Predicting Data Breach Severity

Daniel Fiume
Brown University
10/21/2024

Github: https://github.com/dfiume1/data-breach-ml.git

Introduction

Problem: Company and Government data breaches are very common. Sometimes a company will state they've been breached, but won't divulge much other information.

Goal: Predict the number of users affected by a data breach based on publicly reported information about the breach. This is a **regression** problem.

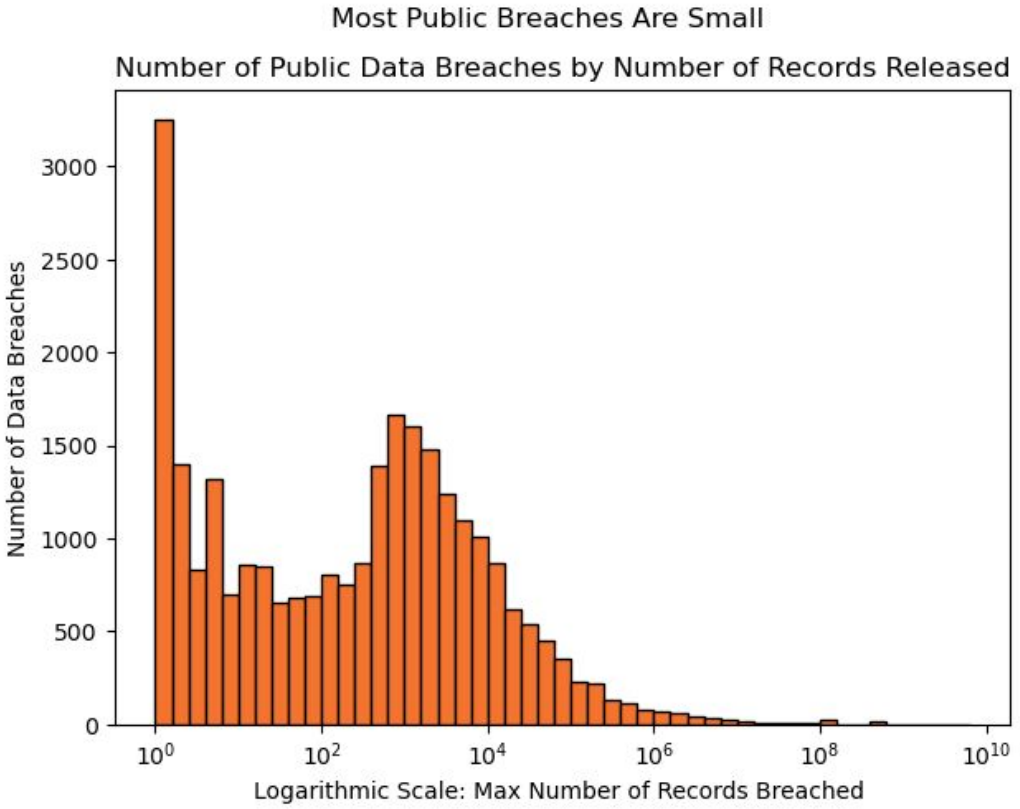
Data Source and Collection: This data was made available to me upon request from the **Privacy Rights Clearinghouse Team**. This data was scraped from public documents, news articles, and websites that announced data breaches. **This is not a public dataset.**

EDA: A First Look

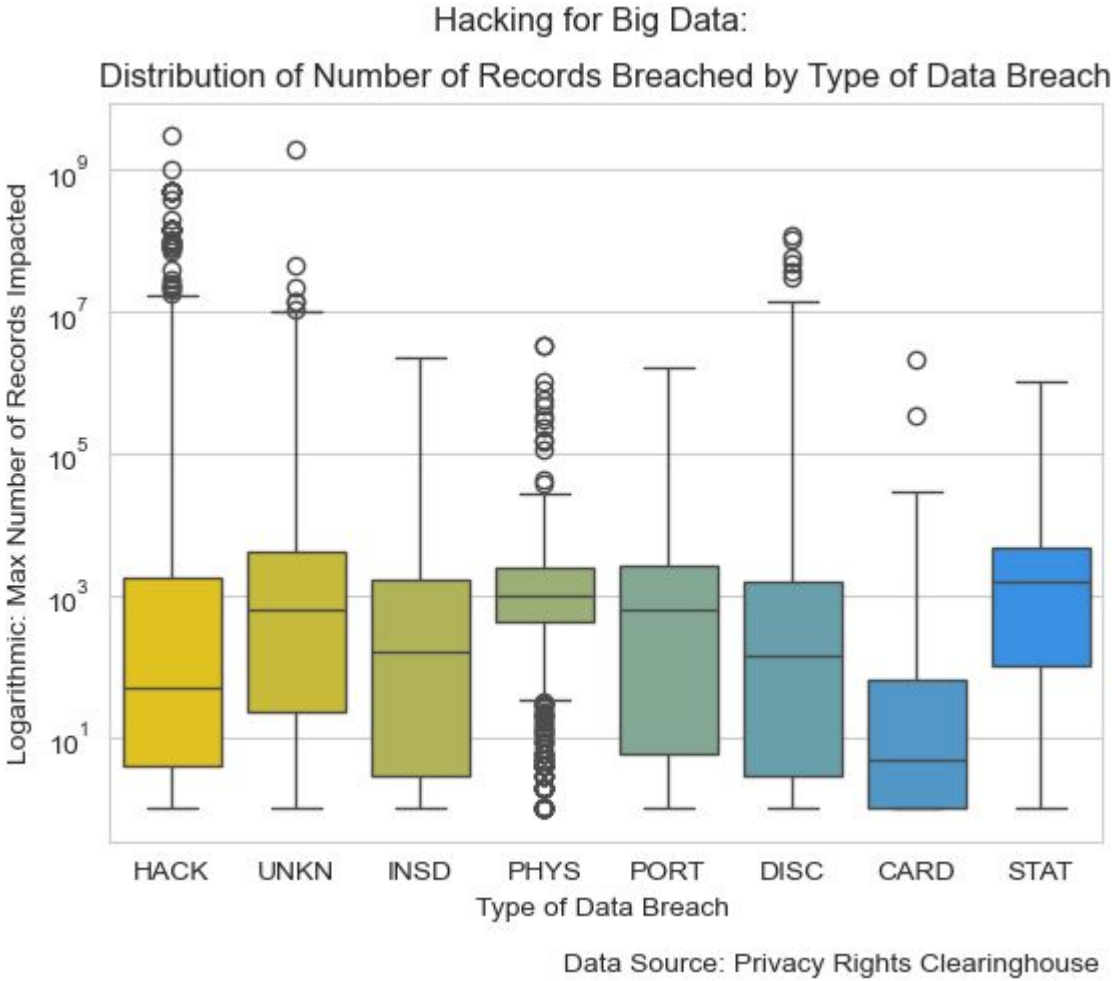
- The raw dataset has **35,167 rows and 28 columns.**
  - A lot of these columns are large chunk of text we are not interested in.
- Columns of Note:
  - Name
  - Source
  - Breach Location (Country, State, City)
  - Reported Date, Date of Breach, and End of Breach
  - Organization Type
  - Breach Type
  - Information Type
  - **Max Records Impacted**
  - State Records Impacted From Source

EDA: Target Variable: Number of Records Released

count	27041.0
mean	638413.8
std	25361928.3
min	0.0
25%	6.0
50%	370.0
75%	2945.0
max	3000000000.0
Name: Max Records Impacted,	



Data Source: Privacy Rights Clearinghouse



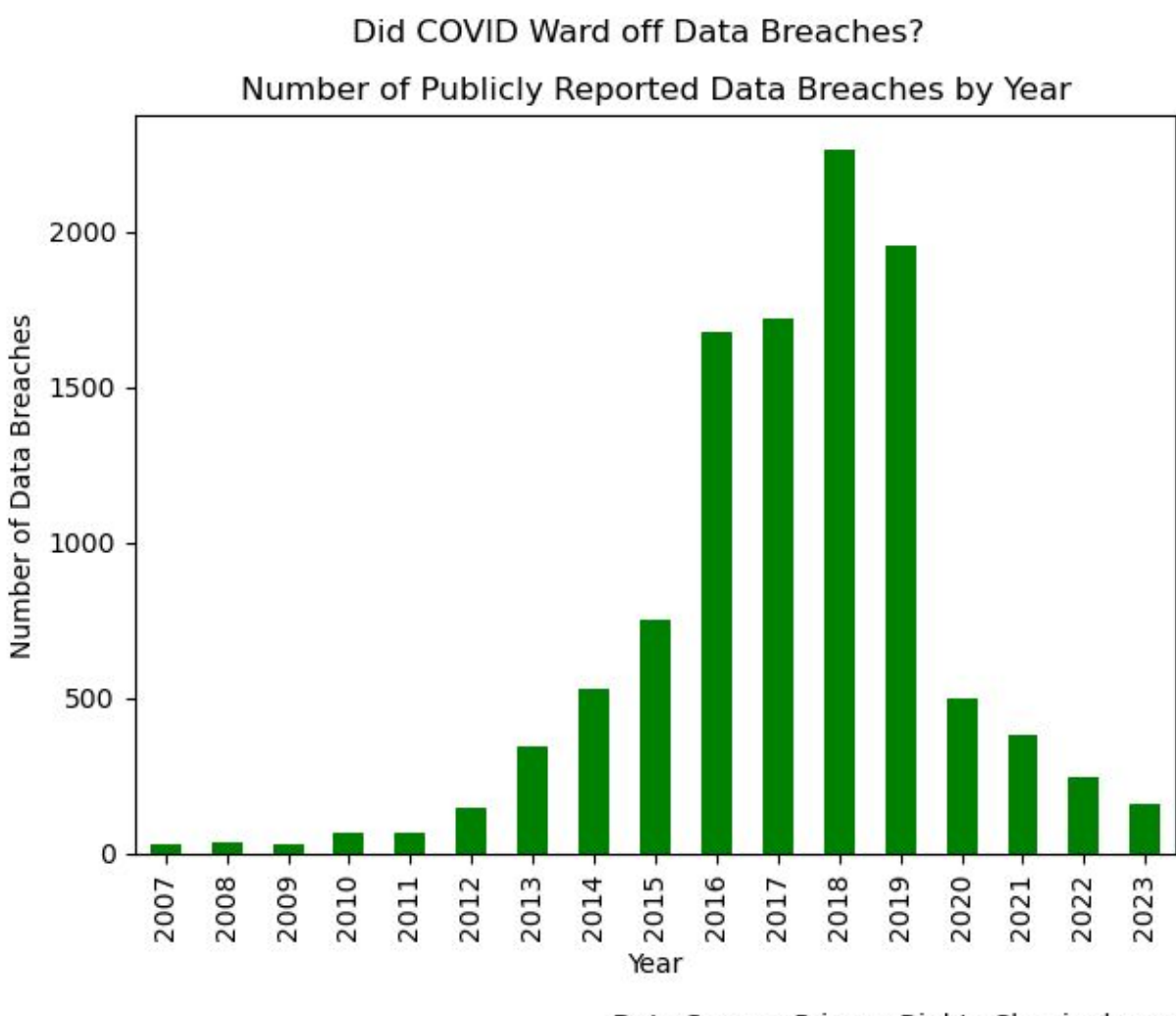
EDA: Type of Breach

Breach Type	
UNKN	13520
HACK	9085
DISC	1812
PORT	959
INSD	914
PHYS	493
STAT	179
CARD	79

EDA: Type of Breach

- **CARD**: Breaches involving credit or debit card fraud, not related to hacking activities.
- **HACK**: Breaches due to hacking or malware attacks by external parties.
- **INSD**: Insider breaches caused by employees, contractors, or customers.
- **PHYS**: Physical breaches involving paper documents that are lost, discarded, or stolen.
- **PORT**: Breaches involving the loss or theft of portable devices like laptops, smartphones, memory sticks, etc.
- **STAT**: Loss or theft of stationary computers or servers not designed for mobility.
- **DISC**: Unintended disclosures not involving hacking, intentional breaches, or physical loss.
- **UNKN (Unknown)**: Used when the breach type cannot be definitively determined.

EDA: Date of Breach:

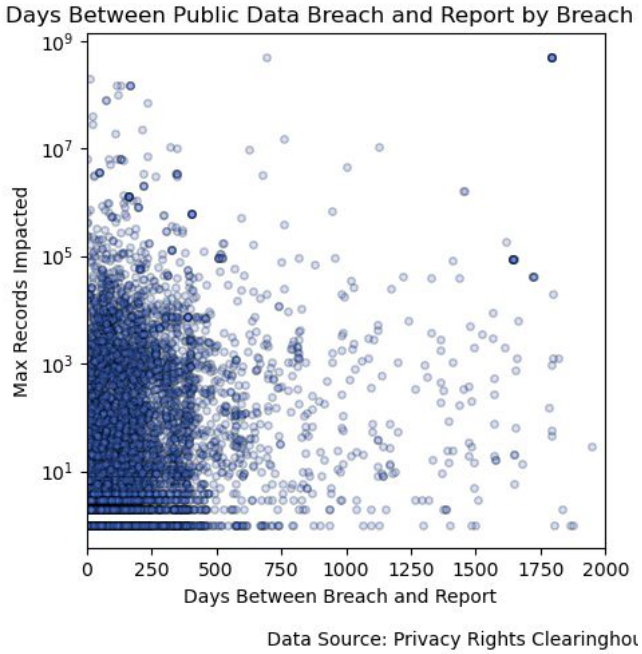


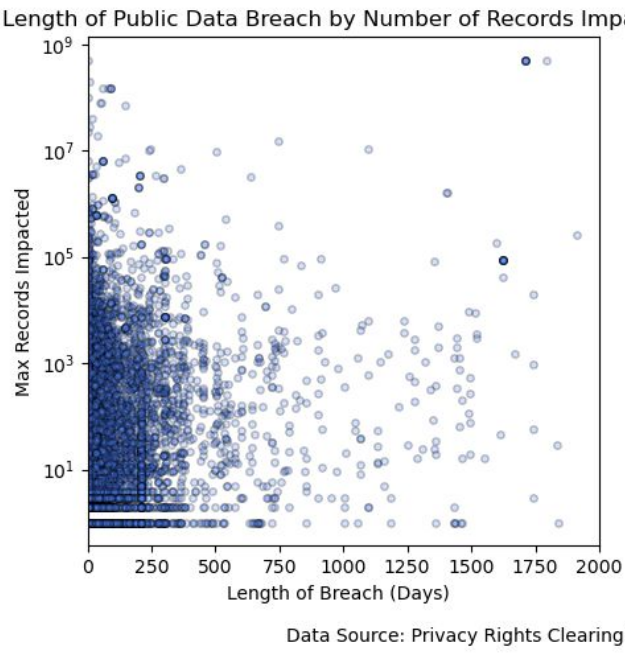
Feature Engineering: New Date Features

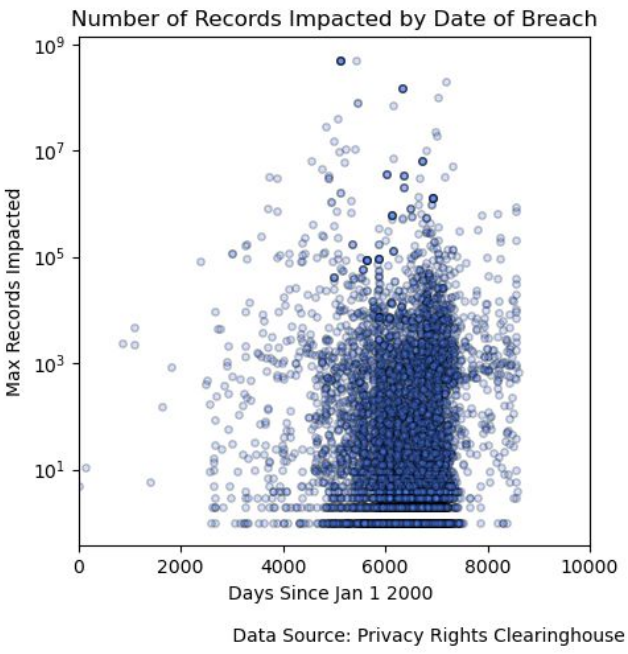
- It would be repetitive to include breach start, end, and reported date.
- But it may be interesting to know the length of breach or time to report.

	Reported Date	Date of Breach	Date of Breach End	Length of Breach (Days)	Days Until Reported	Day # of Breach
0	2020-07-17	2020-02-10	2020-02-11	1	158	8446
3	2018-02-13	2017-11-21	2017-12-08	17	84	7561
5	2017-08-04	2017-02-16	2017-06-25	129	169	7368
8	2018-02-05	2018-02-05	2018-02-05	0	0	7553
9	2022-02-01	2021-06-01	2021-06-18	17	245	9010

Feature Engineering: New Date Features







EDA: Correlations

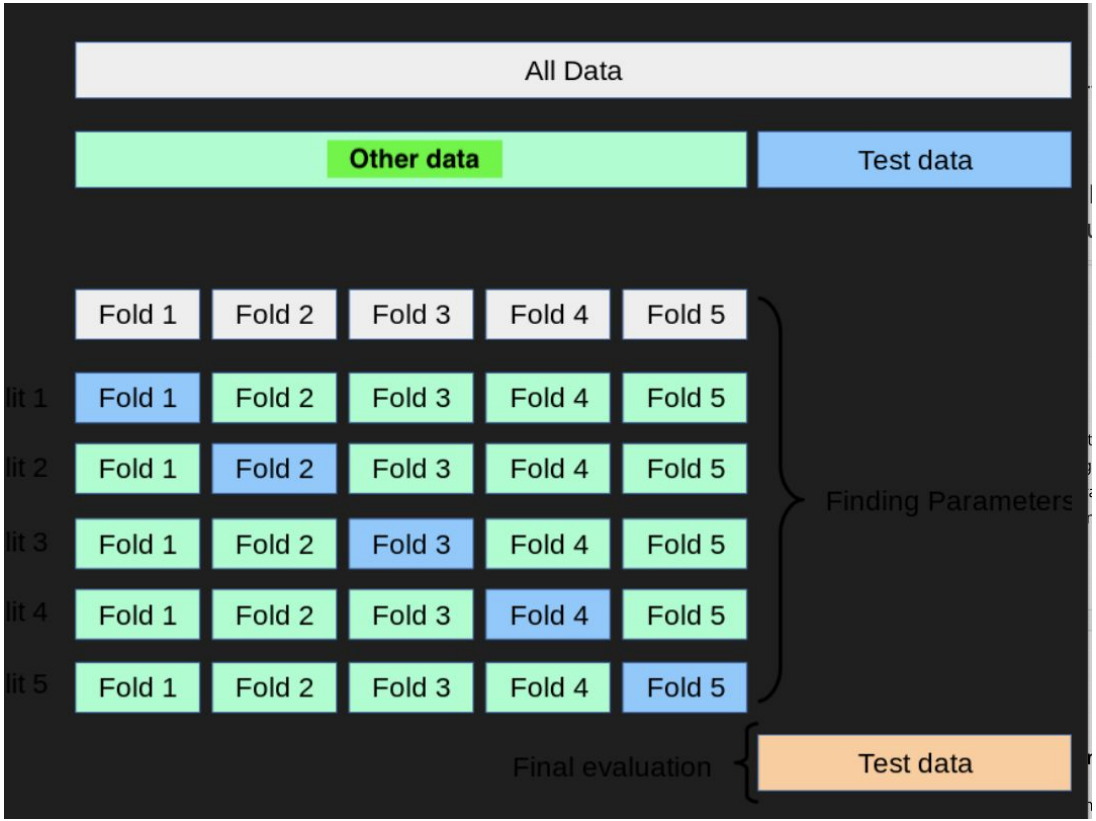
TODO

Splitting:

Train/Validation: 80%

Num folds: 5

Test: 20%



Preprocessing: Missing Values

- First, I removed rows that had missing values for my target variable
  - 35,167 rows -> 27,041 rows

- A lot of my categorical features contain anywhere from 3-60% missing values. Those rows are not removed and instead “UNKN” is treated as a new category.

- My Date Features contain a lot of missing values
  - 27,041 rows -> 10,704 rows if they were to be removed

Preprocessors

- I used Standard Scaler's for my continuous features, as they had tailed distributions.
  - Max Records Impacted
  - Breach Date
  - Length of Breach
  - Days Between Breach and Report
- I used OneHotEncoders for the rest of the features, which are categorical.
  - Breach Type
  - Organization Type
  - Breach Source
  - Breach Location
- # Of Features: 28 —--------> 8
- # Of Rows: 27,041 —----------> 7023
  - This will be fixed when XGBoost is used

Questions?