txt5） >> [c for in c txt5]
['o', 'u', 'a', 'g', 'a', 'd', 'o',
'u', 'g', 'o', 'u']
```


Cleaning text

```
>>> txt8 = '    A quick brown fox jumped over the lazy dog. '
>>> txt8.split(' ')
['', '', '\t', 'A', 'quick', 'brown', 'fox', 'jumped', 'over', 'the', 'lazy', 'dog.', '']
>>> txt8.strip()
'A quick brown fox jumped over the lazy dog.'
```


清理文本

```
>>> txt8 = '一只敏捷的棕色狐狸跳过了懒惰的狗。' >>> txt8.split（' '）
['', '', '\t', 'A', 'quick', 'brown', 'fox', 'jumped', 'over', 'the',
'lazy', 'dog.', ''] >>> txt8.strip（） '一只快速的棕色狐狸跳过了懒惰的
狗。'
```

Changing text

```
>>> txt9 = txt8.strip()
>>> txt9
'A quick brown fox jumped over the lazy dog.'
>>> txt9.find('o')
10
>>> txt9.rfind('o')
40
>>> txt9.replace('o', 'O')
'A quick brOwn fOx jumped Over the lazy dOg.'
```


更改文本

```
>>> txt9 = txt8.strip（）
>>> txt9 '一只敏捷的棕色狐狸跳过了懒惰的狗.'
>>> txt9.find（'o'） 10 >>> txt9.rfind（'o'）
40 >>> txt9.replace（'o'， 'O'） '一个快速的
brOwn fOx 跳过了懒惰的 dOg。”
```


Handling larger texts

- **Reading files line by line**

```
>>> f = open('/Users/Pree/Desktop/notre dame.txt')
>>> f.readline()
'Notre-Dame de Paris, often referred to simply as Notre-Dame, is a medieval Catholic cathedral on the ÃŽle de la CitÃ© in the 4th arrondissement of Paris, France.\n'
```

- **Reading the full file**

```
>>> f.seek(0)
0
>>> txt12 = f.read()
>>> len(txt12)
```


处理较大的文本

- 逐行读取文件

```
>>> f = open（'/Users/Pree/Desktop/notre dame.txt'） >>> f.readline（） '巴黎圣母院，通常简称为巴黎圣母院，是一座中世纪的天主教大教堂，位于法国巴黎第四区的西©城。
```

- 读取完整文件

```
>>> f.seek（0）
0
>>> txt12 = f.read（）
仅限>>>（TXT12）
```

Handling larger texts (cont.)

```
>>> txt13 = txt12.splitlines()
>>> len(txt13)

>>> txt13
['Notre-Dame de Paris, often referred to simply as Notre-Dame, is a medieval Catholic cathedral on the ÃŽle de la CitÃ© in the 4th arrondissement of Paris, France.', '', 'The cathedral is consecrated to the Virgin Mary and considered to be one of the finest examples of French Gothic architecture.']
>>> txt13[0]
'Notre-Dame de Paris, often referred to simply as Notre-Dame, is a medieval Catholic cathedral on the ÃŽle de la CitÃ© in the 4th arrondissement of Paris, France.'
```


处理较大的文本（续）

```
>>> txt13 =
txt12.splitlines（） 仅
>>> （txt13）
>>> txt13 ['巴黎圣母院，通常简称为巴黎圣母院，是法国>>>巴黎第四区西西区的一座中世纪天主教大教堂©。 是一座中世纪天主教大教堂，位于法国巴黎第四区的 ÃŽle de la CitÃ©。
```


File operations

- f = open(filename, mode)
- f.readline(); f.read(); f.read(n)
- for line in f: doSomething(line)
- f.seek(n)
- f.write(message)
- f.close() // close the file handler
- f.closed// check if the file close

文件操作

- f = open（文件名，模式）
- f.readline（）;f.read（）;f.读取（n）
- 对于 f 中的行：doSomething（line）
- f.seek（n）
- f.write（消息）
- f.close（） // 关闭文件处理程序
- f.closed// 检查文件是否关闭

Processing free-text

>>> txt10 = '#DrainTheSwamp - @GreggJarrett: No one takes anything Schiff says seriously because he lost all credibility. For 2 years he claimed there was a mound of criminal evidence. Where is it? Show us… because it doesn't exist. #MAGA #AmericaFirst #Dobbs'

How to find out the callouts and hashtags?

处理自由文本

>>> txt10 = '#DrainTheSwamp - @GreggJarrett：没有人认真对待希夫说的任何话，因为他失去了所有的可信度。两年来，他声称有一堆犯罪证据。它在哪里？向我们展示...因为它不存在。#MAGA #AmericaFirst #Dobbs'

如何找出标注和主题标签?

Finding specific words

Hashtags

>>> [w for w in txt10.split() if w.startswith('#')]

['#DrainTheSwamp', '#MAGA', '#AmericaFirst', '#Dobbs']

Callouts

>>> w for w in txt10.split() if w.startswith('@')]

['@GreggJarrett:', '@']

查找特定单词

主题标签

>>> [w for w in txt10.split（） if w.startswith（'#'）]

['#DrainTheSwamp'， '#MAGA'， '#AmericaFirst'，'#Dobbs']

标注

在 txt10.split（） 中>>> w for w if w.startswith（'@'）] ['@GreggJarrett：'， '@']

Regular expression

- Known as regex" or "regexp", are a powerful tool used in computing for pattern matching within strings.
- *Searching and Extracting* – emails, URLs, etc.
- *Data Validation* – checking input format
- *String Manipulation*- split string

正则表达式

- 称为正则表达式“或”正则表达式“，是用于计算字符串内模式匹配的强大工具。
- 搜索和提取 – 电子邮件、URL 等。
- 数据验证 – 检查输入格式
- 字符串操作 - 拆分字符串

Finding patterns with regular expressions

- Callouts are more than just tokens starting with ‘@’
- @username @UK_Spokesperon
- Match something after ‘@’
  - Alphabets
  - Numbers
  - Special symbols such as ‘_’

使用正则表达式查找模式

- 标注不仅仅是以“@”开头的标记
- @username @UK_Spokesperon
- 匹配“@”之后的内容
  - 字母
  - 数字
  - 特殊符号，例如“_”

Finding patterns with regular expressions

Callouts

>>> w for w in txt10.split() if w.startswith('@')]
['@GreggJarrett:', '**@**']

Import regular expressions first

import re
[w for w in txt10.split(' ') if re.search('@[A-Za-z0-9_]+', w)]
['@GreggJarrett:']

使用正则表达式查找模式

标注

在 txt10.split（） 中>>> w for w if w.startswith（'@'）] ['@GreggJarrett：'， '@']

首先导入正则表达式

import re [w for w in txt10.split（' '） if re.search（'@[A-Za-z0-9_]+'， w）] ['@GreggJarrett：']

Parsing the callout regular expression

- @[A-Za-z0-9_]+
- Starts with @
- Followed by any alphabet (upper or lower case), digit, underscore
- That repeats at least once, but any number of times

解析标注正则表达式

- @[A-Za-z0-9_]+
- 开头为@
- 后跟任何字母（大写或小写）、数字、下划线
- 这至少重复一次，但次数不限

Metacharacters

Metacharacters are characters with a special meaning:

Character	Description	Example
[]	A set of characters	“[a-m]”
\	Signals a special sequence	“\d”
.	Any character(except newline character)	“he..o”
^	Starts with	“^hello”
$	Ends with	“world$”
* (repetitions)	Zero or more occurrences	“aix*”
+ (repetition) ? (repetition)	One or more occurrences Zero or one occurrences	“aix+”
{}	Exactly the specified number of occurrences	“al{2}”
|	Either or	“falls|stays”

元字符

元字符是具有特殊含义的字符：

字符描述示例 [] 一组字符“[a-m]”		
\ 发出特殊序列“\d”的信号		
.任何字符（换行符除外）		“呵呵……o"
^ 以“^hello”开头		
$ 以“world$”结尾		
（重复）零次或多次出现“aix”		
+（重复） ?（重复）	一次或多次出现 零次或一次出现	“aix+”
{} 正好是指定数量的	事件	“艾尔{2}”
|要么或“跌倒|停留”		

Metacharacters

- \d – any digit
- \D – any non-digit
- \s – any white space char, [\t\n\r\f\v]
- \S – opposite the above
- \w – Alphanumeric character , [a-zA-Z0-9_]
- \W – [^ a-zA-Z0-9_]

元字符

- \d – 任意数字
- \D – 任何非数字
- \s – 任何空格字符，[\t\n\r\f\v]
- \S – 与上述相反
- \w – 字母数字字符， [a-zA-Z0-9_]
- \W – [^ a-zA-Z0-9_]

Sets

A set is a set of characters inside a pair of square brackets

Set	Description
[arn]	Returns a match where one of the specified characters (a, r, or n) are present
[a-n]	Returns a match for any lower-case character, alphabetically between a and n
[^arn]	Returns a match for any character EXCEPT a, r, and n
[0123]	Returns a match where any of specified digits(0, 1, 2, or 3) are present
[0-9]	Returns a match for any digit between 0 and 9
[0-5][0-9]	Returns a match for any two-digit number from 00 and 59
[a-zA-Z]	Returns a match for any character alphabetically between a and z, lower case OR upper case

Sets

集合是一对方括号内的一组字符

套装说明	
[阿恩]	返回一个匹配项，其中指定字符之一（ a, r 或 n）存在
[a-n]返回任何小写字符的匹配项，按字母顺序在 a 和 n [^arn] 返回除 a、r 和 n 之外的任何字符的匹配项 [0123] 返回指定数字（0、1、2 或 3）中的任何 一个	
现在 [0-9] 返回 0 到 9 之间任何数字的匹配项 [0-5][0-9] 返回 00 到 59 之间的任何两位数的匹配项 [a-zA-Z] 返回 a 和 z 之间按字母顺序排列的任何字符的匹配项，	
	小写或大写

Example 1 – search() function

The search() function searches the string for a match, and returns a Match object if there is a match.

If there is more than one match, only the first occurrence of the match will be returned:

```
import re

#Check if the string starts with "The" and ends with " ChiangMai":

txt = "The rain in ChaingMai"
x = re.search("^The.*ChiangMai$", txt)

if (x):
  print("YES! We have a match!")
else:
  print("No match")
```


示例 1 – search（） 函数

search（） 函数在字符串中搜索匹配项，如果存在匹配项，则返回 Match 对象。

如果有多个匹配项，则仅返回匹配项的第一次匹配：

```
导入 RE

#Check 字符串是否以“The”开头，以“ChiangMai”结尾：

txt = “清迈的雨” x = re.search
（“^The.*ChiangMai$”， txt）

if （x）： print（“是的！我们
有一场比赛！else： print（“不
匹配”）
```

Example 2 – findall() function

- The findall() function returns a list containing all matches.
- import re

#Return a list containing every occurrence of "ai":

```
str = "The rain in Chaing Mai"
x = re.findall("ai", str)
print(x)
['ai', 'ai']
```


示例 2 – findall（） 函数

- findall（） 函数返回一个包含所有匹配项的列表。
- 导入 RE

#Return 包含“ai”的每次出现的列表：

```
str = “清迈的雨” x =
re.findall（“ai”， str） print
（x） ['ai'， 'ai']
```


Example 2.1 findall() function

- Finding specific characters

```
>>> txt12 = 'ouagadougou'
>>> re.findall(r'[aeiou]', txt12)
['o', 'u', 'a', 'a', 'o', 'u', 'o', 'u’]
>>> re.findall(r'[^aeiou]', txt1)
['g', 'd', 'g']
```


例 2.1 findall（） 函数

- 查找特定字符

```
>>>txt12 = '瓦加杜古' >>>
re.findall（r'[aeiou]'， txt12）
['o'， 'u'， 'a'， 'a'， 'o'， 'u']
>>> re.findall（r'[^aeiou]',
txt1） ['g'， 'd'， 'g']
```

Example 3 – sub() function

• The sub() function replaces the matches with the text of your choice:

import re

str = "The rain in Chaing Mai"
x = re.sub("\s", "9", str)
print(x)

The9rain9in9Chiang Mai

示例 3 – sub（） 函数

sub（） 函数将匹配项替换为您选择的文本：import re

str = “清迈的雨” x = re.sub
（“\s”， “9”， str） print
（x） The9rain9in9清迈

Discussion and class activity
Step 1 KDD (Or most of the DM/ML Process)

- **Initial Selection (data understanding)**
- **Select one of the dataset week3.1 or week3.2 that suit your background.**
  - Read the dataset (Python or R)
  - Use your domain knowledge
  - What is your DS task?
  - Identify type of data (nominal, ordi…)
  - How many samples and feature?
  - Remove feature(s) that is not relevant for your model
  - Extra credit (is there any instance/row) that is useless. How many customer in the dataset should you get rid off?

讨论和课堂活动 第 1 步：KDD（或大部分 DM/ML 流程）

- 初始选择（数据理解）
- 选择适合您背景的数据集 week3.1 或 week3.2 之一。
  - 读取数据集（Python 或 R）
  - 使用您的领域知识
  - 您的 DS 任务是什么?
  - 识别数据类型（标称值、正数等）
  - 有多少个样本和功能?
  - 移除与模型无关的特征
  - 无用的额外信用（是否有任何实例/行）。您应该删除数据集中的多少客户?

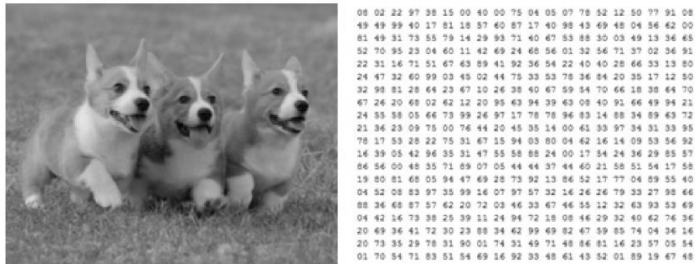
What we have learned so far

- First stage of KDD
- Data pre-processing
  - Handling multi-variance attrs
  - Handling text sentences
  - Splitting sentences into words
  - Splitting words into characters
  - Finding unique words
  - Intro to Regular Expression and operations
  - Words to vectors
  - Word tokenization
  - Feature extraction
- BOW Modelling

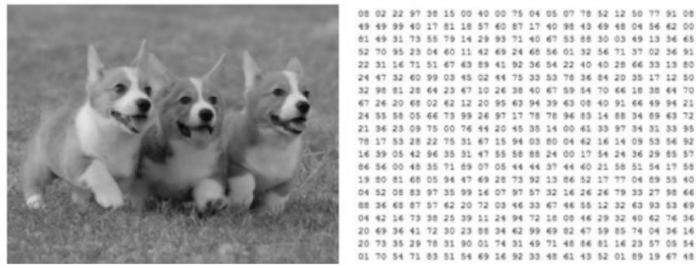
到目前为止，我们学到了什么

- KDD的第一阶段
- 数据预处理
  - 处理多方差属性
  - 处理文本句子
  - 将句子拆分为单词
  - 将单词拆分为字符
  - 查找独特的单词
  - 正则表达式和操作简介
  - 词到向量
  - 单词标记化
  - 特征提取
- 弓形建模

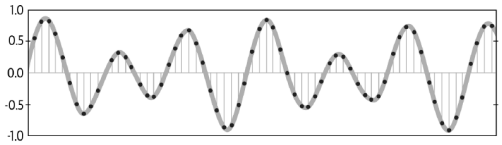
Human vs Computer (Image)



人与计算机（图片）

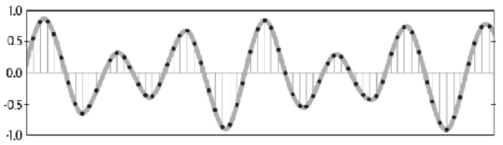


Human vs Computer (Time serie)




```
[-1274, -1252, -1160, -986, -792, -692, -614, -429, -286, -134, -57, -41,
-169, -456, -450, -541, -761, -1067, -1231, -1047, -952, -645, -489, -448,
-397, -212, 193, 114, -17, -110, 128, 261, 198, 390, 461, 772, 948, 1451,
1974, 2624, 3793, 4968, 5939, 6057, 6581, 7302, 7640, 7223, 6119, 5461,
4820, 4353, 3611, 2740, 2004, 1349, 1178, 1085, 901, 301, -262, -499,
-488, -707, -1406, -1997, -2377, -2494, -2605, -2675, -2627, -2500, -2148,
-1648, -970, -364, 13, 260, 494, 788, 1011, 938, 717, 507, 323, 324, 325,
350, 103, -113, 64, 176, 93, -249, -461, -606, -909, -1159, -1307, -1544]
```

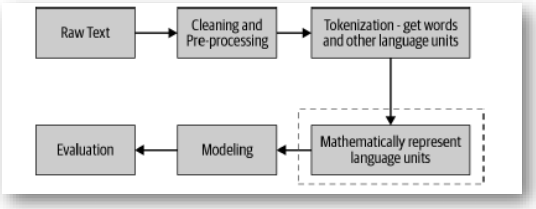
人类与计算机（时间系列）




```
[-1274, -1252, -1160, -986, -792, -692, -614, -429, -286, -134, -57, -41,
-169, -456, -450, -541, -761, -1067, -1231, -1047, -952, -645, -489, -448,
-397, -212, 193, 114, -17, -110, 128, 261, 198, 390, 461, 772, 948, 1451,
1974, 2624, 3793, 4968, 5939, 6057, 6581, 7362, 7640, 7223, 6119, 5461,
4820, 4353, 3611, 2740, 2004, 1349, 1178, 1085, 901, 301, -262, -499,
-488, -707, -1406, -1997, -2377, -2494, -2605, -2675, -2627, -2500, -2148,
-1648, -970, -364, 13, 260, 494, 788, 1011, 938, 717, 507, 323, 324, 325,
350, 103, -113, 64, 176, 93, -249, -461, -606, -909, -1159, -1307, -1544]
```

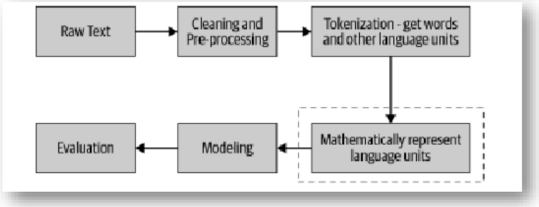
Text representations

- One of the most important step in ML problem
- Poor feature -> poor results



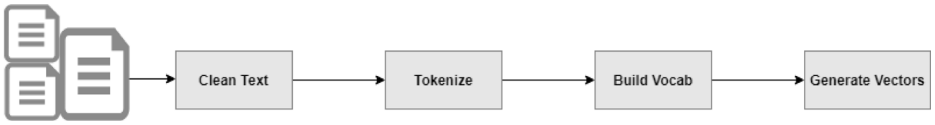
文本表示形式

- ML问题中最重要的步骤之一
- 功能差 - >结果差



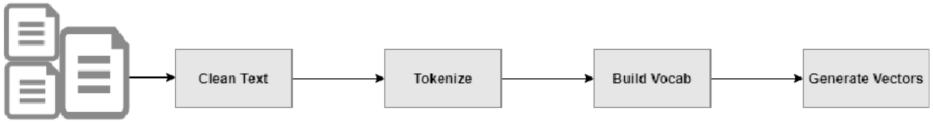
Text pre-processing pipeline with BOW

- Clean the text (white space, upper case, etc.)
- Tokenizing
- Build dictionary
- Filter is also useful (sorting histogram and take top 50)
- The vectors will be use in ML algorithms for document classification or clustering.



使用 BOW 的文本预处理管道

- 清理文本（空格、大写等）
- 代币化
- 构建字典
- 过滤器也很有用（对直方图进行排序并取前 50 名）
- 这些向量将用于 ML 算法中的文档分类或聚类。



SE 953482 Natural Language Processing for SE
66/2
Text Extractions and representation

SE 953482 自然语言

SE 的处理

66/2

文本提取和表示

Midterm coverage

1. NLP Overview
2. Data science methodology
3. Word Tokenization, Text preprocessing
4. Text extraction methods

中期保险

1. NLP概述
2. 数据科学方法论
3.
4. 文本提取方法

Regular expression

- Known as **regex**" or "**regexp**", are a powerful tool used in computing for pattern matching within strings.
- Benefits
  - *Searching and Extracting* – emails, URLs, etc.
  - *Data Validation* – checking input format
  - *String Manipulation*- split string

正则表达式

- 称为正则表达式“或”正则表达式“，是用于计算字符串内模式匹配的强大工具。
- 好处
  - 搜索和提取 – 电子邮件、URL 等。
  - 数据验证 – 检查输入格式
  - 字符串操作 - 拆分字符串

Email validation

```python
import re

# Sample string
text = "Please contact us at support@example.com or sales@example.net."

# Regular expression for matching email addresses
email_pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'

# Find all matches
emails = re.findall(email_pattern, text)

print(emails)
```

电子邮件验证


```python
import re

# Sample string
text = "Please contact us at support@example.com or sales@example.net."

# Regular expression for matching email addresses
email_pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'

# Find all matches
emails = re.findall(email_pattern, text)

print(emails)
```


Email validation

- **\b** at the beginning of the pattern ensures that the email address is not preceded by another word character.

- text = "The cat chased the mouse."
- pattern = r'\bcat\b'

- matches = re.findall(pattern, text)
- print(matches) # Output: ['cat']

电子邮件验证

- \b 确保电子邮件地址前面没有其他单词字符。

- text = “猫追老鼠。”
- 模式 = r'\bcat\b'

- 匹配 = re.findall（pattern， text）
- print（matches） # 输出： ['cat']

URL validation

```python
import re

# Regular expression pattern for a URL
url_pattern = r'http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\\(\\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+'

# Sample string
text = "Check out this website: https://www.example.com or http://example.net."

# Find all matches
urls = re.findall(url_pattern, text)

print(urls)
```

URL 验证

Class exercise *Regular expression practice*

- Write a regular expression to find the word "Python" in a string.
- Write a regular expression to match postal codes like 50300

课堂练习 正则表达式练习

- 编写一个正则表达式以在字符串中查找单词“Python”。
- 编写一个正则表达式来匹配邮政编码，如 50300

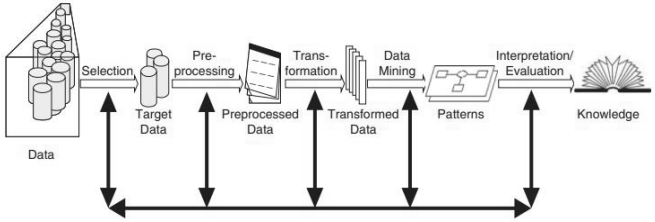


Figure 5 Overview of the steps constituting the knowledge discovery in databases (KDD) process (Fayyad *et al.*, 1996b)

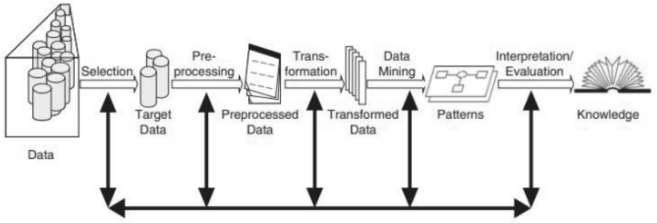


Figure 5 Overview of the steps constituting the knowledge discovery in databases (KDD) process (Fayyad *et al.*, 1996b)

Recall Processes of KDD

1. Learning application domain (initial selection)
2. Data Cleaning
3. Data Integration – where multiple data sources may be combined (heterogenous info. sources)
4. Data transformation
5. Data reduction / feature selection
6. Selecting function of data mining/ml
    1. Prediction/ classification/ associate / clustering

KDD的召回过程

1. 学习应用领域（初始选择） 2.数据清理 3.数据集成 – 可以组合多个数据源（异构信息源）

4. 数据转换
5. 数据缩减/功能选择
6. 数据挖掘/ml的选择功能
    1. 预测/分类/关联/聚类

Recall Processes of KDD (Cont.)

7. Selecting the mining / machine learning algorithms
   Depends on the 6 step
8. Evaluation of the data mining/ml algorithm
9. Result interpretation – visualization of the model, main finding, etc.
10. Action (use of discover knowledge -> public policies, intelligent systems)

KDD的召回过程（续）

7. 选择挖矿/机器学习算法取决于 6 步骤 8.数据挖掘/ml 算法的评估 9.结果解释 – 模型的可视化、主要发现等 10.行动（使用发现知识 - >公共政策、智能系统）

Data cleaning

- 60-70% of the time spending on cleaning data in the Data Mining processes
- “57% of data scientists regard cleaning and organizing data as the least enjoyable part of their work and 19% say this about collecting data sets” (Forbes, 2016)

Source: https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#1852a116f637, (Accessed, August 2018)

数据清理

- 60-70% 的时间用于清理数据挖掘过程中的数据
- “57% 的数据科学家认为清理和整理数据是他们工作中最不愉快的部分，19% 的人这样说收集数据数据集 ”（福布斯，2016 年）

资料来源：https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#1852a116f637，（访问，2018年8月）

Data cleaning – Common error

- Interpretation error: person age >=300
- Inconsistencies : Gender female = [Female, F, Fe]

Error pointing to false value within one dataset

Error	Solution
Redundant white space	Use string functions
Impossible values	Manual overrules
Mistakes during data entry	Manual overrules
Missing values	Remove observation or value
Outliers	Validate and, if erroneous, treat as missing value (remove or insert)

数据清理 – 常见错误

- 解释错误：年龄>=300
- 不一致： 性别 女 = [女， F， Fe]

指向一个数据集中的错误值

错误解决方案	
冗余空格 使用字符串函数 不可能的值 手动覆盖 数据输入过程中的错误 手动覆盖	
缺失值 删除观测值或值 异常值 验证，如果错误，则视为	
	缺失值（移除或插入）

Data cleaning - Common error

Error pointing to inconsistencies between data sets

Error	solution
Deviations from a code book	Match on keys or else use manual overrules
Different units of measurement	Recalculate

- Ignored the sample/case and variables/features
  - case that contain more than 15 % of miss values should be ignored.
  - Variables missing at least 10 % of data were candidates for deletion
  - Ignore the sample, usually perform when target class is missing

数据清理 - 常见错误

指向数据集之间不一致的错误

错误解决方案 与代码本的偏差 按键匹配或使用手动	
推翻不同的测量单位 重新计算	

- 忽略了样本/案例和变量/特征
  - 包含超过 15% 的未命中率值的情况应忽略。
  - 缺少至少 10% 数据的变量是删除的候选变量
  - 忽略示例，通常在缺少目标类时执行

Data Cleaning – (cont.)

Value	Count
Male	156
Female	140
Femalw	12
Malw	10
Malee	5
F	42
M	45

If-else rule
S1 = “Female”
S2 = “Male”
If x == “F”:
X==“Female”

White space: “ Male”, M ale”, Male “

Capital mismatch: “mAle”,

Impossible value:
Check = 0<=age<=120

数据清理 –（续）

值计数	
男 156 女 140 女 12 男 10 男 5 楼 42 米 45	

If-else 规则
S1 = “女性”
S2 = “公”
如果 x == “F”：
X== “女”

空白： “Male”， M ale “， Male ”

资本错配： “mAle”，

不可能的值：检查 =
0<=年龄<=120

Data Cleaning- Handling missing value

Technique	Advantage	Disadvantage
Omit the values	Easy to perform	Lose the information
Use NULL	Easy to perform	Not many algorithms can handle null values
Impute a static value such as 0 or mean	Easy to perform	Lead to false estimation from a model
Model the value	Doesn’t disturb the model too much	Hard to execute Make data assumption

数据清理 - 处理缺失值

技术优势 劣势		
省略值 易于执行 丢失信息 使用 NULL 易于执行 没有多少算法可以处理		
		null 值
插补静态值，例如 0 或平均值	易于执行 导致错误估计	型
对值进行建模 也不会干扰模型	much	难以执行 做出数据假设

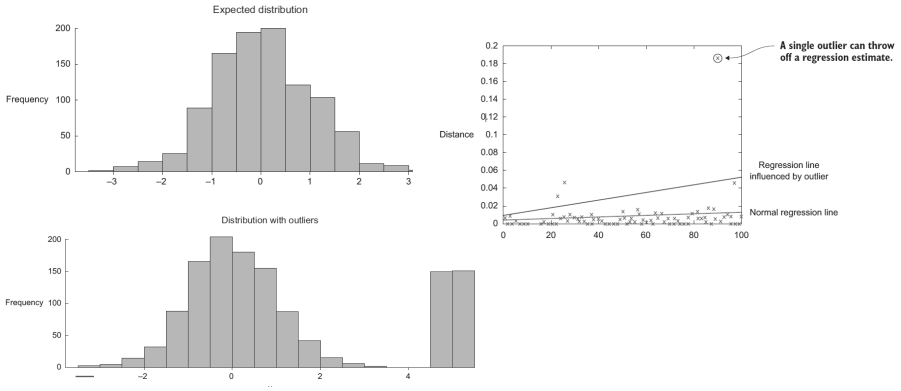
Data Cleaning – (cont.)

- Variables that are Missing At Random (MAR)
  - Use imputation methods
    - Mean or mode substitution (easy to implement)
- Identify outlier and extreme values
    - Binning approach – the most basic technique (sort data to equal bin, then smooth by mean or median
    - Semi-Automated approach - Automate script and domain expert to correct inconsistent data.
    - Clustering approach – using clustering algorithm to group common values,
- Use domain knowledge expert to correct the missing value
  - E.g. Let the weather climate expert comes to check those values.

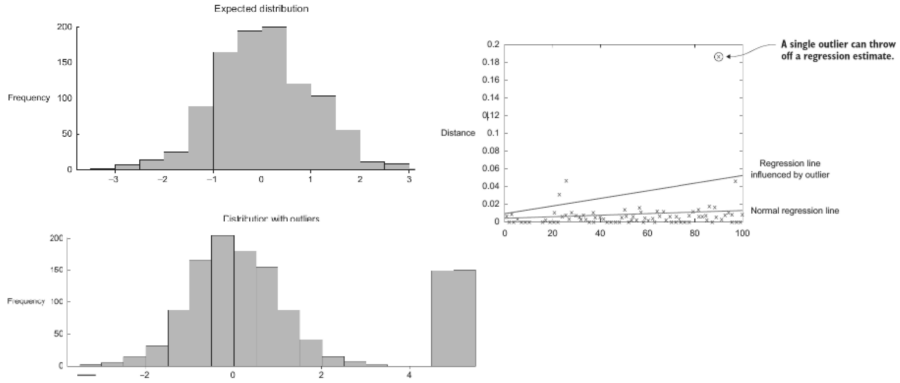
数据清理 –（续）

- 随机缺失的变量 （MAR）
  - 使用插补方法
    - 均值或模态替代（易于实现）
- 识别异常值和极值
    - 分箱方法 – 最基本的技术（将数据排序为相等的分档，然后按平均值或中位数平滑
    - 半自动化方法 - 自动执行脚本和域专家以更正不一致的数据。
    - 聚类方法 – 使用聚类算法对共同值进行分组，
- 使用领域知识专家纠正缺失值
  - 例如，让天气气候专家来检查这些值。

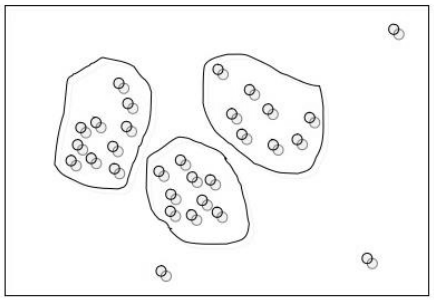
Data cleaning - outlier



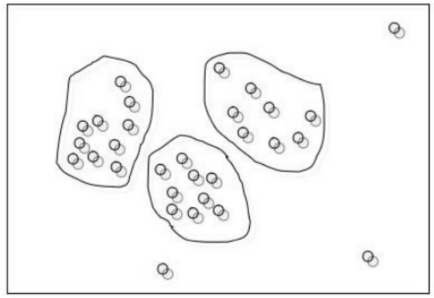
数据清理 - 异常值



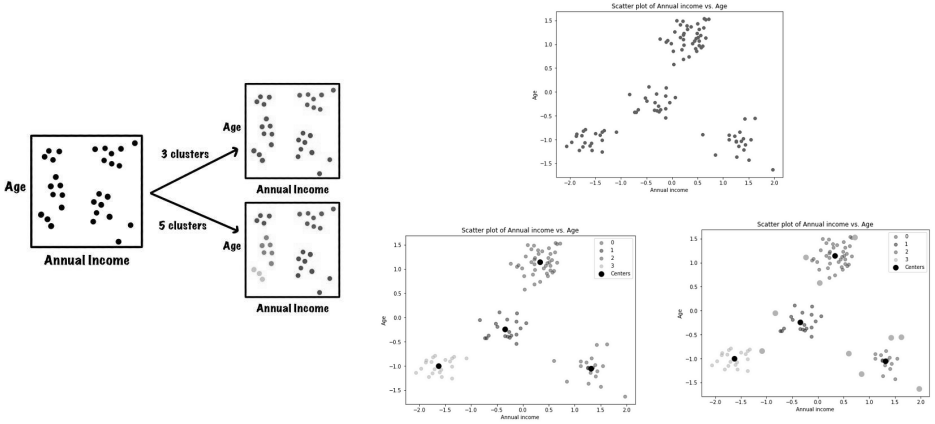
Data Cleaning – using Cluster analysis (Identify outliers/extreme values)



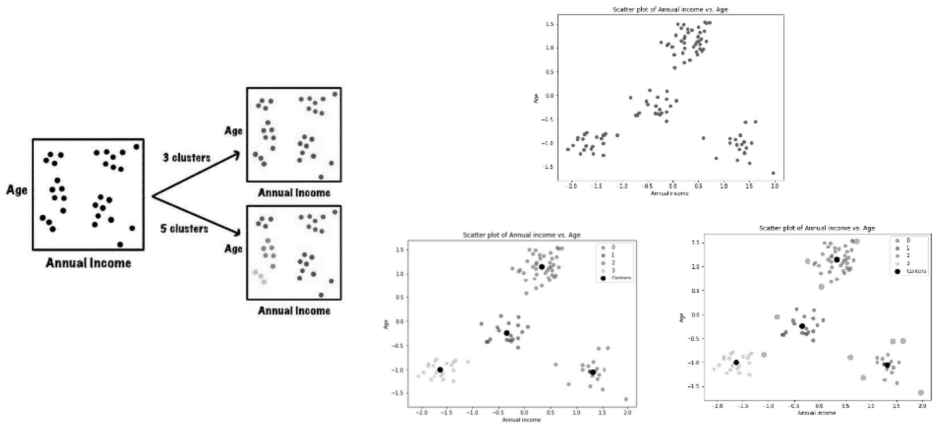
数据清理 – 使用聚类分析（识别异常值/极值）



Outlier Detection Using K-means Clustering



使用 K 均值聚类进行异常值检测



Binning approach – Smooth data

- Sort data : age = [4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34]
- Partition into equal-depth / equal frequency bins: bin_number = 3
  - Bin1 = [4, 8, 9, 15]
  - Bin 2 = [21, 21, 24, 25]
  - Bin 3 = [26, 28, 29, 34]
- Smooth by mean: bin1= [9, 9, 9, 9] , bin 2 = [23, 23, 23, 23], bin3 = [29, 29, 29, 29]
- Smooth by bin boundaries: bin1 = [4, 4, 4, 15], bin2 = [21, 21, 25, 25], bin3= [26, 26, 26, 34]

价格（美元）： 4， 8， 9， 15， 21， 21， 24， 25， 26， 28， 29， 34 4， 8，
9， 15， 21， 21， 24， 25， 26， 28， 29， 34 4， 8， 9， 15

分箱方法 – 平滑数据

- 排序数据： age = [4， 8， 9， 15， 21， 21， 24， 25， 26， 28， 29，
- 划分为等深度/等频条柱：bin_number = 3
  - Bin1 = [4， 8， 9， 15]
  - 箱 2 = [21， 21， 24， 25]
  - 箱 3 = [26， 28， 29， 34]
- 平滑平均值：bin1= [9， 9， 9， 9] ， bin 2 = [23， 23， 23， 23]， bin3 = [29， 29， 29， 29]
- 按条柱边界平滑：bin1 = [4， 4， 4， 15]， bin2 = [21， 21， 25， 25]， bin3= [26， 26， 26， 34]

Basic Text processing (Stop word)

- Remove most common words: and", "the", "is", "in", "on", "that", and "with".
- Noise and irreverent

基本文本处理（非索引字）

- 删除最常见的单词：and “、” the “、” is “、” in “、” on “、” that “和” with”。
- 噪音和不敬

Basic Text processing (Stop words)

- from nltk.corpus import stopwords
- from nltk.tokenize import word_tokenize
-
- sentence = ‘Machine learning is cool!
-
- stop_words = set(stopwords.words('english'))
- word_tokens = word_tokenize(sentence)
-
- filtered_sentence = [w for w in word_tokens if not w in stop_words]
- print(filtered_sentence)

基本文本处理（停用词）

- 从 nltk.corpus 导入非索引字
- 从 nltk.tokenize 导入word_tokenize
-
- sentence = '机器学习很酷!
-
- stop_words = set（stopwords.words（'english'））
- word_tokens = word_tokenize（句子）
-
- filtered_sentence = [w 表示 w in word_tokens 如果不是 w in stop_words]
- 打印（filtered_sentence）

Basic Text processing (Stemming)

- a process of transforming a word to its root form
  - Improve computing process
  - Reduce complexity

Original	Stemming	Lemmatization
New	New	New
York	York	York
is	is	be
the	the	the
most	most	most
densely	dens	densely
populated	popul	populated
city	citi	city
in	in	in
the	the	the
United	Unite	United
States	State	States

基本文本处理（词干提取）

- 将单词转换为其词根形式的过程
  - 改进计算过程
  - 降低复杂性

Original	Stemming	Lemmatization
New	New	New
York	York	York
is	is	be
the	the	the
most	most	most
densely	dens	densely
populated	popul	populated
city	citi	city
in	in	in
the	the	the
United	Unite	United
States	State	States

Basic Text processing (Stemming)

- import nltk
- from nltk.stem import PorterStemmer
- ps = PorterStemmer()

- sentence = "Machine Learning is cool"

- for word in sentence.split():
- print(ps.stem(word)

基本文本处理（词干提取）

- 导入 NLTK
- 从 nltk.stem 导入 PorterStemmer
- ps = PorterStemmer（）

- sentence = “机器学习很酷”

- 对于 sentence.split（） 中的单词：
- 打印（ps.stem（word）

Basic Text processing (Lemmatizing)

- import nltk
- from nltk.stem import WordNetLemmatizer

- lemmatizer = WordNetLemmatizer()

- print(lemmatizer.lemmatize("Machine", pos='n'))
- # pos: parts of speech tag, verb
- print(lemmatizer.lemmatize("caring", pos='v'))

基本文本处理（Lemmatizing）

- 导入 NLTK
- 从 nltk.stem 导入 WordNetLemmatizer

- 词形还原器 = WordNetLemmatizer（）

- print（lemmatizer.lemmatize（“机器”， pos='n'））
- # pos： 词性标签、动词
- print（lemmatizer.lemmatize（“关心”， pos='v'））

Data transformation

- Normalization
  - Scaling attribute values to fall with in a specified range (binning again)
  - Contract or replace with new attribute
    - e.g. measure 3 times, we construct average of the three column
    - e.g. we replace Celsius to Fahrenheit
    - Replace target variable majority vote
  - Dealing with derived attribute (e.g. date of employee)

数据转换

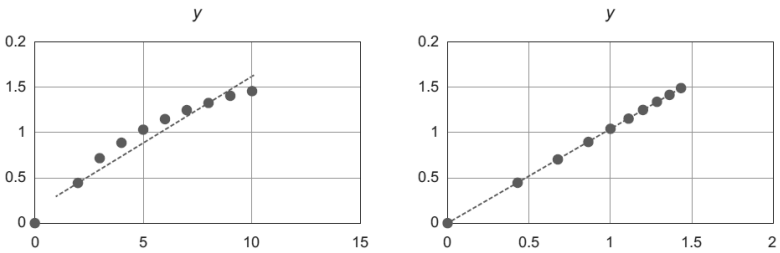
- 正常化
  - 缩放属性值以落在指定范围内（再次装箱）
  - 收缩或替换为新属性
    - 例如，测量 3 次，我们构造三列的平均值
    - 例如，我们将摄氏度替换为华氏度
    - 替换目标变量多数票
  - 处理派生属性（例如员工日期）

Data transformation – continuous value

数据转换 – 持续价值

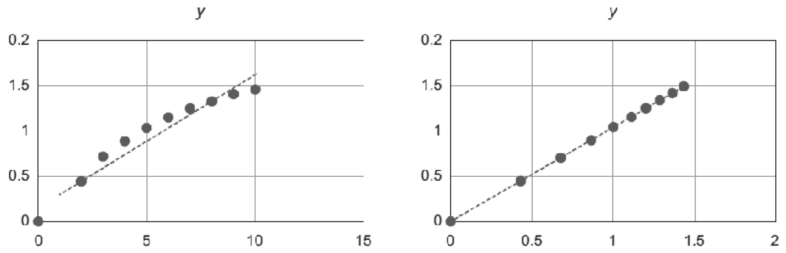
Log normalization

x	1	2	3	4	5	6	7	8	9	10
log(x)	0.00	0.43	0.68	0.86	1.00	1.11	1.21	1.29	1.37	1.43
y	0.00	0.44	0.69	0.87	1.02	1.11	1.24	1.32	1.38	1.46



日志规范化

x	1	2	3	4	5	6	7	8	9	10
log(x)	0.00	0.43	0.68	0.86	1.00	1.11	1.21	1.29	1.37	1.43
y	0.00	0.44	0.69	0.87	1.02	1.11	1.24	1.32	1.38	1.46



Data transformation (cont.)

- Normalization using
  - Decimal Scaling (for NN, SVM) – move the decimal point of value of attribute
  - Min-max function (for NN, SVM) – move the attribute value in the specific ran
  - Select normalization techniques depends on machine learning algorithm and nature of data set (try and try till you get good results)

$$s' = \frac{s - Min}{Max - Min} \qquad z = \frac{x - \mu}{\sigma}$$

数据转换（续）

- 规范化使用
  - 十进制缩放（适用于 NN、SVM）— 移动属性值的小数点
  - Min-max 函数（用于 NN、SVM）— 在特定范围内移动属性值
  - 选择归一化技术取决于机器学习算法和数据集的性质（尝试并尝试，直到获得良好的结果）

$$s' = \frac{s - Min}{Max - Min} \qquad z = \frac{x - \mu}{\sigma}$$

Decimal scaling normalization

- Suppose that the recorded values of x range from − 986 to 917.
- The maximum absolute value of x is 986.
- To normalize by decimal scaling, we therefore divide each value by 1,000
- so that −986 normalizes to −0.986 and 917 normalizes to 0.917.

十进制缩放归一化

- 假设 x 的记录值范围为 − 986 到 917。
- x 的最大绝对值为 986。
- 因此，为了通过十进制缩放进行归一化，我们将每个值除以 1,000
- 因此，−986 归一化为 −0.986,917 归一化为 0.917。

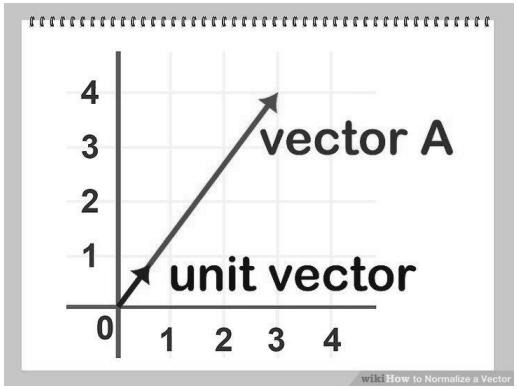
Min-max normalization

- Suppose that the minimum and maximum values for the feature income are 12,000 and 98,000, respectively. We would like to map income to the range 0.0,1.0 . By min-max normalization function. a value of $73,600 for income is transformed to:
- $$s' = \frac{s - Min}{Max - Min}$$
- $\frac{73600-12000}{98000-12000}(1.0 - 0) + 0 = 0.716$

最小值-最大值归一化

- 假设特征收入的最小值和最大值分别为 12,000 和 98,000。我们想将收入映射到 0.0,1.0 的范围。通过最小-最大归一化函数。收入价值 73,600 美元转换为：
- $$s' = \frac{s - Min}{Max - Min}$$
- $\frac{73600 - 12000}{98000-12000}(1.0 - 0 + 0 = 0.716$

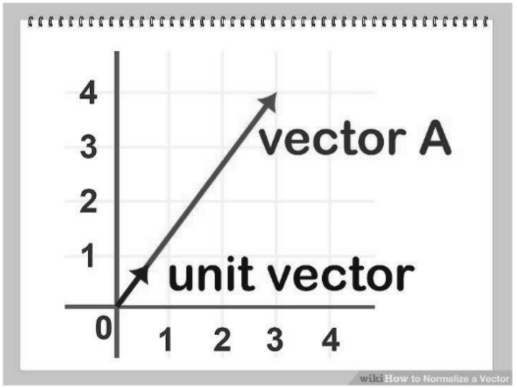
Unit Vector normalize



When your feature have large range,
e.g. dot product can return and overflow,
So, scaling the vector can be benefit.

$$\vec{u} = \frac{\vec{v}}{|\vec{v}|} = \frac{(3,4)}{\sqrt{3^2+4^2}} = \frac{(3,4)}{5} = \left(\frac{3}{5}, \frac{4}{5}\right)$$

单位向量归一化



当您的特征范围很大时，例如点积可以返回和溢出，因此，缩放向量可能是有益的。

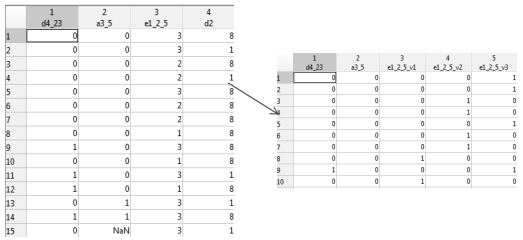
$$\vec{u} = \frac{\vec{v}}{|\vec{v}|} = \frac{(3,4)}{\sqrt{3^2+4^2}} = \frac{(3,4)}{5} = \left(\frac{3}{5}, \frac{4}{5}\right)$$

Data transformation-Discrete value

数据转换 - 离散值

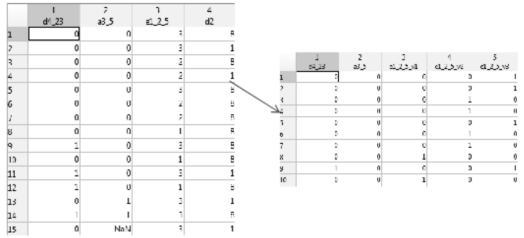
Data transformation- encoding

- Nominal features (one-of-n encoding)
- Ordinal features (Thermometer encoding)



数据转换 - 编码

- 标称特征（其中之一 n 编码）
- 序号特征（温度计编码）



One-of n/one-hot encoding

- Categorical variables need to be converted into forma that could provided machine learning algorithms to perform better

Compay name	Type	Price
BMW	1	220,000
FORD	2	780,000
Toyoya	3	670,000
Toyoya	3	640,000

CN_1	CN_2	CN_3	Price
1	0	0	220,000
0	1	0	780,000
0	0	1	670,000
0	0	1	640,000

n/one-hot 编码之一

- 分类变量需要转换为可以提供机器学习算法以更好地执行的形式

康佩 name	类型 价格	
宝马 1 220,000		
福特 2 780,000		
丰谷 3 670,000		
丰谷 3 640,000		

CN_1 CN_2 CN_3价格		
1 0 0 220,000		
0 1 0 780,000		
0 0 1 670,000		
0 0 1 640,000		

Thermometer Encoding

Compay name	Type	Price
BMW	1	220,000
FORD	2	780,000
Toyoya	3	670,000
Toyoya	3	640,000

CN_1	CN_2	CN_3	Price
0	0	1	220,000
0	1	1	780,000
1	1	1	670,000
1	1	1	640,000

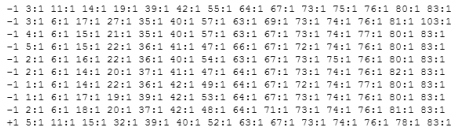
温度计编码

康佩 name	类型 价格	
000 宝马 1 220,000		
福特 2 780,000		
丰谷 3 670,000		
丰谷 3 640,000		

CN_1 CN_2 CN_3价格		
0 0 1 220,000		
0 1 1 780,000		
1 1 1 670,000		
1 1 1 640,000		

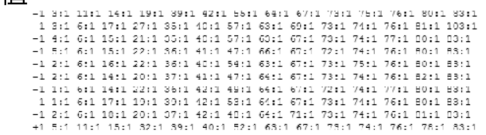
Data transformation – LibSVM format

- Every data mining software require specific data format in order to use in data mining processes.
- Transform data to the LibSVM format.
- <target> <index 1>:<value 1> <index 2>:<value 2>…<index n>.



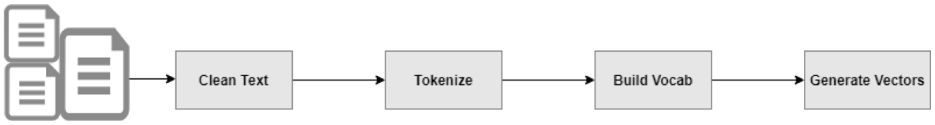
数据转换 – LibSVM 格式

- 每个数据挖掘软件都需要特定的数据格式才能在数据挖掘过程中使用。
- 将数据转换为 LibSVM 格式。
- <索引 1>：<值 1> <索引 2>：<值 2>…<索引 n>。



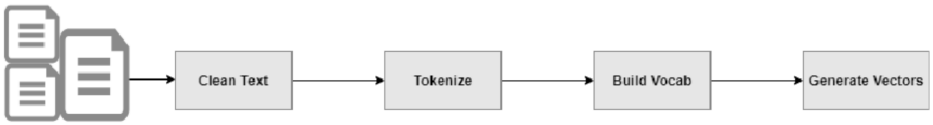
Text pre-processing pipeline with BOW

- Clean the text (white space, upper case, etc.)
- Tokenizing
- Build dictionary
- Filter is also useful (sorting histogram and take top 50)
- The vectors will be use in ML algorithms for document classification or clustering.



使用 BOW 的文本预处理管道

- 清理文本（空格、大写等）
- 代币化
- 构建字典
- 过滤器也很有用（对直方图进行排序并取前 50 名）
- 这些向量将用于 ML 算法中的文档分类或聚类。

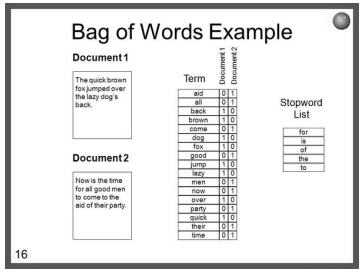


Data transformation-text value

数据转换 - 文本值

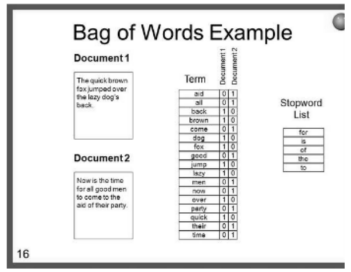
Bag of Words (BOW)

- Machine learning can’t work with raw text
- Text must be convert to numbers, vectors of numbers
- Usually this is called “feature extraction”
- Bag of words is the most simple approach.
- Represent of string based on frequency of entities



词袋 （BOW）

- 机器学习无法处理原始文本
- 文本必须转换为数字，数字向量
- 通常这称为“特征提取”
- 一袋字是最简单的方法。
- 基于实体频率的字符串表示形式



BOW (cont.)

- Sentence 1: "I love apples."
- Sentence 2: "I do not love oranges.“
- Vocab = ["I", "love", "apples", "do", "not", "oranges"]
- Vector for Sentence 1: [1, 1, 1, 0, 0, 0]
- Vector for Sentence 2: [1, 1, 0, 1, 1, 1]

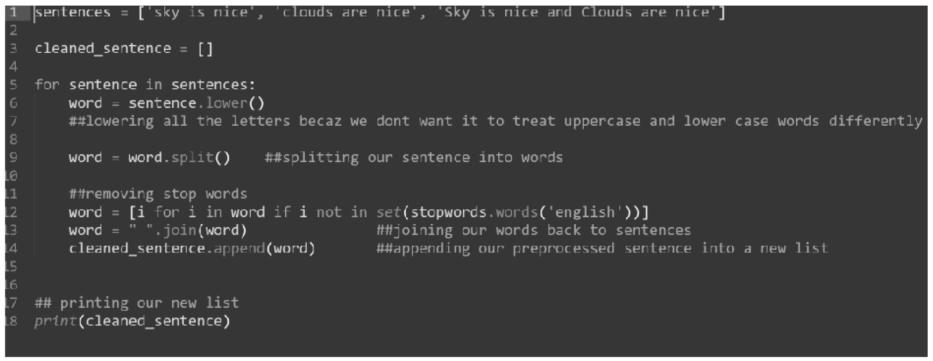
弓（续）

- 第 1 句： “我喜欢苹果。
- 第 2 句： “我不喜欢橘子。
- 词汇 = [“我”， “爱”， “苹果”， “做”， “不”， “橙子”]
- 句子 1 的向量：[1， 1， 1， 0， 0， 0]
- 句子 2 的向量：[1， 1， 0， 1， 1， 1]

BOW in action

```
sentences = ['sky is nice', 'clouds are nice', 'Sky is nice and Clouds are nice']

cleaned_sentence = []

for sentence in sentences:
    word = sentence.lower()
    ##lowering all the letters becaz we dont want it to treat uppercase and lower case words differently

    word = word.split()    ##splitting our sentence into words

    ##removing stop words
    word = [i for i in word if i not in set(stopwords.words('english'))]
    word = " ".join(word)              ##joining our words back to sentences
    cleaned_sentence.append(word)      ##appending our preprocessed sentence into a new list


## printing our new list
print(cleaned_sentence)
```

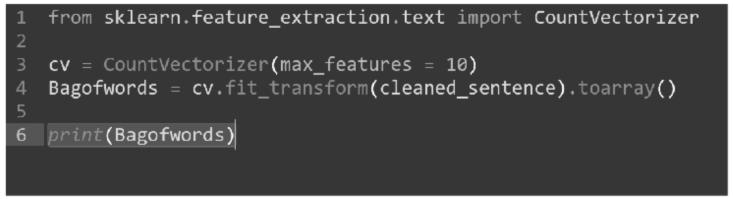
BOW在行动



BOW in action

```
from sklearn.feature_extraction.text import CountVectorizer

cv = CountVectorizer(max_features = 10)
Bagofwords = cv.fit_transform(cleaned_sentence).toarray()

print(Bagofwords)
```

BOW在行动



BOW modeling in action

```
wordfreq = {} # create dict and iterate through each sentence
for sentence in corpus:
    tokens = nltk.word_tokenize(sentence)
    for token in tokens:
        if token not in wordfreq.keys():
            wordfreq[token] = 1
        else:
            wordfreq[token] += 1

# filter dimension to 300
import heapq
most_freq = heapq.nlargest(300, wordfreq, key=wordfreq.get)

sentence_vectors = [] # create sentence list and iterate throgh each sentence
for sentence in corpus:
    sentence_tokens = nltk.word_tokenize(sentence)
    sent_vec = []
    for token in most_freq:
        if token in sentence_tokens:
            sent_vec.append(1)
        else:
            sent_vec.append(0)
```

```
from sklearn.feature_extraction.text import CountVectorizer
matrix = CountVectorizer(max_features=300)
X = matrix.fit_transform(data).toarray()
```


BOW 建模的实际应用


```
wordfreq = {} # create dict and iterate through each sentence
for sentence in corpus:
    tokens = nltk.word_tokenize(sentence)
    for token in tokens:
        if token not in wordfreq.keys():
            wordfreq[token] = 1
        else:
            wordfreq[token] += 1

# filter dimension to 300
import heapq
most_freq = heapq.nlargest(300, wordfreq, key=wordfreq.get)

sentence_vectors = [] # create sentence list and iterate throgh each sentence
for sentence in corpus:
    sentence_tokens = nltk.word_tokenize(sentence)
    sent_vec = []
    for token in most_freq:
        if token in sentence_tokens:
            sent_vec.append(1)
        else:
            sent_vec.append(0)
```

```
from sklearn.feature_extraction.text import CountVectorizer
matrix = CountVectorizer(max_features=300)
X = matrix.fit_transform(data).toarray()
```


Limitation of BOW

- **High Dimensionality:** With a large vocabulary, the feature space becomes very large, leading to issues like the curse of dimensionality.
- **Context Ignorance:** doesn't look at the order or context of words, so lines with different meanings but the same words would be represented the same way.
- **Sparsity:** The vectors tend to be sparse (mostly zeros), which can be inefficient for computation, especially with large vocabularies.

BOW的局限性

- 高维性：词汇量大时，特征空间变得非常大，导致维度诅咒等问题。
- 上下文忽略：不查看单词的顺序或上下文，因此具有不同含义但相同单词的行将以相同的方式表示。
- 稀疏性：向量往往是稀疏的（主要是零），这对于计算来说可能效率低下，尤其是在词汇量大的情况下。

Bag of Ngrams (BoN)

- N-grams - contiguous sequences of 'n' items from a given sample of text or speech
- Unigram – “The quick brown fox"
- Bi-gram - "The quick", "quick brown", "brown fox"
- Tri-gram - "The quick brown", "quick brown fox"

一袋 Ngrams （BoN）

- N-grams - 来自给定文本或语音样本的“n”项的连续序列

- Unigram – “快速的棕色狐狸”
- Bi-gram - “快速”、“快速棕色”、“棕色狐狸”
- 三角星 - “快速棕色”，“快速棕色狐狸”

BoN in action

```python
def generate_ngrams(text, n):
    # Split the text into words
    words = text.split()
    # Create n-grams
    ngrams = zip(*[words[i:] for i in range(n)])
    return [" ".join(ngram) for ngram in ngrams]

# Example usage
text = "The quick brown fox jumps over the lazy dog"
bigrams = generate_ngrams(text, 2)
trigrams = generate_ngrams(text, 3)

print("Bigrams:", bigrams)
print("Trigrams:", trigrams
```

BoN 在行动


```python
def generate_ngrams(text, n):
    # Split the text into words
    words = text.split()
    # Create n-grams
    ngrams = zip(*[words[i:] for i in range(n)])
    return [" ".join(ngram) for ngram in ngrams]

# Example usage
text = "The quick brown fox jumps over the lazy dog"
bigrams = generate_ngrams(text, 2)
trigrams = generate_ngrams(text, 3)

print("Bigrams:", bigrams)
print("Trigrams:", trigrams
```

Workshop 2 (Basic Processing and Representation)

- 1 Use the previous dataset “spam dictation”
- 2 Preprocess text including:
  - Remove white space
  - Remove anything that is not English
  - Calculate word length and added with column name “length”
- 3 Create new column name “text2”

	label	text	length
1463	ham	ok good later come lucky told earlier later pp...	114
2313	ham	guys	23
3290	ham	smoking people use wylie smokes justify ruinin...	85
2604	ham	times job today ok umma ask speed	57
4018	spam	ve selected stay british hotels holiday valued...	159

工作坊 2（基本处理和表示）

- 1 使用上一个数据集“垃圾邮件听写”
- 2 预处理文本，包括：
  - 删除空白
  - 删除任何非英语内容
  - 计算字长并加上列名“length”
- 3 创建新列名称“text2”

	label	text	length
1463	ham	ok good later come lucky told earlier later pp...	114
2313	ham	guys	23
3290	ham	smoking people use wylie smokes justify ruinin...	85
2604	ham	times job today ok umma ask speed	57
4018	spam	ve selected stay british hotels holiday valued...	159

Workshop 2 (Basic Processing and Representation)

- Todays Voda numbers ending 1225 are selected to receive a å£50award. \
- If you have a match please call 08712300220 quoting claim code 3100 standard rates app")
- 'todays voda numbers ending selected receive award match quoting claim code standard rates app'

- 4. Use labelEncoder method to convert class target
- 5. Use CountVectorize to perform BOW
- 6. List Top 5 and bottom 5 of transform sample to show the results and submit your works to MS team

工作坊 2（基本处理和表示）

- 今天以 1225 结尾的 Voda 号码被选中获得 50 英镑的奖励。\
- 如果您有匹配，请致电08712300220报价索赔代码 3100 标准费率应用程序“）
- “今天的 Voda 号码结束选定的接收奖励匹配报价索赔代码标准费率应用程序”

- 4.使用 labelEncoder 方法转换类目标
- 5. 使用 CountVectorize 执行 BOW
- 6. 列出转换样本的前 5 名和后 5 名以显示结果并将您的作品提交给 MS 团队

What's you techniques or steps?

(Write your own algorithm to determine "Positive" or "Negative")

Prize: Ling Tote Bags!

你的技术或步骤是什么?

(编写自己的算法来确定"正"或"负")

奖品:Ling Tote Bags!

Tokenization:

- breaking text into smaller units, usually words or subwords
- facilitates analysis and processing of text

"The quick brown fox jumps over the lazy dog"

Turns into

["The", "quick", "brown", "fox", "jumps", "over", "the", "lazy", "dog"]

代币化:

- 将文本分解为更小的单元,通常是单词或子单词
- 便于文本的分析和处理

"敏捷的棕色狐狸跳过懒惰的狗"

变成

["The", "Quick", "Brown", "Fox", "Jumps", "Over", "the", "lazy",

Stopwords Removal:

- common words that are often considered irrelevant in the context
- Common ones in English are “a”, “an”, “the”, “is”, etc.

"The quick brown fox jumps over the lazy dog"

Turns into

"~~The~~ quick brown fox jumps over ~~the~~ lazy dog"

停用词删除：

- 在上下文中通常被认为无关紧要的常用词
- 英语中常见的有“a”、“an”、“the”、“is”等。

“敏捷的棕色狐狸跳过懒惰的狗”

变成

“敏捷的棕色狐狸跳过懒惰的狗”

Stemming:

- removing prefixes or suffixes
- aiding in analysis and understanding

The runners are running in the race, and the fastest runner will win.

Turns into

The **runner** are **run** in the race, and the fastest runner will win.

堵塞：

- 删除前缀或后缀 - 帮助分析和理解 跑步者在比赛中奔跑，跑得最快的人将获胜。

变成

跑步者在比赛中奔跑，跑得最快的跑步者将获胜。

Lemmatization:

- reducing words to their base or root form
- aiding in analysis and understanding

"Is" -> "be"
"Better" -> "good"

"She is happy". -> "She be happy".
"They are happy." -> "They be happy."

词形还原：

- 将单词简化为基本形式或词根形式 - 有助于分析和理解

"是" -> "是"
"更好" -> "好"

"她很开心" 。-> "她很快乐" 。 "他们很开心。" -> "他们很开心。"

Or use pre-built solutions?

With the programming language of your choice, feed the sentences to Google Cloud API!

With this API, can you separate "negative" and "positive" comments?

Prize: Ling T-Shirts!

还是使用预构建的解决方案?

使用您选择的编程语言，将句子提供给 Google Cloud API！

使用此 API，您可以区分"负面"和"正面"评论吗?

奖品：Ling T恤！