# 

# **Prototype / MVP Document**

## 1. Introduction

The MVP we plan to suggest is a comprehensive presentation that outlines the key components of our data processing pipeline, from infrastructure setup to the final dataset. This aims to provide better insights on our approach to managing and processing data to produce a quality dataset. This presentation will demonstrate our codebase encompassing our infrastructure design, code, and the data pipeline. Additionally, the presentation will also include the results of our data pipeline which is a prototype dataset with roughly around 30 features that have been cleaned, synthesised and enriched.

## 2. Prototype Overview

### 2.1 Prototype concept

The prototype consists of a detailed presentation that highlights two critical components:

1. **Codebase**: A comprehensive explanation of our GitHub repository, showcasing the infrastructure design, data processing scripts, and automation workflows that constitute the data pipeline.
2. **Final Dataset:** An initial version of the enriched dataset containing around 30 key features essential for effective fraud detection in insurance claims.

### 2.2 Key features

1. **Infrastructure as Code:** Utilising Terraform to define and manage Azure infrastructure, ensuring reproducibility and manageability.
2. **Automated Deployment:** Implementing GitHub Actions scripts for seamless deployment of infrastructure and Azure Functions to the cloud.
3. **Data Loading:** Ensuring efficient and secure data transfer with PowerShell and Azure Data Lake Gen2 REST API to ingest raw data.
4. **Data Processing Pipeline:** The pipeline includes scripts for data cleaning (such as renaming and removing columns), synthesising, enriching, and merging data from multiple sources. At this stage, the code is not yet deployed on Azure. Instead, we are developing and testing it locally, simulating the pipeline that will eventually be deployed on the Azure platform. All scripts are version-controlled on GitHub and are currently being executed manually.
5. **Final Dataset:** A dataset with essential columns that will be used by the AI team to train their initial prototype.

## 3. Current project Codebase & Dataset

### 3.1 Codebase

The most up to date version of the codebase can always be found at this link: <https://github.com/AlanDataPortfolio/ey-azure-fn-pipeline>.

The folder structure cannot be shown with a screenshot so here is the best way to visualise the current codebase.

## **Sponsor meeting, feedback and response to feedback**

## **Prototype Feedback Adjustment**

## **Analysis, Design + Testing Documentation for DATA SCIENCE projects**

# **Feature engineering:** (deciding what you are looking for and how to go about it) - Adam

· From the data, are there any trends / ranges to look for?

e.g. if you are analysing time-based movement data, what accelerometer and gyroscope ranges / trends do you want to look for to identify someone that has fallen / collapsed?

e.g. if you are looking at financial data, what does an "upward sale trend" look like?

e.g. if you are looking at population data, or education rates, what "characteristics" would you want to look at and why?

· Describe what data characteristics are being looked for, and how your data pipeline is being processed to generate these features.

· Give each different "feature" or "characteristic" a name and then attribute some form of data ranges / statistical definition.

1.1 Trends and ranges - Identifying the trends and ranges is crucial for feature engineering. The patterns will help to better understand the dataset. For our case, when detecting fraudulent insurance claims, it is important to look at any anomalies in the data. For example, if the ‘totalClaimAmount’ is high and the ‘incidentSeverity’ is minor, it is most likely to be a fraudulent claim. The ‘totalClaimAmount’ column itself can be a good indicator of a fraudulent claim. If the amount is too high and is significantly more than the average claim amount, that claim will be flagged for a fraudulent claim.

Columns such as ‘driverAge’, ‘driverExperience’, and ‘licenseType’ can give us information about the driver. These columns can also provide insights on the driver behaviour. For example, younger drivers, who are less experienced, are more likely to be driving rough and engage in speeding. Whereas, middle-aged drivers are more experienced and are more likely to drive safely. Therefore, these columns can help to identify the type of driver and help in interpreting whether the claim is fraudulent or not.

1.2 Data characteristics and pipeline - As the dataset is created with the intention to use it to train an AI model that will detect fraudulent claims, fraud indicators will need to be considered. Therefore, we would need a fraud column and columns that can help explain a fraudulent claim. Information relating to the incident and the customer details will also be needed to determine a false claim.

Our data pipeline consists of cleaning, enriching, synthesizing and merging to get the combined dataset. In data cleaning, the columns were renamed

1.3 Feature definition - Name every feature and explain them

# **Solution Architecture:** (choice of macro architecture / pipeline) - Alan

· Provide an overall description of each section of the pipeline including the data in and out of each "stage". The data details can be properly described in the "Detailed data descriptions" section.

· This would most likely be a more detailed version of the overall pipeline presented in the team's scoping document as some of the stages would now be implemented / finalised.

· Include any resources available / resource processing constraints to each of the sections in the pipeline (e.g. processing / timing limits)

# **Algorithms / models methods:** (detail what is in each part of the solution architecture, including models used and initial conditions / config settings) - Ninuri

· selection of model... there are many different approaches: predictive, supervised / unsupervised, classifiers, ... which are going to be used? why? and why chose those over other approaches?

· Are there any settings needed (eg. in a KNN model, what is the number of the nearest nodes being used in the application of the classifier?)

3.1. KNN Model

3.1.1. Configuration Settings

3.1.2. Reasons for Model selection

3.2. Random Forest Models

3.2.1. Configuration Settings

3.2.2. Reasons for Model selection

3.3. Why these models were chosen over other approached?

# 

# **Detailed Data descriptions: Noor**

Data being used, data being generated, data being stored, as well as any summaries and/or reports.

Raw Datasets

The raw datasets being used are 3 Kaggle Datasets which contain historic data on automobile insurance claims. The reason these 3 datasets were primarily chosen was because they contain a column which indicates whether that row was a fraud or not. This is specifically useful to train the AI and ML model which will be a fraud detection tool.

The original datasets are raw, unclean and with many missing values and non-standardised values.

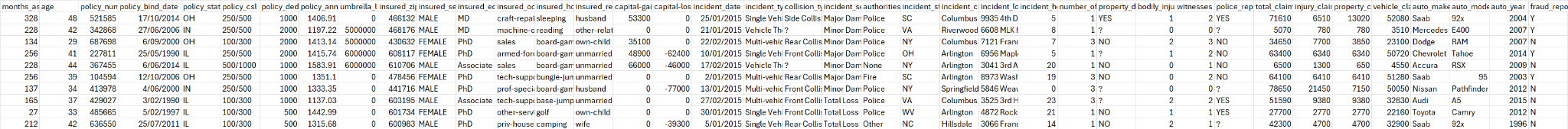
Dataset 1 is a good dataset and has 1000 rows. Many different websites and sources have this exact same dataset. We got it of kaggle as that’s open source. Link to it is:

Dataset 2 has 10000 rows but is about all types of fraud, only about 1500 of these is for automobile fraud which is what we’re targeting. So dataset 2 has 1500 rows.

Dataset 3 hs 10000 rows but is the most messiest one and has random un standardised values and alotta missing and blank values.

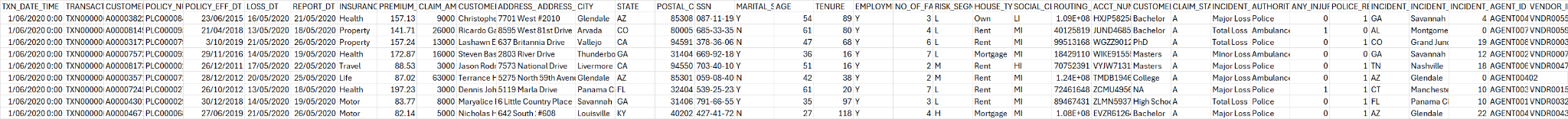
Columns in Raw Datasets:

Raw Dataset 1:



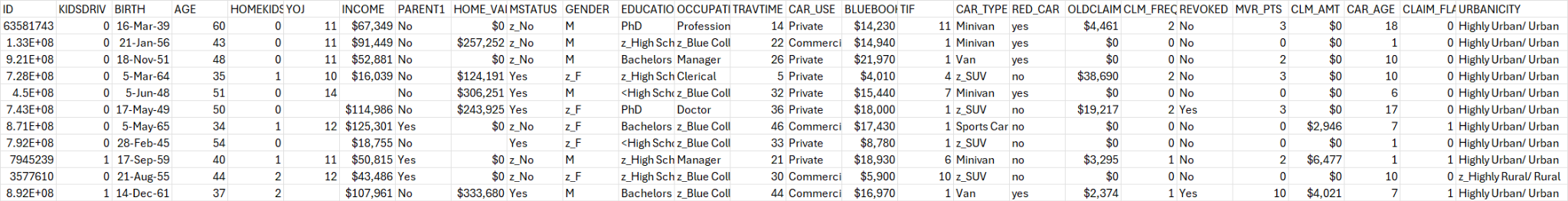
1. months\_as\_customer: Number of months the individual has been a customer (Numerical).
2. age: Age of the customer (Numerical).
3. policy\_number: Unique identifier for the insurance policy (Categorical).
4. policy\_bind\_date: The date when the policy was created/bound (Date).
5. policy\_state: The state where the policy was issued (Categorical).
6. policy\_csl: Combined Single Limit for the policy (Categorical).
7. policy\_deductable: The amount of deductible in the policy (Numerical).
8. policy\_annual\_premium: The annual premium paid for the insurance policy (Numerical).
9. umbrella\_limit: The umbrella limit in the policy (Numerical).
10. insured\_zip: Zip code of the insured person (Categorical).
11. insured\_sex: Gender of the insured person (Categorical).
12. insured\_education\_level: Education level of the insured person (Categorical).
13. insured\_occupation: Occupation of the insured person (Categorical).
14. insured\_hobbies: Hobbies of the insured person (Categorical).
15. insured\_relationship: Relationship status of the insured person (Categorical).
16. capital-gains: Capital gains of the insured person (Numerical).
17. capital-loss: Capital loss of the insured person (Numerical).
18. incident\_date: Date of the incident (Date).
19. incident\_type: Type of incident (Categorical).
20. collision\_type: Type of collision (Categorical).
21. incident\_severity: Severity of the incident (Categorical).
22. authorities\_contacted: Whether authorities were contacted (Categorical).
23. incident\_state: State where the incident occurred (Categorical).
24. incident\_city: City where the incident occurred (Categorical).
25. incident\_location: Specific location of the incident (Categorical).
26. incident\_hour\_of\_the\_day: Hour of the day when the incident occurred (Numerical).
27. number\_of\_vehicles\_involved: Number of vehicles involved in the incident (Numerical).
28. property\_damage: Whether property damage occurred (Categorical).
29. bodily\_injuries: Number of bodily injuries (Numerical).
30. witnesses: Number of witnesses to the incident (Numerical).
31. police\_report\_available: Whether a police report is available (Boolean).
32. total\_claim\_amount: The total amount claimed (Numerical).
33. injury\_claim: Claim amount for injuries (Numerical).
34. property\_claim: Claim amount for property damage (Numerical).
35. vehicle\_claim: Claim amount for vehicle damage (Numerical).
36. auto\_make: Make of the vehicle involved (Categorical).
37. auto\_model: Model of the vehicle involved (Categorical).
38. auto\_year: Year of the vehicle involved (Numerical).
39. fraud\_reported: Whether fraud was reported (Boolean).

Raw Dataset 2:



1. TXN\_DATE\_TIME: The date and time of the transaction (Date/Time).
2. TRANSACTION\_ID: Unique identifier for the transaction (Categorical).
3. CUSTOMER\_ID: Unique identifier for the customer (Categorical).
4. POLICY\_NUMBER: Unique identifier for the policy (Categorical).
5. POLICY\_EFF\_DT: The policy effective date (Date).
6. LOSS\_DT: Date when the loss occurred (Date).
7. REPORT\_DT: Date when the incident was reported (Date).
8. INSURANCE\_TYPE: Type of insurance (Categorical).
9. PREMIUM\_AMOUNT: Premium amount for the insurance policy (Numerical).
10. CLAIM\_AMOUNT: Amount claimed (Numerical).
11. CUSTOMER\_NAME: Name of the customer (Categorical).
12. ADDRESS\_LINE1: Address line 1 of the customer (Categorical).
13. ADDRESS\_LINE2: Address line 2 of the customer (Categorical).
14. CITY: City of the customer (Categorical).
15. STATE: State of the customer (Categorical).
16. POSTAL\_CODE: Postal code of the customer (Categorical).
17. SSN: Social Security Number (Categorical).
18. MARITAL\_STATUS: Marital status of the customer (Categorical).
19. AGE: Age of the customer (Numerical).
20. TENURE: Number of years the customer has been with the insurance company (Numerical).
21. EMPLOYMENT\_STATUS: Employment status of the customer (Categorical).
22. NO\_OF\_FAMILY\_MEMBERS: Number of family members (Numerical).
23. RISK\_SEGMENTATION: Risk category assigned to the customer (Categorical).
24. HOUSE\_TYPE: Type of house the customer owns or rents (Categorical).
25. SOCIAL\_CLASS: Social class of the customer (Categorical).
26. ROUTING\_NUMBER: Customer’s bank routing number (Categorical).
27. ACCT\_NUMBER: Customer’s bank account number (Categorical).
28. CUSTOMER\_EDUCATION\_LEVEL: Education level of the customer (Categorical).
29. CLAIM\_STATUS: Status of the claim (Categorical).
30. INCIDENT\_SEVERITY: Severity of the incident (Categorical).
31. AUTHORITY\_CONTACTED: Whether the authorities were contacted (Boolean).
32. ANY\_INJURY: Whether any injury was reported (Boolean).
33. POLICE\_REPORT\_AVAILABLE: Whether a police report is available (Boolean).
34. INCIDENT\_STATE: State where the incident occurred (Categorical).
35. INCIDENT\_CITY: City where the incident occurred (Categorical).
36. INCIDENT\_HOUR\_OF\_THE\_DAY: Hour of the day when the incident occurred (Numerical).
37. AGENT\_ID: Unique identifier for the insurance agent (Categorical).
38. VENDOR\_ID: Unique identifier for the vendor (Categorical).

Raw Dataset 3:



1. ID: Unique identifier for the customer (Categorical).
2. KIDSDRIV: Number of kids driving in the household (Numerical).
3. BIRTH: Birth date of the customer (Date).
4. AGE: Age of the customer (Numerical).
5. HOMEKIDS: Number of kids living in the household (Numerical).
6. YOJ: Years on the job (Numerical).
7. INCOME: Income of the customer (Numerical, Currency).
8. PARENT1: Whether the customer is a single parent (Categorical).
9. HOME\_VAL: Value of the home (Numerical, Currency).
10. MSTATUS: Marital status (Categorical).
11. GENDER: Gender of the customer (Categorical).
12. EDUCATION: Education level of the customer (Categorical).
13. OCCUPATION: Occupation of the customer (Categorical).
14. TRAVTIME: Travel time to work (Numerical).
15. CAR\_USE: Whether the car is used for private or commercial purposes (Categorical).
16. BLUEBOOK: Value of the car (Numerical, Currency).
17. TIF: Time in force of the insurance policy (Numerical).
18. CAR\_TYPE: Type of car (Categorical).
19. RED\_CAR: Whether the car is red (Boolean).
20. OLDCLAIM: Amount claimed in prior incidents (Numerical, Currency).
21. CLM\_FREQ: Frequency of claims (Numerical).
22. REVOKED: Whether the driver's licence has been revoked (Boolean).
23. MVR\_PTS: Motor Vehicle Record points (Numerical).
24. CLM\_AMT: Amount claimed in the current incident (Numerical, Currency).
25. CAR\_AGE: Age of the car (Numerical).
26. CLAIM\_FLAG: Whether a claim was filed (Boolean).
27. URBANICITY: Urban/rural classification (Categorical).

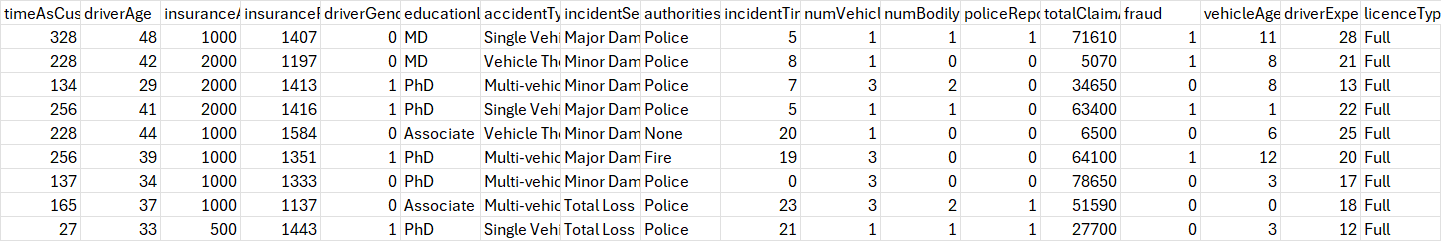
Datasets 1 & 2

After the cleaning and enrichment process, Dataset 1 (1,000 rows) and Dataset 2 (approximately 1,500 rows) were refined by selecting 16 key columns and generating 2 additional columns using the original data. These newly created columns, Driver Experience and Licence Type, were calculated using straightforward mathematical formulas.

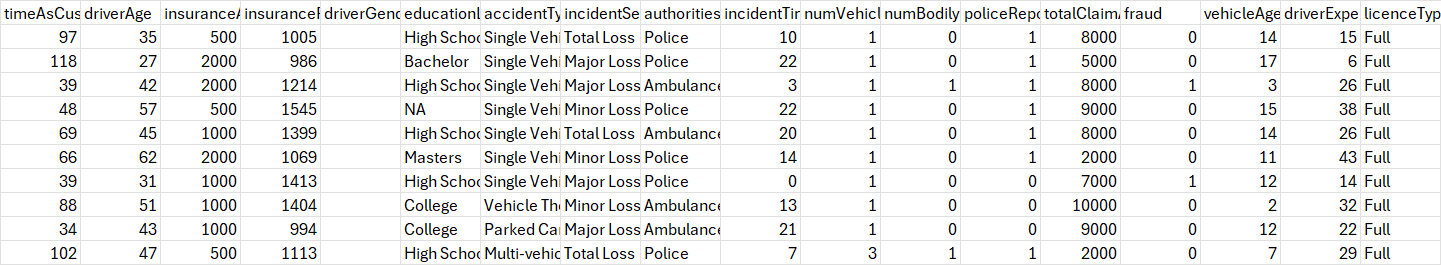
* Driver Experience: This column is derived from the driver's age, using the formula:  
  driver experience = driver age - 16 + R,  
  where R is a random variable between 0 and 6. The subtraction of 16 accounts for the minimum age to obtain a license, while the random variable represents the variability in the time drivers might wait before getting their license.
* Licence Type: The licence type is determined by a simple set of conditional statements based on the driver’s experience:
  + Less than 1 year of experience: Ls
  + 1 to 2 years of experience: P1
  + 2 to 4 years of experience: P2
  + More than 4 years of experience: Full

For Dataset 2, no gender data was available, so the driverGender column remains blank for that dataset, ensuring consistency without introducing any erroneous or bias inducing data.

Cleaned & Enriched Dataset 1:



1. timeAsCustomer: Number of months the individual has been a customer (Numerical).
2. driverAge: Age of the driver (Numerical).
3. insuranceAccess: The level of access to insurance (Numerical).
4. insurancePremium: The premium amount for the insurance policy (Numerical).
5. driverGender: Gender of the driver (Boolean: 0 = Male, 1 = Female).
6. educationLevel: Education level of the driver (Categorical).
7. accidentType: Type of accident (Categorical).
8. incidentSeverity: Severity of the incident (Categorical).
9. authoritiesInvolved: Whether authorities were involved (Categorical).
10. incidentTime: Time of the incident (Numerical).
11. numVehiclesInvolved: Number of vehicles involved in the accident (Numerical).
12. numBodilyInjuries: Number of bodily injuries (Numerical).
13. policeReportBool: Whether a police report was available (Boolean: 0 = No, 1 = Yes).
14. totalClaimAmount: The total claim amount (Numerical).
15. fraud: Whether the claim was fraudulent (Boolean: 0 = No, 1 = Yes).
16. vehicleAge: Age of the vehicle (Numerical).
17. driverExperience: Number of years of driving experience (Numerical).
18. licenceType: Type of licence held by the driver (Categorical).

Cleaned & Enriched Dataset 2:  


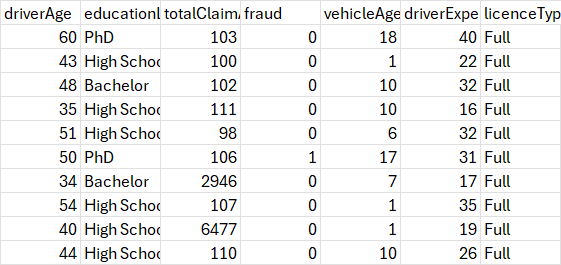
1. timeAsCustomer: Number of months the individual has been a customer (Numerical).
2. driverAge: Age of the driver (Numerical).
3. insuranceAccess: The level of access to insurance (Numerical).
4. insurancePremium: The premium amount for the insurance policy (Numerical).
5. driverGender: Gender of the driver (Boolean: 0 = Male, 1 = Female, empty & not populated).
6. educationLevel: Education level of the driver (Categorical).
7. accidentType: Type of accident (Categorical).
8. incidentSeverity: Severity of the incident (Categorical).
9. authoritiesInvolved: Whether authorities were involved (Categorical).
10. incidentTime: Time of the incident (Numerical).
11. numVehiclesInvolved: Number of vehicles involved in the accident (Numerical).
12. numBodilyInjuries: Number of bodily injuries (Numerical).
13. policeReportBool: Whether a police report was available (Boolean: 0 = No, 1 = Yes).
14. totalClaimAmount: The total claim amount (Numerical).
15. fraud: Whether the claim was fraudulent (Boolean: 0 = No, 1 = Yes).
16. vehicleAge: Age of the vehicle (Numerical).
17. driverExperience: Number of years of driving experience (Numerical).
18. licenceType: Type of licence held by the driver (Categorical).

Dataset 3

Dataset 3, which is the largest dataset (10000) rows, was missing many of the columns present in the other datasets. Instead of synthesising all the columns it was missing, it was decided to just get the columns that match those and leave the rest empty. This was to prevent such a large dataset becoming biased and skewed from data from smaller datasets. It increased variation and variety of data, and would assist in helping the AI model learn what to do if not all rows of data were provided.

Thus, Dataset 3’s selected columns are (included LicenceType and DriverExperience which were enriched using same formula as previous datasets):

Cleaned & Enriched Dataset 3:



1. driverAge: Age of the driver (Numerical).
2. educationLevel: Education level of the driver (Categorical).
3. totalClaimAmount: The total claim amount (Numerical).
4. fraud: Whether the claim was fraudulent (Boolean: 0 = No, 1 = Yes).
5. vehicleAge: Age of the vehicle (Numerical).
6. driverExperience: Number of years of driving experience (Numerical).
7. licenceType: Type of licence held by the driver (Categorical).

Generating Data:

Data being generated is 4000 rows of synthetic data from generation method 1 and 4000 rows of synthetic data from generation method 2.

Method 1:

Synthetic Data Method 1 generates additional data by sampling from the original dataset, maintaining the same distribution and characteristics of the real data. This method is used to expand the dataset, especially when more data is needed for training machine learning models, but without introducing random or irrelevant data.

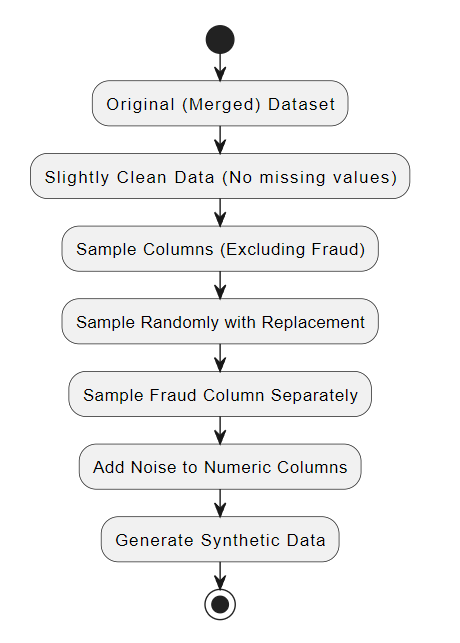
The synthetic data is generated by first cleaning any missing or blank values (such as replacing blanks in authoritiesInvolved with "none"). Then, we sample the original dataset to create new rows. The important point here is that the fraud column is kept reliable and unchanged. While other columns are sampled and noise is added to numeric fields to make the data varied, the fraud column is handled separately to ensure that its distribution remains consistent with the original data.

For the fraud column, we directly sample from the original fraud values and then add them to the synthetic data without any modification or randomization. This ensures that the synthetic data reflects the same patterns of fraudulent and non-fraudulent cases found in the original dataset, maintaining its integrity and reliability.

By preserving the fraud column’s original structure and distribution, this method ensures that synthetic data can be confidently used for analysis or machine learning without introducing artificial patterns into the fraud labels.

Steps:

1. Original Dataset: The raw data with columns like driverAge, accidentType, fraud, etc.
2. Clean Data: Any missing values (like in authoritiesInvolved) are replaced with "none".
3. Sample Columns (Excluding Fraud): Randomly sample all columns except fraud.
4. Sample Fraud Column Separately: The fraud column is sampled independently.
5. Add Noise to Numeric Columns: Add small noise to numeric columns to vary values.
6. Generate Synthetic Data: Final synthetic dataset output.



Method 2:

Synthetic Data Method 2 uses a more advanced approach compared to Method 1. It leverages a generative model called CTGAN (Conditional Tabular GAN), which is designed to generate highly realistic synthetic data by learning the underlying patterns and relationships in the original dataset. This method is particularly effective when dealing with complex datasets that include both categorical and continuous variables.

The process begins by loading the original dataset and defining its metadata. The CTGANSynthesizer is then used to train a model on the original data. This model learns the data's distribution and structure over several epochs (1,800 in this case), ensuring that the synthetic data it generates closely mimics the patterns in the original dataset.

One of the key strengths of Method 2 is that it not only captures basic distributions, but it also models the relationships between different columns. This means that the generated data retains important dependencies between variables, making it more robust for tasks like machine learning or data analysis.

Fraud Column Reliability:

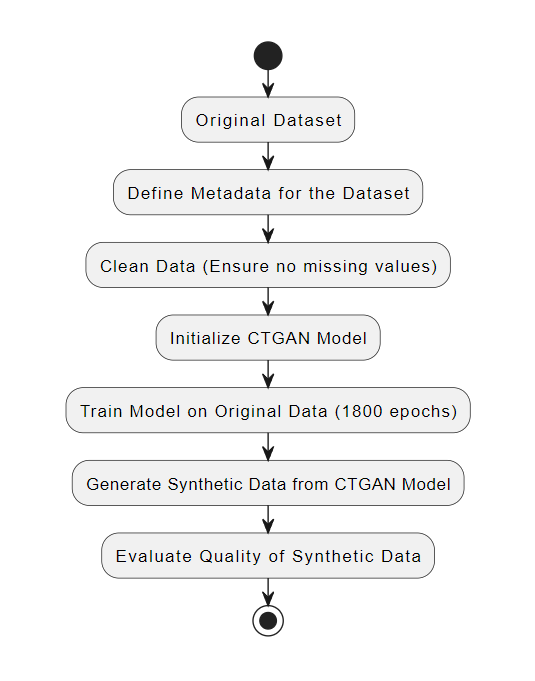
Similar to Method 1, the fraud column in Method 2 is kept reliable. Since the CTGAN model learns from the actual data distribution, it maintains the integrity of the fraud column and generates fraud labels in line with the original dataset. Unlike random sampling, this method ensures that the synthetic data generated by the GAN reflects the real-world proportions and relationships between fraudulent and non-fraudulent cases.

Once the synthetic data is generated, it is evaluated for quality using a tool that compares it to the real data. This ensures that the synthetic data is both accurate and usable for further analysis. Finally, the data is saved, with the index reset to start from 1, ensuring a clean output.

In summary, Synthetic Data Method 2 provides an advanced and reliable approach for generating synthetic data by using CTGAN. It captures the complex relationships in the dataset while ensuring that the fraud column remains accurate and trustworthy.

Steps:

1. Original Dataset: Load the original dataset with columns such as driverAge, accidentType, and fraud.
2. Define Metadata: The structure and relationships of the columns are defined in the metadata, which helps guide the model's learning.
3. Clean Data: Ensure that missing values are handled, for example, replacing empty values in authoritiesInvolved with "none".
4. Initialize CTGAN Model: The CTGAN model is initialised with hyperparameters such as the number of training epochs.
5. Train Model on Original Data: The model is trained on the original dataset, learning the underlying patterns and relationships.
6. Generate Synthetic Data: After training, the model generates new synthetic data that mirrors the original data in terms of structure and distribution.
7. Evaluate Quality: The generated synthetic data is evaluated against the original data to ensure quality and reliability, particularly for sensitive columns like fraud.



Merging Datasets:

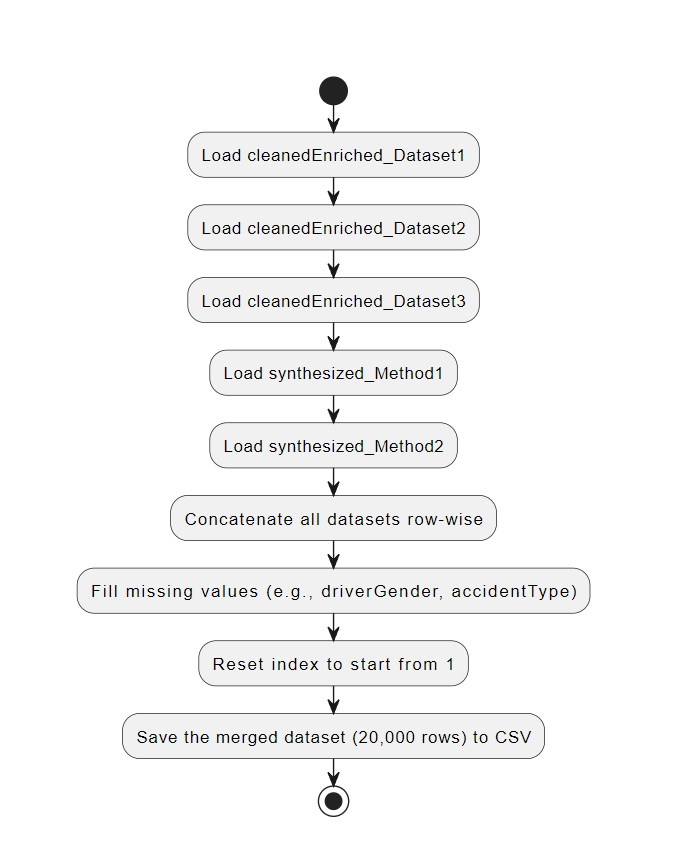
The dataset merging process involves combining three cleaned and enriched datasets with two synthesised datasets into a single dataset containing ~ 20,500 rows. This process ensures the data is consolidated, structured, and ready for further analysis or machine learning tasks.

Steps in the Dataset Merger:

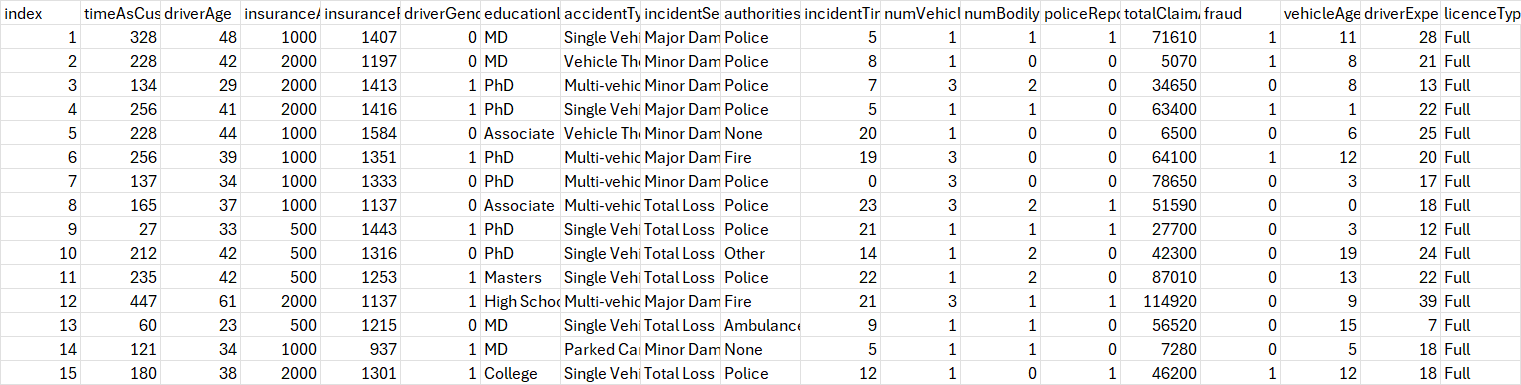
1. Loading the Datasets:
   * Three cleaned and enriched datasets (cleanedEnriched\_Dataset1, cleanedEnriched\_Dataset2, and cleanedEnriched\_Dataset3) are loaded. These datasets have undergone data cleaning and enrichment processes to ensure completeness and quality.
   * Two synthesized datasets (Synthesized\_Method1 and Synthesized\_Method2) are also loaded, representing synthetic data generated through advanced methods.
2. Concatenating the Datasets:
   * The datasets are concatenated row-wise, combining rows from all five datasets into one. This ensures that the merged dataset includes a complete set of data points from all the cleaned, enriched, and synthesised sources.
3. Filling Missing Values:
   * Columns such as driverGender, accidentType, incidentSeverity, and authoritiesInvolved are checked for missing or blank values. Any blanks are filled with "none" to ensure that the dataset is fully populated. This is critical for ensuring the integrity of the dataset when performing further analysis.
4. Resetting the Index:
   * A new index is generated for the merged dataset, starting from 1. This provides a unique identifier for each row in the dataset, ensuring that it is well-structured and organised.
5. Saving the Merged Dataset:
   * The final merged dataset, containing 20,000 rows, is saved to a CSV file. This dataset is now ready for further use, such as in machine learning models or data analysis.

Steps in the Diagram:

1. Load cleanedEnriched\_Dataset1: The first cleaned and enriched dataset is loaded.
2. Load cleanedEnriched\_Dataset2: The second cleaned and enriched dataset is loaded.
3. Load cleanedEnriched\_Dataset3: The third cleaned and enriched dataset is loaded.
4. Load synthesized\_Method1: The first synthesised dataset is loaded.
5. Load synthesized\_Method2: The second synthesised dataset is loaded.
6. Concatenate Datasets: The datasets are merged row-wise, appending rows from all datasets into one.
7. Fill Missing Values: Missing values, particularly in columns like driverGender, are filled with "none".
8. Reset Index: A new index is generated, starting from 1.
9. Save the Merged Dataset: The merged dataset, containing 20,000 rows, is saved to a CSV file for further use.

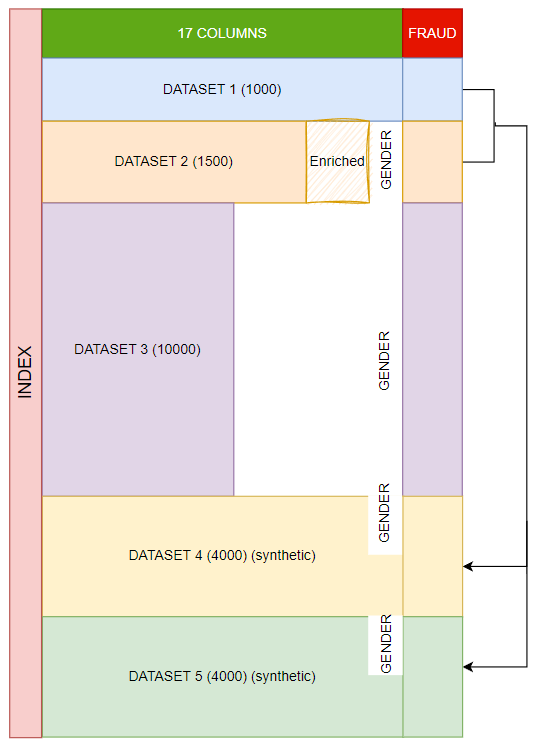


Merged Dataset (~20500 rows):



1. index: Row index for tracking purposes (Numerical).
2. timeAsCustomer: Number of months the individual has been a customer (Numerical).
3. driverAge: Age of the driver (Numerical).
4. insuranceAccess: The level of access to insurance (Numerical).
5. insurancePremium: The premium amount for the insurance policy (Numerical).
6. driverGender: Gender of the driver (Boolean: 0 = Male, 1 = Female).
7. educationLevel: Education level of the driver (Categorical).
8. accidentType: Type of accident (Categorical).
9. incidentSeverity: Severity of the incident (Categorical).
10. authoritiesInvolved: Whether authorities were involved (Categorical).
11. incidentTime: Time of the incident (Numerical).
12. numVehiclesInvolved: Number of vehicles involved in the accident (Numerical).
13. numBodilyInjuries: Number of bodily injuries (Numerical).
14. policeReportBool: Whether a police report was available (Boolean: 0 = No, 1 = Yes).
15. totalClaimAmount: The total claim amount (Numerical).
16. fraud: Whether fraud was reported (Boolean: 0 = No, 1 = Yes).
17. vehicleAge: Age of the vehicle (Numerical).
18. driverExperience: Number of years of driving experience (Numerical).
19. licenceType: Type of licence held by the driver (Categorical).

Structure of resulting dataset after merging:



# **TEST SPECIFICATION**

## **Model evaluation: - Tash**

· how will any data / outputs be compared / tested / evaluated for correctness and accuracy?

· if you are choosing between models, how will the models be compared / contrasted to see which one has a better performance (e.g. if you are comparing different classifier models, on what basis are you comparing them? Detail each comparison.

-

-

## **Performance evaluation results: - Adam/Noor**

what are the results of any tests run so far and what are the future planned tests for future iterations? (e.g. if this is Deliverable 3, what is planned for Deliverable 4? IF this is Deliverable 4, what is planned before the handover?)

Refer to the rubric / marking scheme (“All Teams: Deliverable 3 and 4 marking schemes”) for sections to be included in the design and testing documentation.

It is possible to know that the synthesising process is reliable and accurate because both synthesising methods and processes were different and carried out independently. However, their output results were similar, indicating that both need to have been correct to have gotten similar outputs. I.e. it couldn’t have been a coincidence.

An indicator of their similar output is the occurrence of fraud in the fraud column. 466 occurrences of fraud in method 1, and 417 occurrences in method 2. The total rows synthesised by each method is 4000, so in 4000 rows being this close to each other certainly means that both methods of synthesising are valid and reliable.

