Report_Julian Bürkle_Assignment 4&5:

My Github: https://github.com/jbuerkle93/PPOD and the file is: PPOD Assigment 4&5.ipynb, I used JupyterLab to create and run it.

3.1 Textual Data Anonymisation

Dataset:

The dataset I chose for task 3.1 is called "Tweets Emotions" dataset:
https://www.kaggle.com/datasets/pashupatigupta/emotion-detection-from-text?resource=download

This Data set contains tweets and their classification based on an emotion detection/sentiment analysis algorithm. Also it stores the ID of each tweet.

	tweet_id	sentiment	content
0	1956967341	empty	@tiffanylue i know i was listenin to bad habi...
1	1956967666	sadness	Layin n bed with a headache ughhhh...waitin o...
2	1956967696	sadness	Funeral ceremony...gloomy friday...
3	1956967789	enthusiasm	wants to hang out with friends SOON!
4	1956968416	neutral	@dannycastillo We want to trade with someone w...

Figure 1 - .head() of dataset

3.1.1 Research

In order to understand what needs to be analyzed first let's check what columns contain risky data in relation to privacy.

Primarily, let's check if the tweet_id. This id could be linked to a profile, which could contain PII. Let's check if it is really unique. After applying python code, I can say yes they are unique. Thus they might pose a data privacy risk. The id identfies the tweet. A quick google search revealed that one can find the original tweet by typing this url with the correct tweet_id but incorrect username. twitter.com/anyuser/status/541278904204668929. Twitter/X will automatically fetch the correct username and display the original tweet.

However, I did not work with the tweet_ids stored in this dataset. Perhaps they are old (dataset has been change last 3 years ago) or the author had applied some changes (I cannot find a hint whether the authors has done that, though). However, as it's generally possible to fetch the original tweet including the username. I assume there are ways to also fetch the original tweet with the id stored in this dataset. Hence it should be anonymised in my opinion.

Secondly, let's check the same for the sentiment column: The sentiment column contains many non-unique values. Thus, in my opinion does not pose a privacy risk, thus must not be anonymized.

Third, let's analyse the column which contains the raw tweet. Within the text data of the column could be PII as well as Quasi-Identifiers. Looking at the data we can quickly see that user names appear in many of the tweets. They always start with an @. After checking the twitter/X website I can confirm that: "Your username — also known as your handle — begins with the “@” symbol, is **unique** to your account, and appears in your profile URL" (https://help.twitter.com/en/managing-your-account/change-x-handle). Hence, this usernames could pose a privacy risk and therefore should be anonymized.

Additionally, there are at least some first names, so it could be that there sometimes is the full name which would be a PII. However, only the first name combined with other quasi-identifiers could be used to identifiy an individual. Other quasi-identifiers I see inlcude while manually looking for the data are: country names. Saying that, I guess there are also city names in the tweets, so these should definitely also be anonymized. Let's see in the next exercise what other PII or quasi-identifiers we can find using NLP.

3.1.2 Using spaCy

I used spaCy to scan the text for PII. I used the given categories of spaCy to search for the content (tweets) column of the dataset: 'PERSON', 'DATE', 'EVENT', 'GPE', 'ORG'. spaCy discovered the following occurences for each category:

PERSON: 8403

DATE: 7360

EVENT: 79

GPE: 3264

ORG: 7517

Additionally I scanned the content column for email addresses and found four. Two of these emails contain a first and last name and thus can be classified as PII. Hence, they should be anonymized.

While analyzing, I also noticed that some links like http://tr.im/mLou are included in the content column. Which could also link to other websites which contain PII.

3.1.3 Algorithm

To modify the detected PII I defined the following algorithm:

The Id column should be randomly modified using randomisation -> I do this because as described in 3.1.1 it might be used to retrieve the actual tweet and the user who tweeted it. To make it appear like the current Id I keep its length and just randomize the numbers of the Id.

I replace all usernames with the generic handle @username -> In this way no one can search for the mentioned users handle on twitter and discover the original tweet or more about the person who posted it or who was mentioned. This data modification also pseudonymizes the four occurrences of email

addresses found. Additionally, I noticed that usernames where very often classified as ORG. That's why I decided to start with randomizing all the handles first place in the following algorithm. Just to be safe.

All links (everything that starts with "http" will be replaced with http://link_hidden.

All person names will be randomized using the same length but replacing the name. As spaCy falsely detected many twitter handles as person names I added a check to only randomize the string when the name is not a twitter handle

All dates will be pseudonymized with hidden_date

All organisations will be pseudonymized with hidden_org

All geopolitical events will be pseudonymzied with hidden_gpe

3.1.4 Analysis

After transformation all of the quasi-identifiers detected with the help of spaCy are pseudonymized or randomized. Leaving information-loss concerns aside this means that the likelihood for especially linkage attacks is dramatically reduced. From the detected PII and quasi-identifiers there should be no threat or a very minor threat for attacks that breach privacy. In addition, to spaCy's discoveries I checked for other potentially PII like email and pseudonymzed them as well.

Saying that I must mention that in the vast amount of text there could be other potential PII or quasi identifiers, which neither I or spaCy have scanned the data for. For instance, numerical data such as a tax number or something, could remain in the text data, posing potential threats.

3.2 De-anonymising a dataset

3.2.1 Standard search mechanisms

The dataset I used was the fatal police shooting dataset anonymized using bayesian interference. It contains the following columns:

	date	manner_of_death	armed	age	gender	race	city	state	signs_of_mental_illness	threat_level	flee	body_camera	longitude	latitude	is_geocoding_exact
0	2022-11-06	shot	gun	20.0	M	H	Baltimore	AZ	True	attack	Not fleeing	False	-83.663	37.472	True
1	2016-01-20	shot	gun	32.0	M	W	Kerrville	ME	False	attack	Not fleeing	False	-86.043	38.879	True
2	2020-08-29	shot and Tasered	unarmed	56.0	M	B	Dearborn Heights	NC	False	other	Not fleeing	False	-96.642	32.958	True
3	2020-07-12	shot	toy weapon	52.0	F	B	Butler Township	FL	True	attack	Not fleeing	False	-82.765	28.150	True
4	2019-04-29	shot	nail gun	19.0	M	B	Edmond	MS	False	attack	Not fleeing	False	-93.746	32.492	True

Figure 2 - .head() of dataset

The dataset contains no PII. Quasi-Identifiers could be a combination of date, age, gender, city, state, latitude, longitude as well as potentially manner of death, armed, signs of mental illness, fleeing, body camera. Hereby I classify date, armed, latitude & longitude, race, gender, city & state as the most promising quasi-identifiers. A combination of them could lead to find a link to the real incident.

As I don't know what has been anonymized I tried to search for columns which had not been analyzed. I used a linkage attack approach to find out which data has not been modified. I tried to several google searches using atypical, rare occurrences in the data like that the shot person was armed with a toy gun to identify this (Simulated linkage attack). Combining this with quasi-identifiers like city, state and the exact coordinates (longitude and latitude) of the shootings, I hoped to identify a real shooting and conclude what in the dataset has not been modified yet.

Unfortunately, I did not find any real police shooting matching the data - congrats to my colleague well done. However, what I've found is that the city and state do not correspond with the coordinates. This could mean that one of the two data groups has not been anonymized or that both have been. Considering what I would have done to anonymize the dataset, I assume he or she has anonymized all quasi-identifiers.

When we look closer at the data we can say that for instance the city column most definitely has been anonymized. As you can see in the screenshot Albuquerque (Located in New Mexico) appears with many different states, suggesting that this column or both have been anonymized for sure

```
Albertville, CO - Count: 1
Albuquerque, AL - Count: 1
Albuquerque, AZ - Count: 5
Albuquerque, CA - Count: 6
Albuquerque, CO - Count: 3
Albuquerque, FL - Count: 2
Albuquerque, ID - Count: 1
Albuquerque, IL - Count: 1
Albuquerque, KY - Count: 2
Albuquerque, MD - Count: 1
Albuquerque, ME - Count: 1
Albuquerque, MI - Count: 1
Albuquerque, MO - Count: 2
Albuquerque, MT - Count: 1
Albuquerque, NC - Count: 2
Albuquerque, NY - Count: 2
Albuquerque, OH - Count: 2
Albuquerque, OK - Count: 2
```

Figure 3 - Occurrences of Albuquerque in city column with corresponding date in state column

3.2.2 Design a de-anonymisation algorithm

Dataset of colleague a) & b)

a) & b)

In general, I assume that a de-anonymisation algorithm has to rely on human input. At least, for understanding what the PII and quasi-identifiers are as these require the understanding of context. So the following algorithm is based on getting feedback by humans throughout the process.

1. Feed the algorithm what columns contain PII and what contain quasi-identifier.
2. Prepare the data for better analysis.
    1. E.g. transform categorical columns into numerical ones
    2. Find and remove outliers
3. Understand the algorithm used to anonymize (In this case I understand and assume that we know the method used)
    1. Analyse distribution, frequencies etc
    2. Make test attacks with the new learnings
4. Evaluate the de-anonymisation potential and strategy
    1. Includes testing linkage attacks & recognizing anonymisation errors
5. De-anonymize
6. Test with linkage attacks again

Step 1 & 2 see the python code in git.

Step 3:

As my linkage attempts in the previous exercise using standard search mechanisms were not fruitful I aimed at understanding better how bayesian interference works to anonymize (In the previous exercises I had not used it, yet understood it) in order to create a de-anonymisation algorithm.

I understood that Bayesian interference works the following way:

Roughly speaking, it involves two steps:

1. Using the real dataset, to learn (the parameters for) a model of the data.

2. Using the learned model, draw synthetic data points from its probability distribution and release these.

To implement this you make some assumptions about the dataset to have the Bayesian Network structure. For instance, assume there is connected dependencies within the dataset e.g between 'city' - 'state,' i.e state can be inferred from the city of an individual.

This is then followed by the forward_sample method to create synthetic samples that mirrors the statistical characteristics of the original dataset. This synthetic data is then used to replace the sensitive columns in the original dataset.

OK, so it seems that original data has been replaced by synthetic data, which contains the same probability distribution. That makes it very hard to de-anonymize as probably all of the original data has been replaced by synthetic data. So now wonder that my linkage attack attempts revealed nothing.

Step 4:

As the data is most likely completely synthetic I tried to search for columns left out or some other errors.

Analyzing the statistics of the values showed that some values are very frequent. F.ex. the date 2018-01-06 appears 18 times. Based on this I tried to use google for another linkage attack, but did not find police shootings on that day. However, I found the original dataset, but will not look at it before task 3.2.2.c.

Looking at the rare occasions of how the other person was armed I searched for the one victim which was armed with a stone. I found that this could be 35yo Antonio Zambrano-Montes, who was shot in 2015. https://www.nbcnews.com/news/crime-courts/rock-throwing-man-killed-police-pasco-had-no-other-weapons-n306181.

Looking at the complete data I can deduct that also the columns longitude and latitude have been anonymized.

Another interesting weapon of a later victim, was the one time occurrence of a railroad spike. Searching on google I found: Tyrone Bass, 21, also shot in 2015. https://www.nola.com/news/crime_police/man-stabbed-st-bernard-deputy-with-railroad-spike-before-being-shot-state-police-say/article_546d5aa0-661f-5049-ad77-0e9c127b0d8f.html

Yet another rare occasion was this 19yo woman who carried a cordless drill which seemed like it was an Uzi weapon. However, this was in 2014.

https://www.cbsnews.com/sanfrancisco/news/woman-carrying-cordless-drill-believed-to-be-gun-shot-and-killed-by-san-jose-police/

However, I discovered the original dataset on Kaggle https://www.kaggle.com/datasets/kwullum/fatal-police-shootings-in-the-us?resource=download (But did not look at it in detail yet). And there the overview of the dataset says it includes police shootings from 01.01.2015 and that it was updated 6 years ago. So I assume this either random or it was still included in the dataset.

When I understand the used anonymisation method correctly I assume that stone and cordless drill have been in the original data but either 1.) all other columns have been anonymized using bayesian interference or 2.) all columns including "armed" have been anonymized that way. I imagine that my colleague found out the original distribution and shuffled the values around, without actually creating new synthetic data. I assume my colleague wanted to keep the usefulness of the data. So instead of replacing all unique values of the "armed" column I think he shuffled them around in the row, so that other researchers can still makes sense of the data. E.g. instead of replacing cordless drill or stone, with synthetic values like kitchen knife or roof tile, my colleague must have kept the original values and swapped them around.

So it means that with the other data in the row of armed with stone, I can not deduct anything useful in terms of privacy but with a linkage attack I can find out the name and could for example contact the victims family and write an awesome story as a journalist about this unusual weapon used by the victim, which lead to his death through a police bullet.

With this I think I have discovered an error made in the anonymisation of my colleague's dataset which I had not found in 3.2.1.

Looking at the outliers in age I also found this case where a 74yo got killed by police: https://www.wric.com/news/local-news/the-tri-cities/officials-investigating-shooting-involving-police-officer-in-hopewell/

However, this was in 2023.. So this was probably random. I did not find any other case using this linkage strategy approach. So I assume the age column has been perturbated (Well done colleague).

I did not find any other weakness using this strategy.

Step 5: Evaluate the de-anonymisation potential

Based on what I've found I think the bayesian interference method used is very solid and protects the overall privacy very well. As the data was either replaced with new synthetic data or shuffled around in the rows, while keeping the original distribution, I could not come up with a way to reverse this. How should I know which data originally truly belonged to which row? Or what the original non-synthetic values were? Only with more knowledge about the strategy my colleague used or perhaps using advanced machine learning strategies to identify the anonymisation patterns, I think it could be possible to de-anonymize it.

Yet, in 3.2.2 c) I will propose a method to improve the only weak spot I've found: The outliers, especially in the "armed" column.

My dataset a) & b)

Step 1:

In my dataset almost all QI columns have been suppressed only Gender, Marital Status and the CPGA score are not

Step 2: see python code

Step 3:

The algorithm applied to the dataset contained the suppression of age, the course name and the year of study. CPGA was aggregated, marital status was left the same, as well as the gender. I did not find any outliers in the dataset as we only have categorical columns left.

Step 4:

The algorithm applied to the dataset contained the suppression of age, the course name and the year of study. It is not possible to regain these values. I could come up with new synthetic values, but this would not allow me to recognize a person in the dataset and breach their privacy. Potentially a background knowledge attack would still be possible if I happen to know that the person did this study and is female has a CGPA in the ranges and is married, but as we applied a k-anonymous algorithm there a few potential persons. So in theory it could be possible with immense background knowledge to identify a person and learn something about their mental health issues.

However, I did not find any other datasets or information on Google to reproduce such an attack.

Due to the huge number of suppressed columns I did not come up with an even theoretic approach to de-anonymize this dataset.

Step 5&6:

not possible

Dataset of colleague c)

3.2.2 c)

The main weakness of the dataset was that there are many occurrences of arm types which are rare. Knowing this and combining it with other data sources linkage attacks are possible.

To prevent that I would design an algorithm with the following steps:

1. Apply the same bayesian interference technique as my colleague.
    1. Unfortunately, I was not able to find out how he did it. So I took the anonymized dataset he send me as the starting point and applied the following steps
2. Detect outliers in the armed column

1. I used plots and visual inspection to detect the rare occurrences of arm types in the data.
2. I decided to define the threshold at everything that has less than 6 occurrences.
3. I applied a replace algorithm to anonymize the gun types with low occurrences by replacing them with new synthetic gun types. I decided to do so because in this way the original distribution is kept but its certified that no linkage attacks can be done anymore, as the new gun values are synthetic.
    1. So for example if nail gun was in a gun type that occurred two times time and thus was a weakness that could be exploited by linkage attacks, now both occurrences are replaced with the same new synthetic gun type. Cordless-drill which occurred one time was replaced by one new synthetic gun type as well. The new synthetic gun types can only be assigned to one of the old gun types, to keep the original distribution.

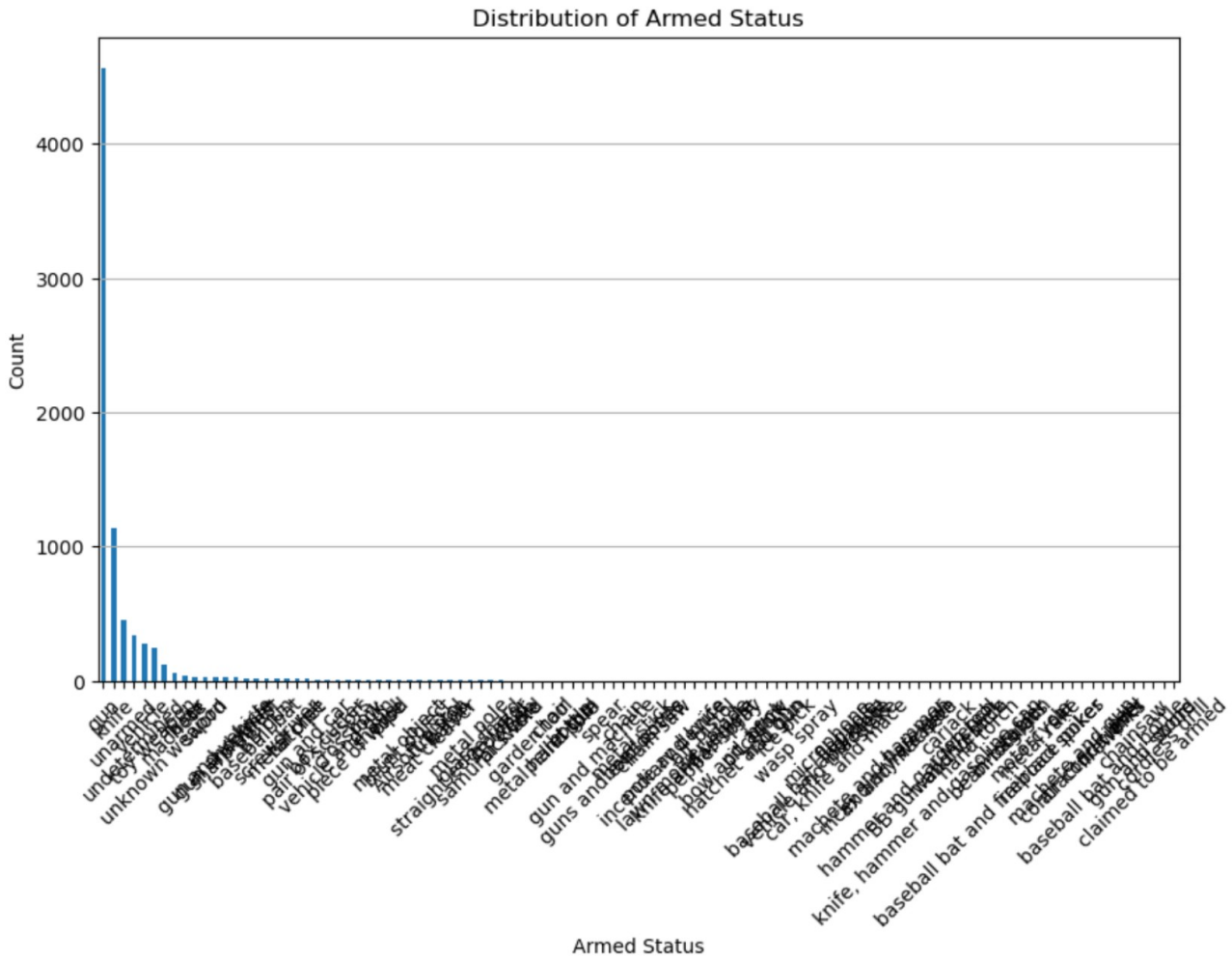

Figure 4 - The distribution of the armed column with the new synthetic values

New values

My dataset c)

From what I know now I think I could have used the bayesian interference algorithm myself to anonymize my dataset in a better way.

In this way, the original distribution would have been kept while the information loss of the data would be rather low. My very high data loss makes the data practically "unreadable" for researcher. A bayesian interference approach would eliminate this disadvantage of my algorithm.

Unfortunately, I was not able to implement bayesian interference in Python myself.

3.3 Experiments -> police shootings dataset

3.3.1 Data Utility

Assumption: Data utility means the usefulness of the anonymized data for statistical analyses by end users as well as the validity of these analyses when performed on the anonymized data. The Fatal Police Shootings dataset contains data about the victim's demographics, how the killing was done and what happened during the killings, as well as geographical data of the killings.

A useful case study would be to find out in what area persons are likely to have certain more or less dangerous arms. I will investigate if you can analyse the **the Relationship Between the type of arm and the location.** From that we could withdraw strategies for the police, which for instance could include that in this area guns are very common, which does not mean that a person holding a gun there must be perceived as very dangerous and shot by police. Resulting in less killings.

Researcher having the original dataset would be very likely to investigate the dataset with the following hypotheses:

Null Hypothesis (H0): There is no significant association between the geographical location (city or state) and the armed status of individuals involved in police incidents.

Alternative Hypothesis (H1): There is a significant association between the geographical location (city or state) and the armed status of individuals involved in police incidents.

This test can be performed using an ANOVA with the original dataset.

3.3.2 Analysis

Based on the knowledge I've built over the other exercises the new anonymised dataset has no risks for de-anonymisation I'm aware of. Exercise 3.2.2 showed that I was not able to de-anonymise the dataset. The new algorithm then replaced replaced the only column I've found which probably kept the original values and distribution and shuffled them simply around. So now its not possible to know which columns contain original values and try to come up with a de-anonymisation approach which makes use of this insight.

However, I could imagine that there might have been a pattern of how the Bayesian interference used previously, which a machine learning algorithm could detect and from there rearrange the data. Yet, this could only be done assuming that the algorithm shuffled around the previous values instead of replacing them with synthetic values.

Speaking of that, I can imagine that a machine learning algorithm could potentially also detect if synthetic values were used or not.

Unfortunately, I was not able to come up with such an algorithm

3.3.3 Method to assess risk of disclosure

A measure to compare the risk of disclosure would be for example to use k-anonymous result of a dataset. If the it is less than two, there is a higher risk for disclosure. In my anonymized student mental health dataset for instance it is higher than one. Making the disclosure of an individual more unlikely. In the anonymized dataset of the police shooting dataset of my colleague and me, it is actually k=1. Which is in theory worse, however as bayesian interference was used the actual disclosure potential is very low. However, for many anonymized dataset k-anonymous is a suitable metric to assess the risk of disclosure.