Exploring your data

Fraida Fund

Contents

Garbage in, garbage out

If you remember nothing else from this semester, remember this!

If you use "garbage" to train a machine learning model, you will only ever get "garbage" out. Also, since you are testing on the same data, you might not even realize it is "garbage" until the model is in production!

Example: author citation data (1)

Data analysis: use PubMed, and identify the year of first publication for the 100,000 most cited authors.

What are our expectations about what this should look like?

Example: author citation data (2)

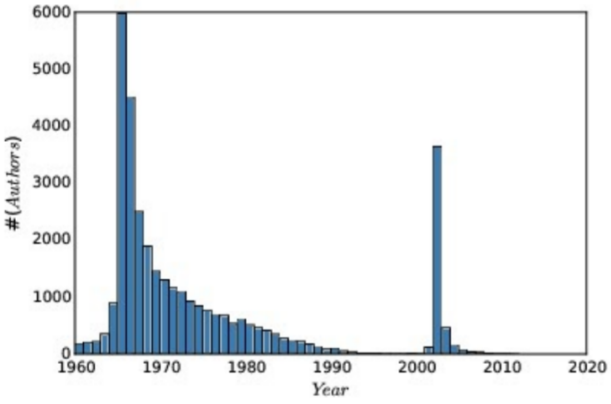


Figure 1: Does this look reasonable?

We can think of many potential explanations for this pattern, even though it is actually a data artifact.

The true explanation: in 2002, PubMed started using full first names in authors instead of just initials. The same author is represented in the dataset as a "new" author with a first date of publication in 2002.

Example: author citation data (3)

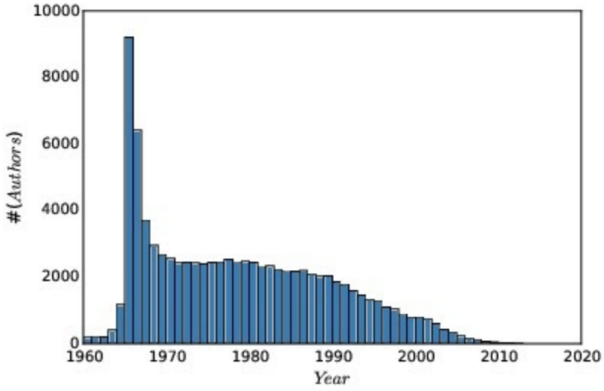


Figure 2: The real distribution, after name unification. Example via Steven Skiena @ SBU.

Example: anomalous voting data (1)

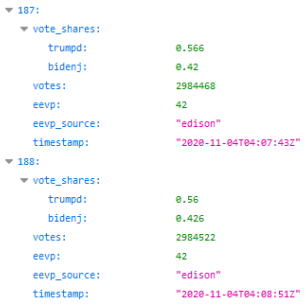


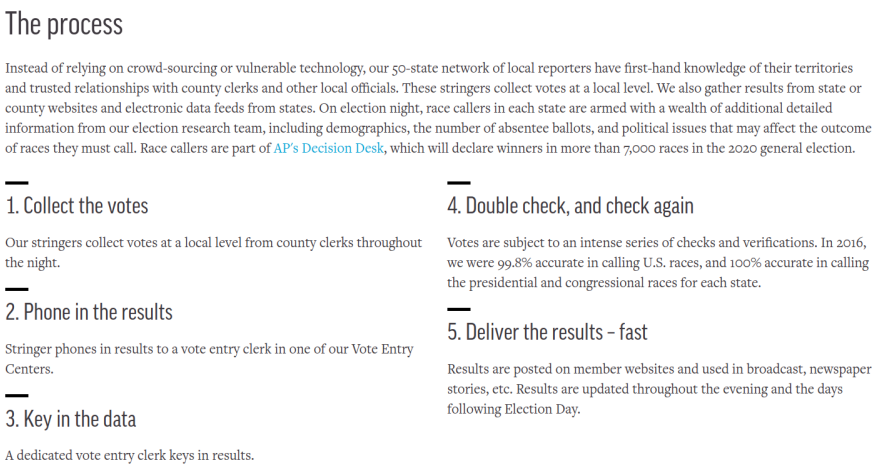
Figure 3: Data like this was widely (wrongly) used as evidence of anomaly in the 2020 U.S. Presidential election.

What are our assumptions about election night data, and how are they violated here?

We expect that per-candidate vote totals (computed by multiplying total votes and vote share) should increase as more votes are counted, but never decrease.

What are possible explanations?

Example: anomalous voting data (2)



The process

Instead of relying on crowd-sourcing or vulnerable technology, our 50-state network of local reporters have first-hand knowledge of their territories and trusted relationships with county clerks and other local officials. These stringers collect votes at a local level. We also gather results from state or county websites and electronic data feeds from states. On election night, race callers in each state are armed with a wealth of additional detailed information from our election research team, including demographics, the number of absentee ballots, and political issues that may affect the outcome of races they must call. Race callers are part of AP's Decision Desk, which will declare winners in more than 7,000 races in the 2020 general election.

1. Collect the votes

Our stringers collect votes at a local level from county clerks throughout the night.

2. Phone in the results

Stringer phones in results to a vote entry clerk in one of our Vote Entry Centers.

3. Key in the data

A dedicated vote entry clerk keys in results.

4. Double check, and check again

Votes are subject to an intense series of checks and verifications. In 2016, we were 99.8% accurate in calling U.S. races, and 100% accurate in calling the presidential and congressional races for each state.

5. Deliver the results – fast

Results are posted on member websites and used in broadcast, newspaper stories, etc. Results are updated throughout the evening and the days following Election Day.

Figure 4: Process by which data is collected by Edison and AP.

This anomaly makes a lot of sense as a correction of a data entry or duplicate entry error.

How Edison/AP collects the data for their Election Night feed:

- There are "stringers" (temporary reporters) at various elections offices who call results into their phone center
- They have people who look at official government websites for new results that they manually enter into the system
- They have people who monitor results sent by fax from counties and cities

all working as fast as they can! Data entry and duplicate entry errors are not only likely, they are almost guaranteed. When they are corrected, vote totals may decrease.

Source: AP, Edison

What kinds of data problems?

What kind of problems might you encounter? (1)

- Rows where some fields are missing data
- Missing data encoded as zero
- Different units, time zones, etc. in different rows
- Same value represented several different ways (e.g. names, dates)
- Unreasonable values

How should you handle little bits of missing data? It always depends on the data and the circumstances. Some possibilities include:

- omit the row
- fill with mean or mode
- fill back/forward (ordered rows)
- train a model on the rest of the data to "predict" the missing value

How should you handle unreasonable values or outliers?

- e.g. suppose in a dataset of voter information, some have impossible year of birth - would make the voter a child, or some indicate the voter is 120 years old. (Voters with no known DOB, who registered before DOB was required, are often encoded with a January 1900 DOB.)
- **not** a good idea to just remove outliers unless you are sure they are a data entry error or otherwise not a "true" value.
- Even if an outlier is due to some sort of error, if you remove them, you may skew the dataset (as in the 1/1/1900 voters example).

What kind of problems might you encounter? (2)

- Rows that are completely missing
- Data is not sampled evenly
- Data or labels reflect human bias
- Data is not representative of your target situation
- Data or situation changes over time

Examples:

- Twitter API terms of use don't allow researchers to share tweets directly, only message IDs (except for limited distribution, e.g. by email). To reproduce the dataset, you use the Twitter API to download messages using their IDs. But, tweets that have been removed are not available - the distribution of removed tweets is not flat! (For example: you might end up with a dataset that has offensive posts but few "obvious" offensive posts.)
- Many social media datasets used for "offensive post" classification have biased labels (especially if they were produced without adequate training procedures in place). For example, they may label posts containing African-American dialects of English as "offensive" much more often. Source, User-friendly article
- A dataset of Tweets following Hurricane Sandy makes it looks like Manhattan was the hub of the disaster, because of power blackouts and limited cell service in the most affected areas. Source
- The City of Boston released a smartphone app that uses accelerometer and GPS data to detect potholes and report them automatically. But, low income and older residents are less likely to have smartphones, so this dataset presents a skewed view of where potholes are. Source

Change over time: Imagine you train a machine learning model to classify loan applications. However, if the economy changes, applicants that were previously considered credit-worthy might not be anymore despite having the same income, as the lender becomes more risk-averse. Similarly, if wages increase across the board, the income standard for a loan would increase.

What kind of problems might you encounter? (3)

- Data ethics fails
- Data leakage

Some data ethics fails:

- On the anonymity of the Facebook dataset
- 70,000 OkCupid Users Just Had Their Data Published; OkCupid Study Reveals the Perils of Big-Data Science; Ethics, scientific consent and OKCupid
- IBM didn't inform people when it used their Flickr photos for facial recognition training

Data leakage

In machine learning, we train models on a training set of data, then evaluate their performance on a set of data that was not used in training.

Sometimes, information from the training set can "leak" into the evaluation - this is called data leakage.

Or, information from the target variable (which should not be available during inference) leaks into the feature data.

Some types of data leakage

- Learning from adjacent temporal data
- Learning from duplicate data
- Learning from features that are not available at prediction time (e.g. data from the future)
- Learning from a feature that is a proxy for target variable, but that doesn't generalize

COVID-19 chest radiography (1)

- **Problem**: diagnose COVID-19 from chest radiography images
- **Input**: image of chest X-ray (or other radiography)
- **Target variable**: COVID or no COVID

COVID-19 chest radiography (2)

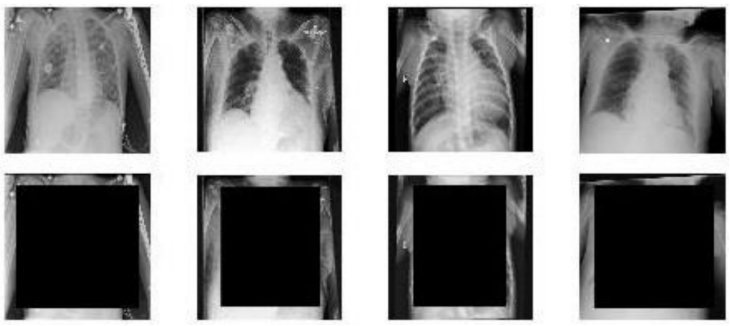


Fig. 5. Original and transformed samples from the 4 datasets, 300 sized black square (Left to right: COV, NIH, CHE, KAG)

Figure 5: Neural networks can classify the source dataset of these chest X-ray images, even *without lungs*! Source

Between January and October 2020, more than 2000 papers were published that claimed to use machine learning to diagnose COVID-19 patients based on chest X-rays or other radiography. But a later review found that "none of the models identified are of potential clinical use due to methodological flaws and/or underlying biases".

To train these models, people used an emerging COVID-19 chest X-ray dataset, along with one or more existing chest X-ray dataset, for example a pre-existing dataset used to try and classify viral vs. bacterial pneumonia.

The problem is that the chest X-rays for each dataset were so "distinctive" to that dataset, that a neural network could be trained with high accuracy to classify an image into its source dataset, even without the lungs showing!

COVID-19 chest radiography (2)

Findings:

- some non-COVID datasets were pediatric images, COVID images were adult
- there were dataset-level differences in patient positioning
- many COVID images came from screenshots of published papers, which often had text, arrows, or other annotations over the images. (Some non-COVID images did, too.)

COVID-19 chest radiography (3)

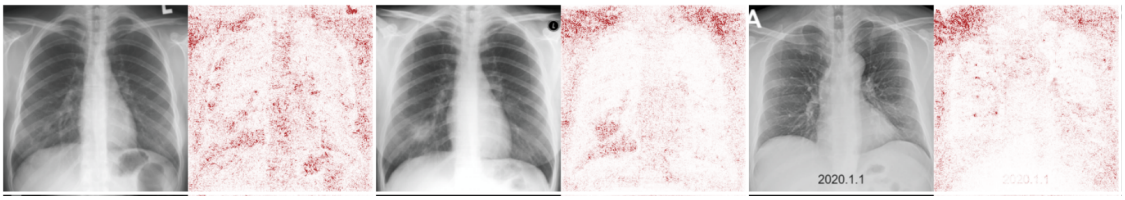


Figure 6: Saliency map showing the "important" pixels for classification. Source

These findings are based on techniques like

- saliency maps, where the model is made to highlight the part of the image (the pixels) that it considered most relevant to its decision.
- using generative models and asking it to take a COVID-negative X-ray and make it positive (or v.v.)

Many of the findings are not easy to understand without domain knowledge (e.g. knowing what part of the X-ray *should* be important and what part should not be.) For example: should the diaphragm area be helpful?

Signs of potential data leakage (after training)

- Performance is "too good to be true"
- Unexpected behavior of model (e.g. learns from a feature that shouldn't help)

Detecting data leakage

- Exploratory data analysis
- Study the data before, during, and after you use it!
- Explainable ML methods
- Early testing in production