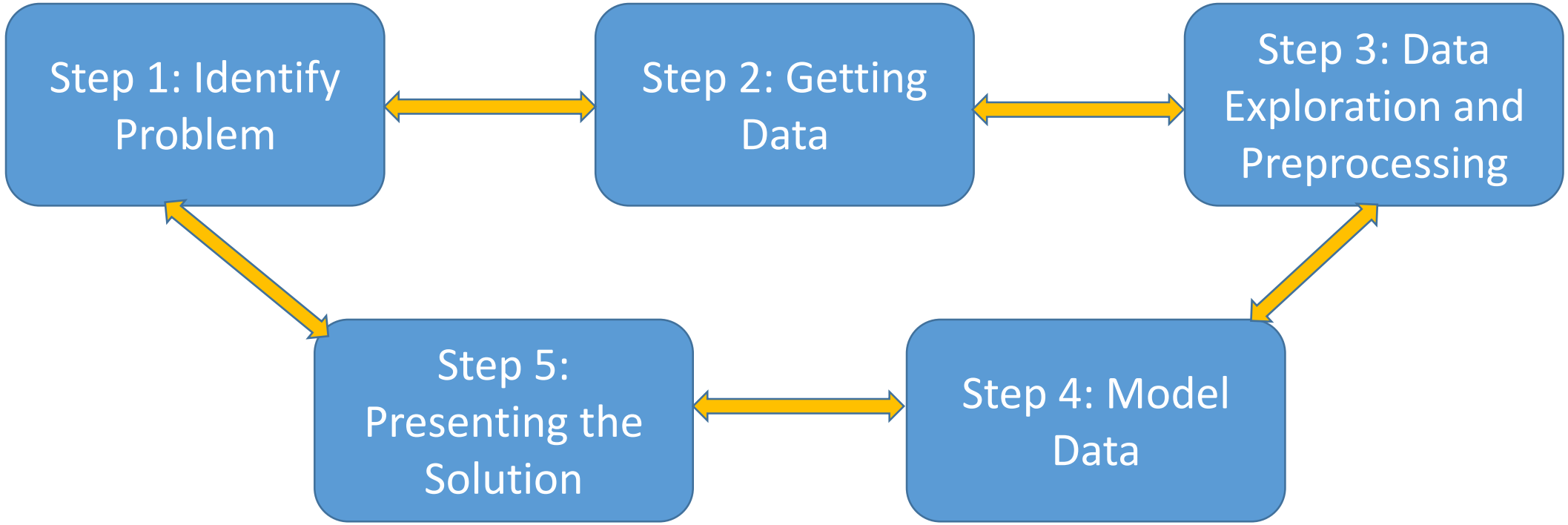
Identify the Problem Getting the Data

Dr. Qiwei Gan

Machine Learning Workflow



Machine Learning Workflow: Step 1

Step 1: Identify Problem

- Understand what problems you are trying to tackle.
- What is the best way to phrase my question(s) as a machine learning problem?
- Always keep the big picture in mind.
- It is easy to lose sight of the ultimate goal.
- It is more focused on business perspective rather than technical perspective.

Step 1: Identify Problem

- Brief introduction in class
- It does NOT mean it is not important!
- In fact, it is more than important than other steps sometimes.

- In essence:
  - Big picture, big picture, big picture.
  - Context, context, contest.

Summary of the article

- What Kinds of Business Problems Machine Learning Can Handle
  - **Is the prediction you're trying to make (or decision you're trying to make) complex enough to warrant ML in the first place?**
  - **How are you framing your problems into ML problems?**
  - **Do you have new data and clean data?**
  - **Does your data have existing labels to help a machine make sense of it?**
  - **Can your solution to this problem afford for some allowance of error?**
- Pointers for Apply Machine Learning to Business Problems
  - **Begin with a priority problem, not a toy problem**
  - **You can give it data, but all of the context must come from you**
  - **Expect to tinker, tweak, and adjust to find ROI**

How to Build up ML Sense?

- Try to imagine applying ML in every of your problem (mental exercise)
- Consider pros and cons, including time and cost
- Good start is to divide your complex problem into smaller tasks and see if they can be dealt with ML.
- Search solutions of type of your problem online, see if others have ever used ML and how successfully it was.

ML Step 2: Getting Data

- Sources of data:
  - Structured (more like tabulate data)
    - Databases
    - Structured files: csv, json, Excel, tab-delimited text files, SAS, SPSS, etc.
    - APIs: Twitter, Facebook, Pinterest, Google, Yahoo Finance, etc.
    - Sensor data from smart devices: accelerometer, gyroscope, etc.
  - Unstructured (free style data)
    - Text files without specific tabulated structure, articles…
    - Other types of files without explicit structures, or you need extra preprocessing to extract structures: images, video, sounds, etc.
    - Webpages

ML Step 2: Getting Data

- State of data:
  - Static: exists when you fetch them,
  - Dynamic (streaming data): may not exists when you fetch them and will be sent to you when they become available.

- Type of data:
  - Numerical
  - Categorical
  - Text
  - Date/time
  - Etc.

ML Step 2: Getting Data

- For most of our ML problems in this course, the scikit learn ML algorithms take in numerical and structured data.
- Keep this in mind when you get your data

- Also keep in mind structured data do not necessarily mean that they are “clean data”, or “complete data”.

- Quote from my statistics professor:
  - Model produces data, and NOT data produces model!

ML Step 2: Getting Data

- Demonstration of web scraping
  - Consider the size of information on the Internet
  - Accumulation of new information: Every second, approximately 6,000 tweets are tweeted; more than 40,000 Google queries are searched; and more than 2 million emails are sent.
  - Surface web vs. deep web

ML Step 2: Getting Data

- Web scraping with requests and beatifulsoup
  - import requests
  - web=requests.get(your url,parameters)
  - Content=web.content or web.text

  - from bs4 import BeautifulSoup
  - soup= BeautifulSoup(content, parser)
  - soup.tag
  - soup.tag[entities]
  - soup.find(tag,[entities])
  - soup.find_all(tag,[entities])

ML Step 2: Getting Data

- Getting data from APIs
- Twitter API
- Tweepy wrapper

Content of a tweet

- The key attributes are the following (in json format):
  - text: the text of the tweet itself
  - created_at: the date of creation
  - favorite_count, retweet_count: the number of favourites and retweets
  - favorited, retweeted: boolean stating whether the authenticated user (you) have favourited or retweeted this tweet
  - lang: acronym for the language (e.g. “en” for english)
  - id: the tweet identifier
  - place, coordinates, geo: geo-location information if available
  - user: the author’s full profile
  - entities: list of entities like URLs, @-mentions, hashtags and symbols
  - in_reply_to_user_id: user identifier if the tweet is a reply to a specific user
  - in_reply_to_status_id: status identifier id the tweet is a reply to a specific status

Prerequisite: Get Access Token

- You have to have access token to use twitter API
  - Register a twitter developer account
  - Create an app
  - Get your keys and tokens
  - https://themepacific.com/how-to-generate-api-key-consumer-token-access-key-for-twitter-oauth/994/

- Understand rate limits
  - If everyone is constantly sending requests to twitter servers, they will crash.
  - For GET method, the limit window is about 15 minutes
  - For each type of requests, the rate limits are different, see web for info:
    - For public query, 180 per window
    - https://dev.twitter.com/rest/public/rate-limits

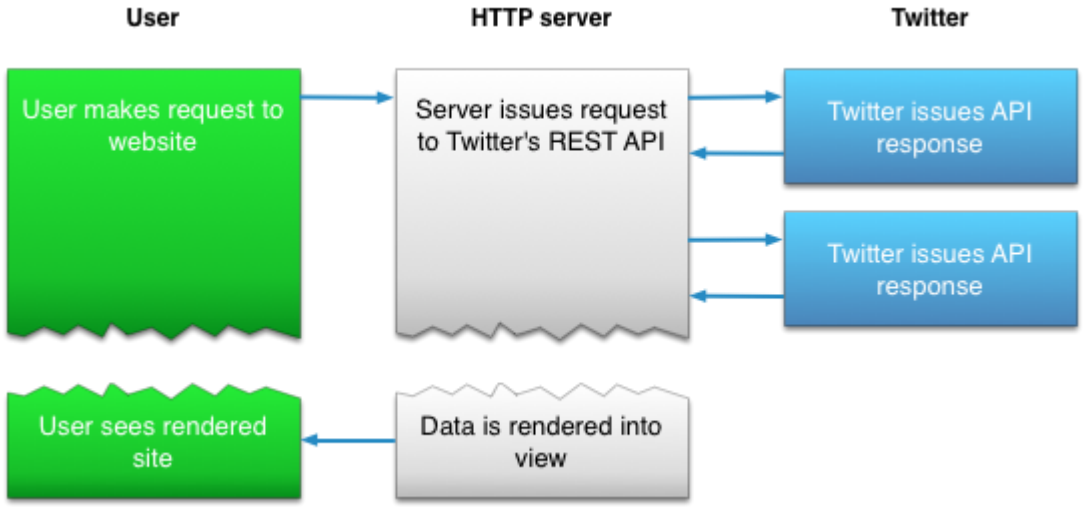
Tweepy

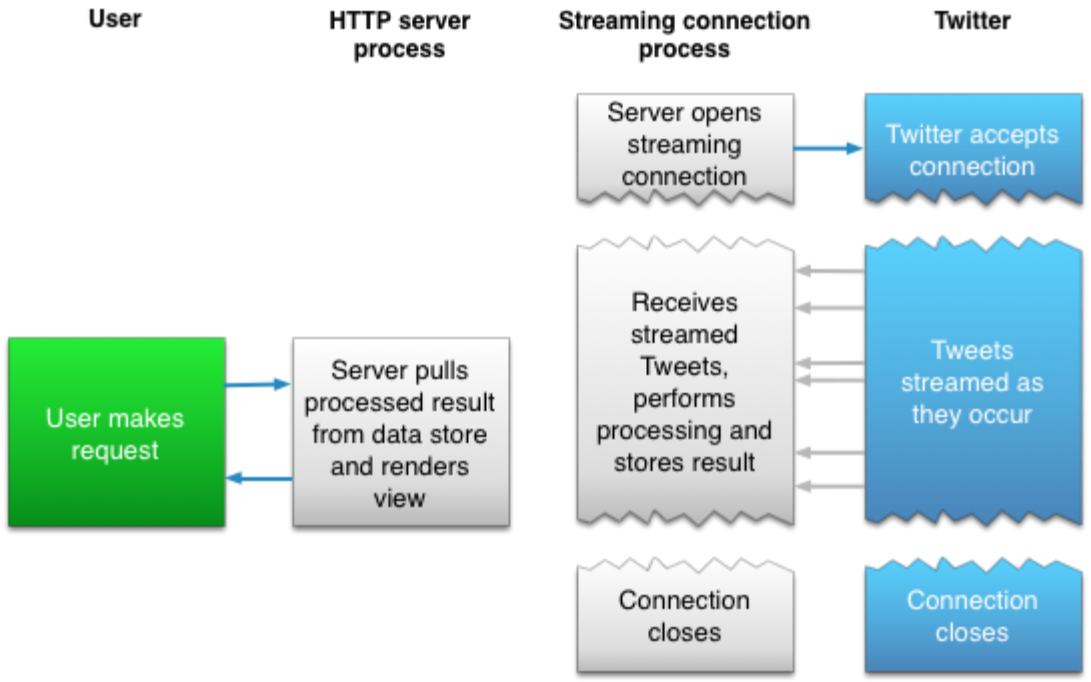
```
import tweepy

consumer_key = ‘’
consumer_secret = ‘’
access_token_key = ‘’
access_token_secret = ‘’

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token_key, access_token_secret)

api = tweepy.API(auth)
```

A Few Common Used Methods

- api.me()
- api.get_user(***id/user_id/screen_name***)
- api.followers([id/screen_name/user_id][, cursor])
- api.search(q[, lang][, locale][, rpp][, page][, since_id][, geocode][, sho w_user])

Twitter REST vs Streaming API





Twitter Streaming API

- Filter method
  - Track: key word lists
  - Follow: user id lists,
  - Locations: list of longitude,latitude pairs

Other Libraries for Web Scrapping

- The Farm: Requests
- The Stew: Beautiful Soup 4
- The Salad: lxml
- The Restaurant: Selenium
- The Chef: Scrapy

Questions?