**Assignments #4-5:**

**Anonymising Textual Data and De-Anonymisation**

**Introduction:**

For Assignments #4 and #5, the focus revolves around the anonymization and subsequent de-anonymization of textual data. In this endeavor, the Titanic dataset serves as the primary subject. The anonymization process involves concealing and modifying certain attributes within the dataset to safeguard individual privacy, while the de-anonymization process endeavors to reverse these modifications and unveil the original information.

**Tasks:**

**3.1 Textual Data Anonymisation**

**3.1.1 Research on Anonymization in the Titanic Dataset:**

* **Name:**Names are personal identifiers that directly link individuals to their data. Anonymizing names mitigates the risk of re-identification and protects individuals' identities.
* **Age:**Age is a sensitive attribute that, when combined with other information, can potentially identify individuals. Anonymizing age through techniques such as aggregation and generalization helps obscure specific details while retaining overall demographic insights.
* **Nationality or Country:**Revealing the exact nationality or country of individuals may lead to the identification of passengers, especially in small datasets. Anonymizing this information by replacing it with generic labels helps conceal individual origins.
* **Numerical Attributes (e.g., Fare, Ticket Number)**:Adding noise or perturbation to numerical attributes protects against precise identification of individuals. Without anonymization, these values could potentially be exploited for re-identification or profiling.
* **Grouped Information (e.g., Passenger Class)**:Randomization within groups, such as passenger classes, helps prevent the inference of individual characteristics based on group patterns. It adds a layer of privacy by obscuring the relationships between individuals and their shared attributes.

**3.1.2 Analysis of Personally Identifiable Information (PII) using spaCy - Explanation**

To identify PII in textual data from the Titanic dataset, I employed spaCy, a Natural Language Processing (NLP) library in Python. Below is an explanation of the process, and you can find the corresponding code in a Python script named pii.py.

* **Generating Sentences from Titanic Dataset (File: sentence.py):**

I began by loading the Titanic dataset from titanic.csv, which contains information about each passenger.For each passenger in the dataset, I crafted a sentence encompassing details like their name, country, gender, age, ticket number, fare, and survival status.All these personalized sentences were then saved to a text file named sentences.txt. This file acts as the input for the subsequent analysis of Personally Identifiable Information (PII).

* **Analyzing PII using spaCy (File: pii.py):**

Then, I created another script named pii.py. In this script, I loaded spaCy's language model (en\_core\_web\_sm) to analyze text.Using a sample sentence structured similarly to those in sentences.txt, I employed spaCy to identify entities like names, countries, etc.The identified entities, representing potential PII, were printed to the console, offering insights into sensitive information within the dataset.

**3.1.3 Applying Masking or Transformation to Detected PII Elements (File: anonymisation.py):**

In the context of anonymizing the Titanic dataset, various transformations were applied to different data types to safeguard sensitive information. Let's delve into the details of these transformations:

**1. Pseudonymization using Faker Library:**

**Operation:** The 'name' column underwent pseudonymization using the Faker library.

Faker is a Python library that generates realistic fake data, including names. By applying the Faker.name() function, real passenger names were replaced with synthetic yet authentic-sounding names. This transformation ensures that individual identities are concealed while maintaining the overall structure of the dataset.

**2. Anonymization of Nationalities with Generic Labels:**

**Operation:** Nationalities in the 'country' column were anonymized using generic labels.

A mapping dictionary (generic\_labels\_mapping) was created to assign generic labels to nationalities encountered in the dataset. The apply function was utilized to replace specific country values with their corresponding generic labels. This transformation ensures that the actual countries of origin are concealed, contributing to the privacy of passengers.

**3. Aggregation and Generalization of Age:**

**Operation:** The 'age' column was aggregated into bins, applying a generalization technique.

The pd.cut function was employed to group individual ages into bins (e.g., '0-20', '20-40'). This transformation provides a level of privacy by generalizing age information. The resulting bins offer broader demographic insights while protecting specific age details.

**4. Perturbation on Numeric Columns (e.g., Fare, Ticket Number):**

**Operation:** Random noise was added to numerical columns for controlled perturbation.

Numeric columns, such as 'fare' and 'ticketno', underwent perturbation. For each numeric value, a percentage of random noise was introduced, computed as 10% of the column's mean. This transformation introduces variability, making it challenging to discern precise values while preserving the overall statistical characteristics of the data.

**5. Randomization Within a Group:**

**Operation:**Data within specific groups, like 'ticketno' within passenger classes, underwent randomization.The groupby and transform functions were utilized to shuffle data within groups. Specifically, the 'ticketno' column was randomized within each passenger class, disrupting any patterns or correlations within those groups. This transformation adds complexity, enhancing the dataset's resilience against re-identification attempts.

These transformations collectively contribute to a robust anonymization strategy, ensuring a delicate balance between preserving individual privacy and retaining the utility of the Titanic dataset.

**3.1.4 Text Analysis Post-Transformation Process:**

After applying a series of anonymization and privacy-preserving transformations to the Titanic dataset, a comprehensive analysis was conducted to assess the efficacy of these measures in concealing sensitive information. The transformations encompassed pseudonymization, anonymization, aggregation, perturbation, and randomization techniques across various data types.

**Findings and Conclusions:**

**Pseudonymization of Names:** The original passenger names were successfully replaced with synthetic names using the Faker library. Pseudonymization proved effective in concealing individual identities, ensuring that real names are no longer accessible in the dataset.

**Anonymization of Nationalities:** Nationalities were anonymized by assigning generic labels. The mapping of nationalities to generic labels obscures specific country information, contributing to privacy preservation.

**Aggregation and Generalization of Age:**Ages were aggregated into bins (e.g., '0-20', '20-40') to generalize individual age details.Generalization of age information protects specific details while retaining broader demographic insights, enhancing privacy.

**Perturbation on Numeric Columns:**Random noise was added to numeric columns like 'fare' and 'ticketno.' Perturbation introduces variability, making it challenging to identify precise values while preserving overall statistical characteristics.

**Randomization Within a Group:**Data, especially 'ticketno,' was randomized within groups (e.g., passenger classes). Randomization disrupts patterns within groups, enhancing the dataset's resilience against re-identification attempts.

**Overall Assessment:**

The combination of these anonymization techniques contributes to a robust privacy-enhancing strategy for the Titanic dataset. The analysis indicates that sensitive information such as names, nationalities, and specific age details has been effectively obfuscated. The dataset maintains its analytical utility while significantly reducing the risk of re-identification. These transformations strike a balance between safeguarding individual privacy and retaining the value of the data for research purposes. The conclusions drawn underscore the success of the applied measures in mitigating privacy risks associated with the original dataset.

**3.2 De-Anonymising a dataset**

**3.2.1 Elements that are De-Anonymisable**The anonymization methods applied to the Titanic dataset reveal potential vulnerabilities in various fields that could be exploited for re-identification. These include the 'Name' field, where the application of cosine similarity with CountVectorizer might unveil patterns in pseudonymous names. Similarly, the 'Age' field, subjected to linear regression, might be susceptible to deanonymization if patterns in the relationship between pseudonymous and original ages are discerned. The 'Nationality' field, anonymized through reverse mapping from generic labels, may be at risk if the mapping process is exposed. Additionally, the 'Fare' and 'TicketNo' fields, perturbed with random noise, could be reversed if the perturbation algorithm is known. Lastly, the 'Randomization Within a Group' applied to 'Class' and 'TicketNo' might be vulnerable if the randomization method is understood. These potential deanonymization points underscore the importance of thorough risk assessment and continuous vigilance in privacy-preserving measures.

**3.2.2 Applying De-Anonymisation algorithms**

So I have applied various De-Anonymising algorithms both on my dataset and on the received dataset

**On My Dataset:**

* **Name Deanonymization using Cosine Similarity(File: deanonymise\_name.ipynb):**

This deanonymization algorithm leverages cosine similarity to associate pseudonymous names with their most similar original names. The original and pseudonymous datasets are loaded, and the 'name' columns are extracted. A CountVectorizer is employed to convert the names into vectors, and cosine similarity is calculated between the pseudonymous and original name vectors. The most similar original names are determined for each pseudonymous name. The algorithm creates a mapping between pseudonymous and original names based on cosine similarity, updating the 'name' column in the pseudonymous dataset. The result is a deanonymized dataset ('titanic\_name\_deanonymized.csv'), revealing the likely original names corresponding to the pseudonymous entries. This method utilizes the semantic similarity of names for effective deanonymization.

* **Age Deanonymization using Linear Regression** **(File: deanonymise\_age.ipynb):**

The algorithm employs linear regression to deanonymize ages in a pseudonymous dataset. It first creates a labeled dataset by concatenating 'age' columns from the pseudonymous and original datasets. Using sklearn's LabelEncoder, it encodes the 'pseudonymous\_age' column and handles NaN values in the 'original\_age' column through mean imputation. A linear regression model is trained on the labeled dataset to predict 'original\_age' based on 'pseudonymous\_age'. Subsequently, the pseudonymous dataset is loaded, and 'pseudonymous\_age' is encoded using the same LabelEncoder. The model predicts the original ages for pseudonymous ages, rounding the values to integers. The 'age' column in the pseudonymous dataset is replaced with the predicted values, resulting in the deanonymized dataset ('titanic\_age\_deanonymized.csv'). This method balances simplicity and effectiveness, considering linear relationships for age prediction.

* **Nationality Deanonymization using Generic Labels Mapping (File: deanonymise\_country.py):**

This deanonymization method focuses on reversing the anonymization of nationality information using a generic labels mapping. The pseudonymous dataset ('titanic\_anonymized.csv') is loaded, and the generic labels mapping is retrieved from the previously saved JSON file ('generic\_labels\_mapping.json'). The script then reverses the keys and values in the mapping to create a reverse mapping. Assuming 'country' is the column to deanonymize, the algorithm applies the reverse mapping to the 'country' column in the pseudonymous dataset. The result is a deanonymized dataset ('titanic\_country\_deanonymized.csv') where generic labels are replaced with the likely original nationalities. This approach showcases the importance of maintaining mappings during anonymization to enable effective reverse transformations.

* **Numeric Columns Deanonymization using Perturbation Reversal (File: deanonymise\_fare\_ticketno.py):**

This deanonymization method targets perturbed numeric columns, specifically 'fare' and 'ticketno,' in the pseudonymous dataset ('titanic\_deanonymized.csv'). The algorithm reverses the perturbation applied during anonymization by subtracting the previously added random noise. Assuming a 10% noise level, the perturbation reversal involves subtracting 10% of the mean of each column multiplied by a random factor from the corresponding column values. The reversed dataset is then saved as 'titanic\_fare\_ticketno\_deanonymized.csv.' This straightforward yet effective approach restores the original numerical values in perturbed columns, contributing to the deanonymization process.

* **Randomization Within Group Deanonymization (File: deanonymise\_randomisation within class.py):**

This deanonymization method addresses the randomization applied within groups, specifically within each passenger class. The pseudonymous dataset ('titanic\_deanonymized.csv') is loaded, and the algorithm reverses the randomization by sorting the dataset within each 'class' based on the original 'ticketno' order. This restores the original order within each class, effectively reversing the randomization process. The reversed dataset is then saved as 'titanic\_deanonymise\_randomization within class.csv.' This approach highlights the importance of understanding the original grouping mechanisms during deanonymization to successfully reverse randomizations applied for privacy protection.

**(a) Discoveries Using De-anonymization Algorithm:**

The de-anonymization algorithm successfully uncovered specific attributes in the dataset. Notably, the reverse mappings for fields like nationality, fare, ticket numbers, and class were accurate, allowing for the retrieval of original values. However, the de-anonymization of names proved challenging, with slight deviations observed. Age, while deanonymized, exhibited some discrepancies but maintained overall accuracy.

**(b) Comparison to Q1:**

When comparing these discoveries with the results from Q1, it's evident that certain fields, including nationality, fare, ticket numbers, and class, were accurately re-identified, aligning with the findings from the original analysis. However, the deanonymization of names and ages showed varying levels of success. Names remained challenging to fully decipher, while age exhibited reasonable accuracy with slight deviations. This comparison underscores the nuanced effectiveness of anonymization techniques, emphasizing the need for continuous refinement to strike a balance between data utility and privacy protection.

**On Received Dataset:**The received Dataset is an anonymised version of fatal-police-shootings-dataset included in the file ‘output\_task.csv’

* **Reversing Pseudonymization: Deanonymizing Date Field in Received Dataset:(File: deanonymise\_date.py):**

In the context of the received dataset ('output\_task.csv'), a reverse pseudonymization approach was applied to the 'date' field. The script utilized the Faker library to generate random dates within the last 30 years, assuming that the original dates fell within this temporal range. This process aimed to revert the pseudonymous dates to a format resembling the original data. The success of this method is contingent on the accuracy of assumptions made during the anonymization phase and the feasibility of recreating the original temporal context. The deanonymized dataset was then saved as 'deanonymized\_output\_task\_date.csv'.

* **Reversing Perturbation: Deanonymizing Age Field in Received Dataset:(File: deanonymise\_age.py):**

The process of deanonymizing the 'age' field in the received dataset involved implementing a reverse perturbation algorithm. The goal was to counteract the effects of anonymization, specifically the addition of 10% noise to the original age values. This was achieved by subtracting 10% of the age from each entry, effectively reversing the perturbation. The resulting dataset, now deanonymized in terms of age, was then saved for subsequent examination and comparative analysis with the original dataset. This approach aimed to restore the accuracy of age information while respecting privacy measures implemented during anonymization.

* **De-Anonymizing Location Data: Reversing Perturbation in 'Longitude' and 'Latitude':(File: deanonymise\_location.py)**

The deanonymization of location information, specifically 'longitude' and 'latitude,' in the received dataset involved applying a reverse perturbation algorithm. During the anonymization process, a 10% noise was added to these coordinates to protect the privacy of individuals. To reverse this perturbation, 10% of each coordinate was subtracted. This approach aimed to restore the accuracy of location data while maintaining the privacy considerations introduced during anonymization. The resulting dataset, now deanonymized in terms of location information, was saved for further analysis and comparison with the original dataset.

* **De-anonymizing Categorical Fields: A Mode-Based Approach:(File: deanonymise\_gender.py,deanonymise\_race.py,deanonymise\_state.py,deanonymise\_city.py)**

The deanonymization process focused on the 'gender', 'race', 'city', and 'state' fields using a grouped mode approach. The grouped mode involves identifying the most frequently occurring value within specific groups, thereby reversing the anonymization. For 'gender', 'race', 'city', and 'state', the dataset was grouped based on relevant criteria such as specific categories or regions. Within each group, the mode (most frequent value) was determined for each field, and this mode was then assigned to the corresponding instances within that group. This technique helps to restore the original categorical information for these attributes. The deanonymized dataset was then saved, preserving the integrity of the original characteristics while providing a more interpretable and identifiable representation of 'gender', 'race', 'city', and 'state'.

**(a) Discoveries Using De-anonymization Algorithm:**

Date Deanonymization: By reversing the pseudonymization applied to the 'date' field, we were able to approximate the original dates of events. However, it's important to note that this might not be precise due to the nature of pseudonymization.

Age Deanonymization: We successfully reversed the perturbation applied to the 'age' field, providing an estimation of the original ages. The accuracy depends on the extent of noise added during anonymization.

Location Deanonymization: Perturbations in 'longitude' and 'latitude' were reversed, offering insights into the approximate original geographical coordinates. The precision relies on the level of noise introduced during the anonymization process.

Gender, Race, City, State Deanonymization: Applying reverse mechanisms using grouped mode, we attempted to deanonymize categorical fields like 'gender,' 'race,' 'city,' and 'state.' However, the success of this process is contingent upon having access to the original mappings used during anonymization.

**(b) Comparison to Q1:**

In comparison to the results from Q1, the discoveries in the received dataset exhibit similar patterns. The pseudonymization and perturbation reversal techniques applied were effective in revealing approximate original values. However, the accuracy is influenced by factors such as the type and amount of anonymization applied. Categorical fields, when anonymized using methods like grouped mode, present challenges in achieving complete deanonymization without access to the original mapping information.

1. **Alternate Anonymisation Approach:Enhanced Data Protection through Fernet Encryption(File: alternate\_anonymization.py):**

An alternative anonymization approach was applied to the 'fatal-police-shootings-data.csv' dataset, focusing on utilizing Fernet encryption for heightened data privacy. This method centers around securing sensitive information like names, ages, and geographical coordinates through robust cryptographic transformations, providing a resilient defense against unauthorized access.

Implemented using the 'cryptography' library in Python, the process involved generating a unique encryption key and applying Fernet encryption to specified columns ('name,' 'age,' 'longitude,' and 'latitude'). This encryption renders the raw values indecipherable without the corresponding key, significantly strengthening the security of the dataset.

The encrypted dataset, named 'encrypted\_fatal\_police\_shootings\_data.csv,' maintains utility for authorized users with the ability to decrypt the data when needed, ensuring reversible anonymization. The approach not only aligns with contemporary data protection standards but also addresses the increasing need for secure handling of sensitive information in compliance with privacy regulations. Fernet encryption emerges as a powerful method for fortifying data security and safeguarding against potential privacy breaches.

**3.3 Experiments**

**3.3.1 The utility of data generated:**

The utility of the data generated using the proposed alternate anonymization scheme, Fernet encryption, lies in its ability to strike a balance between enhanced data privacy and practical usability. Fernet encryption provides a robust layer of security, ensuring that sensitive information such as names, ages, and geographical coordinates is protected against unauthorized access. The encrypted dataset, 'encrypted\_fatal\_police\_shootings\_data.csv,' maintains its utility by allowing authorized users with the corresponding decryption key to revert the data to its original form.

One key advantage is the reversible nature of Fernet encryption, enabling seamless access to the raw data for legitimate and authorized purposes. This reversibility ensures that while the data remains secure, it remains functional for users who possess the necessary credentials. This characteristic distinguishes Fernet encryption from irreversible anonymization techniques, providing a valuable compromise between privacy preservation and data utility.

Moreover, the implementation of Fernet encryption aligns with contemporary data protection standards, making it well-suited for compliance with privacy regulations and ensuring that the dataset meets ethical and legal considerations. The utility of the dataset is thus enhanced, offering a secure and flexible solution for handling sensitive information in scenarios where privacy and usability are both paramount concerns.

**3.3.2 Risks of De-Anonymization:**

Analyzing the new anonymized dataset generated using Fernet encryption is crucial to identify potential risks of de-anonymization. While Fernet encryption provides a strong layer of security, it's essential to evaluate the dataset's susceptibility to various de-anonymization attacks.

One potential risk lies in the correlation of certain features within the dataset. For example, if specific combinations of attributes, such as age, gender, and armed status, exhibit unique patterns, an adversary might exploit these patterns to re-identify individuals. While Fernet encryption protects against direct exposure of sensitive information, patterns and correlations within the dataset may still pose risks.

Another aspect to consider is the impact of variable data distributions. If the encrypted dataset maintains similar statistical properties as the original, an attacker might leverage external knowledge or auxiliary information to infer specific characteristics, potentially leading to de-anonymization. Evaluating the distributional characteristics of the anonymized features is crucial in understanding these risks.

Additionally, the risk of frequency-based attacks should be assessed. If certain categories or values occur with distinct frequencies, an adversary could use statistical analysis to infer the likelihood of specific attributes, compromising the anonymity of individuals within the dataset.

It is imperative to conduct a thorough risk analysis, considering factors like attribute uniqueness, patterns, and distributional characteristics. Employing techniques such as k-anonymity, l-diversity, or t-closeness can enhance the dataset's resilience against de-anonymization risks. Regularly updating encryption keys and periodically reassessing the dataset's vulnerability to evolving de-anonymization methods are crucial steps to maintain a high level of security.

**3.3.3 Method of Assessing risk of disclosure:(File: risk\_of\_disclosure.py)**

To assess the risk of disclosure (de-anonymization) in anonymized datasets, one effective method is to employ the concept of k-anonymity. K-anonymity is a privacy-preserving property that ensures each record in a dataset is indistinguishable from at least k-1 other records based on a set of quasi-identifiers. Quasi-identifiers are attributes that, when combined, can potentially re-identify individuals.

Here is a step-by-step approach to assess the risk and apply the k-anonymity metric:

**Identify Quasi-Identifiers:**Determine the set of quasi-identifiers in the dataset. These are attributes or a combination of attributes that could potentially lead to the re-identification of individuals.

**Define Sensitivity Threshold (k):**Choose an appropriate sensitivity threshold (k) based on the desired level of anonymity. A higher value of k implies a higher degree of anonymity but may sacrifice data utility.

**Group Records:**Group records in the dataset based on the quasi-identifiers. Ensure that each group (equivalence class) has at least k-1 other records.

**Evaluate Anonymization:**Assess the dataset to determine if it satisfies the k-anonymity property. If each record is indistinguishable from at least k-1 others within the same group, the dataset is considered k-anonymous.

**Risk Analysis:**Evaluate the risk of disclosure by considering factors such as the uniqueness of quasi-identifiers, the potential for attribute disclosure, and the impact of external information.

Apply Generalization or Suppression:

If the dataset does not meet the desired k-anonymity threshold, apply further anonymization techniques such as generalization or suppression to enhance anonymity.

Iterative Improvement:

Iterate through steps 3-6 until the dataset achieves the desired level of k-anonymity while balancing data utility.

By employing k-anonymity, this method provides a quantitative measure of the risk of disclosure. Regularly re-evaluating and adjusting the anonymization process based on evolving risks and threats ensures ongoing protection of sensitive information in the dataset.

**Conclusion:**

I found this method of assessing risk of disclosure very effective and accurate  
  
For my own anonymised dataset it shows that "The dataset satisfies k-anonymity."

But the anonymised dataset which I received from my friend shows "The dataset does not satisfy k-anonymity. Further anonymization is needed."

The dataset I obtained after my alternate approach of deanonymising using fernet encryption also shows "The dataset satisfies k-anonymity."  
  
Some datasets from the previous assignments were found to be satisfying while some are not satisfying the metrics

In conclusion, anonymization plays a crucial role in safeguarding data privacy and mitigating security risks. By transforming sensitive information into a format that conceals individual identities while preserving overall data utility, anonymization serves as a fundamental measure to protect against unauthorized access and disclosure. It enhances data safety, enabling organizations to share valuable insights without compromising the privacy of individuals. However, it is essential to continuously evaluate and improve anonymization methods to stay ahead of evolving security challenges and ensure the continued effectiveness of data protection measures.