**3. PROJECT ABSTRACT**

**3.1 ABOUT THE PROJECT**

Data is a crucial part of every individual’s daily life. According to a report by DOMO in 2020, it was estimated that every minute, the following data is generated on the internet:

* 347,222 Instagram stories are posted
* 500 hours of video are uploaded to YouTube
* 41.6 million messages are sent on WhatsApp
* 4.5 million videos are viewed on Snapchat
* 4.7 billion searches are made on Google
* 188 million emails are sent

It would not be a stretch to say that we live in the world of data. However, with a lot of data, comes a lot of responsibilities. It is becoming increasingly difficult to safeguard the identity of the user data and protect the sensitive information of the user online as a lot of tools already exist which can “connect” the different parts of data with one another to reveal information that should have been kept private. The given project suggests an end to end data anonymization pipeline using static anonymization techniques such as K-Anonymity, L-diversity and T-closeness to safeguard the user data. Unique thing about the proposed approach is that the data, after anonymization, still retains a lot of statistical significance and can be used to develop state of art research models and generate insights from the data.

**3.2 OBJECTIVES:**

There have been various goals while designing the solution for this project but the main idea boils down to making sure that the solution brings about a change in the field of data anonymization as there is need for implementation of more state-of the art methods keeping the modern database structure in mind

* To provide a solution such that the data of the clients is safeguarded and personal
* To design the algorithm keeping the security aspect of the data in mind and to make sure that none of the data gets leaked during any stage of the software development lifecycle
* To design a solution in such a way that it supports the clients native data structure
* To design the solution in such a way that the statistical nuance of the data is not lost

**3.3 PROJECT SCOPE**

The given solution is implemented in python 3.10.0 but can be easily replaced and modified to suit almost all the versions of python. Moreover, the additional option to combine the API with the UI integration opens up the scope for a lot more end users to benefit from the project

**3.4 LITERATURE REVIEW**

All of the given algorithms for the project have been implemented using python and some parts of implementation has also been done using Java

The project was developed using SCRUM and JIRA technologies where the ticket system has been prevalent

**3.4. 1 Overview of the technologies**

*Python:* Python is an open source object oriented programming language that is used in variety of tasks such as Machine Learning, Data Science, Web Development, Cryptography, Game development etc

**3.4.2 Review of the algorithms**

K Anonymity was first introduced by L.Sweeney of the Carlie Melington university where she proposed a paper titled “K-ANONYMITY, A MODEL FOR PROTECTING PRIVACY”

This project takes a lot of inspiration from the pseudo code presented in that chapter. Moreover, we also acknowledge that since data attacking techniques have gotten a lot more sophisticated, we implement a lot of other versions than just the basic top-down greedy approach

Building upon the previous approach, mondrian (basic and advanced) are implemented based upon the paper titled Mondrian Multidimensional K Anonymity by the University of Madison

We also further develop the project by allowing the option to leave in insensitive attributes and sensitive names using relaxed mondrian approach given in Pycon 2019

**3.5 PROJECT PLANNING**

**3.5.1 Project Development plan and implementation**

| **Index** | **Milestone** | **Expected Completion date** | **Actual Completion date** |
| --- | --- | --- | --- |
| 1 | Basic K-anonymity using top down greedy approach | 20/02/2023 | 20/02/2023 |
| 2 | Implementing Basic Mondrian algorithm | 05/03/2023 | 09/03/2023 |
| 3 | Implementing K Anonymity using clustering and NCP | 08/03/2023 | 13/03/2023 |
| 4 | Implementing relaxed Mondrian | 12/03/2023 | 15/03/2023 |
| 5 | Testing of the algorithms | 12/03/2023 | 16/03/2023 |
| 6 | Critical analysis of the algorithms | 15/03/2023 | 16/03/2023 |
| 7 | Hierarchy Generation | 17/03/2023 | 20/03/2023 |
| 8 | Finalising the datasets | 21/03/2023 | 21/03/2023 |
| 9 | Adding support for various integrations | 14/04/2023 | 17/04/2023 |
| 10 | Creating User Interface | 01/05/2023 | 29/04/2023 |
| 11 | Testing and model building | 04/05/2023 | 03/05/2023 |

Table 3.4 Project Development Plan with goals and Milestones

**3.5 COST ANALYSIS AND ESTIMATION**

**3.5.1 Development cost:**

The development cost associated with the project depends on the algorithm that we are trying to develop. The development cost comes with three factors:

* The technology being used:

The technology influences the cost of the project heavily as open source languages like python are a lot more likely to have support integration rather than closed source tools like MATLAB

* Algorithms being implemented:

1. NP-HARD: NP-HARD problems are usually compute intensive and hence are difficult to optimise moreover they require significant prerequisites

* Features being added along with the algorithms:

More the features, more cost it will be to add them to the project

Development cost = (No of NP-HARD algorithms\* features being added)

**3.5.3 Deployment cost:**

The cost associated with the project comes with the size of the dataset and the number of the iterations and the depth of the data associated. Moreover, snowflake offers 400 USD credits off for the first month. After that, the users are charged on a per-second basis and the charge depends on the tier that the user chooses for their snowflake account and the amount of the data they have

cost= (size of data)\*seconds the data was running\*69

**3.6 PROJECT EXCERPT**

The given project is a derivative of a client’s project which was later revised and is now available as an open-source Github repository open for anyone to use and contribute to.

The client in question is a US-Based healthcare startup/hospital which has to deal with patients on a daily basis and hence a lot of legacy data. A good chunk of the data contains sensitive information about patients such as their names, social security number, zip codes, email-addresses etc. A lot of the data is potentially really helpful to the research community and can be used to derive scientific insights from the same. The task here is to build an end-to-end data anonymization pipeline which takes care of the data anonymization and data validation before releasing the data to the public. The proposed solution uses snowflake to store and retrieve legacy data and python script to anonymize the given patients data. The data is then stored to another public-facing database where users can integrate it with no-code machine learning and data visualisation platforms like Mindsdb and Powerbi (API support is provided for the same).. The given project implements anonymization algorithms mentioned above in python from scratch without needing the arx api/local arx server and effectively eliminates a lot of the constraints that came alongside using the module

**3.7 AN INTRODUCTION TO DATA PRIVACY AND ANONYMIZATION:**

Data privacy is a fundamental right of every individual in this day and age. With people’s online presence increasing, data privacy is important now more than ever as handling large amounts of data and protecting the rights of an individual is a complex task

Data Anonymization is a form of privatising the data before making it public or sending it to the intended audience.Data anonymization refers to masking sensitive user data in a way that the identity of the user is maintained and if the data is released, the data can not be traced back to the user. Data anonymization is the tradeoff between usability of the data in terms of statistical parameters and privacy of the users and to maintain the same statistical models and domain knowledge are combined to make sure that the anonymization is done properly.

## **3.7.1 Why is data anonymization needed?**

One of the biggest examples which states the importance of anonymizing the data is the 2006 AOL data search leak. Here are some interesting statistics about the data leak:

* On 4th of August 2006, search data of approximately 650,000 users along with 20 Million search results were leaked
* The data was removed relatively quickly on 7th of August 2006
* The AOL did not identify the users in the data as the names of the users were not explicitly mentioned in the data
* However, a popular newspaper magazine called New York times were able to identify the users by cross referencing them with other sources like phone book listing

Another popular data breach is that of the Netflix where the data was leaked and the researchers at the university were able to trace it back to the users

Although proper care should be taken that the data does not get leaked in the first place, it is equally necessary to make sure that if the data gets leaked, the sensitive information is not revealed to the users.

### To ensure the privacy of the data, a five parameter framework is used:

1. Ensure that the data is safe
2. Ensure that the people working on the data are safe
3. Ensure that the scope of the project is viable
4. Ensure that the proper compliant standards are set up to ensure safety in place
5. Disclose of the output data can be monitored to ensure that sensitive data is not leaked

**3.7.1.1 Types of anonymization**

There are two types of anonymization:

1. *Static*
2. *Dynamic*

***Static anonymization*:** Static anonymization refers to the anonymization of the data all at once and then the data is being released to the public or the third party source or vendor. In static anonymization, often a subset of the original data is released after anonymization to the users. Popular static anonymization tools and softwares are ARX, Amnesia etc

***Dynamic Anonymization****:* Dynamic anonymization refers to the anonymization of the data using queries. Often the full dataset is released to the public and the anonymization of the dataset is happening in real-time. Dynamic anonymization is considered more reliable than the static one because static anonymization needs to specify the anonymization techniques very carefully otherwise the subset of the data can be extracted multiple times to paint the picture of the actual data. Some popular dynamic anonymization techniques are: The popular R package diffpriv, Google’s RAPPOR etc

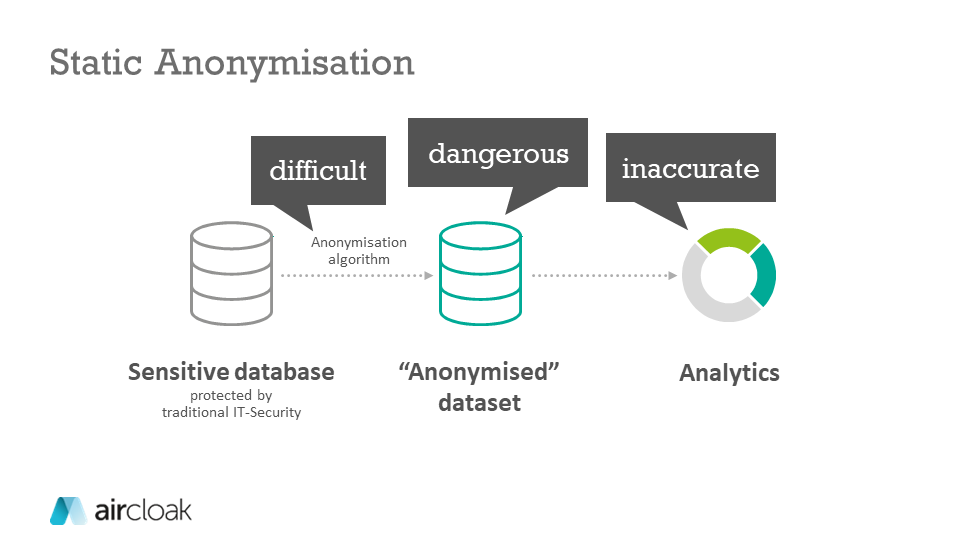


Figure 2 Static Anonymization

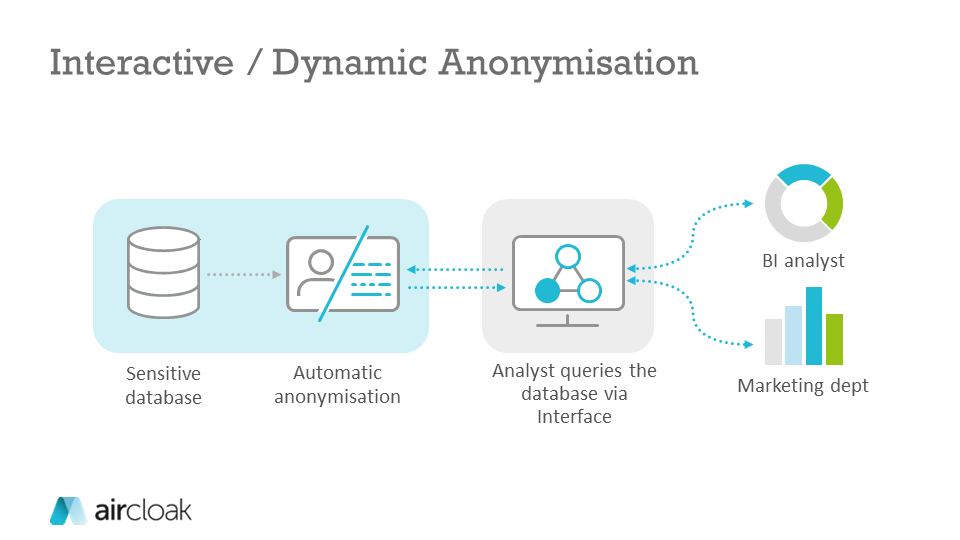


Figure 3 Dynamic Anonymization

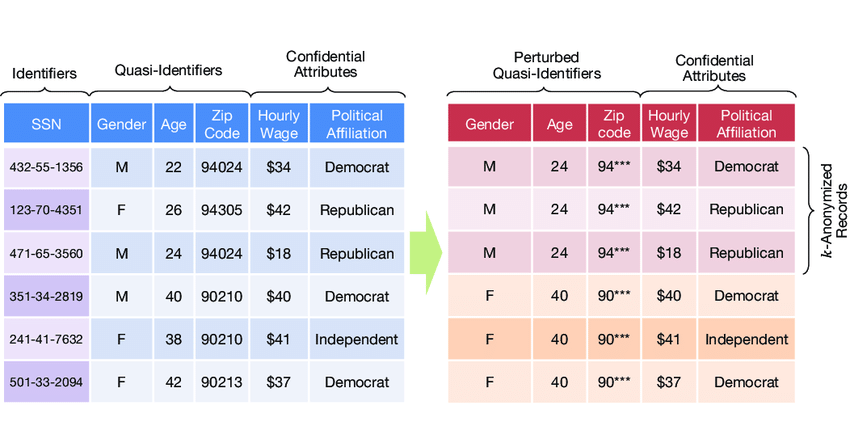


Figure 4 The process of anonymization

**3.8 SOLUTIONS PROPOSED**

**3.8.1 Proposed solution 1:**

*Anonymization of the data using aws databrew*

**Diagram**

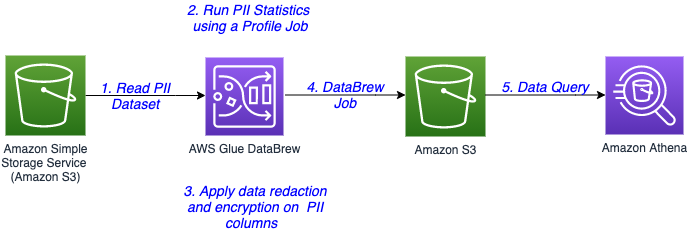


Figure 5 Architectural Diagram of Data Anonymization using AWS Databrew

**Components:**

1. **AWS S3:** As explained in the previous solution, s3 is an object based service used to store the data
2. **AWS Glue DataBrew:** Is a service by amazon that aids data scientists and machine learning engineers in cleaning and normalising the data.
3. **AWS Athena**: Amazon Athena is a serverless, interactive analytics service built on open-source frameworks, supporting open-table and file formats. Athena provides a simplified, flexible way to analyse petabytes of data where it lives.

**Explanation:**

the data in question to be anonymized is stored in an s3 bucket. Moreover, macie can be used to detect sensitive data from the already present data. The data from s3 is then loaded into the aws glue data brew which has jobs which mask the sensitive PII column-wise and after the data is masked, the masked data is then stored into s3 which creates a external table on top of athena

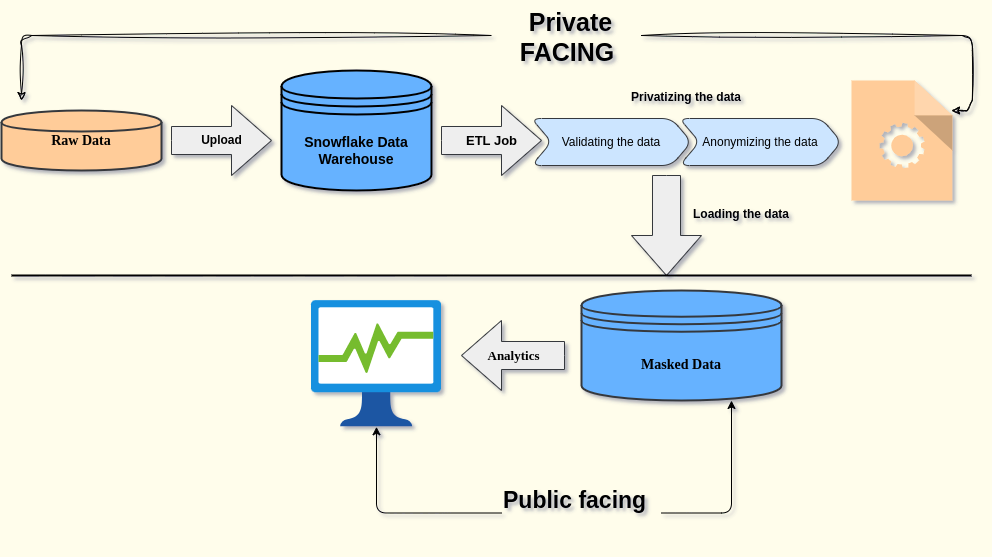
**Feedback from the mentor and drawbacks:**

Using aws macie to detect sensitive data might work for use cases where the user might have accidentally entered the sensitive data such as credit card information inside the s3 bucket. In our use case, the data that we want to mask is just not sensitive data, but the data that can be used in conjunction with other data to identify the users as well which is something where macie fails. Additionally, the aws data brew has limited support for masking the sensitive data and if the attacker knows the method or the technique using which the data was masked, it might be easy for the attacker to re-identify the data. Overall, we need a solution that is fast, has versioning capability, is able to handle legacy data and also uses a predefined algorithm to anonymize a data with a known schema while being cost optimised

**3.8.2 Proposed solution 2**

### *Data Anonymization Pipeline using Snowflake and airflow*

Diagram



**Components:**

**Snowflake:** Snowflake is a SaaS data warehouse solution designed solely for the cloud, it supports popular cloud providers like gcp, aws and azure. The snowflake can be used to build data warehouse and data lake right from your browser as snowflake is a managed service

**PYARXAAS:** Is a popular python module which is used as a static anonymization tool. Pyarxaas provides a wrapper to access functions for your local arx instance.

**Apache Airflow:** Apache airflow is an open source, etl tool that is used to automate the data loading and data transforming pipeline

**Explanation:**

The following solution provides an end to end pipeline for anonymizing data that gives users the flexibility to scale the solution as per their demand or data. Here is the process of pipeline:

1. Takes in the raw data either from the source or from a sql database. What makes the solution truly customizable is that snowflake can take in data from multiple sources Snowflake natively supports AVRo, Parquet, CSV, JSON and ORC hence data from pretty much any source can be taken
2. The raw data is then loaded into the snowflake to create a data warehouse. The access to the given snowflake is heavily restricted. This step is also customizable and scalable as if you want to give certain users access to the raw data warehouse, you can simply assign their role access and they can be managed using snowflake organisation. Thus, reducing the costs and increasing the flexibility
3. After that, an ETL job is run on the data warehouse which validates the data if need, sets the hierarchy of anonymization for the data and actually anonymizes the data. The anonymization being used here is a function written manually and will be triggered after a certain time when the data is loaded in the data warehouse. Here, since we already know the project columns and all the details about the data being present inside we will use static anonymization instead of dynamic but dynamic anonymization can be used as well
4. The anonymized data is then stored in another snowflake data table or this can be also be connected to the cloud provider of your choice but an interesting thing about snowflake is that it really lets you create stunning visualisations using the snowflake console itself
5. The masked data which is not public facing can be used to generate charts, analysis and visualisation

**Comments from the mentor and drawbacks:**

One of the biggest advantages of this solution is the scalability in terms of compute and storage. SInce snowflake is a hybrid between the shared nothing and shared everything architecture, it is really easy to scale up or down. The manual data anonymization gives the users greater flexibility to implement their own algorithm. However, one potential challenge to solve is trying to anonymize the data automatically without having the schema of the data. This is still a challenge since pyarxaas is a static anonymization algorithm References:

[https://aircloak.com/data-anonymisation-software-differences-between-static-and-interactive-anonymisatio](https://aircloak.com/data-anonymisation-software-differences-between-static-and-interactive-anonymisation/)n

**3.9 SCOPE OF THE PROJECT: WHAT IT CAN AND CAN NOT DO**

The data anonymization solution provided in the given project will have a humble impact as it can be modified and applied to a variety of sql databases, csv files, excel spreadsheets and dataframes. Moreover, support is also provided for snowflake tables and s3 instances. However, these will require a lot more configuration.

The given project not only suppresses the sensitive and identifying attributes in the dataset but it also takes care of the quasi-identifying attributes.

The project also takes care of both categorical and numerical variables. However, the user will need to define the categorical variables themselves

The given project can not generate the hierarchies for anonymization with sophistication, only basic functions are provided to generate the hierarchies. It is recommended that users generate their own hierarchies and then use the project to anonymize the data. Moreover, the project is more suited towards social science and healthcare data but can still be applied to other types of data