Data Mining and Business Intelligence

Lecture 2: Data Structure, Data Reduction, and Data Acquisition

Jing Peng
University of Connecticut

1/30/20

Quick Questions

- A dataset has 80% positive examples and 20% negative examples, what is the AUC if a classifier predict every example as positive?
  - 80%
  - 50%
  - 20%

- A dataset has 80% positive examples and 20% negative examples, what is the accuracy of random guess (50% positive & 50% negative)?
  - 80%
  - 50%
  - 20%

Data Structure

Common Data Structures

- Cross-sectional Data
- Time Series Data
- Panel Data
- Network Data

Cross-Sectional Data

- Measurements of many subjects at one point or period of time

- Examples

	A	B	C	D	E	F	G	H
1	**issue_d**	**loan_amnt**	**funded_amnt**	**term**	**int_rate**	**installment**	**grade**	**loan_status**
2	Jan-2015	6000	6000	36 months	6.99	185.24	A	Current
3	Nov-2015	15000	15000	36 months	15.41	523	D	Current
4	Sep-2014	6000	6000	36 months	12.99	202.14	C	In Grace Period
5	Feb-2015	20000	20000	36 months	14.65	689.89	C	Current
6	Jul-2014	35000	35000	36 months	13.98	1195.88	C	Current
7	Jan-2014	20000	20000	36 months	13.53	679	B	Charged Off
8	Jun-2015	27175	27175	60 months	17.57	683.73	D	Current
9	Aug-2014	18825	18825	36 months	15.61	658.22	D	Fully Paid
10	Sep-2015	5000	5000	36 months	8.18	157.1	B	Current
11	Jul-2014	10000	10000	36 months	12.49	334.49	B	Charged Off
12	Apr-2014	20000	20000	60 months	11.99	444.79	B	Current
13	Dec-2012	5000	5000	36 months	10.16	161.72	B	Fully Paid

Time Series Data

- Measurements of one or more subjects at various points of time

Panel Data

- Repeated measurements of many subjects over time (time series and cross-sectional)

MRPP balanced panel:

person	year	income	age	sex
1	2016	1300	27	1
1	2017	1600	28	1
1	2018	2000	29	1
2	2016	2000	38	2
2	2017	2300	39	2
2	2018	2400	40	2

MRPP unbalanced panel:

person	year	income	age	sex
1	2016	1600	23	1
1	2017	1500	24	1
2	2016	1900	41	2
2	2017	2000	42	2
2	2018	2100	43	2
3	2017	3300	34	1

Source: https://en.wikipedia.org/wiki/Panel_data

Network Data

- Measurements of relationships among subjects

Who reports liking whom?

	Choice:			
Chooser:	Bob	Carol	Ted	Alice
Bob	---	0	1	1
Carol	1	---	0	1
Ted	0	1	---	1
Alice	1	0	0	---

- Bipartite networks: customer-product network, user-movie network

Which Data Structure to Use?

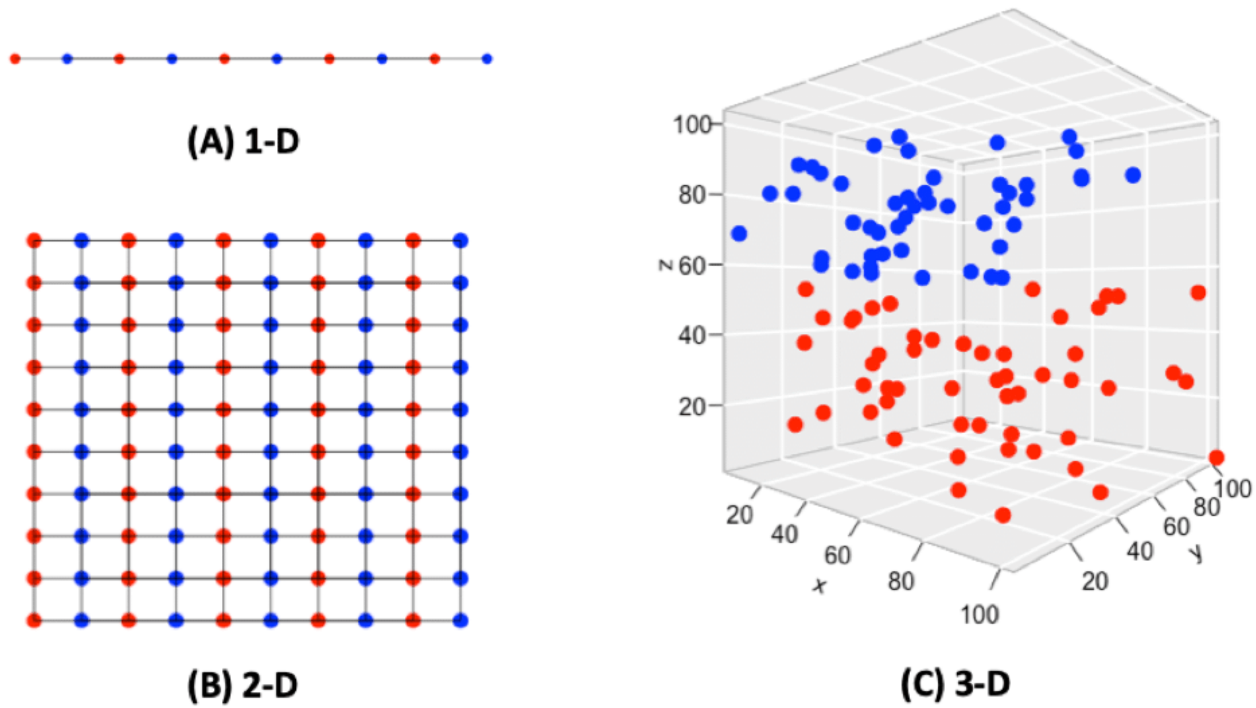
- It depends on a lot of things
  - The nature of your data
  - The purpose of your analysis
  - The unit of analysis
  - The model you plan to use
    - Data mining: ?
    - Time series: ?

Data Reduction

Data Reduction

- Too many observations (rows)
  - Computational burden
- Too many features (columns)
  - Curse of dimensionality
  - Redundant or irrelevant information
- Objective of data reduction: obtain a reduced set (sample) of data that is much smaller in volume yet produce the same (or almost the same) results

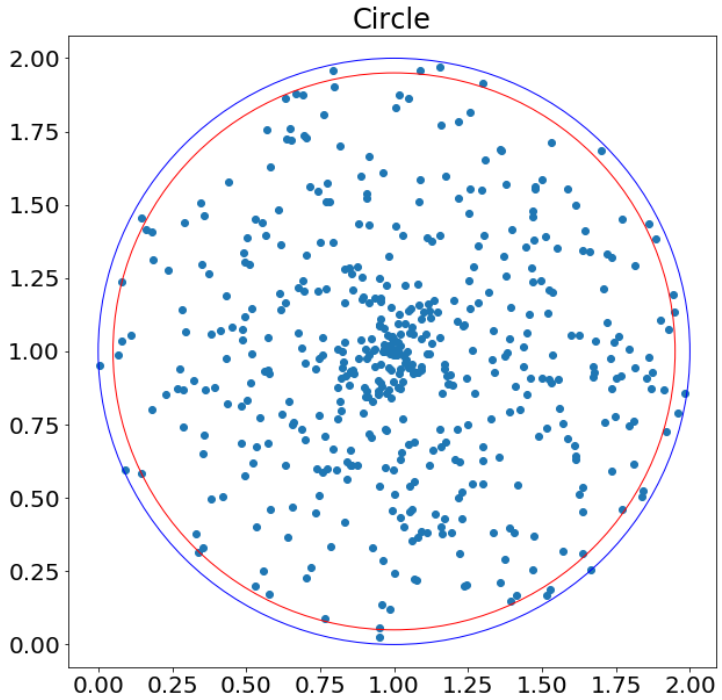
Curse of Dimensionality



Curse of dimensionality: various phenomena that arise in high-dimensional spaces (e.g., hundreds or thousands of dimensions)

Curse of Dimensionality: Distance Metric

- Euclidean distance is appropriately equal for any two data points!



Volume of Hypersphere: $V = cR^d$
Volume of unit hypersphere: $V = c$
Shell of unit hypersphere: $V_{shell} = \mathrm{c} - cr^d$

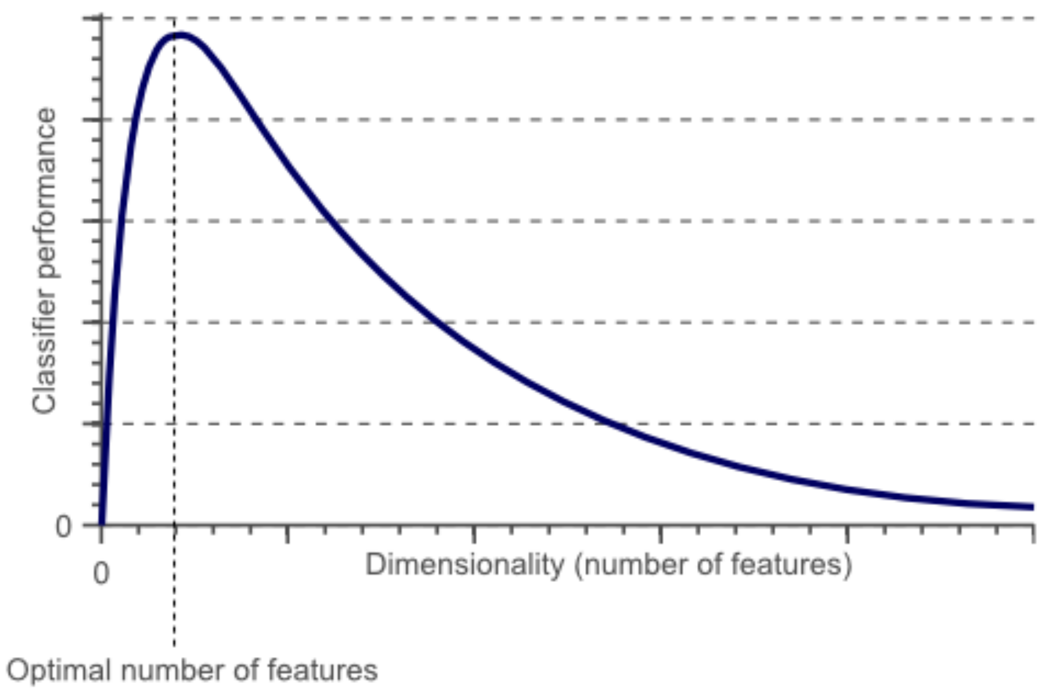
Though the radius of the inner hypersphere r is very close to 1, $cr^d \rightarrow 0$ when d is large

Demo: high_dim_distance.R

https://towardsdatascience.com/on-the-curse-of-dimensionality-b91a3a51268
https://towardsdatascience.com/the-curse-of-dimensionality-50dc6e49aa1e

Curse of Dimensionality: Hughes phenomenon

- With a fixed number of training samples, the predictive power of a classifier or regressor first increases with number of dimensions/features but then decreases



Curse of Dimensionality

- The amount of data needed to produce reliable results grows exponentially with the number of dimensions, because we need several examples for each possible combination of values

Dimensionality Reduction

- Feature projection
  - Principal component analysis (unsupervised)
  - Linear discriminant analysis (supervised)
  - Autoencoder (unsupervised, nonlinear)

- Feature selection
  - Filter method
  - Wrapper method
  - Embedded method

https://en.wikipedia.org/wiki/Dimensionality_reduction

Feature Selection: Filter Method

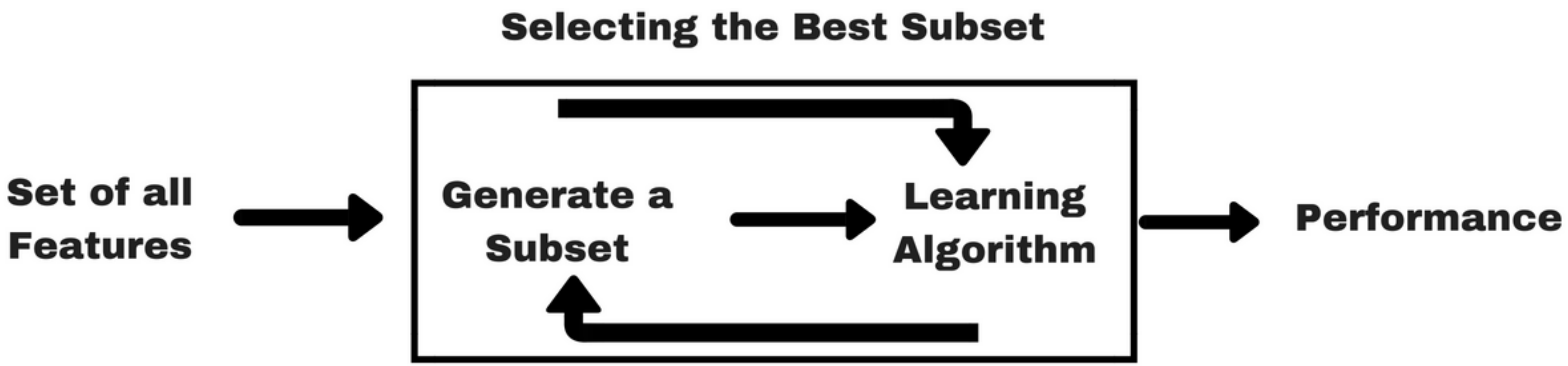
- Select variables without training any prediction model. Instead, they select the top-K features based on their relationships with the outcome, such as
  - Correlation
  - Chi-square test
  - Information gain

- Decisions to make
  - which measure to use
  - how many features to keep

https://en.wikipedia.org/wiki/Feature_selection#Filter_method
https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/

Feature Selection: Wrapper Method

- Use prediction performance on the holdout sample to find the best combination of features, with different strategies:
  - **Step Forward**: keep adding features that best improves the current model until no performance improvement
  - **Step Backward**: start with all features and keep removing features from the current model until no performance improvement
  - **Exhaustive**: exhaust all potential feature combinations and pick the best one



https://en.wikipedia.org/wiki/Feature_selection#Filter_method
https://stackabuse.com/applying-wrapper-methods-in-python-for-feature-selection/

Feature Selection: Embedded Method

- Perform feature selection as part of the model construction process, typically by penalizing large regression coefficients (regularization)
  - Lasso (L1 penalty)
  - Ridge (L2 penalty)
- Estimates tend to be more stable with the presence of penalty terms, though they are not unbiased anymore (larger bias, smaller variance)

https://en.wikipedia.org/wiki/Feature_selection#Filter_method
https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/

Regularization: Ridge vs. LASSO

- Ridge Regression for OLS

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

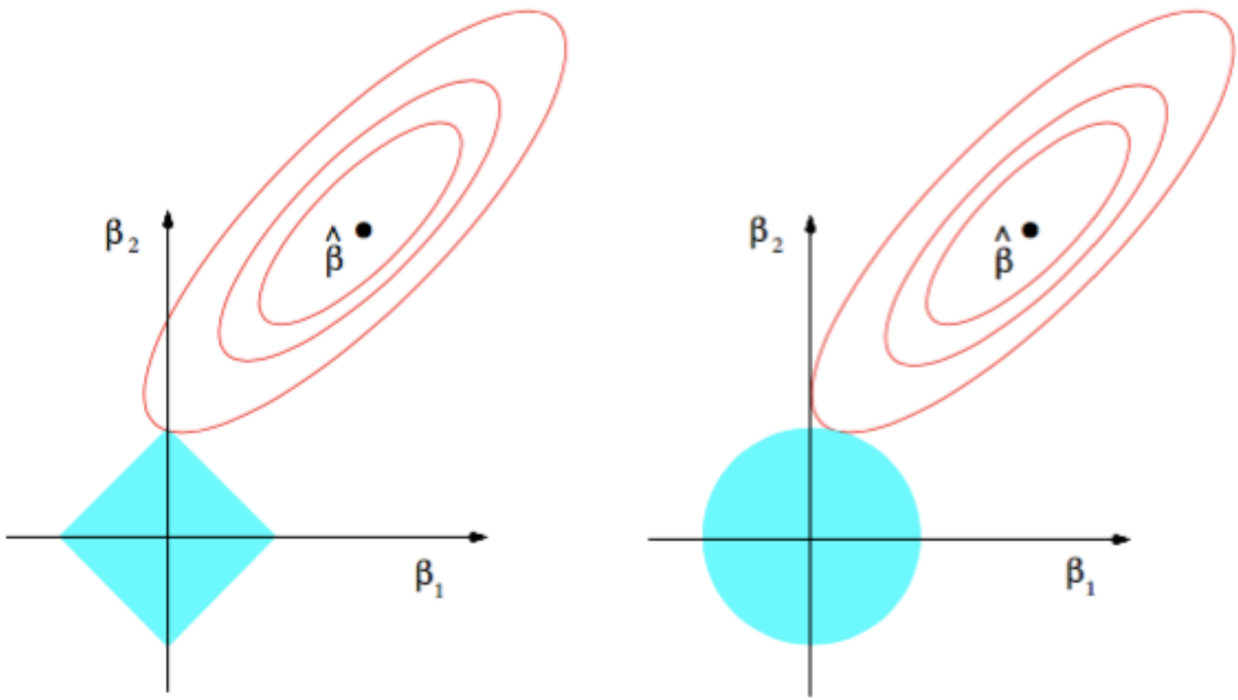
  - Equivalent to minimize $\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2$, subject to $\sum_{j=1}^{p} \beta_j^2 \leq C$

- LASSO Regression for OLS

$$\text{Minimize: } \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

  - Equivalent to minimize $\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2$, subject to $\sum_{j=1}^{p}|\beta_j| \leq C$

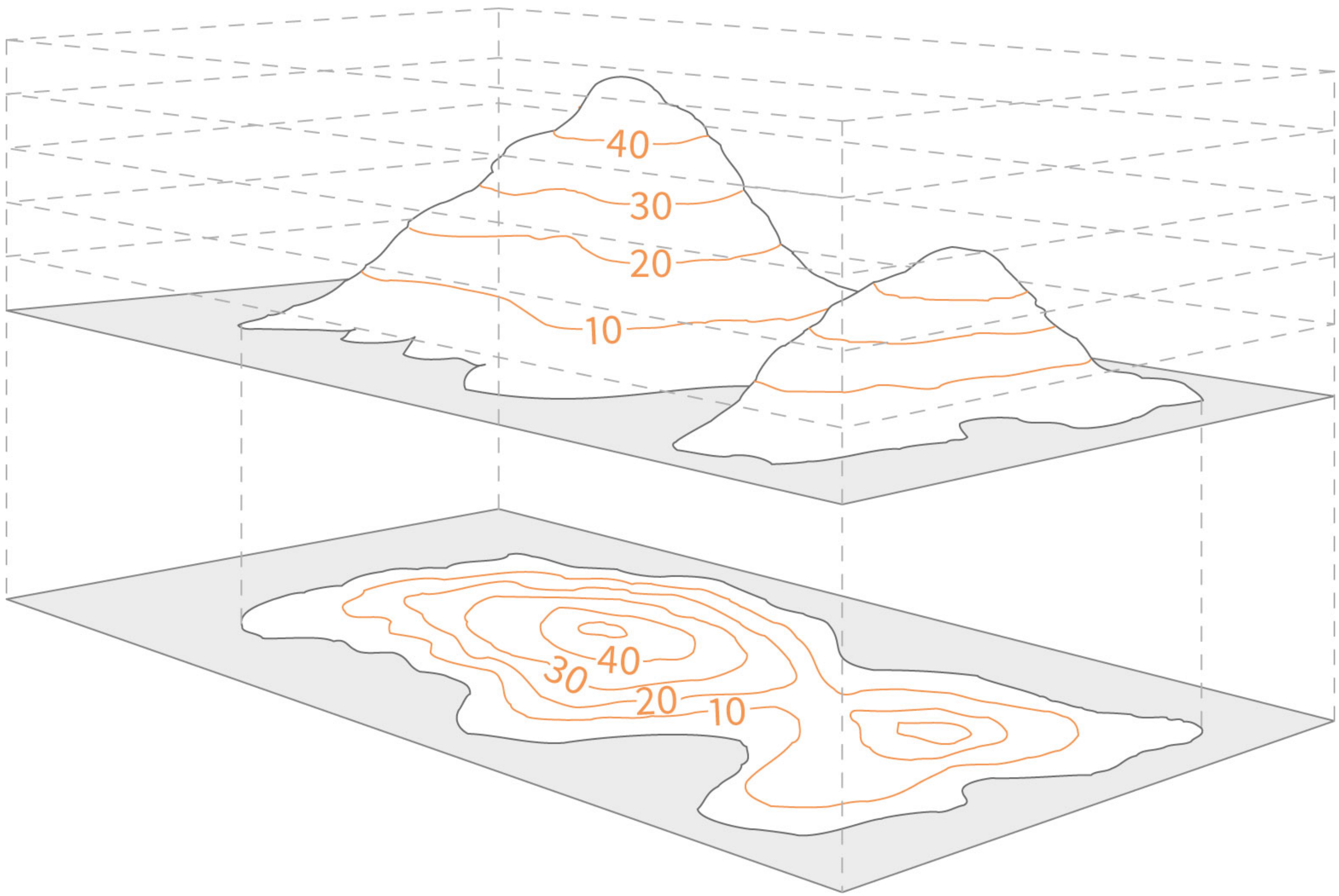
Lasso tends to use less features



Lasso Regression

Ridge Regression

Example: Height as a Function of X and Y



Take Away

- Filter
  - Speed: fast
  - Robust to overfitting
  - Tend to select redundant features
  - Can be used as an initial step to reduce the number of raw features

- Wrapper
  - Can produce the best performance in theory
  - Speed: slow
  - Not as robust to overfitting

- Embedded
  - To some extend, it combines the advantages of Filter and Wrapper
  - Speed: Moderate

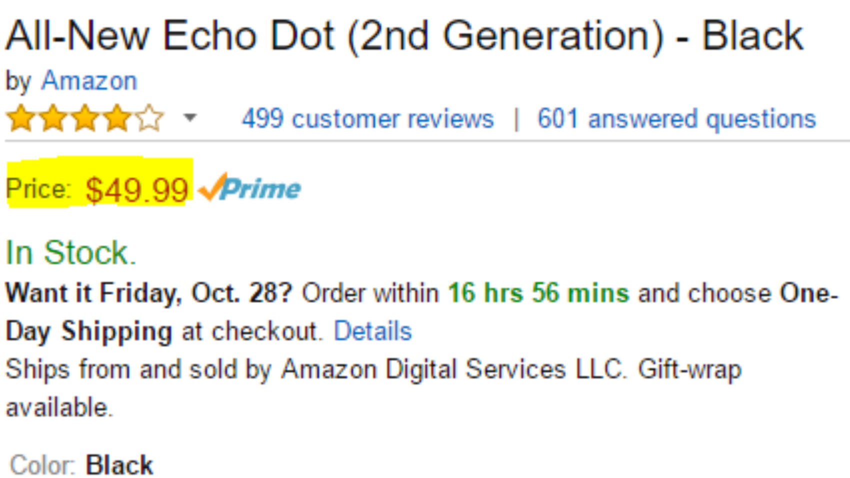
Data Acquisition

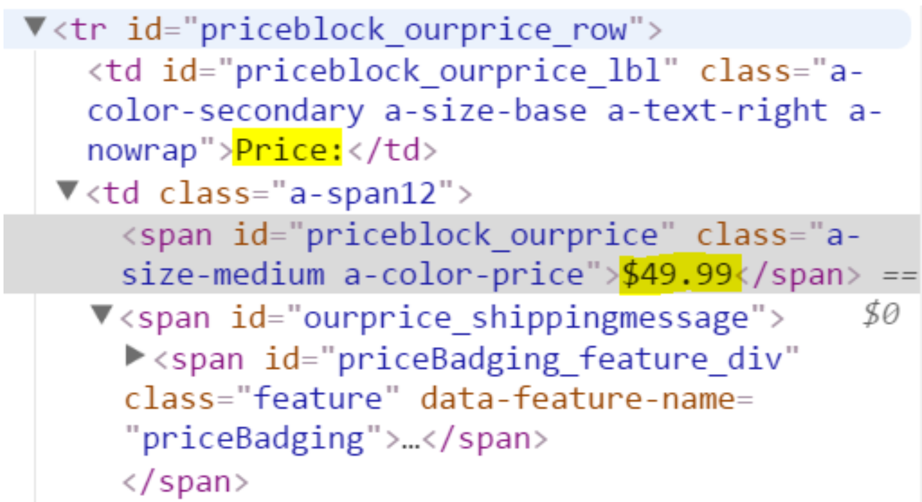
Two Common Data Acquisition Methods

- HTML Scraping

- API Requests

HTML Scraping





```
▼<tr id="priceblock_ourprice_row">
   <td id="priceblock_ourprice_lbl" class="a-color-secondary a-size-base a-text-right a-nowrap">Price:</td>
 ▼<td class="a-span12">
     <span id="priceblock_ourprice" class="a-size-medium a-color-price">$49.99</span> == $0
   ▼<span id="ourprice_shippingmessage">
     ▶<span id="priceBadging_feature_div" class="feature" data-feature-name="priceBadging">…</span>
     </span>
```

Web Page ← Rendered by Browser — HTML Source Code

HTML Scraping Tools

- Static HTML
  - Regular Expression
  - Tools: R, Python, Java
- Dynamic HTML with Javascript
  - Mimic browser to execute Javascript
  - Tool: selenium, scrapy
- Cloud-based scraping: e.g., https://scrapinghub.com/

API Requests

- Application Programming Interface (API) provides programmatic access to read and write data on a platform (e.g., Twitter, Facebook)
  - Primarily for third-party application developers
  - Access requires authentication
  - Different types of APIs to access different types of data
  - Data are returned in structured format (JSON or XML, no need to parse HTML)
  - Often imposes rate limit on requests

API Requests

Example Request

```
curl --request GET --url
https://stream.twitter.com/1.1/statuses/sample.json --header
'authorization: OAuth oauth_consumer_key="CONSUMER_KEY",
oauth_nonce="CONSUMER_SECRET",
oauth_signature="GENERATED_SIGNATURE",
oauth_signature_method="HMAC-SHA1",
oauth_timestamp="GENERATED_TIMESTAMP", oauth_token="ACCESS_TOKEN",
oauth_version="1.0"'
```

Example Response

```
  {
  "created_at": "Tue Feb 27 21:11:40 +0000 2018",
  "id": 968594506663669800,
  "id_str": "968594506663669760",
  "text": "RT @honeydrop_506: 180222 ICN #백현 #BAEKHYUNnn나의 겨울과 너nn#iHeartAwards #BestFanArmy #EXOL @weareoneEXO http
s://t.co/hg7I3xAlBg",
  "source": "<a href='"http://twitter.com"' rel='"nofollow"'>Twitter Web Client</a>",
  "truncated": false,
  "in_reply_to_status_id": null,
  "in_reply_to_status_id_str": null,
  "in_reply_to_user_id": null,
  "in_reply_to_user_id_str": null,
  "in_reply_to_screen_name": null,
  "user": {
    "id": 4448809940,
    "id_str": "4448809940",
    "name": "ayah",
```


Source: https://developer.twitter.com/en/docs/tweets/sample-realtime/api-reference/get-statuses-sample

Twitter Rate Limit Chart

Standard API Rate limits per window

Standard

The standard API rate limits described in this table refer to GET (read) endpoints. Note that endpoints not listed in the chart default to 15 requests per allotted user. All request windows are 15 minutes in length. These rate limits apply to the standard API endpoints only, does not apply to premium APIs.

For POST (create and delete) operations, refer to Twitter's Account Limits support page in order to understand the daily limits that apply on a per-user basis.

Endpoint	Resource family	Requests / window (user auth)	Requests / window (app auth)
GET account/verify_credentials	application	75	0
GET application/rate_limit_status	application	180	180
GET favorites/list	favorites	75	75
GET followers/ids	followers	15	15
GET followers/list	followers	15	15
GET friends/ids	friends	15	15
GET friends/list	friends	15	15
GET friendships/show	friendships	180	15

Which Websites Provide APIs?

- Most social media websites offer APIs
  - Twitter: https://dev.twitter.com/docs
  - Facebook: https://developers.facebook.com/docs/graph-api/reference/
  - Youtube: https://developers.google.com/youtube/v3/docs/
  - Flickr: https://www.flickr.com/services/api/
  - Reddit: https://www.reddit.com/dev/api/

Twitter APIs

- Streaming API (real-time)
  - Standard (sampled, approximately 1%)
  - Firehose (full sample)
- Rest APIs (historical)
  - user_timeline (recent 200 tweets/retweets/replies)
  - retweets (recent 100)
  - followers/friends (5000 per request, time consuming)
  - users/lookup (100 per request)
- Rate Limits
  - Streaming API: not an issue
  - Rest APIs: bottleneck

Twitter API [doc]

Demos Using rtweet Package

- https://rtweet.info/

- collection.R

Readings

- Textbook ABA Chapter 2
- Get familiar with OPIM Virtual Desktop and SAS Enterprise Miner