******

***COMP3850 Project Deliverable Certificate***

| **Name of Deliverable** | *Deliverable 4* |
| --- | --- |
| **Date Submitted** | *17/ 10 / 2024* |
| **Project Group Number** | *14* |
| **Rubric stream being followed for this deliverable (highlight one)**  ***Note: the feasibility study has the same rubric for all streams.*** | *SOFTWARE Rubric*  *GAMES Rubric*  *CYBERSECURITY Rubric*  *DATA SCIENCE Rubric* |

We, the undersigned members of the above Project Group, collectively and individually certify that the above Project Deliverable, as submitted, **is entirely our own work**, other than where explicitly indicated in the deliverable documentation.

| **INITIALS** | **SURNAME** | **GIVEN NAME** | **STUDENT NUMBER** | **SIGNATURE  *(IN-PERSON OR DIGITAL)*** |
| --- | --- | --- | --- | --- |
| NM | Mahagoda | Ninuri | 46592156 | *NinuriM* |
| QN | Nguyen | Quoc Hung (Alan) | 47578130 | *AlanN* |
| AC | Chowdhury | Aasnayem Gazzali (Adam) | 46325824 | *Aasnayem* |
| NK | Khan | Noorullah | 47197404 | *N Khan* |
| TV | Vilathgamuwa | Tashiya | 46992162 | *Tvilathg* |

*NB: please write all details clearly (if handwritten).*

*© Macquarie University, 2021*

**List of tasks completed for the deliverable and activities since last deliverable certificate with totals for each individual team member and whole team** *(copy individual total row for each member and copy pages if more pages needed)*

| **Performed by  *(Names)*** | **Duration *(hrs)*** | **Complexity  *(L, M, H)*** | **Name of task** | **Checked by  *(Initials)*** |
| --- | --- | --- | --- | --- |
| Noorullah Khan | 1 | High | Meeting: With EY Team & Clients (01/10) | TV |
| Noorullah Khan | 3 | High | Meeting: With team after EY (01/10) | TV |
| Noorullah Khan | 0.25 | Mid | Meeting: With Alan & AI Team (03/10) | TV |
| Noorullah Khan | 0.25 | Mid | Meeting: With Alan about splitting and Json (03/10) | AC |
| Noorullah Khan | 0.5 | Mid | Project: Working on PM tool and configs (03/10) | AC |
| Noorullah Khan | 0.25 | Mid | Meeting: With AI Team "Combined Team Meaning" (07/10) | TV |
| Noorullah Khan | 0.25 | High | Meeting: With Adam and Alan desiging final mvp structure ideas (07/10) | TV |
| Noorullah Khan | 1 | High | Meeting: with Adam fixing Data Cleaning Notes Document for Mentor (07/10) | AC |
| Noorullah Khan | 1 | Mid | Meeting: Pre EY meeting (07/10) | TV |
| Noorullah Khan | 1 | High | Meeting: with client EY (08/10) | TV |
| Noorullah Khan | 3 | High | Meeting: with team & AI team after EY (08/10) | TV |
| Noorullah Khan | 1 | High | Development: Trying to Install Next.js and node.js (08/10) | AC |
| Noorullah Khan | 1 | High | Development: Working on Next.js and node.js (09/10) | AC |
| Noorullah Khan | 4 | High | Development: Building website (09/10) | AC |
| Noorullah Khan | 2 | High | Development: Further Webstie Development, Showcasing & Testing (09/10) | AC |
| Noorullah Khan | 1 | High | Research: Future development paths for website (10/10) | QN |
| Noorullah Khan | 0.5 | Mid | Meeting: with data team and Dan after Uni (10/10) | QN |
| Noorullah Khan | 1 | Mid | Research: Prinicples & Test how to integrate websites with Azure AI API (11/10) | QN |
| Noorullah Khan | 1 | Mid | Project: Deliverable 4 making website section (12/10) | NM |
| Noorullah Khan | 1 | Mid | Project: Deliverable 4 work (12/10) | NM |
| Noorullah Khan | 0.75 | Mid | Meeting: Saturday Team Meeting (12/10) | TV |
| Noorullah Khan | 1 | High | Project: updating PM tool (12/10) | QN |
| Noorullah Khan | 0.25 | Mid | Project: updating GitHub Projects (12/10) | QN |
| Noorullah Khan | 0.5 | Mid | Development: Adding/Updating CLaim Descriptions (12/10) | AC |
| Noorullah Khan | 1 | Mid | Meeting: Pre EY meeting (14/10) | AC |
| Noorullah Khan | 1 | High | Meeting: With EY Team & Clients (15/10) | TV |
| Noorullah Khan | 3 | High | Meeting: With team after EY (15/10) | TV |
| Noorullah Khan | 0.5 | High | Project: Re-configiuring GitHub configuration & sharing claims.csv with AI Team (Mingxin) (16/10) | TV |
| Noorullah Khan | 1 | High | Development: Adding fraudChance, fraudAnalysis & claimNotes columns to claims.csv (16/10) | AC |
| Noorullah Khan | 3 | High | Development: Updating website functionality (16/10) | AC |
| Noorullah Khan | 3 | High | Development: WebApp functionality login, logout, notes (16/10) | AC |
| Noorullah Khan | 1 | Mid | Development: Reviewing WebApp code & scripts (17/10) | AC |
| Noorullah Khan | 2 | High | Research: Attended MQ AI Conference and learnt more ways to approach fraud detection AI (17/10) | AC |
| Noorullah Khan | 1 | High | Research: Attended MQ AI Conference and researched about Anomaly detection algorithms (17/10) | AC |
| Noorullah Khan | 0.25 | Mid | Project: Deliverable 4 Updating | NM |
| Noorullah Khan | 0.5 | Mid | Project: Reviewing & slightly editing Deliverable 4 New Content | NM |
| NInuri Mahagoda | 1 | High | Meeting: With EY Team & Clients (01/10) | NM |
| NInuri Mahagoda | 3 | High | Meeting: With team after EY (01/10) | TV |
| NInuri Mahagoda | 1 | Mid | Research: looked into NLTK library (05/10) | TV |
| NInuri Mahagoda | 0.5 | Mid | Research: thought of MVP ideas for Del 4 (06/10) | TV |
| NInuri Mahagoda | 0.25 | Mid | Meeting: With AI Team "Combined Team Meaning" (07/10) | QN |
| NInuri Mahagoda | 1 | Mid | Meeting: Pre EY meeting (07/10) | QN |
| NInuri Mahagoda | 1 | Mid | EY: editing stand-up slides (07/10) | TV |
| NInuri Mahagoda | 1 | High | Meeting: with client EY (08/10) | QN |
| NInuri Mahagoda | 3 | High | Meeting: with team & ai team after EY (08/10) | QN |
| NInuri Mahagoda | 0.5 | Mid | Meeting: with data team and Dan after Uni (10/10) | AC |
| NInuri Mahagoda | 0.75 | Mid | Meeting: Saturday Team Meeting (12/10) | AC |
| NInuri Mahagoda | 2 | Mid | Deliverable 4: worked on troubleshooting and end-user sections with Task (13/10) | TV |
| NInuri Mahagoda | 1 | Mid | Meeting: Pre EY meeting (14/10) | AC |
| NInuri Mahagoda | 0.5 | Low | EY: editing stand-up slides (14/10) | NK |
| NInuri Mahagoda | 1 | Mid | Deliverable 4: edited the draft document with Tash (14/10) | TV |
| NInuri Mahagoda | 1 | High | Meeting: With EY Team & Clients (15/10) | NK |
| NInuri Mahagoda | 3 | High | Meeting: With team after EY (15/10) | NK |
| NInuri Mahagoda | 1.5 | Mid | Deliverable 4: ran through updates for Deliverable 3 Document with Tash (15/10) | TV |
| NInuri Mahagoda | 1.5 | Mid | Deliverable 4: reviewed mentor feedback for Deliverable 4 new section with Tash (15/10) | TV |
| NInuri Mahagoda | 3 | Mid | Deliverable 4: Updated deliverable 3 for deliverable 4 (16/10) | TV |
| NInuri Mahagoda | 2 | Mid | Deliverable 4: created a user manuals for end users and FAQ section (16/10) | TV |
| NInuri Mahagoda | 2 | Mid | Deliverable 4: reviewing other memmers content and adding notes/comments (16/10) | TV |
| NInuri Mahagoda | 4 | Mid | Deliverable 4: formatted and edited with Tash (17/10) | TV |
| Tashiya Vilathgamuwa | 1 | High | Meeting: With EY Team & Clients (01/10) | NK |
| Tashiya Vilathgamuwa | 1 | Mid | Research: Looked into NLTK library (05/10) | NM |
| Tashiya Vilathgamuwa | 3 | High | Meeting: With team after EY (01/10) | NK |
| Tashiya Vilathgamuwa | 1 | Mid | Meeting: Pre EY meeting (07/10) | NK |
| Tashiya Vilathgamuwa | 1 | Mid | EY: Editing Standup slides (07/10) | NM |
| Tashiya Vilathgamuwa | 1 | High | Meeting: with client EY (08/10) | NK |
| Tashiya Vilathgamuwa | 3 | High | Meeting: with team & ai team after EY (08/10) | NK |
| Tashiya Vilathgamuwa | 0.5 | Mid | Meeting: with data team and Dan after Uni (10/10) | AC |
| Tashiya Vilathgamuwa | 0.75 | Mid | Meeting: Saturday Team Meeting (12/10) | AC |
| Tashiya Vilathgamuwa | 2 | Mid | Deliverable 4: Worked on Troubleshooting and User sections with Ninuri (13/10) | NM |
| Tashiya Vilathgamuwa | 1 | Mid | Meeting: Pre EY meeting (14/10) | QN |
| Tashiya Vilathgamuwa | 0.5 | Low | EY: Editing Standup slides (14/10) | NM |
| Tashiya Vilathgamuwa | 1 | Mid | Deliverable 4: edited the draft document with Ninuri (14/10) | NM |
| Tashiya Vilathgamuwa | 1 | High | Meeting: With EY Team & Clients (15/10) | AC |
| Tashiya Vilathgamuwa | 3 | High | Meeting: With team after EY (15/10) | AC |
| Tashiya Vilathgamuwa | 1.5 | Mid | Deliverable 4: Ran through Updates for deliverable 3 document with Ninuri (15/10) | NM |
| Tashiya Vilathgamuwa | 1.5 | Mid | Deliverable 4: Reviewed the mentor feedback for deliverable 4 new section with Ninuri (15/10) | NM |
| Tashiya Vilathgamuwa | 3 | Mid | Deliverable 4: Updated edited and added new sections to the Deliverable 4 (16/10) | NM |
| Tashiya Vilathgamuwa | 2 | Mid | Deliverable 4: contuned editing and updating overall Delikverable 4 document (16/10) | NM |
| Tashiya Vilathgamuwa | 2 | Mid | Deliverable 4: Created a user manual for end users and FAQ section (16/10) | NM |
| Tashiya Vilathgamuwa | 4 | Mid | Deliverable 4: formatted and edited with Ninuri (17/10) | NM |
| Quoc Hung (Alan) Nguyen | 1 | High | Meeting: With EY Team & Clients (01/10) | TV |
| Quoc Hung (Alan) Nguyen | 3 | High | Meeting: With team after EY (01/10) | TV |
| Quoc Hung (Alan) Nguyen | 0.5 | Mid | Meeting: With Noor & AI Team (03/10) | NK |
| Quoc Hung (Alan) Nguyen | 0.25 | Mid | Meeting: With Noor about splitting and Json (03/10) | NK |
| Quoc Hung (Alan) Nguyen | 0.25 | High | Meeting: With Noor and Adam desiging final mvp structure ideas (07/10) | AC |
| Quoc Hung (Alan) Nguyen | 1 | Mid | Meeting: Pre EY meeting (07/10) | NM |
| Quoc Hung (Alan) Nguyen | 1 | High | Meeting: with client EY (08/10) | NM |
| Quoc Hung (Alan) Nguyen | 3 | High | Meeting: with team & ai team after EY (08/10) | NM |
| Quoc Hung (Alan) Nguyen | 0.5 | Mid | Meeting: with data team and Dan after Uni (10/10) | NM |
| Quoc Hung (Alan) Nguyen | 0.75 | Mid | Meeting: Saturday Team Meeting (12/10) | NK |
| Quoc Hung (Alan) Nguyen | 0.5 | Mid | Meeting: with Adam to discuss about Deliverable 4 | NK |
| Quoc Hung (Alan) Nguyen | 1 | Mid | Meeting: Pre EY meeting (14/10) | NK |
| Quoc Hung (Alan) Nguyen | 1 | High | Meeting: With EY Team & Clients (15/10) | TV |
| Quoc Hung (Alan) Nguyen | 1.5 | High | Meeting: With team after EY (15/10) | TV |
| Quoc Hung (Alan) Nguyen | 0.5 | Mid | Meeting: With Adam to discuss about SQL database | AC |
| Quoc Hung (Alan) Nguyen | 1.5 | Mid | Development: Pull request review for ticket 43 | Configure github actions | AC |
| Quoc Hung (Alan) Nguyen | 2 | High | Development: Transform data to json ticket 58 | AC |
| Quoc Hung (Alan) Nguyen | 0.5 | High | Development: Pull request review for ticket 58 | Transform data to json | AC |
| Quoc Hung (Alan) Nguyen | 2 | High | Development: Pull request review for ticket 69 | Review and remove redundant files | AC |
| Quoc Hung (Alan) Nguyen | 1.5 | High | Development: Pull request review for ticket 66 | Terraform | NK |
| Quoc Hung (Alan) Nguyen | 2 | High | Development: Pull request review for ticket 34 | Terraform | NK |
| Quoc Hung (Alan) Nguyen | 1.5 | Mid | Development: Pull request review for ticket 56 | Infrastructure re-configuration | NK |
| Quoc Hung (Alan) Nguyen | 1 | Mid | Deliverable 4: Updated old content | TV |
| Quoc Hung (Alan) Nguyen | 2 | Mid | Deliverable 4: New contents | NM |
| Aasnayem Gazzali Chowdhury(Adam) | 1 | High | Meeting: With EY Team & Clients (01/10) | NM |
| Aasnayem Gazzali Chowdhury(Adam) | 3 | High | Meeting: With team after EY (01/10) | NM |
| Aasnayem Gazzali Chowdhury(Adam) | 0.25 | High | Meeting: With Noor and Alan desiging final mvp structure ideas (07/10) | NK |
| Aasnayem Gazzali Chowdhury(Adam) | 1 | High | Meeting: with Noor fixing Data Cleaning Notes Document for Mentor (07/10) | NK |
| Aasnayem Gazzali Chowdhury(Adam) | 0.25 | Mid | Meeting: With AI Team "Combined Team Meaning" (07/10) | QN |
| Aasnayem Gazzali Chowdhury(Adam) | 1 | Mid | Meeting: Pre EY meeting (07/10) | AC |
| Aasnayem Gazzali Chowdhury(Adam) | 1 | High | Meeting: with client EY (08/10) | AC |
| Aasnayem Gazzali Chowdhury(Adam) | 3 | High | Meeting: with team & ai team after EY (08/10) | TV |
| Aasnayem Gazzali Chowdhury(Adam) | 1.5 | High | Development: Trying to install node.js and next.js | NK |
| Aasnayem Gazzali Chowdhury(Adam) | 0.75 | Mid | Development: Created a testing table | QN |
| Aasnayem Gazzali Chowdhury(Adam) | 0.5 | Mid | Meeting: with data team and Dan after Uni (10/10) | NK |
| Aasnayem Gazzali Chowdhury(Adam) | 3 | Mid | Deliverable 4: New content | NM |
| Aasnayem Gazzali Chowdhury(Adam) | 0.75 | Mid | Meeting: Saturday Team Meeting (12/10) | NM |
| Aasnayem Gazzali Chowdhury(Adam) | 0.5 | Mid | Meeting: with Alan to discuss about Deliverable 4 | QN |
| Aasnayem Gazzali Chowdhury(Adam) | 1 | Mid | Meeting: Pre EY meeting (14/10) | AC |
| Aasnayem Gazzali Chowdhury(Adam) | 1 | High | Meeting: With EY Team & Clients (15/10) | AC |
| Aasnayem Gazzali Chowdhury(Adam) | 3 | High | Meeting: With team after EY (15/10) | NM |
| Aasnayem Gazzali Chowdhury(Adam) | 0.5 | Mid | Meeting: with Alan to discuss about SQL database | NK |
| Aasnayem Gazzali Chowdhury(Adam) | 1 | Mid | Deliverable 4: Updated deliverable 3 | TV |
| Aasnayem Gazzali Chowdhury(Adam) | 2 | High | Development: Troubleshoot MySQL workbench | QN |
| Aasnayem Gazzali Chowdhury(Adam) | 2.5 | Mid | Deliverable 4: Updated deliverable 4 based on feeback | TV |
| Aasnayem Gazzali Chowdhury(Adam) | 0.5 | Mid | Deliverable 4: Fixed the structure | NM |
| Aasnayem Gazzali Chowdhury(Adam) | **1** | Mid | Deliverable 4: updating my content | NM |
| **Ninuri Total** | 35.5 |  |  |  |
| **Tashiya Total** | 34.75 |  |  |  |
| **Noorullah Total** | 43.75 |  |  |  |
| **Alan Total** | 29.75 |  |  |  |
| **Adam Total** | 30 |  |  |  |
| **TEAM TOTAL** | **173.75** |  |  |  |

# 

| **UPDATED PROJECT PLAN**  **Team 14**  Revolutionise Claims Management: Unleash the  Power of GenAI for Peak Efficiency in the Insurance  Sector |
| --- |

# **TABLE OF CONTENTS**

[**Revision History Table**](#_29u1pvk9hcpy) **12**

[**1. Introduction**](#_ey84htrmwg41) **14**

[1.1. Statement Of Purpose](#_qjhs061mrs6d) 14

[1.2. Scope](#_lkut0qrfjjxf) 14

[1.2.1 Justification](#_4uf4b0jgqo2y) 14

[1.2.2. Scope](#_rjyuljif6o41) 14

[1.2.3. Objectives](#_xyjk61ckia3q) 15

[1.2.4. Deliverables](#_54jjshzg3zw2) 15

[1.2.5. Exclusions (Out of Scope)](#_8v8472ht4ytg) 16

[1.2.6. Constraints](#_n265lwlww75s) 16

[1.2.7. Assumptions](#_6rg2mkgr1ak) 16

[**2. Risk Management**](#_lqjosrusuuua) **16**

[2.1 Risk Analysis Table](#_g57y68clqzhr) 16

[2.2 Risk Monitoring](#_la9wnt7cbt6d) 17

[2.3 Risk qualification and prioritisation](#_g1eo1ea9bjrs) 17

[**3. Resource Management**](#_2y3v9jdga453) **17**

[3.1 People:](#_44sxmq3i7ou8) 17

[3.2 Hardware:](#_hwh7gbdfqw4f) 18

[3.3 Software:](#_qob3qjqm8oue) 18

[3.4 Other Resources:](#_mbb3znk6zq6e) 20

[**4. Change Management**](#_cb5ra97yr809) **21**

[4.1 Changes to requirements and scope](#_xeiv48wsckg) 21

4[.2 Changes to documents, code and data](#_to5qqghujx3x) 22

4[.2.1 Branching strategy: Issue-based](#_jedd9268oajn) 23

4[.2.2 Code Reviews: Enforcing Quality with Pull Requests and Github Actions](#_t1t9k8yz6kdg) 23

4[.2.3 Documentation versioning](#_e7o84684gu1a) 23

4[.2.4 Immutable storage:](#_cwyfyno7l3dh) 24

[**5. Quality Management**](#_lekybt1055ne) **24**

[5.1. Objectives](#_h6i4fbli6icu) 24

[5.1.1. Ensuring data quality](#_tp7xfigdsi55) 24

[5.1.2. Model Accuracy](#_dwbw2mcsigel) 25

[5.1.3. Compliance with business requirements](#_gohx5xep14nk) 25

[5.1.4. Customer Focus and User Satisfaction](#_qtbo2vktu0pk) 25

[5.2. Data Quality Management: Monitoring and Assurance](#_d9qynmim2cp7) 26

[5.2.1. Data Collection Process](#_qy1g9rvf6euw) 26

[5.2.2. Exploratory Data Analysis (EDA)](#_6m2b1emo52ya) 26

5[.2.3. Data Cleaning](#_5zdknymy0jxw) 27

[5.2.4. Data Standardisation](#_wtehbtbxtwqy) 28

[5.3. Human Resource Management: Agile Delivery](#_fkxjyeyjsrks) 28

[5.3.1. Sprint Planning](#_71loxbsf5llm) 28

[5.3.2. Iteration Reviews](#_g28oahdueu17) 29

[5.3.3. Continuous Feedback Loop](#_etfw87v3wn8l) 30

[5.4. Development Management](#_8vusy3gogao5) 30

[5.4.1. Code convention](#_obln1aviwgdl) 30

[5.4.2. Prototype](#_fbalu9o460av) 30

[5.5 Website Management](#_71uh2ggvp57)…………………………………………………………………... .31

[5.6. Documentation Management](#_jmrubmu6lojh) 32

[5.6.1. Formatting](#_jv1w7xz3cttp) 32

[5.6.2. Documentation control officers](#_gvcm2nvszrgf) 33

[**6. Schedule**](#_entroc15wpqc) **33**

[6.1. Project Overview](#_hn2tdt7nve5) 33

[6.1.1. Deliverables](#_lrkiwbincapt) 34

[6.2. Project Phases and Key Milestones](#_5j8wvf4ignmv) 34

[6.3. Schedule - Gantt Chart](#_avcw20yzhg5a) 34

[6.4. Schedule Revisions](#_lyy0m6283qgf) 35

[6.5. Resource and Task Allocation](#_ftqw611qqx4g) 36

[6.6. Tools and Technologies](#_2ruauu2u53xh) 36

[6.6.1. Jira](#_7pkb895335j) 36

[6.6.2. GitHub](#_xekn66svh0kl) 36

6.6.3 Discord………………………………………………………………………….…37

[**7. Handover requirements**](#_s9jsgqm9645g) **37**

[7.1 Software and Tools](#_5jm05f50vsvk) 37

[7.2 Data Deliverables](#_hhnycii7yea5) 37

[7.3. Testing and Validation](#_nlw987petja9) 38

[7.4. Model Deliverables: (To be done by the AI Team)](#_jcin8mc98mc4) 38

[7.5 User and Training Manuals](#_cm5otmkttos5) 38

[7.6 Final Report and Evaluations](#_oiv7i7i1gd7t) 38

[7.7 Client Consultation and Acceptance:](#_9v1ioya3ssvh) 38

[7.8 Frequently Asked Questions(FAQs)](#_2y0ssad9lt0r)................................................................................39

[**8. Assumptions**](#_gxmzuce4l6nz) **39**

[8.1. Resources](#_x44v8bi9ibga) 39

[8.2. Availabilities](#_26yw3gq2l76c) 39

[8.3. Tools](#_tj2rlx15qgic) 39

[8.4. Techniques](#_rv41gm4hl33o) 40

[8.5. Standards 4](#_wdmc9n7b1giz)0

[8.6. Communication 4](#_qugceikd6nz1)0

[8.7. Expectations 4](#_m9p6ygrvpycn)1

[8.8. Design 4](#_5d0dgyqkzwra)1

[8.9. Documentation](#_ye5oery78wh1) 41

[8.10. Quality Management 4](#_4yso3u3wlou8)1

[8.11. Risk and Contingency 4](#_v037b9e766o0)1

[8.12 User Interface](#_nfct2wfdq4xq)…………………………………………………………………..……….41

[**9. References**](#_7w2zlxn99xlw) **43**

[**10. Appendices**](#_j35rc0g36jpe) **44**

[Appendix 1. Risk analysis table](#_qy7uoh9zx8ow) 44

[Appendix 2. Risk Indicators Table](#_tyldfwk9mmf4) 47

[Appendix 3 Probability/ Impact Matrix](#_u5cm5re7oncu) 48

[Appendix 4: Project Phases and Key Milestones 5](#_ktl5ycyng7bp)0

[Appendix 5: Sprint 1 Gantt Chart 5](#_xkx4usff5wy7)1

[Appendix 6: Sprint 2 Gantt Chart 5](#_rkrl6hlrdzkh)2

[Appendix 7: Sprint 3 Gantt Chart 5](#_1t7oxiho8kc1)3

[Appendix 8: Sprint 4 Gantt Chart](#_xkxdm5s4bjlb) 54

[Appendix 9: Sprint 5 Gantt Chart](#_gugdm1jztf3g) 55

[Appendix 10: Task Allocation Groups](#_e437qncfqmrd) 58

# **Revision History Table**

| **Version** | **Last Update** | **Author(s)** | **Changes** |
| --- | --- | --- | --- |
| V2\_S1  Introduction | 30/09/24 | Ninuri/  Tashiya | * Expanded on the scope to include information on how we will perform NRMA’s fraud detection in sections 1.2.2. And 1.2.3. |
| V3\_S1 | 16/10/24 | Ninuri / Tashiya | * Removed the point in ‘Exclusions’ - front end development of user interface for claims officer as we developed a front-end interface * Updated 1.2.2 and 1.2.3 to update that we are now developing a front end interface. * Removed 1.2.5 dot point addressing that front end interface is out of scope. |
| V2\_S2  Risk Management | 30/09/24 | Tashiya | * Added More technical and Business related risks on to the Risk analysis table, Risk indicators table and Probability/impact matrix |
| V3\_S2 | 16/10/24 | Tashiya | * Addes more technical risks on the risks analysis table and probability/ impact matrix due to the scope expanding to include a web app |
| V2\_S3  Resource Management | 30/09/24 | Tashiya | * Added the costs for each resource in sections 3.1., 3.2, and 3.4. * Added the new section in 3.3. called software costs |
| V3\_S3 | 16/10/24 | Tashiya | * Updated section 3.3 by adding two new resources under software also mentioning the costs for the new resources. |
| V1\_S4  Change Management | 28/08/24 | Tashiya | * No changes made |
| V2\_S4 | 16/10/2024 | Tashiya | * Updated Section 4.1 by adding to new points. |
| V2\_S5  Quality Management | 30/09/24 | Ninuri | * Defined quality metrics and KPIs for each aspect of managing quality * Added more data standardisation points in section 5.2.4. * Added tables to track the quality metrics * Added information on conducting exploratory data analysis (EDA) called section 5.2.2. * Added more documentation formatting points in section 5.5.1. |
| V2\_S5 | 16/10/24 | Ninuri | * Updated 5.4.2 table and added more rows to describe the expansion of scope * New section added 5.5 Web app management |
| V2\_S6  Schedule | 30/09/24 | Ninuri | * Updated the project schedule to cover changes in structure * Updated the resource allocation to add one for group in section 6.5. * Added a schedule revision table in section 6.4. * Added new project schedule images for completed and in progress sprints in appendix 7, 8 and 9. |
| V3\_S6 | 16/10/24 | Ninuri | * Updated section 6.1.1 by adding more product related deliverables * Updated 6.4 schedule revisions table with more rows * Updated 6.5 Resource task allocation * Added new section 6.6.3 “Discord” |
| V1\_S7  Handover requirements | 30/09/24 | Ninuri/  Tashiya | * Added a new section “Handover requirements” which outlines the methods and requirements we will have in place when handing over this project to our client. |
| V2\_S7 | 16/10/24 | Ninur/  Tashiya | * Added a new section 7.8 “FAQ” in handover requirements. |
| V2\_S8  Assumptions | 30/09/24 | Ninuri | * Added more assumptions on design, requirements, documentation and aspects of the project plan in sections 8.8., 8.9., 8.10,. and 8.11. |
| V3\_S8 | 16/10/24 | Ninuri | * Added a new section 6.12 “User Interface” |

# **Introduction**

## **1.1. Statement Of Purpose**

The purpose of this document is to showcase a comprehensive project plan on the formation of a fraud detection system for NRMA’s claims processing system using Generative AI (GenAI). To address this EY has assigned us to collaborate with Group 13 who have become the AI Team while we are the Data Team and tasked to develop an end-to-end solution upgrading their current fraud detection process. While we work collaboratively with the AI Team (G13), our main task is on different ends of the project as Group 13 build the AI model while we (G14) create a high quality dataset that will be used to train the model. The project plan was created to communicate with the stakeholders on the goals, objectives and tasks needed to complete the project throughout all phases of the project lifecycle.

The Project plan will include:

1. **Risk Management:** Managing risks effectively is crucial for the project's success. This risk management plan will outline how to identify, evaluate and address potential risks at every stage of the project.
2. **Resource Management:** Resource management aims to maximise the efficient use of people, hardware, software, and other resources. This part of the project plan will detail strategies for optimal resource allocation ensuring that the project will have enough resources to meet its objectives while adhering to budget and time limits.
3. **Change Management:** Changes to requirements, scope and updates to documents, code and data will be managed through a well defined process. This section of the project plan will delve deeper into how it will be managed and reviewed systematically.
4. **Quality Management:** Effective quality management is pivotal for the success of this project. The quality management plan outlines the approach to ensuring that all project deliverables meet the established standards and stakeholder expectations.The plan encompasses quality planning, assurance, and control processes designed to guarantee that the project adheres to high standards of quality throughout its lifecycle.
5. **Schedule:** The schedule section will follow the project overview, deliverables, task and resource allocation.
6. **Assumptions:** In this section, any assumptions made during the planning and execution phase of the project that may impact the project’s outcome will be raised.

## **1.2. Scope**

### **1.2.1 Justification**

This project was established as a way of improving NRMA’a current fraud detection system in automobile claim reports by leveraging GenAI as a means of increasing the accuracy and efficiency of the process. This helps create a robust tool for NRMA to streamline the claims process and decreases the amount of operational costs and time spent on reviewing claims.

### **1.2.2. Scope**

The scope of the project includes the development and provision of a dataset designed for training and evaluating a fraud detection model. This entails:

* **Data Acquisition:** Collecting and sourcing relevant automobile claim information
* **Data Preparation:** Cleaning, organising, and enhancing the dataset to ensure it is ready for model training.
* **Dataset Delivery:** Delivering the completed dataset to the AI team, along with necessary documentation and user guides.
* **Fraud Detection Model:** Coordinating with the AI team to create a GenAI model that can effectively detect fraud through machine learning algorithms such as anomaly detection and neural networks.
* **Model Training and Testing:** Collaborating with the AI team to train their model with the provided dataset and evaluating the model’s performance through testing to improve its capabilities.
* **[NEW] User Interface:** Collaborating with the AI team to create an interactive user interface to be used by the claims agents to check for fraudulent claims.
* **[NEW] Deployment on Azure:** Upload all our scripts and datasets onto Azure so it is accessible and scalable for NRMA after the handover

### **1.2.3. Objectives**

* To develop a clean, accurate and comprehensive dataset for the AI model to be trained.
* To improve NRMA’s existing fraud detection system by supplying high quality data for training and evaluating models.
* **[NEW]** To create an interactive user interface that uses GenAI to review insurance claims.
* **[NEW]:** To successfully deploy the fraud detection system on Azure Services as per business requirements.
* To complete the project within the set timeline and budget, ensuring adherence to the defined quality standards.
* To improve the current fraud detection process at NRMA using GenAI to reduce review time and operation costs.

### **1.2.4. Deliverables**

* **Sprint 1**: Create user stories and source datasets
* **Sprint 2**: Project Plan, Requirements Document, Options Analysis Report, Test Data File.
* **Sprint 3**: Design Document, Test Cases Document, Prototype/MVP.
* **Sprint 4**: Updated Prototype, User/Training Manual.
* **Sprint 5**: Final Group Reflective Report.
* **Sprint 6**: Final Product Dataset (Bronze, Silver, Gold Layers), Final Report.

### 

### **1.2.5. Exclusions (Out of Scope)**

* Development of the AI model itself (handled by AI team)
* Post deployment support or maintenance for the AI model or dataset
* Data visualisation (PowerBI, Tableau etc.)

### **1.2.6. Constraints**

* **Timeline**: The project must be finalised within the designated six sprints, with deadlines established for each phase.
* **Budget**: Adherence to the project budget is required, covering all expenses related to tools, resources and any unexpected costs.
* **Data availability:** The quality and comprehensiveness of the dataset depend on the accessibility and availability of relevant data sources.

### **1.2.7. Assumptions**

* The synthesiation techniques that have been chosen are able to synthesise 7,000 rows of high quality data
* Synthesised data will be of defined standard and will be of enough quality to train the AI model
* The Azure Tools, Azure Functions and Data Lake will integrate well with each other and be enough to complete the project

# **2**. **Risk Management**

Ensuring the completion of our project relies heavily on effective risk management practices in place from the start to finish of the project journey ahead of us (Smith,2020). Outlined in the sections are the key risks we've pinpointed within the project scope arranged by type alongside a thorough examination of their likelihoods and potential impacts and strategies geared toward mitigating these risks effectively. Our proactive approach, to identifying and tackling these pitfalls early on is geared towards reducing any possible interruptions and increasing the probability of reaching our project objectives successfully.

## **2.1 Risk Analysis Table**

Appendix 1 below outlines the different risks related with different aspects of the data science project ranging from technology and people related risks to organisational/ business related risks , requirements and estimation risks. Each risk is evaluated based on probability of occurrence and the impact it can have on the project. Moreover, appropriate mitigation strategies are proposed to manage these risks effectively. Refer to Appendix 1.

## **2.2 Risk Monitoring**

Our team has identified numerous indicators to help aiding us to monitor each type of risk. These indicators help us to constantly check whether a risk is becoming more likely. High priority risks are closely monitored, but low priority risks are also being continuously tracked to ensure they do not escalate unexpectedly. The indicators are summarised in the risk indicators table (Appendix 2).

## **2.3 Risk qualification and prioritisation**

To assess the seriousness of the risks highlighted in appendix 3 a combination of probability and impact factors has been integrated into the risk register to assist the project team in ranking risks according to their likelihood and potential impact on the project's outcomes. Project managers make use of a probability and impact matrix to aid in identifying risks that demand attention and action items during project planning and execution phases (Li, 2020). The matrix depicted in Appendix 3 assists the team in identifying the high priority risks marked in the red areas while categorising medium priority risks in orange zones and low priority risks, in green.

# **3. Resource Management**

To ensure the successful execution of our project, a well structured resource allocation and management strategy has been implemented (Armstrong, 2023). This strategy focuses on optimising the use of people, hardware, software and other resources while ensuring flexibility and responsiveness to project needs. [**CHANGES**] Additionally we’ve considered data and budget resources to ensure comprehensive management and readiness.

## **3.1 People:**

People resource management is important as it ensures that tasks are assigned to the most qualified individuals, optimises team performance and adapts to changes effectively.

* **Role Assignment:** Each team member has been assigned specific roles based on their expertise and strengths. The project manager (Noorullah) oversees the overall project , while specialised roles like data engineer, business analyst and cloud architect ensure that tasks are handled by the most qualified individuals.
* **Task Prioritisation:** Tasks are prioritised based on project milestones, with critical tasks assigned to team members with relevant expertise. This ensures that high impact tasks receive the attention they need for timely completion.
* **Cross training:** To mitigate risks associated with resource availability, cross training among team members is encouraged. For example, Business Analysts (Tashiya and Ninuri) are trained in python coding in order to assist Data engineers when needed.

**Management Strategy:**

* **Weekly meetings:** regular face-to-face and online meetings are held to review progress, reassign tasks if necessary and address any bottlenecks. This ensures that the team remains agile and can quickly adapt to changes.
* **Performance tracking:** Individual and team performance are tracked using Jira work management. This allows for real time monitoring of task completion and resource utilisation, enabling timely interventions of resources that need to be reallocated.

**People Costs:**  It's important to note that our team is providing our labour free of charge, as this project serves as a valuable opportunity for gaining practical experience. This arrangement not only enhances our skills but also allows us to contribute meaningfully to the project's success without incurring labour costs.

## **3.2 Hardware:**

For team members to have the essential equipment to execute tasks effectively, effective hardware management is needed.

* **Laptops/ Desktop computers:** Because of the scope of our project, the only mandatory hardware the team needs is their laptops or desktop computers. These devices are essential for development, data modelling and analysis tasks. Each team member’s computer must be capable of handling the necessary processing demands, with a reliable internet connection for accessing resources, collaborating via discord and participating in online meetings. Since all critical infrastructure is cloud based no additional physical hardware is needed.

**Management strategy:**

* **Standardisation:** All team members are required to have laptops or computers that meet minimum standard in terms of processing power, memory and storage capacity to ensure smooth operations and meet Visual studio code version 1.92 requirements. Laptops used need to be able to have a working camera to join group meetings to facilitate open communication and ensure team building.
* **Maintenance and security:** Each member will be responsible for their own devices security through ensuring their devices are updated with the latest security patches and software updates. Consistent backups of important data is also encouraged to prevent any loss of progress in case of any hardware failure that may occur.

**Hardware Costs:**

* **Laptops/Desktops:** If hardware needs to be replaced or upgraded, the cost per device is estimated at $1,200. Total hardware budget allocation: $6,000.
* **Maintenance:** No formal IT support budget, but members are encouraged to perform regular updates. No additional budget allocation required for this.

## **3.3 Software:**

Effective software management is vital as it ensures that the tools and applications used by the team are reliable, up to date and suited for our project needs.

* **Github:** All code and project related documentation are stored in Github repositories. Team members are assigned appropriate access levels to ensure secure and organised appropriate access levels to ensure secure and organised code management. Collaboration is facilitated through pull requests, code reviews, and issue tracking.
* **Jira**: Tasks and project milestones are documented and tracked in Jira. Team members are assigned tasks and responsibilities according to their roles. Jiras Gannt charts and task dependencies help visualise project timelines and manage workflows effectively.
* **Azure Data functions:** Resources are allocated based on the computational requirements of data processing tasks. Azure data functions are scaled automatically according to the workload, ensuring efficient use of resources and cost-effectiveness.
* **Azure Data lake Storage(ADLS2) :** Data is organised and stored in Azure Data Lake Storage according to its classification and access needs. Storage costs are monitored and data is managed to optimise storage usage and ensure security and compliance.
* **[NEW]Figma:** The web app’s user interface (UI) and user experience (UX) design primarily relies upon Figma as the tool for design and prototyping purposes. It facilitates real time collaboration among designers and developers well as stakeholders to review design iterations effectively. Moreover the feature of version history, in Figma keeps track of all design modifications ensuring feedback incorporation.
* **[NEW]next.js:**We decided to use this framework for developing the web application because of its ability to render on the server side and enhance SEO performance effectively; this enablesus to provide speedy and scalable web pages that are user friendly as well.
* **[NEW] Node.js:** Node.js is commonly employed as the environment for managing the server side operations of an application due, to its asynchronous and event driven design that supports scalability and high performance levels when dealing with multiple simultaneous connections effectively.

**Management strategy:**

* **Licensing and access control:** Ensure that all software licences are upto date and that team members have the necessary permissions and access levels to perform tasks effectively.
* **Integration and compatibility:** Regurualry verify that software tools integrate seamlessly and are compatible with each other to avoid disruptions in workflow.
* **Support and maintenance:** Regularly review software performance and make adjustments as needed to meet project requirements. Utilsing support service provided by the software vendors we have used for troubleshooting and resolving any issues that may arise.
* **Training**: Our team will be trained on how to use each software tool effectively. With softwares like Github, How to guides have been created to ensure all team members are using the software correctly and ensuring no issues will occur. Moreover, User guides have been created on how to develop azure functions on local machines as well, therefore helping us on how to set this process up and what to learn to start developing.

**Software Costs:**

* [NEW] Free Services: Github, Jira, Figma, Node,js, Next.js.
* Azure Data Functions - The first 400,000 GB/s of execution and 1,000,000 executions are free. Since we will be processing 10 MB of data, it is unrealistic to use up all of the free-tier capacity. Therefore, Azure Functions will incur a total cost of $0.
* Azure Data Lake Storage (ADLS2): Data is organised and stored in Azure Data Lake Storage according to its classification and access needs.
* **Cost Structure**:
  + Storage: The first 51,200 GB/month is charged at $0.031 per GB.
  + Write Operations: $0.10942 (for every 4 MB, per 10,000 write operations).
  + Read Operations: $0.00874 (for every 4 MB, per 10,000 read operations).

Overall Software cost:

* Total Operations Cost: Write Operations ($0.10942) + Read Operations ($0.00874) = **$0.11816**.
* Since we have three layers of storage (bronze, silver, and gold), the total operations cost for all layers is: **$0.11816 \* 3 = $0.35448**.
* Adding the base storage cost to the operations cost gives us:
* Base Storage: $0.031
* Total Operations Cost: $0.35448
* Total Monthly Cost for ADLS2: $0.031 + $0.35448 = **$0.38548/month** (rounded to **$0.39** for simplicity).

## **3.4 Other Resources:**

In addition to people, hardware and software our project requires the management of various other resources that contribute to the successful completion of the project.

* **Documentation and knowledge repositories:** Centralised documentation that includes project plans, meeting notes, technical documentation and training materials. This is crucial for maintaining consistency, enabling knowledge sharing and ensuring that all team members are aligned. All project documentation is stored in a shared accessible location (Google Drive and Github). Team members are responsible for maintaining up to date documentation and specific roles are assigned to oversee the consistency and accuracy of the knowledge repositories.
* **Communication platforms:** Tools such as Discord and email are used for internal and external communication. These platforms are essential for daily interactions, quick updates and formal communication with mentors. Discord is the primary communication platform for more day to day interactions. While email is used for more formal communication. Specific channels and threads within discord are dedicated to different aspects of the project to maintain organised communication.
* **Data Acquisition (Kaggle):** We used Kaggle to source datasets that meet business requirements

**Management strategy:**

Documentation and knowledge repositories management:

* **Regular updated:** Documentation and knowledge repositories will be regularly updated to reflect the latest project developments. Routine audits will take place to ensure all documentation is accurate, comprehensive and aligns with the project's process.
* **Version control:** Implementation of version control practises in Github and google drive to track changes and maintain the integrity of documentation. This will ensure that any previous versions can be retrieved if necessary.
* **Centralised access**: All team members will have appropriate access to these repositories and the structure will be easy to navigate with the clear guidelines on where and how to store and access documents.

Communication platform management:

* **Channel organisation:** WIthin discord, dedicated channels will be created for different aspects of the project to keep communication organisation and relevant.
* **Clear protocols**: Established protocols when to use discord versus email, ensuring that urgent or day to day updates are communicated through discord while formal communication is through email.
* **Security and privacy:** Ensured that all communication platforms are secured with appropriate privacy settings, and have encouraged the use of encrypted channels for any sensitive information sharing.

**Data and other resource costs:**

* **Data Acquisition:** Data from Kaggle is free, with no associated costs.
* Google Drive, Github and Discord are all free tools.

# **4. Change Management**

## **4.1 Changes to requirements and scope**

During the project, our client revised their requirement to exclude data visualisation as part of the minimum viable product (MVP) due to budget constraints. Initially the project scope included using tools like Power BI and Tableau to create visual representation of the data and several team members were undergoing research and training to utilise these tools effectively.

Impact of the change:

* **Training adjustment:** Team members who were in the process of getting trained on utilising power BI have now refocused their efforts on other critical areas of the project like python, data cleaning and data synthesising.
* **Scope Revision:** The removal of data visualisation from the MVP allows the team to reallocate resources and time to other essential aspects of the project, therefore ensuring the remaining deliverables are completed within the agreed budget and timeline.
* **Client communication:** The decision was made collaboratively with the client to help align the project outcomes with their budget constraints while maintaining the core functionality required for the MVP.
* **[NEW]: New Mentor introduction:**  A new mentor has joined the team to offer support in the updated focus areas such, as Python programming and data refinement methods to help align the teams skills with the projects scope revisions.
* **[NEW]: Updated requirements from the client:** The client has also made changes, to their needs by putting importance on the design of the user interface (UI) making sure that the system is easy to use and intuitive even without visual data representations.This adjustment has led to more attention being given to enhancing the front end interface.
* **[NEW]: New requirements from the AI Team:** The AI team has asked for the data to be managed in JSON and SQL formats to allow for structured data processing flexibility. They need to make changes in how they store and handle the data to ensure it works smoothly with AI models and processes.

Next Steps:

* **Reallocation of resources:** The team will now redirect their focus on tasks that align with the updated project scope.
* **Ongoing communication:** Our project manager will ensure continuous communication with the client to monitor any further changes in requirements or scope and manage them effectively.

Through adapting to this change, the team remains committed to delivering a high quality final product which will meet the clients needs while also meeting the budget limitations.

## **4.2 Changes to documents, code and data**

It is crucial to implement practices that maintain the integrity of the codebase, infrastructure and data when managing a large and complex project. As previously mentioned, Github will be the platform that we will employ to perform version control for our project. Specifically, by adopting a structured branching strategy to enforce consistency in code structure, versioning documentation, and securing critical data with immutable storage, we can create an efficient and secure development environment.

##### 

### **4.2.1 Branching strategy: Issue-based**

For every issue created in Github, a corresponding branch will be named with the issue number. This ensures that each branch is clearly linked to a specific issue. For example, we could have issue number 5 that is titled “Add string validation function” with further description to explain the problem. Then the person that is assigned the issue will create a branch called “ticket-5” and add the asked functionality. In the final commit message, that person must also write “#5 add string validation function” so that when a person checks out the code repo, they can click on the “#5” which is a hyperlink to the issue that was created to read more about the problem.

This approach will also be integrated with Github Actions to automatically run tests whenever code is pushed to a specific branch. On another note, Github Actions will be used in the CI/CD pipeline to ensure infrastructure changes with Terraform are tested first before being automatically deployed and make changes to the actual solution on Azure.

### **4.2.2 Code Reviews: Enforcing Quality with Pull Requests and Github Actions**

A Pull Request (PR) workflow enforces that every new issue or feature must be developed on a separate branch and go through a PR review before merging into the main branch. GitHub's branch protection rules will prevent direct pushes to the main branch, requiring a code review before merging changes from the development branch. Also Github Actions will be used to perform checks on the code before it is ready for code reviews and merging.

To re-iterate on the CI/CD integration with Github Actions previously mentioned, we will automate testing and deployment pipelines with Github Actions, so that every PR triggers tests for Azure Functions and infrastructure changes defined in Terraform. This ensures that the code and infrastructure are peer-reviewed before being merged to the main (production) branch.

Any new feature or fix for Azure Functions will go through the entire process, including checks for code quality and errors using GitHub Actions, before being peer-reviewed and deployed to the main branch, ensuring seamless deployment to our Azure environment.

### **4.2.3 Documentation versioning**

In this project, we’ll treat our documentation as code. This means that all documentation will be stored in the /assets/docs folder. We will use Git for version control of our documentation.While reverting changes to documentation is unlikely, we will continually update it to reflect changes in our system.

Each document will often be updated alongside the corresponding code. If documentation updates are independent of any code changes, a separate issue will be created for them. For PRs, we will not spend extensive time reviewing documentation after it has been written, as two dedicated team members (Tash and Ninuri) will ensure all documentation is up-to-date and follows the established standards.

### **4.2.4 Immutable storage:**

Our solution involves using Azure Data Lake Storage Gen 2, which is essentially Azure Blob storage with support for Immutable Policies, often used for data containers holding critical data. This is still under consideration. Another competing idea is to enable versioning in each data storage container. This means if we change our files, a version tree will allow us to revert to previous versions of the data quickly.

However, since the amount of data we will be sourcing and creating for this project is relatively small, we are also considering creating a versioning system using folders. We are also considering creating a versioning system with our folder. This means that whenever new data is added or the structure of the data changes, we can create a new folder to store the updated data. This approach will simplify our solution, but it is not scalable or realistic in a real-world scenario, as the number of folders would quickly grow and become unmanageable.

# **5. Quality Management**

Quality Management is the continuous improvement of processes and services to satisfy expectations of the client at every stage of the project (Ebrahimi and Sadeghi, 2013). This section highlights the various aspects of quality planning where we discuss the objectives of the project, quality assurance, and quality control methods that will be undertaken to ensure the quality of the project.

## **5.1. Objectives**

### **5.1.1. Ensuring data quality**

With the presence of uncleaned data in the database any results we receive from queries will be inaccurate, incomplete, incorrect, and not up-to-date as they do not “represent real-world entities to which they refer” (Fan and Geerts, 2022). Thus, improving the quality of our data is crucial for this project. Key issues with data quality include data consistency, data accuracy, data currency, data deduplication, and information completeness (Fan and Geerts, 2022). Data quality management is used to effectively detect and correct errors in the data resulting in increased value and accuracy to the project results. To address these concerns exploratory data analysis (EDA) will be implemented in order to analyse data distributions, potential inconsistencies, missing data and outliers. Key performance indicators (KPIs) that will be used to measure data quality include:

* **Data Accuracy:** verifying data against standards or benchmarks
* **Data Consistency:** comparing contradictions with similar datasets
* **Data Completeness:** measuring the percentage of missing values to be less than 20% of the dataset

### **5.1.2. Model Accuracy**

The main objective of the dataset is to train an AI model on insurance claims allowing it to identify fraudulent and non-fraudulent claims. However, the poor quality of training data is known to negatively impact the performance of machine learning models with issues such as underfitting and overfitting affecting the model’s accuracy. Thus, ensuring the quality of the dataset will save the AI team training effort on their AI model, avoid disappointment from machine learning results, as well as reduce setbacks in the project's progress (Mohammed et al., 2024). To test the accuracy of the dataset when train by the model quality metrics such as:

* **Model accuracy:** precision, recall and F1 score with a percentage high than 85%
* **Training efficiency:** number iteration taken by the model to provide good results will be less than 50

Will be used as a way of gaining feedback on the quality of the dataset to be further improved and used again by the AI team in a feedback loop.

### **5.1.3. Compliance with business requirements**

This project has five main business requirements; using Azure services, Agile Delivery, Anonymised Data, User Interface, and being a Gen AI solution. Various control and assurance mechanisms must be placed to ensure that the business requirements are being maintained and adhered to at every step of the development process such as:

* **Azure Usage:** percentage of overall services deployed on Azure is over 90%
* **Agile Compliance:** Assessing the progress of sprints and speed of project delivery to be over 85%
* **Anonymisation:** 100% of PII is removed in all datasets and cases

### **5.1.4. Customer Focus and User Satisfaction**

The main goal of any project is to satisfy the user and create a working product. Thus, it is crucial to involve the client in every stage of the project to ensure that what is being created is what they need. As well as gaining input and feedback on key aspects of the project to confirm that the project meets the client’s requirements whether it be budget, technology used, or procedures undertaken. To track the feedback and satisfaction of clients the following KPIs it be used:

* **Client engagement:** number/frequency of feedback sessions with client is at least once per sprint
* **Satisfaction scores:** given after each sprint on how accurately the product meets the users needs and has to be over 75%

## **5.2. Data Quality Management: Monitoring and Assurance**

### **5.2.1. Data Collection Process**

The source of data represents a logical perspective that encapsulates the methodologies for data collection, data generation, and assessing organisational compliance and integrity of the source (Mohammed et al., 2024). Thus, it focuses on the notion of data provenance such as data provider, data origin, and the other organisations involved in creating and transforming data (Mohammed et al., 2024). The main source of data collection is Kaggle, an online community where data scientists and machine learning engineers can obtain datasets to train AI models as well as publish their datasets for others to use. While the data source (Kaggle) is widely reputable suggesting the quality of the data it is important to conduct assessments on the dataset to ensure that the data source is credible and adheres to the dimensions of data quality.

Some assessments that have been done to the data source during this process are (Mohammed et al., 2024):

* **Origin:** where the data was sourced
* **Transformations:** how was the data changed
* **Traceability:** which focuses on the entities involved in its history
* **Reputation:** which is when the reliability and credibility of the data source are evaluated
* **Reliability:** which reviews the collection methods and if it conforms with established practices.

Furthermore, the data in the dataset has to be evaluated in relation to the data quality dimensions to ensure their validity (Picard et al., 2020):

* **Accuracy:** measuring how correct the data values are
* **Accessibility:** how easily the data can be accessed and user awareness of the data being collected as well as its location
* **Consistency**: the extent key data feature provide correlating information on the same data object
* **Timeliness**: the dataset has current data and information is available on time
* **Traceability**: the data’s history and origin is verifiable
* **Usability**: the scope of which the data can be understood and utilised
* **Relevance**: the data is able to meet the need of the user and even through information may change the collected data should still be relevant

### **5.2.2. Exploratory Data Analysis (EDA)**

EDA will be conducted on each dataset at the start of the dataset preparation stage as it allows for the exploration of data distributions and potential issues. By identifying missing data, inconsistencies, outliers of unusual patterns these issues can be addressed before the cleaning stage.

* **Correlation Analysis:** find correlations between features that could impact the model’s performance
* **Summary statistics:** spotting irregularities by checking mean, median, mode and the standard deviation
* **Data visualisation:** creating graphs and plots to examine the distribution of the data

### **5.2.3. Data Cleaning**

Datasets that are collected from different sources and put in our Bronzer Layer may contain types of dirty data such as missing fields/values, contradicting data, noisy values, data integration issues and cryptic data (Swapna et al., 2016). Additionally, anonymising the data is a key requirement of the project meaning that personally identifiable information should be removed from the dataset such as names, addresses, mobile phone numbers etc. Therefore, the data set must be cleaned to ensure better quality of the dataset for AI model training.

Three main steps in data cleaning include (Swapna et al., 2016):

* **Exploratory data analysis** - As a dataset may contain multiple error types it is important to do a comprehensive exploratory data analysis to identify the types of errors in the dataset.
* **Transformation -** At this stage the identified errors are transformed to clean the dataset. This may include dropping columns with personal information, removing duplicate rows, and filling missing values in the dataset.
* **Verification** - After the dataset is cleaned the new dataset should be evaluated and tested on the AI model with a sample of the dataset to test its accuracy when training the model.

To check the accuracy of the dataset after cleaning these quality metrics will be held to ensure the quality of the final dataset:

* **Model Accuracy:** using accuracy score, MSE, MAE and r^2 the accuracy of the imputed missing variables in the dataset has to be above 85%
* **Transformation Analysis:** number of duplicate rows and missing values has to be 0
* **Verification Check:** results from the AI model have to be above 90%

| **Quality Metric** | **Initial Result** | **Actions Taken** |
| --- | --- | --- |
| **Model Accuracy** | Accuracy score, MSE, MAE and r^2 of certain columns using KNN and random forest were above 95% however some only had a score of 70% | Hyper-tuning was done using GridSearchCV and RandomizedSearchCV to increase the scores |
| **Transformation Analysis** | At the beginning all datasets had missing values. | After imputing missing values there were no missing values in the dataset |
| **Verification Check** | Dataset with missing values resulted in an accuracy of less than 50% | Missing values were imputed or rows were dropped to create a complete dataset |

### **5.2.4. Data Standardisation**

Standardising data is the transformation of data into a common format focusing on interoperability standards and modelling annotations. It is usually performed by referencing a gold standard model that becomes the standard for the parameters of following datasets where non-standard values are replaced with corresponding values that comply with the standard. Data standardisation is crucial to our project and should adhere to the following metrics:

* **File type:** All datasets found should be in .csv format so that datasets can be easily integrated.
* **Data Format:** All data in a specific column must be in the same format across all datasets such as for dates (YYYY-MM-DD) or time (HH:MM:SS). This allows for uniform and clearly defined data in columns for easy understanding and integration.
* **Data Type:** float, string, int, datetime, etc. should be the same for all data in the same column. All columns should be converted to numerical values as the AI model only trains with numerical values
* **Column Name:** All column names have to be the same across datasets
* **Categorical Data Categories:** Data under categorical columns should be named the same across all datasets. E.g. for Education Level all dataset should have the categories High School, Primary School etc. not Secondary School or Tertiary.

## **5.3. Human Resource Management: Agile Delivery**

Utilising Agile methodology for project delivery leads to on-time delivery, customer satisfaction, business value, and product quality (Krehbiel and Miller, 2018). Thus, is an approach for improving and managing quality in delivering a successful project. Agile practices focus on facilitating face-to-face interactions leading to better articulation of goals, encouraging collaboration, improving team dynamic, and supporting innovations and experimentation (Krehbiel and Miller, 2018).

### **5.3.1. Sprint Planning**

Sprint planning is short iterations of project stages to bring value to the customer in smaller and more frequent intervals (Krehbiel and Miller, 2018). This alongside daily stand ups promotes enhanced communication, collaboration, reflection, and continuous improvement (Krehbiel and Miller, 2018). The ability to show work and progress to the client frequently allows for the client to be involved in the process. This improves the quality of the project deliverable as changes are being made real-time to reflect the requirements of the client. The quality of the project process is met as improved communication leads to smoother project completion and less push back from stakeholders. To track the effectiveness of each sprint the following KPIs will be used:

* **Sprint completion rate**: 90% of the sprint is completed within the defined time frame
* **Team velocity:** increasing amount of tasks to be done by the team each sprint

| **Quality Metric** | **Initial Result** | **Actions Taken** |
| --- | --- | --- |
| **Sprint completion rate** | 100% of Sprints 1, 2, 3, 4, 5 have been complete on time | N/A |
| **Team velocity** | Number of tasks in each sprint is greater than the previous | Work is split effectively amongst the team and deadlines are met. |

### **5.3.2. Iteration Reviews**

Having iteration reviews at the end of each sprint allows the team to gather all the feedback from the client, identify any areas for improvement, and re-align themselves with the project goals. This results in improved performance in the following sprint as work reflects the feedback given and inefficiencies were noticed and altered to be more streamlined enhancing the quality of the project delivery.

* **Client satisfaction score:** A satisfactions score of over 85% is given after each sprint review
* **Change request rate:** Less than 3 change requests are made per stand-up
* **Feedback turnaround time:** feedback from client is given with 48 hours so adjust can be made promptly

| **Quality Metric** | **Initial Result** | **Actions Taken** |
| --- | --- | --- |
| **Client satisfaction score** | Client is highly satisfied with progress at each sprint | N/A |
| **Change request rate** | Maximum of 2 changes have been request from all the stand-ups so far | Requests are noted and implement by the next stand-up |
| **Feedback turnaround time** | Minor feedback on items present during stand-up are given immediately. However feedback needed for deliverables are not given on time and may take a week or longer. | Follow up messages and email are sent to client and may need to escalate with teaching staff if the issue continues to arise. |

### **5.3.3. Continuous Feedback Loop**

Engaging in Agile Delivery allows for a continuous feedback loop with the client in which work is shown to the client and the team receives feedback. This happens in short iterations allowing for the project to change and reflect the requirements of the client. This leads to reduced need for a final inspection that may not meet the clients requirements but a more robust product made from various testing and prototypes that can be effectively integrated to the clients system.

* **Client engagement frequency:** interaction with at least 2 mentors per sprint
* **Feedback incorporation rate:** 100% of feedback incorporated within the next sprint

| **Quality Metric** | **Initial Result** | **Actions Taken** |
| --- | --- | --- |
| **Client engagement frequency** | There is at least 3-4 mentors discussing the project across all team members. | Communicate with different mentors for a variety of feedback and ask questions on crucial topics. |
| **Feedback incorporation rate** | 90%-100% of feedback is incorporated within the next sprint or the following sprint. | If feedback cannot be incorporated, discuss with mentors and ask questions for further clarification. |

## **5.4. Development Management**

### **5.4.1. Code convention**

A code convention is a set of guidelines made for a programming language to enhance the software structural quality. Projects that implement these conventions have improved readability of their source code and enable easier software maintenance. Aspects of a language that is included in the convention include, indentation, declarations, naming conventions, comments, etc.

### **5.4.2. Prototype**

Creating a prototype of the dataset allows the team to test the viability of the datasets capability (Burgess, 2024). For this project our team will mainly engage in vertical prototyping where we create test data sets for a specific functionality or requirement (Burgess, 2024). This allows the team to test clean datasets that have been created and see whether they are enough to train the AI model. In cases of over- and under-fitting the dataset can then be modified and new prototypes can be created. Once a prototype of good training capacity is created then the dataset can be scaled up and the model and be trained more robustly.

**[NEW]**

| **Prototype** | **Initial Action** | **Feedback** |
| --- | --- | --- |
| **Prototype 1: Excel file .csv** | Created an excel file with the initial dataset of 30 columns and 12,500 rows | However, it was hard for the AI model to interpret data in excel format so the AI team requested a dataset in .json format |
| **Prototype 2: JSON file .json** | After receiving the request we spent 1 day focusing on creating this file and promptly sending it to the AI team. JSON is data represented in a text-based format which will be easier for an AI model to read and interpret to provide a result. | However, as the AI team changed their AI model from a neural network to a chat gpt 4.0 mini model. Thus, as the team was more familiar with sql the requested an SQL database. |
| **Prototype 3: SQL format** | Following this request we immeadiately started working on creating an SQL database to sent back to the AI team. Having an SQL database allows for real-time predictions to be made from the database. | As this was recently sent we are still waiting for the feedback. |

## **5.5. [NEW] Website Management**

As the fraud detection model is created a user interface is needed to allow the claims officer to interact with the AI model to examine insurance claims for fraud. The website needs to meet the desired standard of usability, performance and accuracy meeting the requirements of the client. Ultimately, managing the quality of the website will enhance user satisfaction, ensure scalability, and delivery long-term value while reducing operations risks that may occur. Quality metrics are a key component in tracking and maintaining the website across numerous features:

* **API success rate**: 90% of claims are accurately fetched, updated and process
* **User operation reliability:** 95% of the time user interactions with features on the website such as fraud detection and claim management must operate reliably
* **Authentication success rate**: 0% of unauthorised users will be able to access the website.
* **Customer satisfaction score**: 85% or higher positive feedback from users on the website.

| **Quality Metric** | **Initial Result** | **Actions Taken** |
| --- | --- | --- |
| **API sucess rate** | Currently all APIs are properly working and claims are properly fetched, updated and processed on the website. | The AI model has still not been linked yet so future issues may arise. To prevent this we are deciding to test the integration locally before putting on Azure. |
| **User operation reliability** | Current feature are properly interacting with the user and providing desired results. | More features will be added to |
| **Authentication success rate** | Initial website design had no authentication however a basic system has been created. | More testing and improvements to the authentication will have to be made to keep the confidential information safe |
| **Client satisfaction score** | Currently client is very satisfied with the progress of the project however, would like to see more improvements on design and authentication. | After feedback was given an authentication system was implemented and the website design is being improved. |

## **5.6. Documentation Management**

### **5.6.1. Formatting**

Having standardised formatting across all documents increases the professionality and quality of the project documentation as it is uniform.

* Font: Times New Roman
* Size: Heading 1 - 20, Heading 2 - 16, Heading 3 - 14, Normal Text -12
* Spacing: 1.15
* Table of Contents: Separate one for each document and must link to each heading
* Cover Page: At the start of each document

### **5.6.2. Documentation control officers**

Document control officers are put in place as a control mechanism to make sure that all documentation is written according to the defined documentation standards. These officers are further explored in the Team Manual where the further roles and responsibilities are discussed.

# **6. Schedule**

## **6.1. Project Overview**

For our project, we are working with our client EY to create a solution using GenAI to examine automobile claim reports and detect whether they are fraud or not fraud. The main aim of this project is to simplify their claims process and enhance their current fraud detection capabilities. To do this we are split into AI and Data Teams, as the Data Team our main project goal is to create a dataset that is clean and accurate to be used by the AI to train and test their AI model. As we follow the agile methodology when undertaking this project we have decided to call our phases sprints as we carry out the project as outlined below:

* Sprint 1: Planning Stage **(completed)**
  + Forming the team, getting to know each others skills and proficiencies, and creating the communication channel
* Sprint 2: Building Stage 1 **(completed)**
  + Brainstorming ideas for our MVP taking into consideration alternatives and feasibility
* Sprint 3: Building Stage 2 **(completed)**
  + Finalising the project plan and creating a test dataset and synthesised data for training
* Sprint 4: Building Stage 3 **(completed)**
  + Hand over prototype dataset to AI team to train their AI model
* Sprint 5: Testing Stage 1 **(completed)**
  + Revise the dataset in relations to accuracy results from AI and reduce any under/over-fitting issues
* Sprint 6: Testing Stage 2 **(in progress)**
  + Hand revised dataset to the AI team for further testing
* Sprint 7: Execution Stage
  + Create front-end of the fraud detection project
* Sprint 8: Deployment Stage
  + Hand over the final deliverable to the client with a user manual and final report

### **6.1.1. Deliverables**

For our deliverables, they have been split into project-related deliverables for the items we present to our project sponsor Deborah, and product-related deliverables for the items presented to our client Joshua Falanga from EY.

**Project-related deliverables:**

* Deliverable 1: Feasibility Report and Team Manual
* Deliverable 2: Project Plan, Requirements/Scoping Document and Update Team Manual
* Deliverable 3: Updated Deliverable 2, Design Document, Test Cases, Prototype/MVP
* Deliverable 4: Updated Increment 1 (Deliverable 3), User/Training Manual
* Deliverable 5: Final Group Reflective Report
* Deliverable 6: Project Presentation/Demonstration
* Individual Contributions forms

**Product-related deliverables:**

So far our deliverables to the client have been:

* User Stories
* High-Level Solution Design
* Options Analysis Report
* **[NEW]** MVP 1 and 2 proposition documents
* **[NEW]** front-end user interface
* Deliverable 3: MVP/Prototype
* Deliverable 7: Final product, the dataset

However, for our project, we have split it into 3 main stages:

* Bronze Layer: Raw datasets
* Silver Layer: Cleaned & enhanced datasets with synthetic data
* Gold Layer: Combined dataset for training AI model

## **6.2. Project Phases and Key Milestones**

See Appendix 4 for a breakdown of the project phases and the key milestones in each phase.

## **6.3. Schedule - Gantt Chart**

The figures below showcase the tasks required for the completion of the project using our project management tool Jira to create Gantt charts to visualise this timeline. The Gantt chart includes the summary of the task, estimated starting and ending dates of the tasks and Sprints, if the task has been completed or not and the dependencies tasks have with other tasks. As we progress through the project and each Sprint, the schedule will be subject to change such as dependencies, unforeseen factors, changes from clients and setbacks from team members. Thus, the schedule will be modified to reflect these changes throughout the project. See Appendix 5, 6, 7, 8 9 amd 10 for the Gantt chart figures

## 

## **6.4. Schedule Revisions**

| **Sprint** | **Task** | **Revision** | **Actions** |
| --- | --- | --- | --- |
| Sprint 3 | Migrate tasks to Azure and test run scripts. | Had to push back on this task as the account was not created yet. | Discuss with mentors in stand-up about getting Azure logins to complete task. |
| Sprint 3 | AI team requirements changed from 1,000 rows of data to 20,000 rows of data so they can train their AI model for their MVP 1. | Had to pull forward certain tasks that were planned in order to meet their requirements in time for MVP 1 | Gave them 6,500 rows after one day and 20,000 after a week. |
| Sprint 3 | Originally building a database using 1,000 rows and 3 synthetic datasets.  However, couldn’t use only 1 dataset as a source as it would add bias and unreliability. Any test run would become not-valid | Had to put a hold on continuing scheduled cleaning and synthesising tasks as we has to source more datasets. Now we use 3 original source datasets and 2 synthetic datasets. | Build 20,000 row dataset using 3 datasets (12,500 rows total) and 2 synthetic datasets (4,000 rows each). |
| Sprint 4 | AI team discovered that they needed no null values in the dataset to run the AI model | Had to go back and revise the dataset and remove the gender column that has the most missing values. | Gave them a temporary dataset with 10,500 rows. |
| **[NEW]**  Sprint 5 | After the Azure account was created and given to us by the client the account had issues with permissions and restrictions of actions. Furthermore, the Azure subscription was cancelled promptly after more than $80 on the account was used. | The issues with getting the Azure account and getting permissions set-back the project by a couple of sprints as we had to wait to run our scripts on Azure. Furthermore, after the cancellation of the Azure subscription no team member was able to do anything on Azure and combined with the lack of response from the client we had to wait ofr our weekly meeting to resolve the issue. | Discussed our issues with the mentors as soon as they occurred and resolved them in the following meeting so we could promptly continue our tasks. |
| **[NEW]**  Sprint 6 | The creation of the front-end user interface. | Have to put an indefinite hold on the creation of the front-end until the AI team have completed their AI model so we have continue with integration and testing of the website. | Have been in constant communications with the AI team and have provided them with requested datasets such as in .json and SQL so the model can be completed promptly. |

## 

## **6.5. Resource and Task Allocation**

Resources for this project include, team members, skills, equipment, and tools and software. Proper resource allocation leads to a “streamlined workflow, mitigation of bottlenecks, simplified project management and resource optimisation” (Martins, 2024). This allows us to improve our project management ensuring that resources are used efficiently, projects are completed on time, finished to the desired quality standards and within budget resulting in a successful project (Martins, 2024). **[CHANGES]** To properly allocate our resources and skills we have split ourselves into seven main groups, Individual Member, Business Intelligence Team, Development Team, AI Team, Web Team, and Whole Team. These tasks and resources are allocated by the Project Manager to specific groups and assigned to one team member to maintain accountability and traceability of the project.As we reach the end of our sprint 4 and begin our sprint 5 we have decided to increase the collaboration with the AI team and have two members work with the AI team temporarily to ensure issues with the AI model and dataset are fixed and problems can be communicated more effectively. **[NEW]** To create our user interface which is a website that allows the client to analyse claims reports and detect fraud we have decided to create a new team called the Web Team to create, develop, integrate and test the website. Refer to Appendix 11 for the Task Allocation groups.

## **6.6. Tools and Technologies**

### **6.6.1. Jira**

Our project management tool is Jira project management software that allows for the tracking, assigning, and managing of tasks. Our project is split into 2-week sprints and the team is assigned tasks by the Project Manager. We have combined our tasks with Group 13 using this tool so their tasks are also placed on our Jira creating a combined timeline and Gantt chart. This allows us and the client to see the progress and timeline of the project as a whole as well as add dependencies to tasks of the other team.

### **6.6.2. GitHub**

GutHub is a version control system that allows for versions of files to be tracked and for source code to be worked on collaboratively when developing. It also allows for teams to discuss and manage tasks within the GitHub allowing for more effective collaboration. When using GitHub each branch will be created by identifying an issue and numbering it. This allows for other members to work on issues and commit work to a certain issue number so each push is organised and sorted. Once an issue has been completed the branch will be closed and the issue number will be removed to reduce misunderstandings and proceed to other issues.

### **6.6.3. [NEW] Discord**

Discord is a communication tool that allows for team members to communicate, host meeting and send-reminders to their fellow members. This platform is being used by all project members as a central hub of communication by the Data team. the AI team and the mentors. This tool allows all members to communicate with each other on what tasks they are working on, ask questions and work collaboratively with each other on the same task so voice channels and screen sharing. This has become the main tool for reaching out for help from other team members and mentors to complete tasks. Furthermore, it allows the team to discuss their weekly commitments and schedules so the team is on the same page about availability and tasks that will be completed.

# **7. Handover requirements**

During the handover process we as the data team have to present to the client NRMA the necessary documentation, resources and tools required to effectively use the fraud detection system after the project has been completed. The following section will cover the vital documentation that is handed over to the NRMA at the end of the project. Through discussions with the client we have confirmed the necessary handover requirements that have to be met in order for them to maintain and develop the system.

## **7.1 Software and Tools**

* **Environmental Setup:** Instructions to set up the Azure environment used throughout the project such as Azure Functions and Data Lake.
* **Automation Scripts:** Details on software requirements such as libraries and packages to run scripts as well as a guide on running scripts that were created to automate model training, data processing and deployment tasks.
* **Deployment Instructions:** Clear documentation on deploying the scripts and fraud detection model with necessary configurations

## **7.2 Data Deliverables**

* **Final Dataset:** The complete and structured dataset utilised to train the fraud detection model, organised into various levels of data quality (Bronze, silver, Gold)
* **Data Documentation:** Detailed documentation outlining the structure, attributes and origins of the dataset including the preprocessing or transformations that were applied throughout the project.
* **Data Pipeline:** If any scripts were utilised to clean,load or structure the data these will be provided along with the guidelines on how to execute them.

## **7.3. Testing and Validation**

* **Validation Scripts:** Descriptions of scripts used to evaluate the accuracy of cleaning scripts or the evaluations of models used to synthesise data
* **Testing Tools:** Manual on testing process used such as environment, datasets and tools so it can be replicated.

## **7.4. Model Deliverables: (To be done by the AI Team)**

* **Fraud Detection Model:** Provide the completed and trained AI model, prepared and ready to be deployed.
* **Model Documentation**:A detailed explanation of the model, including its structure, algorithms, chosen hyperparameters and performance metrics.
* **Model code and Scripts:** All code used for the model training and evaluation, including any python scripts along with the necessary libraries and packages.
* **Model Outputs:** Examples of model outputs that demonstrate how fraudulent claims are flagged, accompanied by performance reports such as precision, recall and F1 score.

## **7.5 User and Training Manuals**

* **User Manual:** Detailed guide on how to use the fraud detection system with key directions to input data, troubleshoot issues and understanding the results.
* **Training Manual:** Documentation on how to use the model and interpret outputs with directions on how to retrain the model with current data.

## **7.6 Final Report and Evaluations**

* **Project Completion Report:** A thorough report that outlines the project’s objectives, methodologies and results, highlighting significant decisions and outcomes.
* **Evaluation Metrics:** Documentation detailing how success was measured, including performance benchmarks and how effectively the system achieved NRMA’s fraud detection objects and requirements.

## **7.7 Client Consultation and Acceptance:**

Before the final handover is conducted a meeting with the stakeholder’s at NRMA will be held to review all the required deliverables and ensure they meet the requirements of the client and project objectives. Confirmation that the NRMA team is able to use the new fraud detection system successfully. Ultimately a final testing process will be run to allow the NRMA team to test the final system in their environment and address and final adjustments before the project conclusion.

## **7.8. [NEW] Frequently Asked Questions (FAQs)**

A FAQ section will be provided to address common queries to help the client operate the fraud detection system effectively after the handover is completed.

**Below are a few common questions that are asked:**

* How can the fraud detection system be accessed?
* What process should be followed to submit new claims for fraud detection?
* What actions should be taken if a claim is incorrectly flagged as fraudulent?
* Who should be contacted when in need of technical support?
* How can user permissions be updated or modified?

The FAQ in the Scripts and Model Execution Document will contain a more comprehensive guide on the questions above and additional commonly asked question to assist the client

# **8. Assumptions**

## **8.1. Resources**

* **Skillset** - team member have the necessary skills and background to complete the project
* Training - training for new tools and technology introduced will be provided during the project
* **Budget** - the project budget is sufficient to to cover the tools needed for the project
* **Funding** - financial support will be provided for the project with no delays in the acquisition tools and services needed for the project

## **8.2. Availabilities**

* **Team availability -** all team members will be available throughout the entirety of the project with no major changes to the team
* **Stakeholder availability** - stakeholders will be available for weekly stand-up meetings and subject matter experts will be present in weekly meetings and feedback sessions.

## **8.3. Tools**

* **Accessibility -** all necessary tools will be available for editing to all team members with no restrictions.
* **Compatibility -** the Azure services used will be compatible with each other and with the services used by the AI team for their AI model.
* **Azure services -** the costs to acquire Data Lake, Azure Functions and Application Insights will not exceed $0.45 per month for the duration of the project.
* **Github -** the team members will use GitHub to commit and push tasks. They will follow the
* **Jira -** will be used by each team member to update task progress. The project manager will assign tasks to each team member and actively update the Jira to reflect the current and future status of the project.
* **Google Drive -** will be used as a central hub for storing documents and presentations so each team member has access and is able to edit it.

## **8.4. Techniques**

* **Data synthesisation techniques** - will create 7,000 rows of clean and good quality synthetic data
* **Data cleaning method -** will clean the raw dataset to remove or replace missing values, duplicate values, and unrequired data columns.
* **Testing techniques** - testing the dataset with the AI model will provide valuable insights on the quality and performance of the dataset on the model and allow for refinement of the dataset to increase accuracy.

## **8.5. Standards**

* **Data Quality Standard** - datasets are chosen in relation to the data quality standards defined and of sufficient quality to develop a reliable dataset.
* **Documentation Standards** - all documents created for the project are created following the defined documentation standards to ensure the professionality and quality of documents created.
* **Code Conventions** - when writing code for the project the team members will adhere to the established coding standards to ensure the consistency and maintainability of the code.

## **8.6. Communication**

* **Communication channels -** team will maintain regular communication with each other to discuss about the project on the Discord server and always react or engage in discussions
* **Regular meetings -** the team will hold regular weekly meetings to discuss task status, future tasks and roadblocks.
* **Issue Communication -** when any issues or problems arise that may impact the progress of the project, team members will promptly discuss the issue with the team to reduce further delay or misunderstandings.
* **Client engagement -** the client will answer questions asked on the discord as well as provide feedback on Deliverable tasks.
* **Sponsor communication -** the project sponsor (Deborah) will reply to any issues regarding the project in a timely manner

## **8.7. Expectations**

* **Project scope** - will be clearly defined and business requirements will be established and communicated to the team.
* **Quality expectations** - project will be completed to a high quality standard from coding to documentation adhering to predefined standards.

## **8.8. Design**

* **Azure Services Compatibility -** assumed that the Azure services used such as Data Lake, Application Insights and Data Lake will be compatible with the infrastructure and Azure tools used by the AI team
* **Synthetic Data Accuracy -** the data synthesisation techniques used wll create 7,000 rows of synthetic data of high quality.

## **8.9. Documentation**

* **Access to documentation -** all team members will have unrestricted access to the Google Drive
* **Adherence to documentation standards** - all team members will follow the guidelines made for documentation and the documentation control officers will check for compliance.

## **8.10. Quality Management**

* **Dataset Quality -** the chosen datasets will be up the predefined quality standards
* **Coding Quality - t**eam members will adhere to coding standards so the codebase is consistent, readable and maintainable.

## **8.11. Risk and Contingency**

* **No delays in acquiring tools -** necessary tools, services and financial aid will be provided in a timely manner
* **Training and adaptation** - team members will be able to learn new tools and technologies introduced during the project quickly to reduce delays
* **No major technical issues -** during the project no major issues relating to tools, coding or Azure will arise.

## **8.12. [NEW] User Interface**

* **Fraud detection system -** the system will generate fraud risk scores as a percentage between 0-100% as well as provide a reason for the score.
* **Internal use only -** the system will only be accessed by NRMA staff and management thus a strong authentication system will be required
* **Data back-up -** fraud system will back-up data daily to ensure that recovery form system crashes or deletions can be recovered.
* **Interface design -** will be minimalistic and only consist on essential buttons and features such as ‘Get Claim’ or ‘Detect Fraud’.
* **Automatic retrieval -** pending claims will be automatically retrieved from the database to minimise menaul entry from users that may lead to errors

# **9. References**

Armstrong, M. (2009). *Armstrong’s Handbook of Human Resource Management Practice*. <http://196.43.179.3:8080/xmlui/handle/123456789/1068> [15/08/2024]

Ebrahimi, M., & Sadeghi, M. (2013). Quality Management and performance: An Annotated Review. *International Journal of Production Research*, *51*(18), 5625–5643. <https://doi.org/10.1080/00207543.2013.793426> [12/08/2024]

Fan, W., & Geerts, F. (2022). *Foundations of Data Quality Management*. Springer Nature. [13/08/2024]

Martins, J. (2024). *Resource Allocation Tips for Project Managers • Asana*. Asana. <https://asana.com/resources/resource-allocation> [22/08/2024]

Li, S. Q., & Chen, Y. S. (2020). Analysis of the probability matrix model for the seismic damage vulnerability of empirical structures. *Natural Hazards*, *104*(1), 705–730. <https://doi.org/10.1007/s11069-020-04187-2> [15/08/2024]

Mohammed, S., Harmouch, H., Naumann, F., & Srivastava, D. (2024). Data Quality Assessment: Challenges and Opportunities. *Data Quality Assessment: Challenges and Opportunities [Vision]. PVLDB*, *17*(1). <https://arxiv.org/pdf/2403.00526> [10/08/2024]

Picard, S., Chapdelaine, C., Cappi, C., Gardes, L., Jenn, E., Lefevre, B., & Soumarmon, T. (2020). *Ensuring Dataset Quality for Machine Learning Certification*. IEEE Xplore. <https://doi.org/10.1109/ISSREW51248.2020.00085> [21/08/2024]

Smith, P. G., & Merritt, G. M. (2002). *Proactive Risk Management: Controlling Uncertainty in Product Development*. <https://www.strategy2market.com/Preston-Smith/Proactive-Risk-Management/PMJ12-03.pdf> [14/08/2024]

Swapna, S., Niranjan, P., Srinivas, B., & Swapna, R. (2016). *Data cleaning for data quality*. IEEE Xplore. <https://ieeexplore.ieee.org/abstract/document/7724284> [16/08/2024]

Timothy, C., & Miller, D. (2018). Article 1 1-1-2018 Part of the Technology and Innovation Commons Recommended Citation Recommended Citation Krehbiel. *Journal of International Technology and Information Management Journal of International Technology and Information Management*, *27*(1). <https://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?article=1342&context=jitim> [20/08/2024]

# **10. Appendices**

## **Appendix 1. Risk analysis table[CHANGES]**

| **Risk Category** | **Risk** | **Description** | **Probability** | **Effect/consequence** | **Impact** | **Mitigation Strategies** |
| --- | --- | --- | --- | --- | --- | --- |
| Technology Risks | System Downtime | Azure Data Functions or GIthub experiencing outages | Medium | Delays in data processing and version control | Medium | Regular backups, use of alternative tools temporarily |
|  | Data Security Breach | Unauthorised access to sensitive data during processing or storage | Low | Potential legal issues, Loss of client trust | High | Implement encryption, regularly update security protocols |
|  | Inadequate Software integration | Difficulty integrating Jira, Github, Azure Data Functions. | Medium | Reduced efficiency, increased project delays. | Medium | Conduct regular integration tests, use middleware to ease integration |
|  | Scalability Issues | Azure functions or data storage limits, slowing down processes as data volumes increase. | Medium | Inefficiency in processing large datasets | Medium | Optimise code, plan for scalability. |
|  | Data Quality Issues | Incomplete or inaccurate data affecting model performance | Medium | Decreased model accuracy | High | Robust data cleaning and validation |
|  | Algorithm complexity | Overly complex algorithms increase cost and time | Medium | Higher Computational requirements | Medium | Simplify algorithms based on data needs |
|  | Infrastructure Limitations | Inadequate computational resources for large datasets | Low | Delayed model training | Medium | Plan for scalable infrastructure |
|  | Model overfitting | The model fits the training data too well but performs poorly on unseen data | Medium | Decreased model generealisationality, which leads to inaccurate predictions in production | High | Implement Cross validation, regularisation techniques and monitor performance on test data |
|  | Data Privacy Compliance | Mishandling of personal data , violating privacy laws | Low | Legal penalties, loss of client trust, project delays | High | Ensure data anonymization, adhering to privacy regulations, regularly audit data handling processes. |
|  | Version Control conflicts | Conflicting Code changes leading to loss of work or errors in the project | Medium | Delayed development progress, bugs introduced into the project | Medium | Establish version control best practises(Regular commits, branch management) |
|  | Insufficient Data for modelling | Lack of sufficient data for effective model training or poor representation of edge cases | Medium | Inadequate model performance, failure to generalise on new data | High | Use data augmentation, synthetic data generation(CTGAN) or seek additional datasets |
|  | Bias in Data | Presence of bias in training data, leading to unfair or unethical model predictions | Medium | Discriminatory model outputs, loss of trust from NRMA | High | Perform bias audits on datasets, ensure diversity in data. |
| **[NEW[** | Cross -Browser Compatibility issues | The wep app not functioning properly across different browsers. | Medium | Poor user experience, reduced accessibility for some users. | Medium | Peform extensive cross-browser testing, use responsive design frameworks |
| **[NEW]** | High Traffic Load | The web app unable to handle a sudden surge in traffic, leading to slow performance or crashes. | Medium | Poor user experience, potential loss of customers. | High | Use load balancers, optimise server resources, implement auto scaling features |
| People Skills | Skill Gaps | Team members lacking expertise in key areas (e.g AI | Medium | Project delays and suboptimal outcomes. | High | Provide training, hire consultants if needed. |
|  | Team collaboration | Poor communication or conflicts among team members. | Medium | Misunderstandings, mistakes, and delays. | High | Foster open communication, regular team meetings. |
|  | Resource availability | Unavailability of key team members during critical phases | Medium | Stalled progress delayed tasks | Medium | Cross train team members, have backup plans. |
|  | Role clarity | Overlapping or unclear responsibilities leading to confusion | Medium | Duplication of efforts or missed tasks | Medium | Clearly define roles and responsibilities |
| Organisation risks / Business risks | Budget constraints | Project exceeded the allocated budget, causing resource shortages. | Medium | Compromise in project scope or quality | High | Regular budget reviews, adjust scope as needed. |
|  | Policy Changes | Changes in organisational policies or priorities disrupting the project. | Low | Delays,need for significant project adjustments. | Medium | Stay informed about policy updates, plan for flexibility. |
|  | Shifting priorities | Clients changing focus on deprioritizing the project | Low | Delays or project termination | Medium | Regular stakeholder communication |
|  | Market Changes | Shifts in market trends affecting the project( economic downturns, supply chain disruptions) affecting business needs or priorities. | Medium | Shift in business goals, project delays | Low | Monitor trends, maintain flexibility, maintain backup plans |
|  | Inadequate change management | Poor management of changes to project scope , objectives or timelines causing confusion and inefficiencies | Medium | Scope creep, delays or suboptimal results | Medium | Implement a structured change management process, ensure proper documentation, and align teams before making changes, |
|  | Failure to meet regulatory compliance | Inability to comply with industry specific regulations or data privacy laws, leading to legal issues. | Low | Legal penalties, reputation damage, or project cancellation. | High | Research on particular laws and document where we would need to ensure we meet compliance and standards. (Eg. Removal of personal identifiable information) when cleaning data. |
|  | Inadequate Resource allocation | Insufficient allocation of human or financial resources leads to slowdowns or compromised project quality | High | Deadlines missed, team burnout, reduced overall output. | High | Conduct thorough resource planning and regularly review resource usage to ensure adequate staffing and budget. |
| Requirements risks | Ambiguous Requirements | Incomplete or unclear project requirements leading to confusion. | Medium | Scope creep,rework, project delays. | High | Detailed requirements gathering, regular reviews. |
|  | Changing requirements | Frequent changes in project scope or objectives disrupting the project flow. | Medium | Increased costs , extended timeline. | High | Establish a change management plan. |
| Estimation Risks | Underestimation complexity | Understanding the complexity of tasks. | Medium | Missed deadlines, budget overruns | High | Break down tasks, seek mentor opinions on estimates |
|  | Timeline overruns | Project tasks longer than estimated, affecting the overall project timeline | Medium | Delays in project completion, Potential scope reduction | High | Regular progress tracking, adjust timeline as needed |
|  | Resource Overcommitment | Over Committing resources to asks leasing | Low | Decreased productivity, potential turnover | Low | Monitor Workload, balance task distribution. |

## **Appendix 2. Risk Indicators Table [CHANGES]**

| **Risk Type** | **Indicators** |
| --- | --- |
| **Technology Risks** | * Slow or unpredictable performance of project related software. * Delays in obtaining necessary software licences * Frequent complaints about software performance from team members * Increased frequency of system crashes or data loss * Inadequate software integration, causing reduced efficiency * Scalability issues due to increased data volume or resource constraints * High frequency of system outages or downtime * Data quality issues leading to inaccurate results or system errors * Inconsistent version control leading to code conflicts or bugs * Security breaches or unauthorised access to sensitive data. |
| **People Risks** | * Decline in team morale * Team members missing or not participating in meetings * Deterioration in team communication and relationships * Overload or burnout of key team members due to poor task distribution |
| **Organisational / Business Risks** | * Weak leadership or lack of clear direction from the project manager * Insufficient support from mentors * Delays in decision making or approvals from stakeholders * Shifting organisational priorities leading to deprioritization of the project. * Budget constraints impacting project resources and scope. * Market changes affecting the relevance or direction of the project. * Inadequate resource allocation leading to bottlenecks or delays. * Failure to manage changes in project scope or requirements effectively. * Policy changes disrupting the flow of the project. |
| **Requirements risks** | * Negative or in insufficient feedback from the client * High frequency of requirement changes * Misalignment between client expectations and project deliverables * Unclear requirements leading to rework |
| **Estimation Risks** | * Missed development milestones * Project lagging behind schedule as per the Gantt chart * Repeated failure to conduct regular team meetings. * Underestimation of time or resources needed for key tasks |

## **Appendix 3 Probability/ Impact Matrix [CHANGES]**

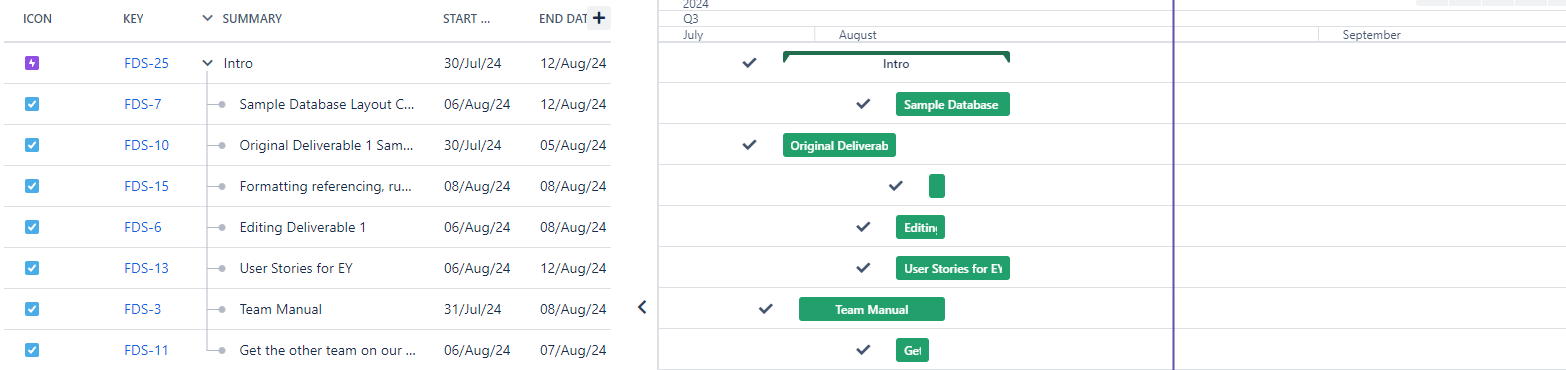
|  | **Impact** | | | |
| --- | --- | --- | --- | --- |
| **Probability** |  | **Low** | **Medium** | **High** |
| **High** | P1: Skills Gaps  R1: Ambiguous Requirements  R2: Changing requirements | P1: Skills Gap | R1:Ambiguous Requirements  R2:Changing Requirements  O7: Inadequate Resource Allocation |
| **Medium** | T4: Scalability Issues  O4: Market changes | P2:Team collaboration  P3: Resource Availability  T1: System downtime  T6: Algorithm Complexity  T9:Version Control Conflicts  O5: Inadequate change management  T14: Cross -Browser Compatibility issues | O1: Budget constraints  E2: Underestimation of Complexity  E3:Timeline Overruns  T5: Data Quality Issues  T7: Model overfitting  T10:Insufficient Data for modelling  T11: Bias in Data  T15: High Traffic Load |
| **Low** | E1: Resource Overcommitment | T3:Inadequate Software integration  E4: Missed Milestones  O3: Shifting priorities | O2: Policy Changes  T2: Data Security Breach  T8:Data Privacy Compliance  O6: Failure to meet regulatory compliance |

## **Appendix 4: Project Phases and Key Milestones**

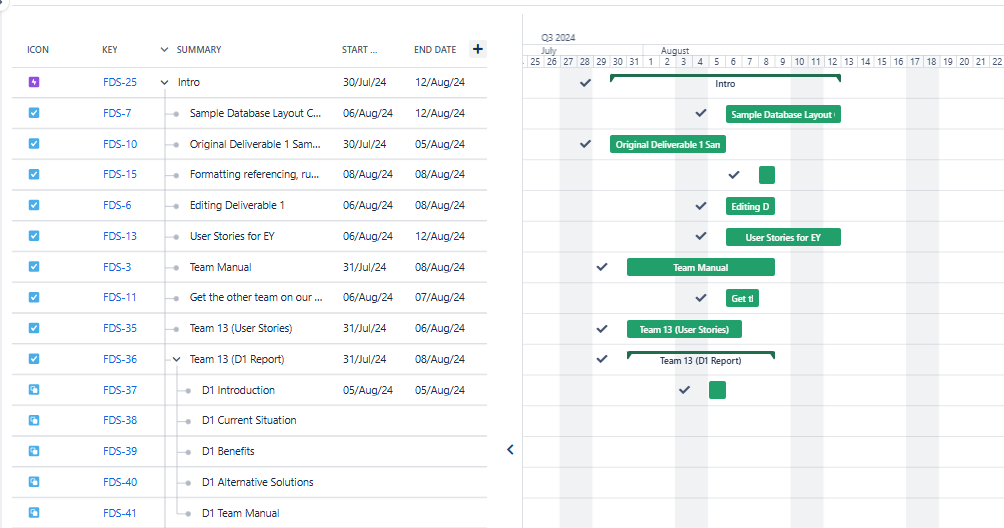
| **Phase** | **Activities** | **Milestones** | **Deliverables** |
| --- | --- | --- | --- |
| Sprint 1  30/07 - 12/08  2024 | * Create discord * Sample database layout creation * Create user stories * Submit Deliverable 1 draft to the client * Complete Deliverable 1 | * Combined discord with mentors and Team 13 * Finalising required columns for the dataset | Project:   * Feasibility Report * Team Manual   Product:   * User Stories |
| Sprint 2  13/08 -  27/08  2024 | * Start sourcing data * Create High-Level Solution Design (HLSD) * Create Options Analysis * Set up GitHub * Set up Azure Infrastructure * Start loading data to the bronze layer * Prepare test data file * Submit Deliverable 2 draft to client * Complete Deliverable 2 | * Obtaining 10,000 rows of data * Github set-up * Feedback from mentors for Deliverable 2 | Project:   * Project Plan * Requirements Document * Updated Team Manual   Product:   * Options Analysis Report * HLSD presentation * Test data file |
| Sprint 3  28/08 - 10/09  2024 | * Obtain more datasets * Create the Code Convention * Research on data synthesization methods * Produce Synthetic Data * Build Azure Infrastructure setup * Build CI-CD pipeline * Clean the datasets | * Cleaned dataset created * Synthesise 7,000 rows of data | Project:   * Individual Contribution form   Product:   * MVP options document * GitHub documentation |
| Sprint 4  11/09 -  24-09  2024 | * Find additional data sources * Clean and enrich dataset 2 * Clean and enrich dataset 3 * Test various imputation methods * Create a MVP Proposition document * Submit deliverable 3 draft to client * Find and test methods for creating synthetic data * Create synthetic data | * Create merge dataset * Send 20,000 rows to the AI team * Creation of MVP | Project:   * N/A   Product:   * MVP1 presentation * Test dataset |
| Sprint 5  25/09/- 8/10  2024 | * Complete deliverable 3 tasks * Refine data pipeline * Resolve Azure permission issues * Move project onto Azure | * Feedback from client on Deliverable 3 * Set up the Azure infrastructure | Project:   * Updated Deliverable 2 * Design Document * Test Cases Document * Prototype/MVP   Product:   * MVP2 options document |
| Sprint 6  9/10 - 22/10 | * Submit deliverable 4 draft to client * Create a MVP Proposition document * Complete Deliverable 4 * Resolve Azure pricing issue * Create an initial web app and develop it * Re-design web app layout * Create a .json formate database * Create a SQL server and database | * Feedback on client for Deliverable 4 * Initial user interface design * Feedback on prototype 2 * Revise Dataset with AI team feedback | Project:   * Updated Increment 1 (Deliverable 3) * User/Training Manual   Product:   * MVP prototype shown to client * User interface |
| Sprint 7  23/10 -  5/11  2024 | * Integrate AI model to the website * Do testing on the websites performance and functionality * Create wireframes * Update website design | * Create final Front end solution * Present initial design to clients | Project:   * Final Group Reflective Report * Project Presentation * Individual Contribution form   Product:   * Web App wireframes |
| Sprint 8  6/11 -  14/11  2024 | * Work on finalising website design * Create final presentation for client and partners | * Handover final project to client. | Project:   * Final Delivery of Project to Client   Product:   * New fraid detection system |

## **Appendix 5: Sprint 1 Gantt Chart**

Gantt Chart for Team 14



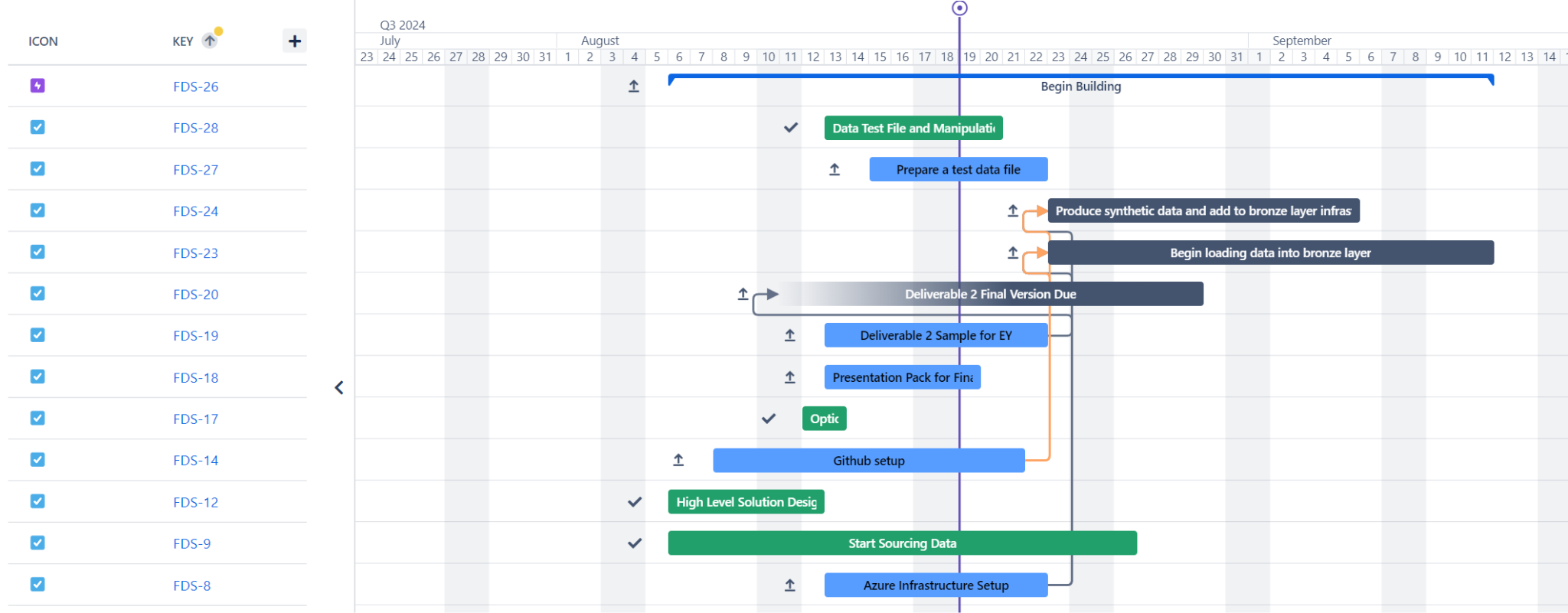
Gantt Chart combined with Team 13



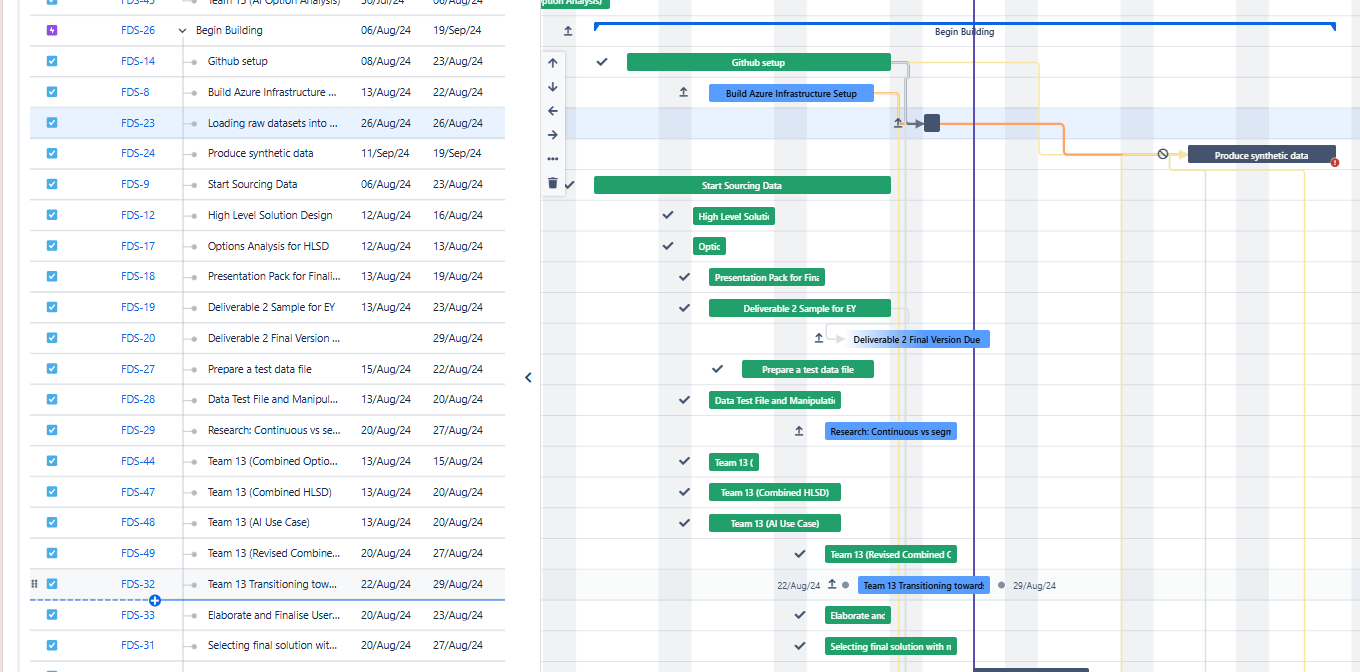
## 

## **Appendix 6: Sprint 2 Gantt Chart**

Gantt Chart for Team 14



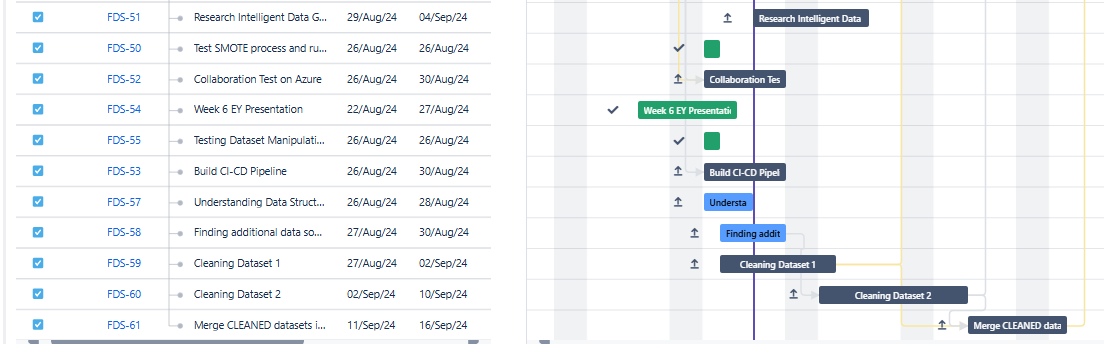
Gantt Chart combined with Team 13

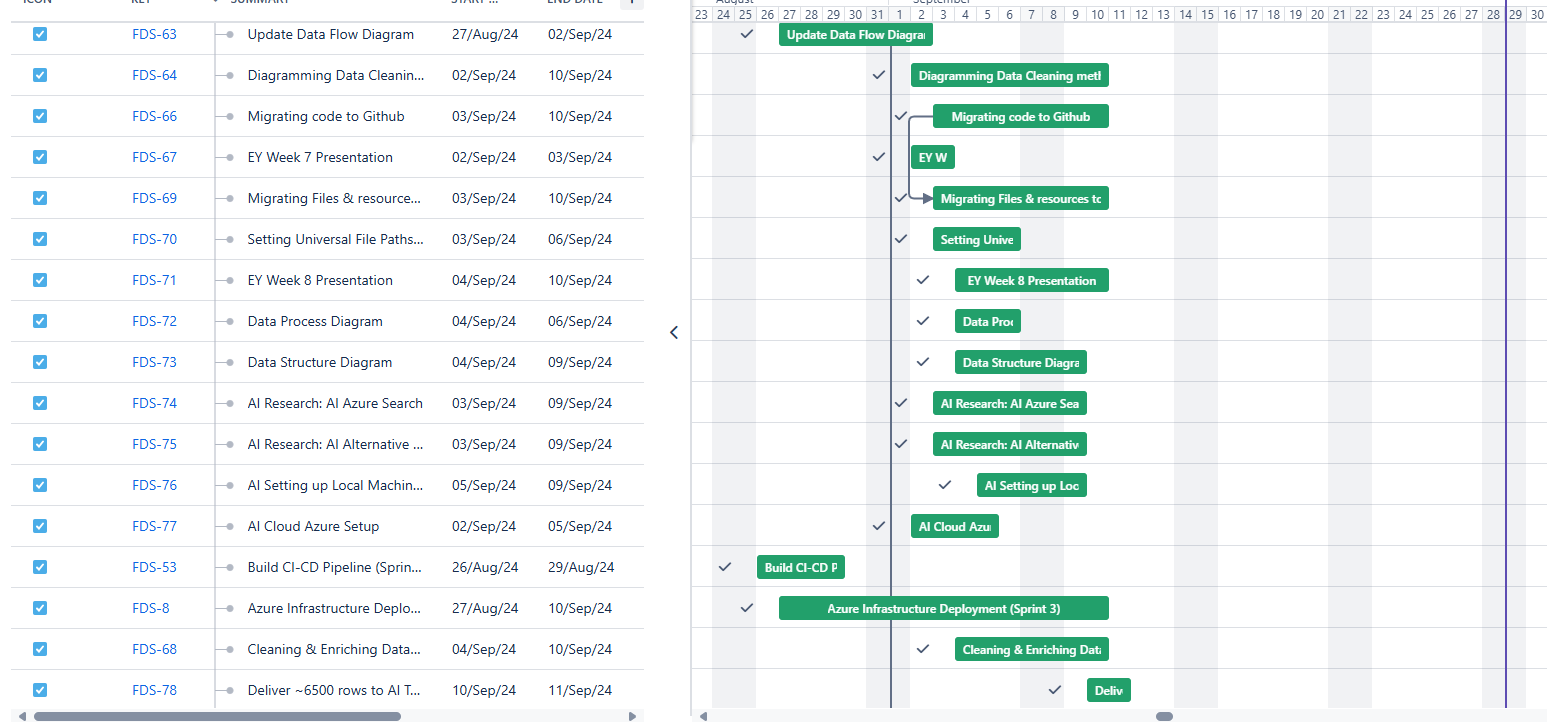


## 

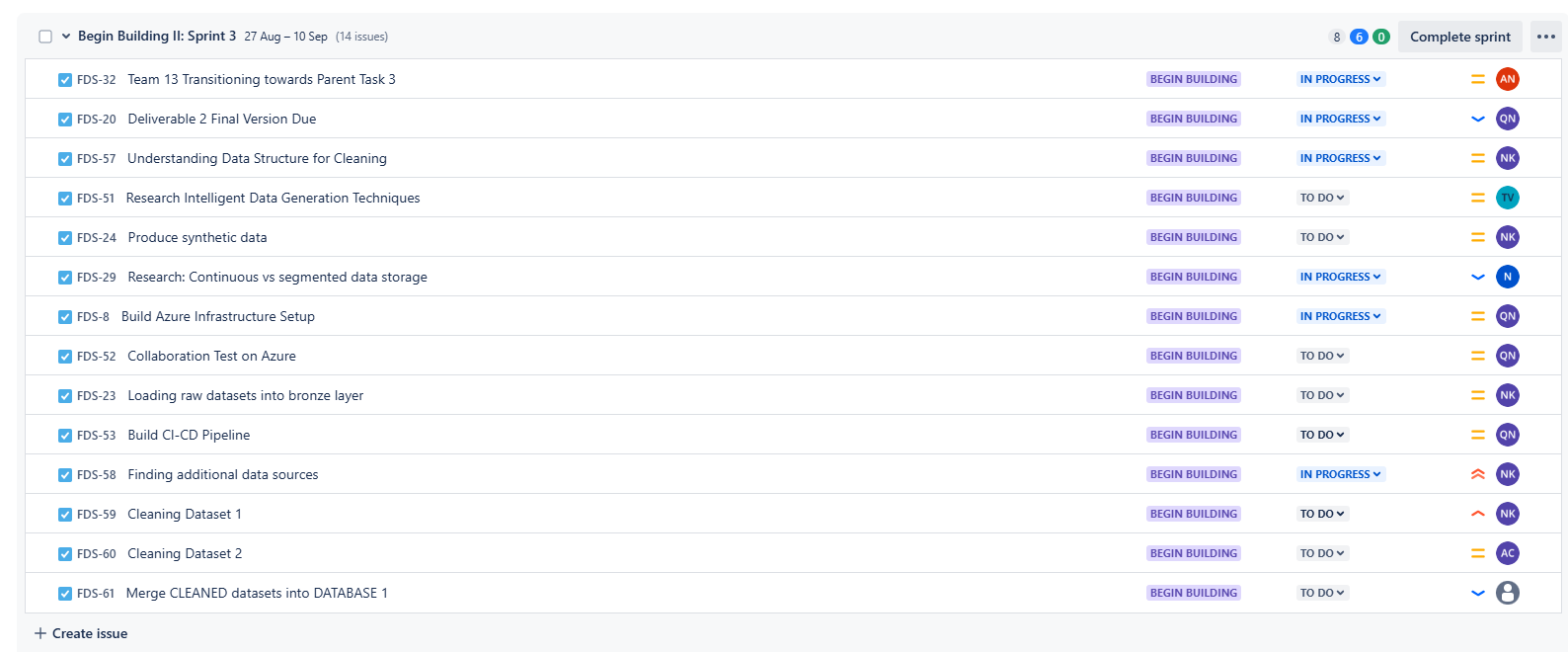
## **Appendix 7: Sprint 3 Gantt Chart**

Gantt Chart combined with Team 13



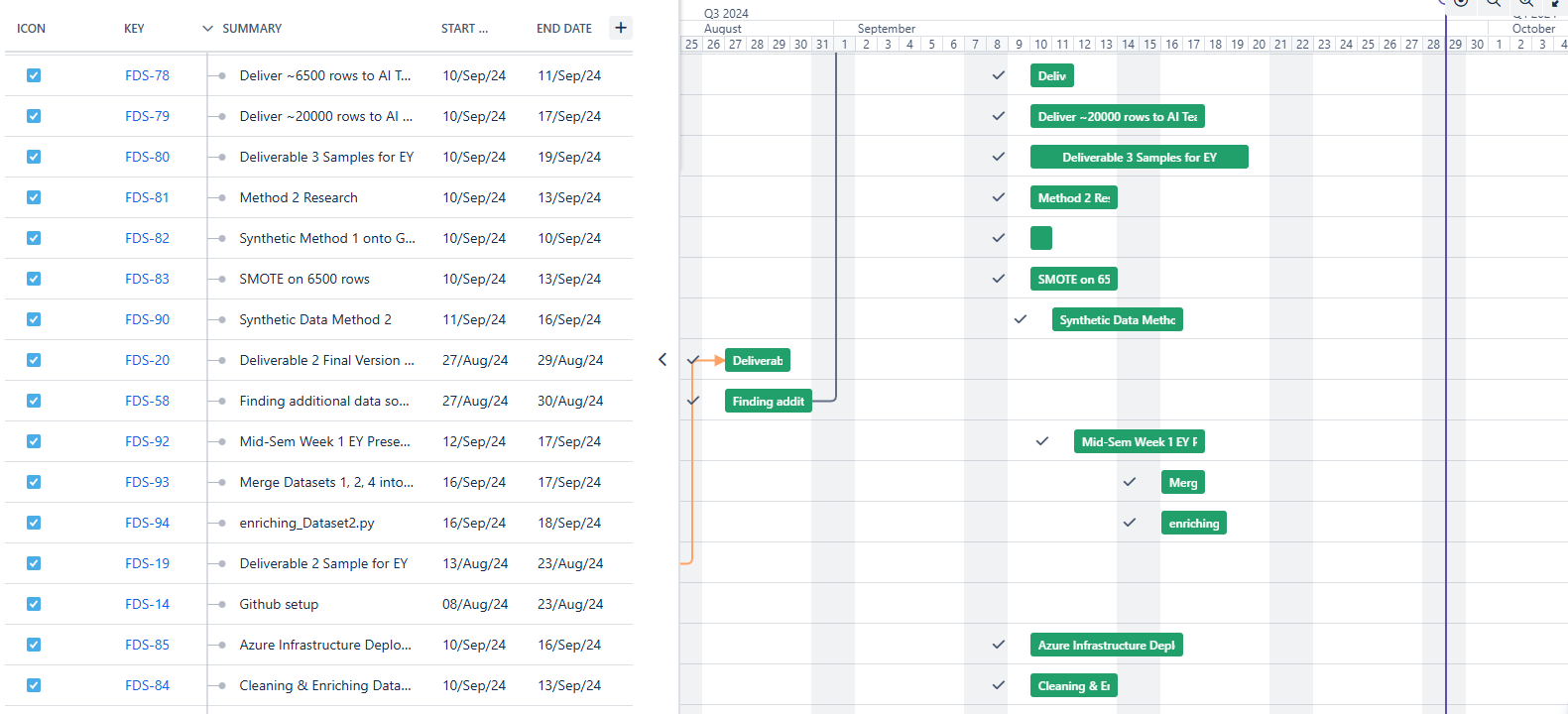


Sprint 3 Backlog



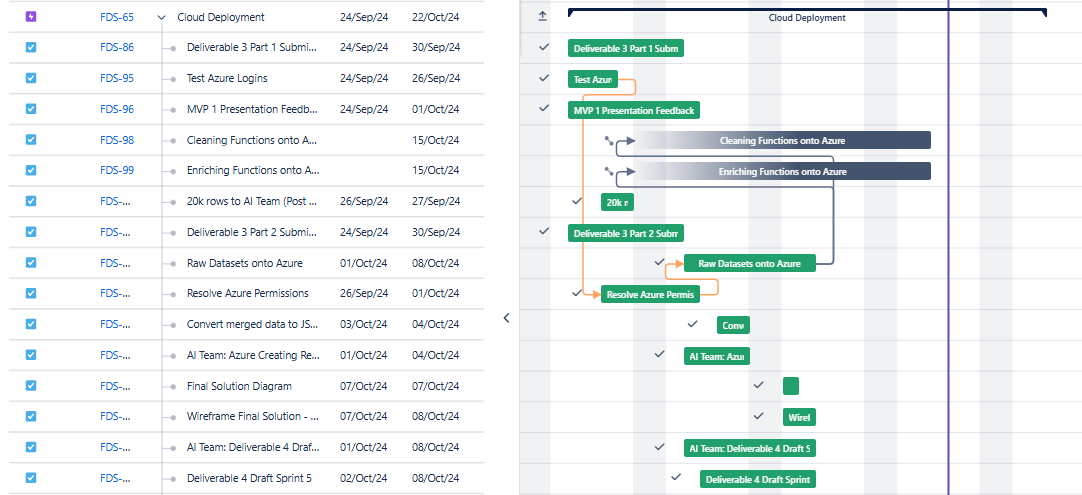
## **Appendix 8: Sprint 4 Gantt Chart**

Gantt Chart combined with Team 13



## **Appendix 9: Sprint 5 Gantt Chart [CHANGES]**

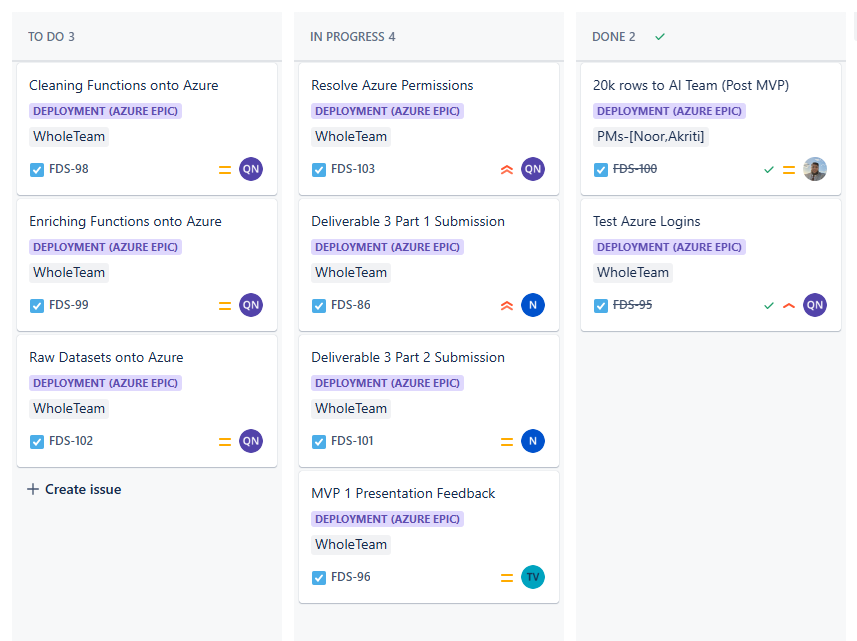
Gantt Chart combined with Team 13 **[NEW]**



Sprint 5 Backlog

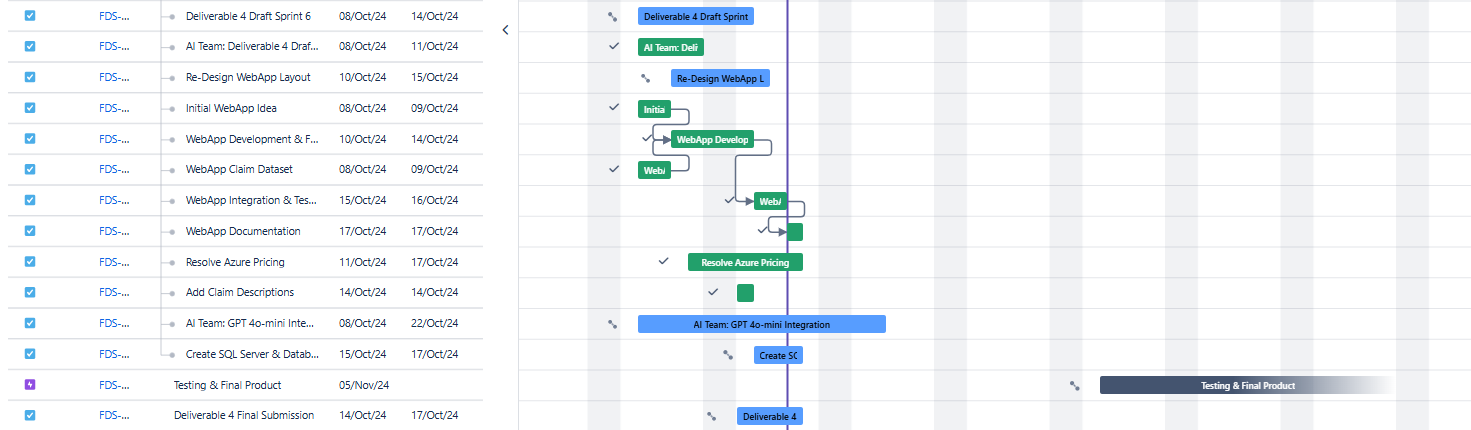


Sprint 5 board

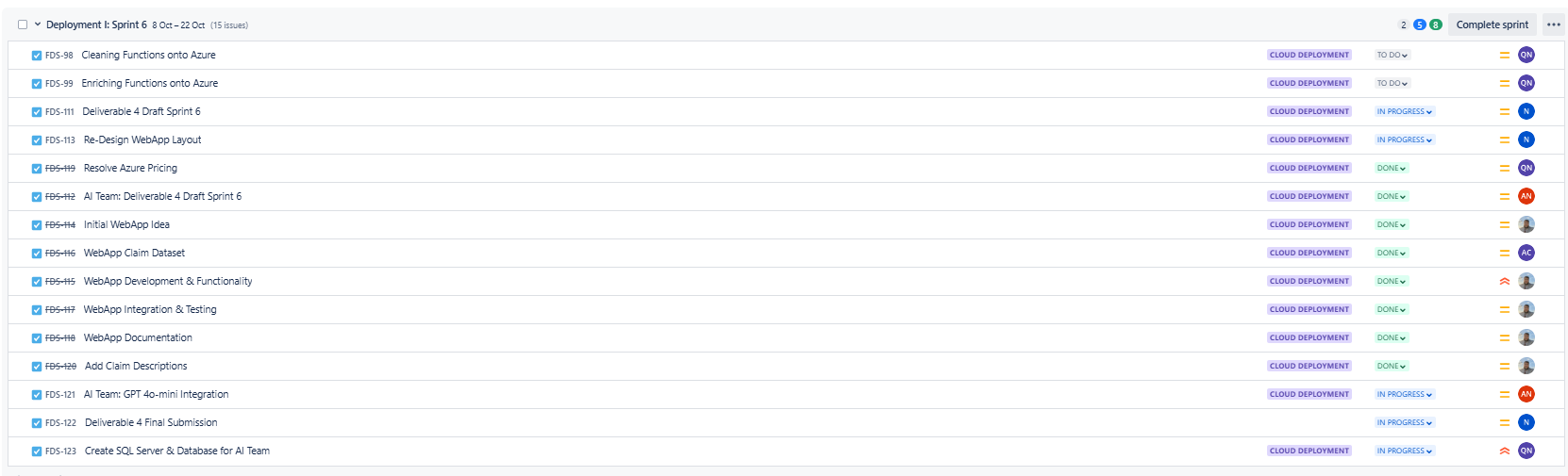


## **Appendix 10: Sprint 6 Gantt Chart [NEW]**

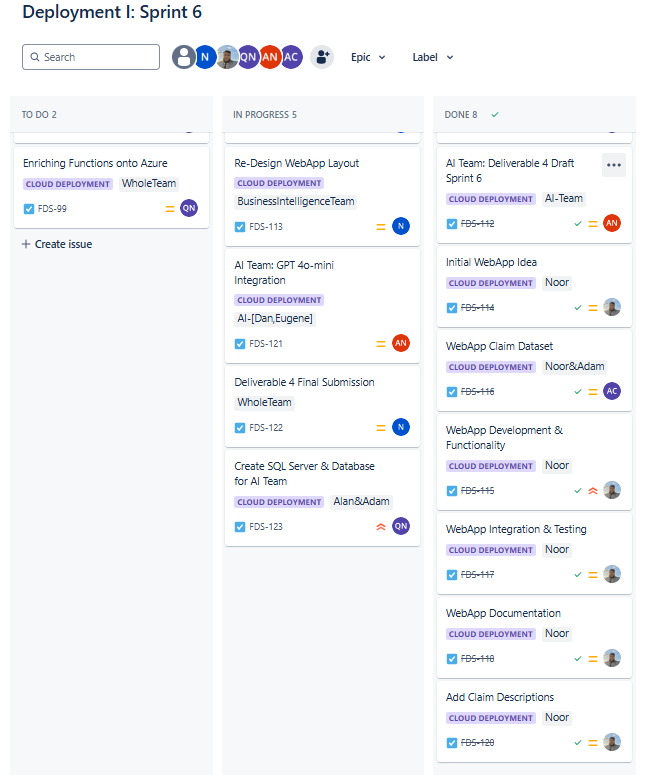
Gantt Chart combined with Team 13



Sprint 6 Backlog

****

Sprint 6 board



## **Appendix 10: Task Allocation Groups**

| **Team** | **Members** | **Tasks** |
| --- | --- | --- |
| Individual Member | Ninuri, Tashiya, Alan, Noorullah, Aasnayem(Adam) | Relating to the skill set of each member which is further explored in our Team Manual. |
| Business Intelligence Team | Ninuri and Tashiya | Research on aspects of the business to integrate with the product. |
| Dev Team | Aasnayem(Adam), Alan, Noorullah, Ninuri and Tash | Mapping dataset structure, pseudocode for cleaning, designing development pipeline |
| AI Team | Alan and Noorullah | Join with the AI team to help integrate the dataset with the AI model and assist in any technical issues that arise with the dataset and the AI model. |
| Web Team **[NEW]** | Noorullah, Ninuri and Tash | Create the web app design, develop, and conduct integration and testing |
| Whole Team | Ninuri, Tashiya, Alan, Noorullah, and Aasnayem(Adam) | Project-related tasks such as the Deliverables handing to the University. |

# 

| **UPDATED REQUIREMENTS AND SCOPING DOCUMENT**  **Team 14**  Revolutionise Claims Management: Unleash the  Power of GenAI for Peak Efficiency in the Insurance  Sector |
| --- |

# **TABLE OF CONTENTS**

[**Revision History Table 6**](#_2hy0duuzboz1)**2**

[**1. Data Understanding 6**](#_1yjkh1anmujp)**4**

[1.1 Initial Data Sources 6](#_nb88drf4p74s)4

[1.2 Data Collection Methods](#_j86cvkk8rvlu) 68

[1.3 Assumed Data Quality, Structure, and Completeness [NEW] 7](#_31r487wyurrz)0

[1.4 Mechanisms for Evaluation 7](#_wsa4qo3jwpe6)1

[**2. Data Preparation 7**](#_78jq27ck99z1)**1**

[2.1. Data Preparation Activities 7](#_cw9vkorct2yx)1

[2.2. Data Processing Pipelines 7](#_vsglpkaap9vo)3

[2.3. Different Preparation Pipelines 7](#_kjgxuavyoxm9)3

[2.4. Justification for Different Models 7](#_4bqm53ndl4c2)3

[2.5. Data Storage Mechanisms 7](#_zh4yyeka4p9s)4

[**3. Modelling 7**](#_2jxsxqh)**4**

[3.1. Modelling Techniques 7](#_ok62gylqhmrx)4

[3.1.1. Random Forest Classifier: 7](#_i534jd8at0dg)5

[3.1.2. Random Forest Regressor: 7](#_ou9mcaqtbq42)5

[3.1.3. Support Vector Regression (SVR): 7](#_37sedja8434b)6

[3.1.4. K-Nearest Neighbours (KNN): 7](#_15du7qhed2hc)6

[3.1.5. Relationship to the Chosen Model: 7](#_4702xt9sbv7y)7

[3.2. Evaluation Metrics](#_4vi4vz8c233c) 77

[3.3. Hyperparameter-tuning](#_hbtzssk9su7t) 78

[3.4. Bias and Fairness](#_b3aukivr97ci) 78

[3.5. Test Dataset](#_1y810tw) 79

[**4. Evaluation 8**](#_2xcytpi)**0**

[4.1. Evaluating Results 8](#_1ci93xb)0

[4.2. Evaluation Techniques 8](#_3ypoxuretb8u)1

[4.3. Review Process 8](#_3whwml4)1

[4.4. Next Steps 8](#_2bn6wsx)2

[**5. Deployment 8**](#_qsh70q)**2**

[5.1. Deployment Plan: 8](#_3as4poj)2

[5.2. Monitoring and Maintenance: 8](#_49x2ik5)3

[5.3. Final Report and Project Review: 8](#_147n2zr)4

[**6. References 8**](#_2wvhept83mj7)**4**

[**7. Feedback and Response 8**](#_o2z4lvrz5okw)**5**

# **Revision History Table**

| **Version** | **Last Update** | **Author(s)** | **Changes** |
| --- | --- | --- | --- |
| V2\_S1  Data Understanding | 28/09/24 | Noorullah | * For each column mentioned data type and column description * Mentioned the size of raw data file and how the different fields are split in the raw data * Added section 1.3. Pn the assumed data quality, structure amd completeness of the original dataset amd synthetic dataset * Incorporated mentor feedback by editing the way column descriptions are presented to be clearer to read |
| V3\_S1 | 17/10/24 | Noorullah | * No changes made |
| V2\_S2  Data Preparations | 25/09/24 | Adam | * Included the steps taken to handle outliers * Incorporated the various data transformation techniques * Discussed how the data was integrated with examples |
| V3\_S2 | 15/10/24 | Adam | * Removed the section called data transformation in 2.1 * Combined the sections for cleaning and enriching together in 2.1 * Updated data formatting with the new file formats of .json and SQL in 2.1 * In section 2.5 included information on back-up plans and security measures |
| V2\_S3  Modelling | 29/09/24 | Ninuri / Tashiya | * Changed modelling techniques to reflect the work done in section 3.1. * Added define evaluation metrics * Described the model training and prediction procedures * Expanded on evaluation parameters and hyperparameter tuning * Discussed bias assessment methods to ensure equitable outcomes * Updated the test dataset standards |
| V3\_S3 | 16/10/24 | Tashiya /  Ninuri | * Changed 3.4 to expand points on bias and fairness so it’s more clearer * Updated 3.5 to include the new data format requirements. |
| V2\_S4  Evaluation | 20/09/24 | Adam | * Added section 4.2. on evaluation techniques such as cross validation |
| V3\_S4 | 13/10/24 | Adam | * In section 4.2 further discussed why addressing null values in important and the trade-offs between dropping and imputing * In section 4.2. mentioned the deployment of Azure and its benefits * In section 4.4. expanded on our next steps to create a Web App |
| V1\_S5  Deployment | 28/08/24 | Noorullah | * No changed made to this section |
| V3\_S5 | 15/10/24 | Noorullah | * In section 5.1 added a new section that expands on our new deliverable to the client which is a user interface * In section 5.3 updated the section to include the user interface |

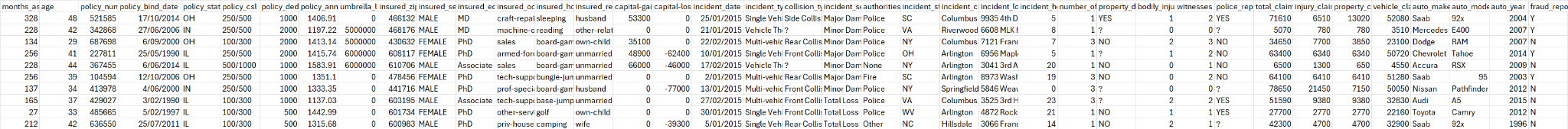
# **1. Data Understanding**

## **1.1 Initial Data Sources**

* Primary Dataset:
  + **Data Source:** 3 Kaggle Datasets. Dataset 1 with 1000 rows, Dataset 2 with 10000 rows (only 1575 relevant to automobile insurance), Dataset 3 with 10303 rows.
  + Content: Datasets include a variety of columns relevant to insurance claims including months\_as\_customer, age, policy\_number, policy\_state, incident\_type, total\_claim\_amount, and a crucial column, fraud\_reported, which indicates whether each entry is fraudulent.

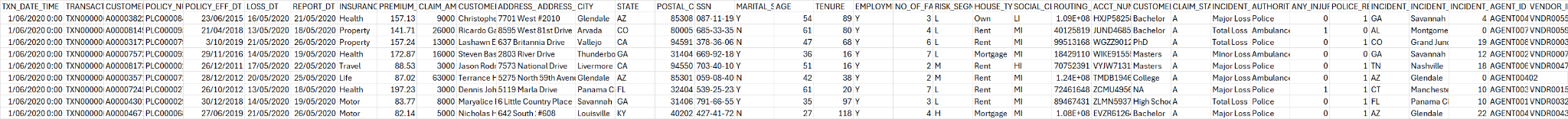
Detailed column Descriptions of each dataset:

**Raw Dataset 1: 1000 rows ~ 261 KB**



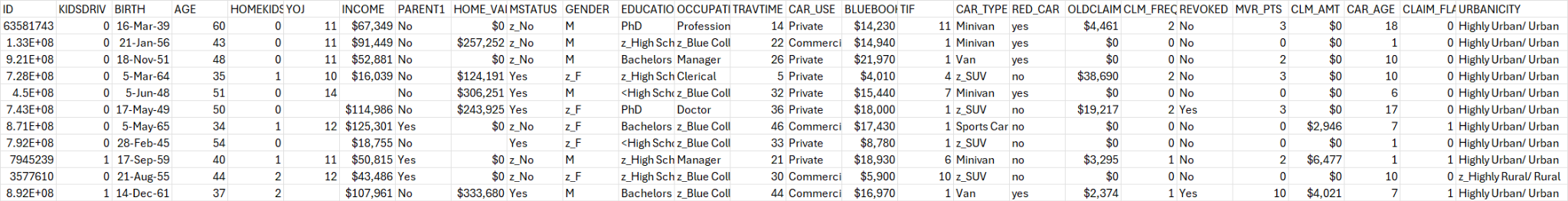
| Column Name | Data Type | Description |
| --- | --- | --- |
| months\_as\_customer | Numerical | Number of months the individual has been a customer |
| age | Numerical | Age of the customer |
| policy\_number | Categorical | Unique identifier for the insurance policy |
| policy\_bind\_date | Date | The date when the policy was created/bound |
| policy\_state | Categorical | The state where the policy was issued |
| policy\_csl | Categorical | Combined Single Limit for the policy |
| policy\_deductable | Numerical | The amount of deductible in the policy |
| policy\_annual\_premium | Numerical | The annual premium paid for the insurance policy |
| umbrella\_limit | Numerical | The umbrella limit in the policy |
| insured\_zip | Categorical | Zip code of the insured person |
| insured\_sex | Categorical | Gender of the insured person |
| insured\_education\_level | Categorical | Education level of the insured person |
| insured\_occupation | Categorical | Occupation of the insured person |
| insured\_hobbies | Categorical | Hobbies of the insured person |
| insured\_relationship | Categorical | Relationship status of the insured person |
| capital-gains | Numerical | Capital gains of the insured person |
| capital-loss | Numerical | Capital loss of the insured person |
| incident\_date | Date | Date of the incident |
| incident\_type | Categorical | Type of incident |
| collision\_type | Categorical | Type of collision |
| incident\_severity | Categorical | Severity of the incident |
| authorities\_contacted | Categorical | Whether authorities were contacted |
| incident\_state | Categorical | State where the incident occurred |
| incident\_city | Categorical | City where the incident occurred |
| incident\_location | Categorical | Specific location of the incident |
| incident\_hour\_of\_the\_day | Numerical | Hour of the day when the incident occurred |
| number\_of\_vehicles\_involved | Numerical | Number of vehicles involved in the incident |
| property\_damage | Categorical | Whether property damage occurred |
| bodily\_injuries | Numerical | Number of bodily injuries |
| witnesses | Numerical | Number of witnesses to the incident |
| police\_report\_available | Boolean | Whether a police report is available |
| total\_claim\_amount | Numerical | The total amount claimed |
| injury\_claim | Numerical | Claim amount for injuries |
| property\_claim | Numerical | Claim amount for property damage |
| vehicle\_claim | Numerical | Claim amount for vehicle damage |
| auto\_make | Categorical | Make of the vehicle involved |
| auto\_model | Categorical | Model of the vehicle involved |
| auto\_year | Numerical | Year of the vehicle involved |
| fraud\_reported | Boolean | Whether fraud was reported |

**Raw Dataset 2: 10000 rows ~ 2870 KB**



| Column Name | Data Type | Description |
| --- | --- | --- |
| TXN\_DATE\_TIME | Date/Time | The date and time of the transaction |
| TRANSACTION\_ID | Categorical | Unique identifier for the transaction |
| CUSTOMER\_ID | Categorical | Unique identifier for the customer |
| POLICY\_NUMBER | Categorical | Unique identifier for the policy |
| POLICY\_EFF\_DT | Date | The policy effective date |
| LOSS\_DT | Date | Date when the loss occurred |
| REPORT\_DT | Date | Date when the incident was reported |
| INSURANCE\_TYPE | Categorical | Type of insurance |
| PREMIUM\_AMOUNT | Numerical | Premium amount for the insurance policy |
| CLAIM\_AMOUNT | Numerical | Amount claimed |
| CUSTOMER\_NAME | Categorical | Name of the customer |
| ADDRESS\_LINE1 | Categorical | Address line 1 of the customer |
| ADDRESS\_LINE2 | Categorical | Address line 2 of the customer |
| CITY | Categorical | City of the customer |
| STATE | Categorical | State of the customer |
| POSTAL\_CODE | Categorical | Postal code of the customer |
| SSN | Categorical | Social Security Number |
| MARITAL\_STATUS | Categorical | Marital status of the customer |
| AGE | Numerical | Age of the customer |
| TENURE | Numerical | Number of years the customer has been with the insurance company |
| EMPLOYMENT\_STATUS | Categorical | Employment status of the customer |
| NO\_OF\_FAMILY\_MEMBERS | Numerical | Number of family members |
| RISK\_SEGMENTATION | Categorical | Risk category assigned to the customer |
| HOUSE\_TYPE | Categorical | Type of house the customer owns or rents |
| SOCIAL\_CLASS | Categorical | Social class of the customer |
| ROUTING\_NUMBER | Categorical | Customer’s bank routing number |
| ACCT\_NUMBER | Categorical | Customer’s bank account number |
| CUSTOMER\_EDUCATION\_LEVEL | Categorical | Education level of the customer |
| CLAIM\_STATUS | Categorical | Status of the claim |
| INCIDENT\_SEVERITY | Categorical | Severity of the incident |
| AUTHORITY\_CONTACTED | Boolean | Whether the authorities were contacted |
| ANY\_INJURY | Boolean | Whether any injury was reported |
| POLICE\_REPORT\_AVAILABLE | Boolean | Whether a police report is available |
| INCIDENT\_STATE | Categorical | State where the incident occurred |
| INCIDENT\_CITY | Categorical | City where the incident occurred |
| INCIDENT\_HOUR\_OF\_THE\_DAY | Numerical | Hour of the day when the incident occurred |
| AGENT\_ID | Categorical | Unique identifier for the insurance agent |
| VENDOR\_ID | Categorical | Unique identifier for the vendor |

**Raw Dataset 3: 10000 rows ~ 1544 KB**



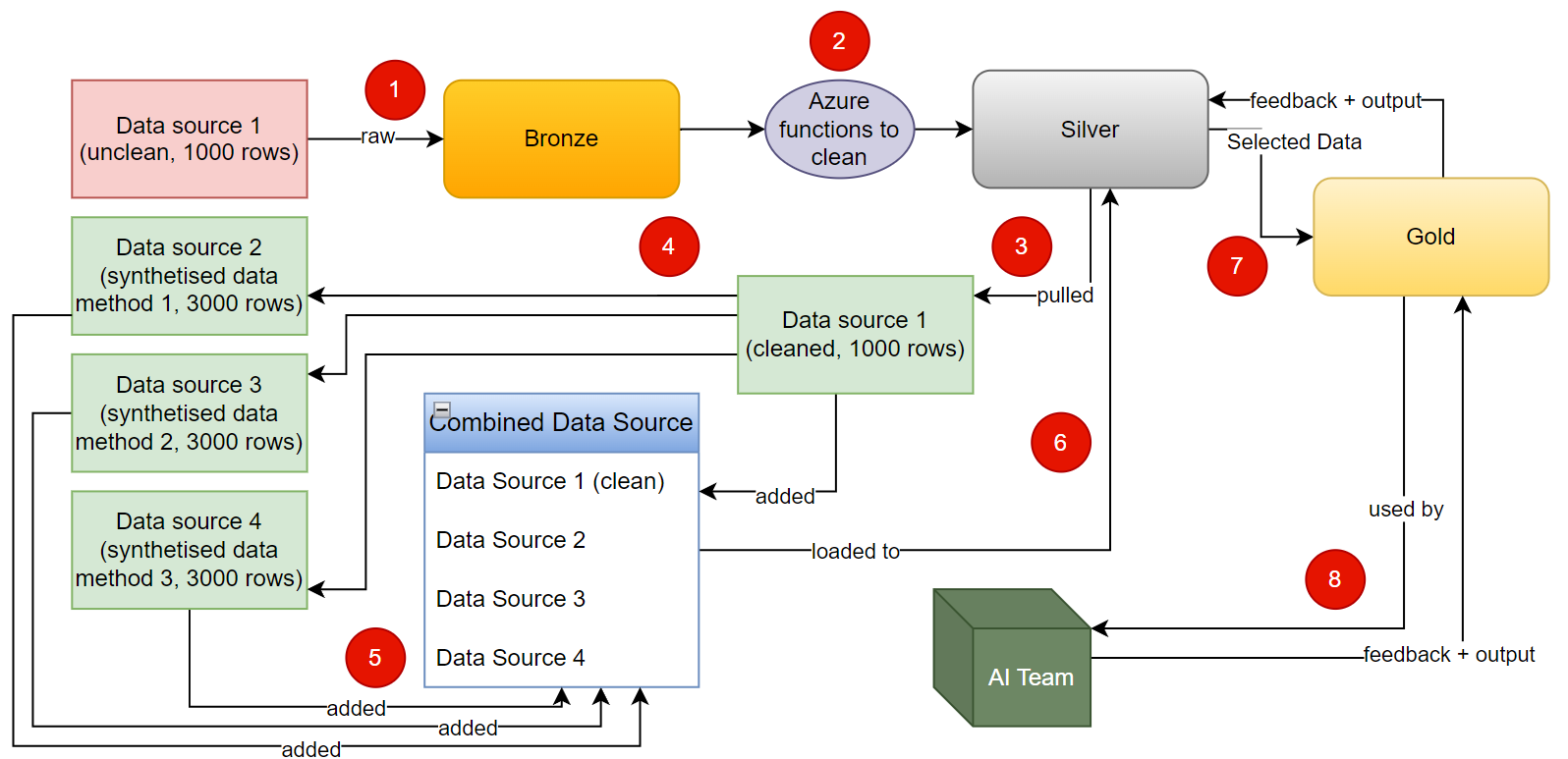
| Column Name | Data Type | Description |
| --- | --- | --- |
| ID | Categorical | Unique identifier for the customer |
| KIDSDRIV | Numerical | Number of kids driving in the household |
| BIRTH | Date | Birth date of the customer |
| AGE | Numerical | Age of the customer |
| HOMEKIDS | Numerical | Number of kids living in the household |
| YOJ | Numerical | Years on the job |
| INCOME | Numerical (Currency) | Income of the customer |
| PARENT1 | Categorical | Whether the customer is a single parent |
| HOME\_VAL | Numerical (Currency) | Value of the home |
| MSTATUS | Categorical | Marital status |
| GENDER | Categorical | Gender of the customer |
| EDUCATION | Categorical | Education level of the customer |
| OCCUPATION | Categorical | Occupation of the customer |
| TRAVTIME | Numerical | Travel time to work |
| CAR\_USE | Categorical | Whether the car is used for private or commercial purposes |
| BLUEBOOK | Numerical (Currency) | Value of the car |
| TIF | Numerical | Time in force of the insurance policy |
| CAR\_TYPE | Categorical | Type of car |
| RED\_CAR | Boolean | Whether the car is red |
| OLDCLAIM | Numerical (Currency) | Amount claimed in prior incidents |
| CLM\_FREQ | Numerical | Frequency of claims |
| REVOKED | Boolean | Whether the driver's licence has been revoked |
| MVR\_PTS | Numerical | Motor Vehicle Record points |
| CLM\_AMT | Numerical (Currency) | Amount claimed in the current incident |
| CAR\_AGE | Numerical | Age of the car |
| CLAIM\_FLAG | Boolean | Whether a claim was filed |
| URBANICITY | Categorical | Urban/rural classification |

* + **Widespread Availability:** It has been discovered that dataset 1 is identical across multiple sources, websites, and forums, suggesting that it is widely used and well-vetted (explained more in 1.3).
  + Despite dataset 1 being a good starting point and it seeming reliable due to multiple independent sources pointing to it, it alone is not enough for the AI team’s model. More data is needed (other datasets). Dataset 1 also needs to still be evaluated and prepared properly through being investigated, understood, mapped out and cleaned. Data cleaning would entail dropping those columns not needed, converting some metadata, and creating/augmenting new/existing columns. These processes might change over the course of the project. The data preparation will be to ensure it meets and services the project’s needs and can assist in fraud detection purposes.
* Additional Data:
  + **Synthetic Data Generation:** To expand the dataset, 8,000 additional rows will be generated using two different data synthesis methods (Remapping + Mathematical Sampling and CTGAN (Generative Adversarial Networks, a Neural Network based method). This will then be combined to give a dataset with a total of about 20,000 rows.

## 

## **1.2 Data Collection Methods**

* Kaggle Dataset:
  + The initial dataset is already collected and readily available for analysis (after preparation). The dataset was sourced (Mwitiderrick,2018)
  + Columns and Structure: The dataset includes the following columns:
    - Months\_as\_customer, age policy\_number, Policy\_bind\_date, Policy\_state, Policy\_csl, Policy\_deductable, Policy\_annual\_premium, Umbrella\_limit, Insured\_zip, Insured\_sex, Insured\_education\_level, Insured\_occupation, Insured\_hobbies, Insured\_relationship, Capital-gains, Capital-loss, Incident\_date, Incident\_type, Collision\_type, Incident\_severity, Authorities\_contacted, Incident\_state, Incident\_city, Incident\_location, Incident\_hour\_of\_the\_day, Number\_of\_vehicles\_involved, Property\_damage, Bodily\_injuries, Witnesses, Police\_report\_available, Total\_claim\_amount, Injury\_claim, Property\_claim, Vehicle\_claim, Auto\_make, Auto\_model, Auto\_year, Fraud\_reported
* Bronze Layer:
  + **Raw Data Storage:** The unclean original dataset (1,000 rows) is stored in the Bronze layer in its raw form. The bronze layer is the initial repository for all data before any cleaning or processing.
  + **Purpose:** Keeping the unclean data in Bronze ensures that the original dataset is always accessible for reference or reprocessing if needed. It also makes the project traceable and reproducible.
* Silver Layer:
  + **Data Cleaning:** The raw dataset from the Bronze layer is cleaned using Azure Functions. This cleaned dataset (1,000 rows) is then stored in the Silver layer as a cleaned version of the original data.
  + **Synthetic Data Combination:** The cleaned dataset from Silver is used as the basis for generating synthetic data (~ 9,000 rows). The combined dataset, now totaling 10,000 rows, is also stored in the Silver layer.
  + **Data Organization:** The Silver layer will contain different folders to organise the original cleaned data and the combined dataset.
* Gold Layer:
  + **Final Data Preparation:** The data in the Silver layer is further refined and stored in the Gold layer. This final data is fully prepared and ready for use by the AI team. It will contain columns relevant and useful to the AI model and will have a feedback loop from the AI Team which will trickle down to the silver layer if our data quality needs to be improved by more cleaning and refining.



*Figure 1: Data Layers Layout Diagram*

## **1.3 Assumed Data Quality, Structure, and Completeness**

* Original Dataset:
  + **Quality:** The original datasets, sourced from Kaggle and other widely referenced platforms, are well-vetted and frequently cited in various studies, reports, and articles. These datasets share many similarities across different platforms, often pointing back to the same original Kaggle source. For instance, the first dataset contains 1000 records with 39 diverse columns covering multiple categories of insurance-related data. Dataset 2 and Dataset 3 further add to the breadth and depth of data, with additional rows and varying fields. The large number of records and diverse data types help support data quality and robustness.
  + **Structure:** These datasets have a generally standardised structure, which simplifies their integration. However, care must be taken to address the nuances and differences between the datasets to ensure that all critical elements, such as those necessary for fraud detection, are accurately captured. For example, Dataset 3, being messier and less standardised, required more rigorous cleaning.
  + **Completeness:** While the original datasets are comprehensive in capturing their intended data points, they may not account for all specific scenarios or edge cases relevant to our project. The synthetic data generation methods applied further enhance the completeness of the datasets, providing additional rows to help cover such edge cases and better balance the data.
* Synthetic Data:
  + **Quality:** The synthetic data, generated from cleaned and enriched source datasets, is expected to be of high quality, as it draws directly from the original data's patterns and relationships. The data has been validated to ensure it introduces meaningful variance while maintaining consistency with key attributes, particularly the fraud column, which remains reliable across both synthetic methods.
  + **Structure:** The synthetic data mirrors the structure of the original cleaned and enriched datasets, ensuring that it integrates smoothly with Dataset 1, Dataset 2, and Dataset 3. This consistent structure ensures that the synthetic rows are compatible with the original datasets, helping to create a comprehensive dataset for AI model training.
  + **Fraud Distribution:** A key part of the evaluation process involved ensuring that the synthetic data accurately reflects the distribution of fraud cases. Both synthetic data generation methods produced a similar number of fraudulent records (around 400 in each method), which aligns with the distribution in the original data. This consistency reinforces the legitimacy of the synthetic data and ensures that fraudulent cases are well represented, providing a robust base for model training.
  + **Imbalances and Biases:** Since fraudulent claims are naturally underrepresented in the data, maintaining a balanced representation in the synthetic data was critical. While no oversampling methods like SMOTE were used, the synthetic data successfully preserved the fraud ratios, ensuring that machine learning models trained on this data will learn to detect fraudulent cases without skewed bias.

## **1.4 Mechanisms for Evaluation**

* **Data Quality Assessment:**
  + **Initial Dataset:** Perform a detailed exploratory data analysis on the initial datasets to identify biases, potential overfitting risks, and any embedded assumptions that might affect the project’s outcomes.
  + **Synthetic Data:** Evaluate the synthetic data against the original datasets to ensure it adds diversity and robustness. The synthetic data should not merely replicate common patterns but should introduce valuable variance. Any purposeful errors will also be evaluated to assess their impact on model training.
* **Data Structure Verification:**
  + Confirm that the dataset’s structure fully supports the analyses planned. Look for any structural limitations in the dataset and address these with synthetic data as needed.
* **Completeness Check:**
  + Ensure that the dataset captures all relevant scenarios, especially for fraud detection. The incorporation of SMOTE elements in the synthetic data generation will help ensure a balanced representation of fraud cases.
* **Additional Data Collection Requirements:**
  + As the project progresses, regularly assess whether more unique or tailored data is needed. Stay flexible to incorporate supplementary data that could provide additional insights or cover overlooked scenarios.

# **2. Data Preparation**

## **2.1. Data Preparation Activities**

* **Data Selection**: In order to select the data, we looked into various data sources and identified the relevant features to include in our dataset. In our case, we went ahead with one dataset and created 100 rows of data using information from that dataset. As a consultant, it is essential to follow the requirements of the client. Therefore, this dataset was chosen according to the problem statement relating to detecting fraudulent insurance claims and according to the requirements of the AI team and the client. We then increment from 100 rows to create 1000 rows of data using this dataset. We are using only one dataset currently as it is much more efficient to just use one base dataset as it will ensure all data is consistent. However, this can change in the future and we are open to include other datasets. After further discussion with the mentors and the clients, we decided to go ahead with 3 datasets instead of the sole dataset. Dataset 1 contains 1000 rows, Dataset 2 contains 1500 rows and Dataset 3 contains around 10800 rows.
* **Data Cleaning and Enriching [CHANGED]:** For data cleaning, we plan to fill in any missing data. We also plan to drop the columns which we will not need and possibly convert the categorical values into numerical ones for example: True will be converted to 1 and False will be converted to 0. We plan to clean the data and store it in the silver layer of our data structure. Outliers can make the model perform poorly.Outliers can be detected by performing statistical tests like the z-score test. Using the results from the test, we can either remove the outliers which are extreme or transform them if they are not too extreme. We used various data transformation techniques while enriching our dataset. We standardised our data so that our data is consistently scaled. An example of this is the gender column in dataset 3. The column had ‘M’ and ‘z\_F’ as the two possible outputs. So, we converted ‘z\_F’ to ‘F’ to match with the other datasets.
* **Data Synthesisation:** We plan to synthesise around 9000 rows of data using data synthesization techniques. We plan to use 3 different methods which will generate 3000 rows of data each. The methods are the “make\_classification” function in the scikit-learn library in python, GAN model, and AI based generation. After discussing with the mentors and the AI team, we decided to go ahead with 2 different synthesisation methods. Each method will produce 4000 rows of synthetic data. The first method is called Remapping and Mathematical Sampling where data is remapped and sampled to generate rows. The second method is called Conditional Tabular Generative Adversarial Network (CTGAN) where relationships and patterns are learnt to produce high quality, synthetic data.
* **Data Integration:** While we are not currently planning to integrate different data sources, we can still do that if we find another valid data source. If we had to merge, we would identify the common keys such as the ClaimID and merge the datasets. We would have to be consistent with the data. If we are using datasets with different formats, we would have to convert the separate formats into a unified structure. Therefore, JSON files will be converted to tables, XML files will be converted to CSV files. We have actually integrated 3 datasets and 2 synthesisation methods to combine it to one complete dataset with 20,900 rows approximately. The 3 datasets that are cleaned and enriched are merged first. Then, the synthesised methods are combined with the 3 datasets. This is done using the ‘pd.concat’ function in the pandas library. We also make sure that we have an index which is continuous starting from 1 to the end.
* **Data Formatting:** Data formatting ensures one particular format for our dataset. For our case we will have a CSV file. We select a CSV file as our dataset is relatively small and CSV is useful for small datasets. We have to ensure that all of the rows in the columns are consistent. For example, the date and time format should be the same for all the rows. **[NEW]** After liaising with the mentors and the AI team, the new model will require a dataset in JSON format and an SQL database. At this stage, we have provided a JSON file and we are working on creating an SQL database.

## **2.2. Data Processing Pipelines**

The data processing pipeline is mentioned above. We will have three different layers which are bronze, silver and gold. The bronze layer will have 1000 rows of data which will not be cleaned. While we are going from the bronze to the silver layer, the data will be cleaned using azure functions. While cleaning the data, we want to drop any columns we will not need and fill in any missing values. In the silver layer, we will also synthesise data using two different methods. Each method will synthesise 4000 rows of data. The methods are Remapping and Random Sampling, CTGAN(Conditional Tabular Generative Adversarial Networks). This will be stored back in the silver layer and then it will be stored in the gold layer where there will be a feedback loop. The feedback loop will also move to the silver layer where the changes will be made. The AI team(Team 13) will review and give suggestions on the dataset and the Data team (Team 14) will adjust the dataset accordingly.

## **2.3. Different Preparation Pipelines**

Different models will require different preparation pipelines. The pipeline will ensure that the data is not mishandled and is optimised. This will lead to an improved performance of the model. If a Decision Tree model was chosen, the data would not need to be cleaned that much as that model can handle raw data well. Outliers would need to be removed if they impact the model negatively. However, if a Neural Network model was chosen, the data would need to be normalised and high quality data would need to be provided. Neural Networks are more sensitive therefore, transformations need to be performed to remove outliers and categorical values would need to be converted to numerical values.

## **2.4. Justification for Different Models**

Having different pipelines/models will enhance efficiency. Unnecessary steps can be avoided if there are separate pipelines. For example, in a single pipeline, cleaned data might still go through a data synthesis step that's really only needed for raw data, which wastes time and resources. With separate pipelines, the cleaned data can skip this step altogether, making the process more efficient. Therefore, less resources and time will be required. Moreover, separate pipelines allow for easy adjustments if new methods are introduced later on. A tailored pipeline will be beneficial as it will optimise the performance of the model. This will improve the accuracy of the model and reduce the risk of biases.

## **2.5. Data Storage Mechanisms**

Our team has decided on using Azure Data Lake Storage Gen2 (ADSL2) to store all of our data. The data will be stored in layers as mentioned above. Azure Data Lake Storage Gen2 is being used as it is cost effective, secure, and scalable. Working in Azure will ensure the privacy of our data due to encryption and authentication while accessing azure. Moreover, as the dataset is not saved locally and saved on a cloud computing service which is Azure, it ensures that the dataset is backed up. Our dataset will not be that large therefore, the cost for ADSL2 will be cheap. Even if our dataset increases, that will not be a problem as ADSL2 can handle large volumes of data. Therefore, this setup ensures high performance and provides a good solution for data storage.

# 

# **3. Modelling**

For the project, we are working in collaboration with Group 13 for EY. In this collaboration, we are split into two teams, Team 14 who focuses on the data side with our main deliverable being a comprehensive dataset and cloud infrastructure, and Team 13 which focuses on the AI side and will be using our dataset to train their Generative AI model. The main outcome of using these AI modelling techniques is to find out if an insurance claim being made to NRMA is fraud or not fraud. This is a classification problem as it aims to categorise the data into predefined classes of fraud and not a fraud. Thus, the AI team will be using machine learning models like neural networks to classify the data. Therefore, while AI modelling does not fall within the scope of our project with EY, below we will be discussing some aspects of machine learning modelling that will be taking place in relation to cleaning the dataset.

## **3.1. Modelling Techniques**

Filling missing values is an essential part of preparing the dataset for subsequent analysis and model training. Various machine learning techniques including Random Forest Classifier, Random Forest Regressor, Support Vector Regression(SVR) and K-Nearest Neighbours(KNN), Can be utilised to estimate and replace these missing values by analysing the patterns present in the available data. These approaches tend to yield more precise imputations than conventional mean,median mode imputation as they take into account the intricate relationships among different features.

### **3.1.1. Random Forest Classifier:**

* **Reason it was Chosen:** it is good for imputing missing values in categorical variables through predicting classes based on patterns in the dataset.
* **Intended Outcome:** Impute categorical values
* **Training procedure:** the dataset will be split into rows with filled categorical values and rows with missing categorical values. The model will then be trained with the dataset with full categorical values and then predict the rows with missing values with the most likely option based on patterns observed in the dataset.
* **Imputation Process:** The most frequent class will be imputed into missing categorical rows

Benefits of Random Forest Classifier:

* Robust for categorical data imputation
* Can handle large datasets with missing categorical data
* Can handle high-dimensional data
* Can manage complex, non-linear relationships

Limitations of Random Forest Classifier:

* Requires sufficient training data with complete values
* Computationally expensive for large datasets

### **3.1.2. Random Forest Regressor :**

* **Reason it was Chosen:** To help estimate missing values in numerical variables. It builds multiple decision trees and combines their outputs to generate predictions, similar to the classification method.
* **Intended Outcome:** Impute numerical values
* **Training procedure:** This regressor is trained using rows with complete numerical data, allowing it to learn the relationships between the target numerical variable and other features. It leverages ensemble learning techniques to forecast missing values.
* **Imputation Process:** Missing numerical values are replaced with the predicted outputs from the regressor, which takes into account the effects of all other features in the dataset.

Benefits of Random Forest Regressor:

* Handles continuous and discrete variables
* Effective in detecting complex data patterns
* Non parametric
* Does not assume in particular distribution of the data

Limitations of Random Forest Regressor:

* It is resource intensive as it requires large computational power and memory.

### **3.1.3. Support Vector Regression (SVR):**

* **Reason it was Chosen:** Highly effective in imputing missing numerical values by maximising the margin of error around the predicted values to identify the best fit.
* **Intended Outcome:** Impute numerical values
* **Training procedure:** SVR is trained on the available data, using methods like the radial basis function(RBF) or polynomial kernels to account for non-linear relationships. The model minimises the difference between actual and predicted values within a set margin, ensuring a good fit.
* **Imputation Process:** When dealing with missing continuous values, the SVR model predicts the most likely value for each gap by analysing the relationships between the existing data points and features.

Benefits of Neural Networks

* SVR works well in situations with non linear interactions and high dimensional data which works great with simpler methods struggle,
* Flexibility allows it to handle more complex data structures.

Limitations of Neural Network

* SVR can be sensitive to extreme values and may require substantial computing power for larger datasets.
* Fine tuning parameters like the type of kernel and regularisation is necessary for optimal performance.

### **3.1.4. K-Nearest Neighbours (KNN) :**

* **Reason it was Chosen:** KNN is a simple yet effective method for filling in missing data, suitable for both numerical and categorical values. It works by finding by finding
* **Intended Outcome:** Impute both categorical and numerical values.
* **Training procedure:** KNN does not require formal training. Instead, it calculates the similarity between data points with gaps and those with complete information. For numerical data, it averages the nearest neighbours, and for categorical data, it chooses the most frequent category.
* **Imputation Process:** For each Missing entry, KNN identifies the closest complete data points and fills in the gaps by considering the values from those neighbouring records.

Benefits of K-Nearest Neighbours(KNN):

* Straightforward to apply
* Works for various types of data without needing assumptions about distribution
* Adaptable to different datasets and types of variables

Limitations of K-Nearest Neighbours(KNN):

* Time consuming when it comes to large datasets, as the algorithm compares each point to others
* Deciding the right number of neighbours(K) and similarity measure is crucial for getting accurate results.

### **3.1.5. Relationship to the Chosen Model:**

The mentioned modelling techniques were chosen due to their abilities to be able to handle large and complex datasets, along with its established history in fraud detection cases. The desired result is to ensure accuracy, precision and recall of fraudulent transaction detection while reducing false positives. Each chosen model is projected to contribute unique strengths:

* **Random Forest Classifier:** Able to handle large datasets and high dimensional data. Is also able to handle complex, non linear relationships.
* **Random Forest Regressor:** For imputing discrete and continuous variables by detecting complex data patterns.
* **Support Vector Regression (SVR):** Is able to work great with nonlinear interactions and high dimensional data. Can handle complex data structures.
* **K-Nearest Neighbours (KNN):** Straightforward approach that can adapt to different datasets and different types of variables which offers flexibility.

## **3.2. Evaluation Metrics:**

To gauge the performance of the imputation techniques, the following metrics will be applied:

* **Mean Squared Error(MSE):** This is used for numerical data, MSE calculates the average of the squared differences between the actual values and the imputed ones, which helps to assess the accuracy of the imputations,
* **Mean Absolute Error (MAE):** This is used for numerical imputations, MAE provides a more intuitive measure of error by averaging the absolute differences between the actual and imputed values.
* **Accuracy:**This metric evaluates how often the imputation model correctly predicts missing categorical values, offering a simple way to measure success for categorical data.
* **F1-Score:** Particularly useful for categorical variables, the F1- score balances precision and recall, ensuring the imputation method works well even with imbalance data.
* **Cross-Validation:** To test the consistency of the imputation models, techniques like K-fold cross validation will be applied. This helps ensure the models performance holds steady when applied to different parts of the dataset.

## **3.3. Hyperparameter-tuning**

To enhance model performance and boost the accuracy of imputations, fine tuning of hyperparameters will be conducted in the following areas:

* **Random Forest Classifier/ Regressor:** Critical parameters to adjust include the number of trees, tree depth, the minimum number of samples required to split a node and the number of features used for splits.
* **SVR:** Key aspects to optimise include selecting the appropriate kernel(e.g. RBF or polynomial) , setting the regularisation strength (C) and determining the epsilon value, which controls the error margin.
* **KNN:** The primary factors to refine are the numbers of neighbours (K) and the type of distance metric to ensure better imputations.

To find the optimal hyperparameter settings for each model, Grid search or random search methods will be utilised.

* **GridsearchCV -** creates a grid of all the possible hyperparameter combinations and selects the combination with the nest performance by evaluating each combination through cross-validation.
* **RandomSearchCV** - randomly selects a predefined number of combinations from the distribution and the best combination is selected through cross validation

## **3.4. Bias and Fairness [CHANGED]**

When handling missing values, it’s important to be mindful of potential biases that may arise, especially if the missing data is not randomly distributed (eg. certain demographic groups may be more missing entries). To address this:

* **Bias Detection:** Make sure to check how missing data is spread out among various demographic categories like age or gender to find any differences that stand out among them all. Once you've filled in the missing data gaps are filled in with estimates or imputation techniques review the outcomes to make sure that no particular group shows more errors or discrepancies than others. Tracking performance measures (such, as accuracy and precision) across these groups can point out any biases that might exist in the models forecasts.
* **Bias Mitigation:** To address bias concerns effectively several approaches can be implemented:
  + **Demographic Sensitive Models:** To better address missing data it may be helpful to train imputation models for various groups, with differing patterns of missing data ensuring that the specific traits of each group are accurately represented.
  + **Weighted Imputation**: Adjust the data weighting during imputation to address groups with representation and more missing data effectively aiming to balance the influence of each group, in the models overall outcome.
  + **Fairness Constraints**: Ensure fairness by adding constraints that adjust predictions to minimize biases and promote outcomes for all groups.

Through identifying and addressing bias, during the imputation process we can guarantee that the model produces impartial and equitable results preventing any skewed outcomes that may unfairly impact certain groups.

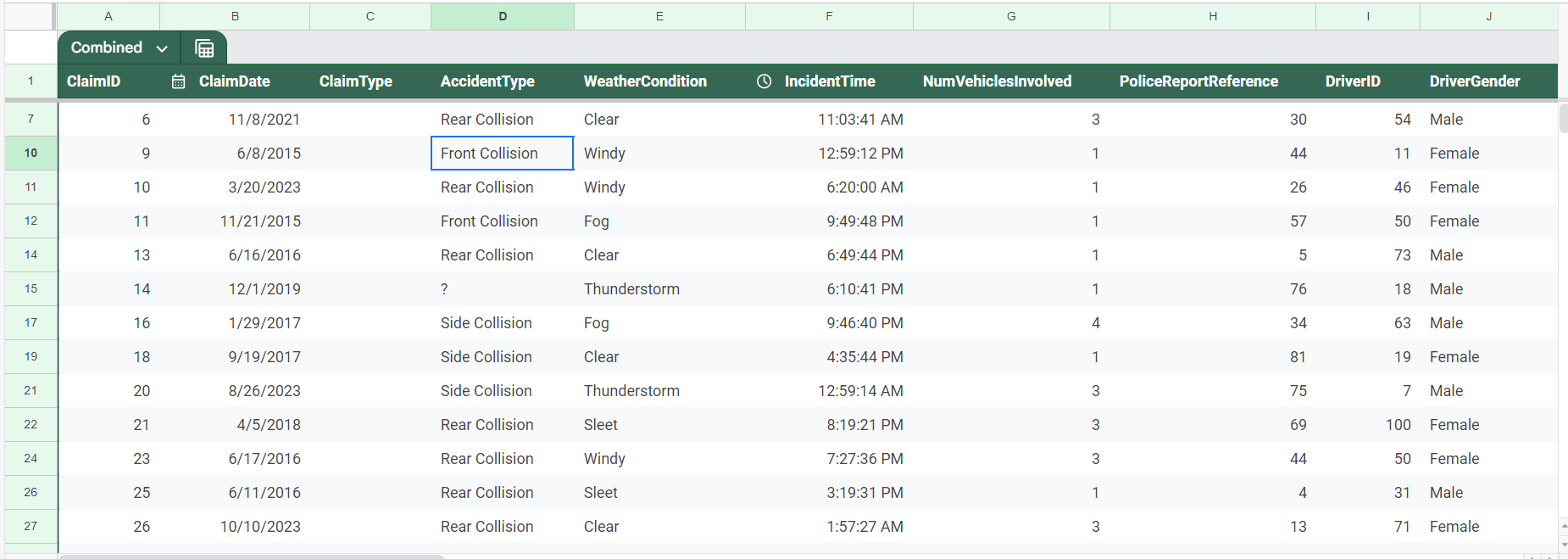
## **3.5. Test Dataset**

The mentioned modelling techniques above all need test a dataset with the following industry standards:

* **Balanced Dataset:** having a dataset with an even distribution of fraudulent and non fraudulent claims ensures that the model can effectively differentiate between the two classes. To verify if a claim is fraudulent or not, the dataset will have a column which will specify whether the claim made is “fraud” or ”not fraud” enabling the AI model to identify patterns in these sets of data.
* **Representative Features:** The features included in the test dataset should be relevant in helping the model detect fraudulent claims, such as police report, claim type, accident type etc.
* **Labelled Data:** The test dataset should be labelled as “Fraud” or “ not fraud” in each instance to evaluate the models accuracy, precision, recall and overall performance.
* **Recent Data:** The test data should be up to date to reflect current fraud patterns ensuring the model stays relevant and effective,

However, with further discussion and testing with the AI team have found that there are a few more requirements:

* **Missing values:** there must be no missing values as the model does not train properly if there are any.
* **Column names:** the names of columns have to be uniform across all datasets including names and upper/lower casing. If not, the code has to be changed to reflect each different column name.
* **Numerical values:** the given data should have no categorical values and must be transformed to numerical values.
* **[NEW]** Data format: The AI team has also mentioned that the dataset needs to be given in formats to easily fit into their model training processes, such, as;
  + **Json**: The data set should be organized in JSON format to make sure that each entry is correctly structured for use, with AI processing tools.
  + **SQL:** The dataset should also be accessible in SQL format, for querying and manipulation to facilitate efficient data management and transformation.



***Figure 2: Example of test data given to AI Team***

# **4. Evaluation**

## **4.1. Evaluating Results**

**Null values:**

While reviewing our results, we should ensure that any rows missing any data should be null and the number of null values should also be counted. This will provide a clear understanding about the context of missing data. To solve the case of the null values, we can drop the observations with missing values. We can also use other data imputation methods such as mean, median, KNN to fill in the null values. **[NEW]** Addressing the null values is important as training a model with a lot of empty values will result in a poor model.

**Volume of Data:**

When we evaluate the volume of data, several details would need to be considered. The overall volume of data can impact the performance of the system. However, for our case, the volume of data is not extensive enough so performance should not be affected. While considering the volume of data, we have to ensure that the infrastructure is scalable. **[NEW]** We are storing our data is Azure which will ensure that our infrastructure is scalable. The number of rows in our dataset can be increased, therefore, it has to be ensured that our system is scalable which it is in our case.

**Data Diversity:**

We have to check and analyse the variety of data. If there is a vast range of data, we can conclude that the data is diverse which will help while training the model as there is more to consider and make a decision. A low diversity of data will not be ideal as it will not consider most situations. A diverse dataset will help to avoid bias that can result in inaccurate outcomes while training the models. It is critical that the data is diverse as diverse data will provide a clearer picture which will lead to accurate outcomes.

**Gold Layer Standards:**

The Gold Layer is our most important layer. In the gold layer, the dataset will be well refined so that it can be trained by the AI team’s model. So, while evaluating the results, it is important to ensure that the data in the gold layer is consistent and organised. So, the data in the gold layer should be cleaned and there should be any biases. The gold layer data should be ready for analytical processes as the machine learning models will be trained on this dataset. So, we have to assess the performance of the gold layer by checking the data retrieval time and query speed. Moreover, the gold layer will also store the output from the AI team. So, it is crucial that the gold layer is capable of handling that process.

**Quality of Each Column:**

It should be ensured that each column represents legitimate data. For example, the dates should be actual calendar dates, times should be in AM or PM according to the clock. Most importantly, the columns should represent relevant data. While cleaning the data, irrelevant columns should be dropped. Moreover, the data should be consistent. The rows in the columns should follow the same method. For example, the dates should be in DD/MM/YYYY format. Each column should be documented which can explain the data types, ranges, transformations, etc and the columns should be clearly explained so that it is easy to understand.

## **4.2. Evaluation Techniques**

We can add additional evaluation techniques to ensure that the performance of the model is good. We can do this by using cross validation techniques which will split the training and the testing data. One specific example of this can be the K-Fold Cross Validation which is a common technique used to split the data. In this process, the dataset is divided into K equal parts where K is the number of equal parts or folds that the dataset is divided into. K can be chosen by us and it determines the number of times the model is trained and tested. After all the training and testing is done, the overall result is averaged to give a more accurate result for the model. Another cross validation technique that can be used is the Stratified K-Fold Cross Validation technique where the concern of imbalanced datasets can be addressed. For example, if the amount of fraudulent cases is insufficient in our dataset, we can use Stratified K-Fold Cross Validation to ensure that each fold has a similar distribution and balance the ‘Fraud’ column. These cross validation methods ensure that the model is reliable and the model is not dependent on a singular train/test split. Multiple splits will help improve the accuracy of the model.

## **4.3. Review Process**

While reviewing the process, it is important to ensure that the process follows the business requirements set by the client. The business rules set by our client were using a GenAI Solution, using Azure Services, Anonymise personal data, a front-end user interface, and Agile Delivery. The dataset will be reviewed by the Data team and then by the AI team who will give feedback and then a feedback loop will form to enhance the dataset. There should be data validation checks to ensure that the data is accurate. The whole process should also be documented so that each and every step in the process can be explained. The code used should also be reviewed to ensure that the process is not error prone. Finally, there should be clear communication between the Data team and the AI team and also with the client to clear any confusion and streamline the process.

## **4.4. Next Steps**

If the results are positive after the review process, we can scale up and add more rows to the dataset. Adding more rows will provide a more accurate solution as there are more rows of data to consider. As our current infrastructure is made to handle larger volumes of data, this should not be a problem and the process will be smooth. However, if the results are not positive after the review process, it may be worth revisiting the strategy and try to fix it or use a different methodology. Collaborating with the AI team and the clients will be crucial to discuss the next steps and setting up a plan for the methodology will be essential. **[NEW]** As a next step, we have created a WebApp which is integrated with the AI model. Essentially, the WebApp can get a claim and provide a fraud risk score and the reason behind the fraud based on the AI model. At this stage, we have not connected the WebApp with the model yet however, this should be done in the near future.

# **5. Deployment**

## **5.1. Deployment Plan:**

* **Deliverables to the Client:**
  + **Gold Layer Dataset:**
    - A fully processed and refined dataset stored in the Gold layer, ready for AI model training.
    - This dataset is cleaned, enriched, and prepared, ensuring optimal use for fraud detection in insurance claims.
    - Key features are extracted from historical data with a claim identifier, providing a comprehensive dataset for model training.
  + **Documentation**:
    - Detailed documentation outlining the entire data preparation process.
    - Includes steps for data cleaning, synthesis, and integration.
    - Guidelines for accessing, querying, and using the dataset.
    - Instructions for collaboration between the client’s team and the AI development team during model training.
  + **Training:** 
    - Training sessions for the client’s team on how to interact with the Gold layer dataset.
    - Walkthroughs of the documentation to ensure a full understanding of the data preparation process.
    - Collaboration-focused workshops on working with the AI team during the model development phase.
  + **[NEW]User interface:**
    - A user-friendly website has been designed to empower claims officers to effectively interact with the data and the AI model, enabling real-time fraud detection. This intuitive interface simplifies the workflow for the officers, helping them make quicker, data-driven decisions regarding the status of insurance claims. The layout and features are tailored to ensure that even non-technical users can easily navigate and access the insights generated by the AI model.
    - The interface includes a dashboard that serves as the central hub for managing claims. The dashboard provides a comprehensive overview of all open and pending claims, along with AI-generated fraud risk scores. It also highlights important alerts for claims that require immediate attention, ensuring that claims officers can prioritise high-risk cases. This centralized view helps streamline the claims process and allows officers to manage multiple cases efficiently.
    - Additionally, the website offers search functionality that allows claims officers to retrieve both past and current claims by entering a Claim ID or selecting a specific claim status. This feature enables officers to quickly access the relevant details of a claim, review its history, and make informed decisions. The search function ensures that claims management is not only efficient but also comprehensive, covering both ongoing and previously closed claims.

## **5.2. Monitoring and Maintenance:**

* **AI Model Performance Monitoring:**
  + The AI model, expected to be trained using OpenAI’s GPT-4o mini, will be closely monitored for performance metrics such as test accuracy. With the threshold acceptability being around 85-90%, if the model's accuracy falls below it, the team will work together to assess this issue.
  + This process will include examining the AI models logs and parameters while also reviewing the quality of the underlying data.
* **Data Re-evaluation and Refinement:**
  + **Inaccuracy Handling:**
    - If the AI model’s accuracy is found to be suboptimal, the data team will revisit the Silver layer, where a version of the combined dataset can be pulled for further cleaning and refinement. This refined dataset will then be stored in a new folder within the Silver layer and subsequently moved to the Gold layer for retraining the AI model.
  + **Overfitting Prevention:**
    - In cases where the AI model appears to be overfitting or performing too accurately, indicating a lack of diversity in the data, the data team will generate additional synthetic data. This may involve deliberately introducing errors or variations during the synthesis process to enhance the dataset's diversity.
* **Data Expansion:**
  + To improve the robustness of the AI model, the dataset may need to be expanded. This could involve generating more synthetic data or employing different synthesis techniques to create a larger, more varied dataset. If a synthesis method is found to be insufficient, alternative techniques will be explored to ensure the highest quality of synthetic data.

#### 

## **5.3. Final Report and Project Review:**

* **Final Report:**
  + A comprehensive final report will be produced, summarising the entire project, including data preparation, synthesis processes, AI model training results, and any refinements made during the deployment phase.
* **Project Review:**
  + A Comprehensive review of the project will be carried out, which will include assessing the effectiveness of the deployment, if the data is of quality and the overall performance of the AI model. Lessons learned will be documented to inform future projects.
* **[NEW] User Interface:** The user interface (UI) will be evaluated for its usability, accessibility and overall user experience. Feedback from clients will be used to identify any potential improvements or enhancements. This evaluation will help ensure that the UI meets user needs and facilitates efficient interaction with the system.

# **6. References**

Mwitiderrick. (2018). insurancedata/insurance\_claims.csv at master ·mwitiderrick/insurancedata. GitHub. <https://github.com/mwitiderrick/insurancedata/blob/master/insurance_claims.csv>

# **7. Feedback and Response**

On Friday the 23rd of August we sent our Requirements/Scoping Document to the client (EY). following this we had a face to face meeting with the client on the 28th of August from 8am to 9am at the EY building to have our weekly stand-up and discuss the feedback for the document

| **Section** | **Feedback** | **Response/Action Points** |
| --- | --- | --- |
| **Data Understanding** | 1. Data Preparation steps were “a little vague” 2. Including a diagram of the layers might be handy to break up large amounts of text. 3. Needed to be specific about why the data is data “well-vetted” and why it has good data quality. | 1. Added more details into the data preparation steps and what it will entail, specifically data cleaning. Also noted that this process might change as the course of the project. 2. Added a detailed, colourful diagram of our data pipeline project idea along with steps which guide viewers on how to follow the pipeline. 3. Added details into how our data is well-vetted and what characteristics and attributes of our data make it high data quality. Including it has 39 columns and 1000 records. |
| **Data Preparation** | 1. Justify the use of a single dataset. 2. Mention that the dataset was chosen according to the specific requirements. 3. Explain the data synthesization techniques in detail for the future deliverables. 4. Specify the certain azure functions. 5. Explain the unnecessary steps avoided if separate pipelines are used. | 1. Explained why we are using a single dataset as it is more consistent. However, this is likely to change in the future. 2. Mentioned that the dataset was chosen following the requirements. 3. Will discuss more about the techniques in the later deliverables. 4. Will be discussed later as it is not decided yet. 5. Explained the unnecessary steps along with an example. |
| **Data Modelling** | 1. Emphasise that this section is not part of our project scope 2. Fix the formatting for the benefits and limitation so it is uniform across all techniques 3. Explain the reasoning behind the test data requirements | 1. Expanded why the modelling section is not part of our project scope and mentioned that the AI team is responsible for this so markers have a better understanding of our project. 2. Fixed any formatting issues so everything is uniform. 3. Expanded on our test dataset requirements to better improve our explanation on how we are validating whether a claim is fraudulent in your dataset? |
| **Evaluation** | No suggestions were received the mentors were satisfied with the contents of this section | Based on the feedback given above we expanded on some of the parts that seemed vague and fixed some formatting errors so the document was uniform. |
| **Deployment** | No suggestions were received the mentors were satisfied with the contents of this section | Based on the feedback given above we expanded on some of the parts that seemed vague and fixed some formatting errors so the document was uniform. |

# 

| **UPDATED TEAM MANUAL**  **Team 14**  Revolutionise Claims Management: Unleash the  Power of GenAI for Peak Efficiency in the Insurance  Sector |
| --- |

# **TABLE OF CONTENTS**

[**Revision History Table 8**](#_mk1gden1ras7)**9**

[**1. Team Organisation and Structure**](#_h51d8fyuz0vb) **90**

[**2. Team Values & ACS Code of Professional Ethics 9**](#_swzr988wnlyn)**1**

[2.1. ACS Code of Professional Conduct 9](#_elv22pnri0u7)1

[2.2. Honesty 9](#_rsft776bqpik)2

[2.3. Trustworthiness 9](#_yvngz9yrynap)2

[2.4. Respect 9](#_p97vmuryl7h0)2

[**3. Communication Plan and Meet Schedule 9**](#_9o4dopri62zt)**3**

[3.1 Communication 9](#_ujlctmz1edxj)3

[3.2 Meet Schedule 9](#_xp07llg59fq2)4

[**4. Conflict Resolution/Negotiation 9**](#_a73whcyuvf9u)**5**

[4.1. Conflict resolution 9](#_rgbgbdugtwqa)5

[4.2. Escalation process 9](#_y1z2y2erfgk5)5

[4.3. Preventative steps: 9](#_8iqsxfr6qphj)5

[4.4 Documentation of Resolutions and Agreed Changes[NEW]: 9](#_ujbn8winycgh)6

[**5. References 9**](#_82tcbiqw0wgb)**7**

# **Revision History Table**

| **Version** | **Last Update** | **Author(s)** | **Changes** |
| --- | --- | --- | --- |
| V2\_S0 | 08.08.24 | Noorullah Khan | * PM tools section removed from team manual |
| V2\_S1 | 27.08.24 | Quoc Hung (Alan) Nguyen | * Ninuri, Tashiya, Noorullah will transition to the role of Data Engineer.while remaining their previous roles * Alan will act as a Dev Ops Engineer while remaining Data Engineer |
| V2\_S2 | 22.08.24 | Ninuri Mahagoda | * Honesty: Opened communication with AI team * Trust: Completed tasks and engaged in continuous learning * Respect: Whole team meeting with Team 13 and push others to learn new concepts relevant to the current stage |
| V2\_S3 | 28.08.24 | Quoc Hung (Alan) Nguyen | * Updated communication through new combined group chat. * Updated communication through the PM channel being public for all to view. * Both teams private chats are now open for viewing to see discussions |
| V2\_S4 | 25.08.24 | Tashiya Vilathgamuwa | * Updated the approach to addressing conflict resolution to reflect the execution stage. * Updated preventative steps to reflect execution stage. |
| V3\_S4 | 28.09.24 | Tashiya Vilathgamuwa | * Added Documentation of resolutions and agreed changes to change management section |
| V4\_S4 | 16.10.24 | Tashiya Vilathgamuwa | * Added new point to section 4.4.1 |

# **1. Team Organisation and Structure**

| **Name** | **Role** |
| --- | --- |
| **Noorullah Khan** | Project Manager / Data Engineer |
| **Tashiya Vilathgamuwa** | Document Control Coordinator / Data Engineer |
| **Ninuri Mahagoda** | Document Control Coordinator / Data Engineer |
| **Aasnayem Gazzali Chowdhury** | Data Engineer |
| **Quoc Hung (Alan) Nguyen** | Devops Engineer / Data Engineer |



As Sprint 2 wraps up, our team has made some changes to the roles of each member moving forward. Despite these changes, we are maintaining a flat hierarchy due to only having 5 members. This structure will simplify our communication pattern and allow for easier management. The key difference this time is that roles have been simplified to align more closely with what each member will be doing. Notably, Tashiya and Ninuri will now be involved in the technical work. Since the project focuses on building a data pipeline, everyone will be working primarily as a Data Engineer, along with any additional responsibilities they may take on. Given the nature of the solution we’re developing, it makes sense for all team members to take on Data Engineering tasks.

**Noorullah Khan:**

1. Noorullah still holds the role of **Project Manager**. He will be responsible for assigning tasks and setting deadlines with other members.
2. Additionally, Noorullah will now become a **Data Engineer** who will be working collaboratively with other team members to build a data pipeline on Azure to build a dataset to train a generative AI model for the group 13 who also work with EY in this program. Noorullah, currently, is working on collecting a large enough raw dataset before we can work with it.

**Tashiya Vilathgamuwa and Ninuri Mahagoda:**

1. Business Analyst’s related work is mostly done, it will just come back every now and then. Therefore, both Tashiya and Ninuri have decided to transition into the technical part and become **Data Engineers**. As above, they will be working collaboratively with other team members to build a data pipeline on Azure to build a dataset. Currently, Tashiya is working on synthesising data to create an even larger dataset or fill missing data. Ninuri, on the other hand, is working on cleaning the data and working on the code convention for the team to follow.
2. They will still be working jointly as **Document Control Coordinators**. This role emphasises that they will be in charge of ensuring that our Deliverables are written in accordance with the definition given by the University. Also, they will be checking to ensure that we are not plagiarising or using AI-generated content.

**Aasnayem Gazzali Chowdhury:**

1. Aasnayem still holds the role of a **Data Engineer** which will be to build a data pipeline on Azure. Aasnayem, at the moment, is working closely with Noorullah to gather raw data sources and decide on methods to clean the data. More importantly, Aasnayem will be spending most of his time researching on data quality standards which would act as the goal for our dataset, so that we can come up with effective solutions to clean the data.

**Quoc Hung (Alan) Nguyen:**

1. Similar to everyone else, Quoc will be working as a **Data Engineer** also which would mean writing scripts to process the data gathered before.
2. Additionally, Quoc will act as a **DevOps Engineer** who will be responsible for managing the infrastructure of the solution on Azure as code and write scripts to automate the process of testing and continuously deploy code on Azure. Lastly, a DevOps Engineer will be responsible for managing the GitHub repository.

## 

# **2. Team Values & ACS Code of Professional Ethics**

## **2.1. ACS Code of Professional Conduct**

As we progress through our project with EY it is important to uphold the ACS Code of Professional Ethics to guide the behaviour and decision-making showcasing our professional conduct to the client. In this updated team manual we will discuss the core values of honesty, trustworthiness, respect for others and respect for the profession in correlation to our current ICT environment (ACS, 2023). This highlights our commitment to professionalism as we engage and support our team values and ethics throughout the course of this project.

## **2.2. Honesty**

Honesty remains one of our key values in this project as we continue to interact within our team and with the client and facilitate open and professional communication (ACS, 2023). As our project advances we have become more entangled with Team 13 as we have started to collaborate more with each when working with the client. To facilitate open and truthful communication with both teams we have opened up our private team channels on the discord so both teams are able to see what is discussed by the other team and what they are currently working on or discussing. Additionally a combined discord chat was created so that both teams can talk to each other on problems and discuss tasks without disrupting the individual team chats. Furthermore, both teams have combined our Gannt charts on Jira so that all team members know the progress of the tasks and any roadblocks we may face as well as an overview of the project status as a whole.

## **2.3. Trustworthiness**

Maintaining trust between the team members and with the client is vital for the success of the project (ACS, 2023). As we have undertaken tasks and roles in the project in line with our communicated abilities we have strived to maintain the trust of our members as we complete tasks to the best of our abilities and knowledge. We have further developed our capabilities to include GitHub, python and data synthesisation techniques as we engage in continuous development. As we have reached the building stage of our project we have cleaned our dataset to remove personal information from the dataset such as Names, Addresses and Phone numbers ensuring the data is anonymised.

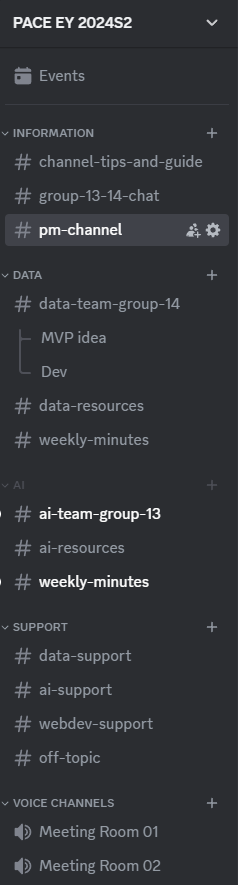
## **2.4. Respect**

Having respect for individuals around us and the profession we do is a fundamental part of collaborative success (ACS, 2023). We continue to have weekly meetings on the project to discuss our ideas and findings to the group and hear the various perspectives and opinions each member has on various topics. We have also started to have whole team meetings with Team 13 on the project as a whole to encourage communication between peers and for more ideas and perspective to be brought on ideas and issues. We have supported our members' advance in “ICT knowledge and competence” as we push our members to learn about data synthesisation techniques, and Terraform (ACS, 2023). Ultimately, our project is based on the notion of “advancing ICT capabilities and systems” for EY as we incorporate Gen AI to improve an existing system for the client (ACS, 2023).

# **3. Communication Plan and Meet Schedule**

## **3.1 Communication**

Our Discord server seems to have reached a stable stage and this structure will likely be kept moving forward. It has been set up as follows:



There are 5 categories including:

1. **INFORMATION:** channels inside this category is mainly used to make announcements and relay information from weekly meeting with sponsor to both teams
   1. channel-tips-and-guide: as the name suggests
   2. group-13-14-chat: this channel will be used for cross-talk between the teams.
   3. pm-channel: this channel is public but reserved for the two Project Managers to communicate with each other.
2. **DATA:** channels inside this category will be data related
   1. data-team-group-14: is now a public chat channel that can be viewed by anyone in this server. However, only the mentors and group 14 can send messages in this channel. We are assigned to work on the Data team. This chat will be our main mode of communication. The reason for the change is to improve the transparency which will help the other team understand the work we are doing better.
      1. MVP idea: is an example of a thread, which is a side discussion we want to have with a few selected members. The usage of a thread will help not clutter the main chat channel
   2. data-resources: is a public chat channel where anyone can post any data related resources, including data sources and blogs about cleaning data.
   3. weekly-minutes: is a public chat channel where we will post meeting minutes for any meeting we might have.
3. **AI:** channels inside this category will be AI related.
   1. data-team-group-14: is now a public chat channel that can be viewed by anyone in this server. However, only the mentors and group 13 can send messages in this channel. This chat will be their main mode of communication. The reason for the change is to improve the transparency which will help our team understand the work they are doing better.
   2. data-resources: is a public chat channel where anyone can post any AI related resources, including where to find open-sources models and how to fine-tune a model.
   3. weekly-minutes: is a public chat channel where group 13 will post meeting minutes for any meeting they might have.
4. **SUPPORT:** channels inside this category will mainly be used to ask mentors for support in specific problems
   1. data-support: is a public chat channel to post data related questions
   2. ai-support: is a public chat channel to post AI related questions
   3. webdev-support: is a public chat channel to post web development related questions
   4. off-topic: is a public chat channel to post any miscellaneous questions
5. **VOICE CHANNELS:** channels inside this category will mainly be used as virtual online meeting rooms where we can organise our weekly meeting.

## **3.2 Meet Schedule**

* Face-to-face meeting with sponsor: 8am to 9am weekly on Tuesday at EY office 200 George street, Sydney 2000
* Face-to-face team meeting & work together: 9:30 to 1pm weekly on Tuesday after the meeting with the sponsor.
* End of week online meeting: 11am to 12:30am weekly on Saturday using Discord

Compared to Deliverable 1, there has been no change to our meeting schedule as it has proven to work very well for our team. The only note is that for the post-EY meeting that we have, we will dedicate roughly 30 minutes to discuss some ideas with group 13. After that, the two teams will split and our team will continue our internal meeting and work together.

In summary, every week, we will have a meeting with our sponsor at their office to showcase what we have done throughout the week and decide on the direction for the next step. This meeting will be our formal way of demonstrating our work to the sponsor besides our informal chat on Discord with them. Following this meeting will be a 90-minute meeting for our team. The purpose of this meeting is to build team rapport and also discuss our plans and work for the week. Lastly, at the end of every week, we will have an online meeting on Saturday to check on the work we have done as well as planning for our next meeting with the sponsor. The purpose of these meetings can be overlapping depending on the tasks that we have at hand. However, we aim to meet with each other 3 times a week to keep everyone accountable and build trust in each other.

# **4. Conflict Resolution/Negotiation**

## **4.1. Conflict resolution**

**Approach to addressing conflict**

In the execution stage, conflicts can occur due to increased pressure, overlapping responsibilities, or differences in work styles. Our approach is to address these conflicts promptly and constructively to avoid any disruption to project progress.

**Conflict resolution procedure**

1. Recognise the conflict among team members
2. Organise a meeting involving the team members
3. Give each member an opportunity to share their viewpoint
4. Pinpoint the cause of the disagreement
5. Brainstorm potential solutions collectively as a team
6. Reach an agreement on a resolution
7. Put the solution into action and do a follow up

**Guidelines for Constructive Conflict Resolution**

* Direct attention to the problem, not personal characteristics
* Use "I" statements to communicate emotions and concerns
* Engage in active listening
* Strive to grasp all viewpoints before suggesting solutions
* Be willing to negotiate
* Maintain professionalism and respect consistently

## **4.2. Escalation process**

If conflicts cannot be resolved among team members:

1. Approach the team leader or project manager to address the matter.
2. If a resolution is not reached, escalate the concern to MQ’s leaders.

#### 

## **4.3. Preventative steps:**

To minimise conflicts within our team:

* Encourage regular team-building activities
* Provide opportunities for open feedback
* Address potential issues early before they escalate
* Regular check ins to identify potential conflicts early on
* Clearly defined processes to ensure all team members understand the workflow to avoid overlap and confusion
* Real time feedback: Encourage team members to provide feedback on issues as they arise rather than waiting for formal meetings.
* Stress management: Note to be mindful of the pressure during execution and team members can help each other through offering support to manage stress.

By applying these updated strategies to overcome any conflicts in the execution stage the team can effectively manage conflict, ensuring that the project stays on track and that the working environment remains positive and productive.

## **4.4 Documentation of Resolutions and Agreed Changes[NEW]:**

In order to maintain transparency and foster team unity in the run, it's crucial to record any solutions reached when dealing with conflicts and agreed upon adjustments between parties involved. This approach can anticipate disagreements in the future, align everyone's understanding, and establish clear directions for making decisions later on.

4.4.1 **Recent Resolutions and Agreed-Upon Changes**

1. **Resolution: Timeline Misalignment for Deliverables**
   1. **Conflict:** A miscommunication occurred regarding the deadlines for submitting the deliverable draft to clients. Some team members thought they had more time than others, leading to delays.
   2. **Resolution:** The team agreed to implement a new, centralised timeline in Jira, ensuring that all deliverables and deadlines are visible to every member in real-time.
   3. **Agreed Change**: A calendar reminder will be sent to all team members at the start of each week, detailing upcoming tasks and deadlines to avoid further misalignments.
2. **Resolution: Communication Delays During Remote Collaboration (Working around everyone's busy schedule)**
   1. **Conflict:** We were finding it difficult to find times for meetings to practise MVP or team discussions due to everyone's different schedules.
   2. **Resolution**: The team members each provided a schedule of their working hours, so others know when they will be non responsive. We also have allocated two specific times each week for a compulsory meeting where all team members have to attend which will be Saturday 11am and Tuesday 9am, where all team members will be free for the continuation of the project. This allowed us to know we could have our discussions, practises, and collaboration during this time.
   3. **Agreed Change:** Change to have set times and days for meetings where everyone is available. Also Team members will now set their availability status on Discord, ensuring everyone is aware of their availability for urgent matters. The weekly meeting also is used to identify and address any potential communication bottlenecks.
3. **Resolution**: Differing Opinions on Project Direction
   1. **Conflict:**During a casual team meeting, two team members shared differing views on the project's direction. One preferred a more innovative approach, while the other advocated for sticking closely to the original plan. Instead of feeling tense, the team saw this as a healthy discussion.
   2. **Resolution:** To delve deeper into both viewpoints​ the group opted to plan a brainstorm meetup​. One by one​ each party shared their thoughts. The team had a lively exchange on the advantages of each angle​. In the end​ they managed to merge both perspectives​ forming a path, for the project that left everyone satisfied​.
   3. **Agreed Change:**In the meetings ahead of us all together decided to set aside a specific time, for exchanging various viewpoints and coming up with new ideas collaboratively and positively to encourage everyones input.

# **5. References**

ACS. (2023). *ACS Code of Professional Ethics*. ACS; Australian Computer Society. https://www.acs.org.au/content/dam/acs/CodeOfProfessionalEthics\_Mar\_2023.pdf

# 

| **PROTOTYPE/MVP DOCUMENT**  **Team 14**  Revolutionise Claims Management: Unleash the  Power of GenAI for Peak Efficiency in the Insurance  Sector |
| --- |

# **TABLE OF CONTENTS**

[**Revision History Table**](#_nryj9qsl7r2i)**………………………………………………………………………………….100**

[**1. Introduction**](#_76bi7k3n5vqc) **101**

[**2. Prototype Overview**](#_45rhujerjs7i) **101**

[2.1 Prototype concept](#_qamvaq39k1oc) 101

[2.2 Key features](#_sqkae9mq3wcf) 101

[**3. Current project Codebase & Dataset**](#_nzxf0shsuxrt) **102**

[3.1 Codebase](#_8qp0bxd3vqwy) 102

[3.1.1 Assets folder 10](#_u0f8fqsif4x1)3

[3.1.2 Source code folder 10](#_5jb9qtqsdqr2)3

[3.1.3 Documentation 10](#_224sfwwhphqd)3

[3.2 Dataset 10](#_e840eoganv66)3

[**4. Sponsor meeting, feedback and response to feedback 10**](#_fkqxnyajnkkl)**4**

[4.1. Prototype Feedback Adjustment 10](#_3x41c17c9lao)4

# 

# **Revision History Table**

| **Version** | **Last Update** | **Author(s)** | **Changes** |
| --- | --- | --- | --- |
| Introduction  V2\_S1 | 16/10/24 | Quoc | * Updated Introduction to ensure markers understand we are the “Data Team” not the “AI Team” (Group 13) and that we are working collaboratively with them. |
| Prototype overview  V1\_S2 | 25/09/24 | Quoc | * No changes made |
| Current project codebase & Dataset  V1\_S3 | 25/09/24 | Quoc | * No changes made |
| Sponsor meeting, feedback and response to feedback  V2\_S4 | 16/10/24 | Quoc | * Updated the introduction of this section * Updated 4.1 “Prototype Feedback Adjustment” section to include the development of the user interface. |

# **1. Introduction**

The MVP document offers a detailed overview of the essential components of our data processing pipeline, covering everything from infrastructure setup to the generation of the final dataset. It provides insights into our approach to data management and processing, aimed at delivering a high-quality dataset. The document outlines our infrastructure design, the codebase, and the data pipeline. It also presents the resulting prototype dataset, consisting of 19 features that have been thoroughly cleaned, synthesised, and enriched. Please note that we are the data team and we are not responsible for delivering the generative model (logic). Rather, we are more focused on building a quality dataset for the AI team (group 13) to fine tune their model.

# **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:** A version of the enriched dataset containing 19 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.

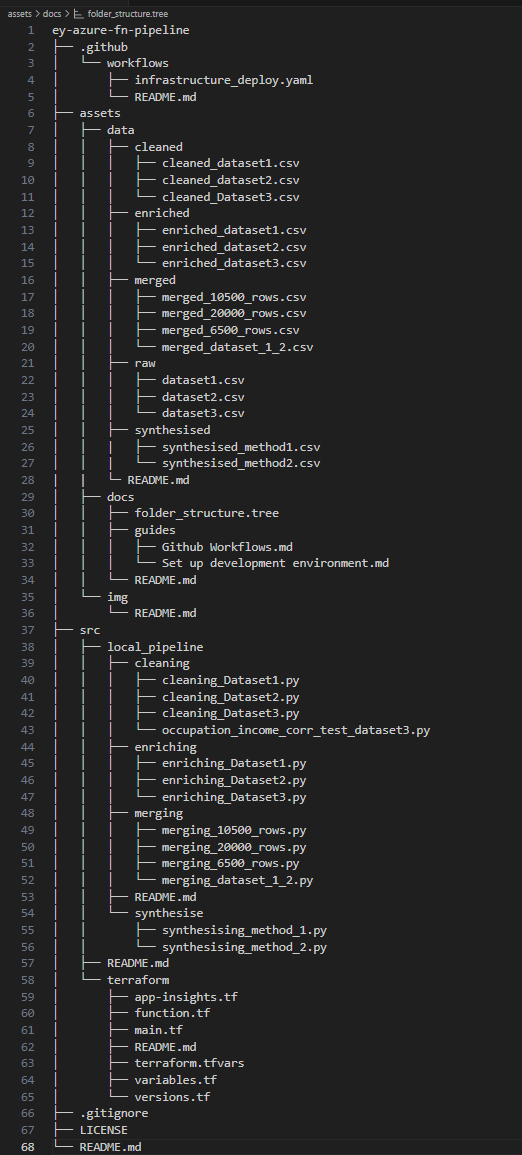
# 

# **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.



*Figure 1: Current Codebase*

### **3.1.1 Assets folder**

The 'assets' folder stores data at various stages of the transformation pipeline. Within the 'data' subfolder, the data is further partitioned to reflect its raw, enriched, synthesised, and merged states. Additionally, documentation related to the project will be stored in this folder. At the time the screenshot was taken, the metadata for the dataset in the gold layer had not yet been completed. This document will be continuously updated to reflect the latest metadata for datasets shared with the AI team. The 'img' folder contains images that will be used in README.md files and other project documentation.

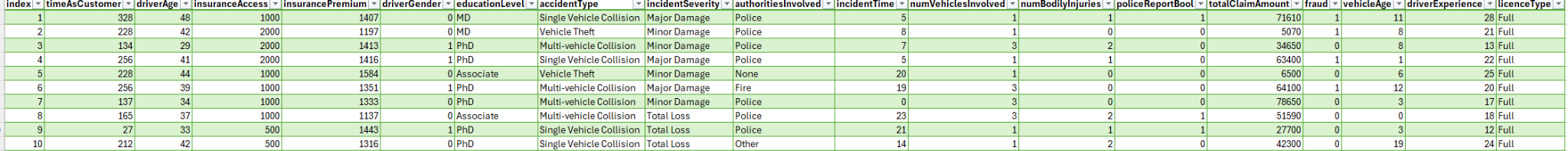
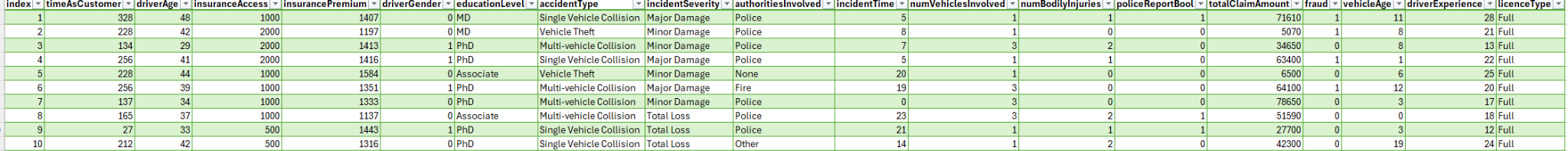
### **3.1.2 Source code folder**

The most crucial component of this prototype is the contents of the 'src' folder, which houses both the source code for setting up the infrastructure and the code that forms the transformation pipeline. As shown in the screenshot above, the folder is organised into two sections: 'terraform,' which is responsible for setting up the cloud infrastructure, and 'local\_pipeline,' which is used to build the transformation pipeline. The pipeline consists of four stages: cleaning, enriching, synthesising, and finally merging the data.

### **3.1.3 Documentation**

Lastly, documentation is not limited to the 'assets' folder but is distributed throughout the codebase within the README.md files. This approach allows for more effective management of the content in each README.md file, ensuring the information remains concise and does not overwhelm the reader. This method also ensures that relevant details are accessible in context, making it easier for team members to reference specific information without needing to navigate through separate documents.

## **3.2 Dataset**

*Figure 2: Final Dataset*

The dataset, which consists of 20,876 rows of data in CSV format, will be elaborated on in more detail later in the document.

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

We had a prototype presentation on Tuesday, 24th September 2024, at EY’s office. After the presentation, one of the mentors questioned our choice of tools, specifically suggesting the use of another library called 'AutoML,' which is essentially a library designed to automate the process of building machine learning models. He mentioned that with this library, the tasks we are handling in the transformation pipeline could be completed in just five lines of code and asked whether we had considered this option. In the next section, **Prototype Feedback Adjustment**, we will explore how we plan to evaluate and potentially incorporate the AutoML library into our workflow to optimise the transformation process.

The mentor also requested the development of a streamlined web application designed for claims managers to interact directly with unprocessed insurance claims. This application would feature AI integration, specifically incorporating a large language model (LLM) to enhance decision-making. The LLM would provide contextual insights on each claim and offer predictions regarding the likelihood of fraud. These predictions would be based on historical claims data, utilising natural language processing capabilities to deliver comprehensive feedback.

Furthermore, the mentor highlighted the need for an interactive feature within the application. This feature would enable claims managers to engage directly with the large language model, allowing them to ask questions related to specific claims or general data queries. The LLM would then provide detailed explanations, supporting the claims manager with deeper insights into each case, thus facilitating more informed decisions during the claims review process.

## **4.1. Prototype Feedback Adjustment**

During the presentation, we outlined two critical objectives to focus on for the next prototype delivery: deploying the data pipeline to Azure and further refining the transformation pipeline. The mentor's feedback has been integrated into the second objective, where we plan to assign 1-2 team members to research the potential of using ‘AutoML’ and explore its application in our pipeline. Our goal is to recreate the pipeline—if not with the exact same methods, at least following the same steps, such as cleaning and synthesising the data. Following this, a team review will be conducted using predefined standards to assess which approach yields the better dataset. Additionally, we will incorporate feedback from the AI team to further determine the most effective method.

The web application will primarily be developed by the data team (which is us, Group 14), while Group 13 (the AI team) will focus on tuning the large language model. We are currently in the process of designing a wireframe for the application to present to the clients for feedback, which will guide the development of the full-scale web application. In the interim, a basic web application has been created with limited functionality to aid in the development and visualisation of the wireframe.

Regarding the additional feature requested by the client/mentor, we are collaborating closely with the AI team to assess its feasibility and determine whether it can be implemented effectively. Should the AI team confirm their ability to realise this functionality, we will integrate it into our wireframe before moving forward with the final delivery of the web application.

# 

| **ANALYSIS, DESIGN + TESTING DOCUMENTATION**  **Team 14**  Revolutionise Claims Management: Unleash the  Power of GenAI for Peak Efficiency in the Insurance  Sector |
| --- |

# **TABLE OF CONTENTS**

**Revision History Table………………………………………...………………………………108**

[**1. Feature engineering: 10**](#_lzfgmtf9iad0)**9**

[1.1 Trends and ranges 10](#_bf9lmhi2r7so)9

[1.2 Data characteristics and pipeline 10](#_3pxj91upzvfc)9

[1.3 Feature definition 1](#_eo4n0cp4jp4n)12

1[.4. Semantic Features[NEW](#_d7xjefub6fp5)].............................................................................................................113

[**2. Solution Architecture 1**](#_ryj7yw870g23)**13**

[2.1 Current architecture 1](#_gnnnqfbl6ne2)13

[2.2 Cloud architecture (not yet deployed) 1](#_k0kki89gevit)15

[**3. Algorithms / models methods 11**](#_l2mjyumkiy8h)**6**

[3.1. KNN Model 11](#_h43q0yu2c5dr)7

[3.1.1. Configuration Settings 11](#_e1rhfjojw0ub)8

[3.1.2. Reasons for Model selection 11](#_egrx6mswua1e)8

[3.2. Random Forest Models 11](#_qvxe909pruj9)8

[3.2.1. Configuration Settings 11](#_m2rb7z33gedi)9

[3.2.2. Reasons for Model selection 1](#_f2liiara9pq)20

[3.3. Why were these models chosen over other approaches? 1](#_lmxr7f5dtxej)20

[3.4. Advanced Techniques [NEW](#_pk7g43ix3skr)]…………………………………………………………………..121

[**4. Detailed Data descriptions: 1**](#_gv4z3eqx89tg)**21**

[4.1 Original Raw Datasets 1](#_vw5e3ri0812r)21

[4.2. Columns in Raw Datasets 1](#_619wrttzrtw8)22

[4.3. Cleaning & Enriching 12](#_dsq15dkeog8x)7

[4.4. Generating Data: 1](#_2jh8fpqb5hmu)31

[4.4.1. Method 1: 1](#_u3h70ysuq300)31

[4.4.2. Method 2: 1](#_m5s659ehbp43)32

[4.4.3. Merging Datasets: 1](#_75744aujprf7)33

**TEST SPECIFICATION**

[**1. Model Evaluation 1**](#_7vyh5dxtpku9)**37**

[1.1. Model Evaluation and Comparison 1](#_3z1wmobrz3b0)37

[1.1.1. Correlation Matrix and ANOVA Analysis: 1](#_obe1pjidf3wk)37

[1.1.2. Validation of Synthesised Data: 13](#_3guxe941bnqt)8

[1.1.3. Imputation Techniques: 13](#_3pao0cpto3a6)8

[1.2. How we compared and evaluated Models and decided what to choose 13](#_5gw1fjam8azd)9

[1.2.1. R² (Coefficient of Determination) 13](#_loviq9tc2461)9

[1.2.2. Accuracy 1](#_rnxiahg0558x)40

[1.2.3. Mean Squared Error (MSE) and Mean Absolute Error (MAE) 1](#_wu6vztkmaiur)40

[1.3. Overall 1](#_449qy26g8h8x)41

[**2. Performance evaluation results: 1**](#_c8ue64c9px0s)**41**

**3**[**. References 1**](#_gyoqyjck1ap0)**43**

# **Revision History Table**

| **Version** | **Last Update** | **Author(s)** | **Changes** |
| --- | --- | --- | --- |
| V2\_S1  Feature Engineering | 11/10/24 | Adam | * Added a section for Semantic Features |
| V2\_S2  Solution Architecture | 15/10/24 | Quoc | * In section 2.2 explained why the cloud architecture is still not deployed yet |
| V2\_S3  Algorithms and Models methods | 16/10/24 | Ninuri | * Created a new section 3.4 Advanced Techniques to discuss about AutoML |
| V2\_S4  Detailed Data Descriptions | 31/10/24 | Noorullah | * No changes made |
| V2\_S5  Model Evaluation | 31/10/24 | Tashiya | * No changes made |
| V2\_S6  Performance Evaluation Results | 14/10/24 | Adam | * Added more information on how the results from these tests will benefit the project and be more outcome focused. |

# **1. Feature engineering:**

## **1.1 Trends and ranges**

Identifying the trends and ranges is crucial for feature engineering as the patterns will help to better understand the dataset by identifying relationships between variables, detecting outliers, and evaluating the data quality. Correlations between variables can be checked to identify potential relationships. Through this approach, our raw data can be converted to a more meaningful format which will be well structured and easier to interpret by the model, therefore leading to a better performance of the model. In our case, when detecting fraudulent claims, it will be key 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.

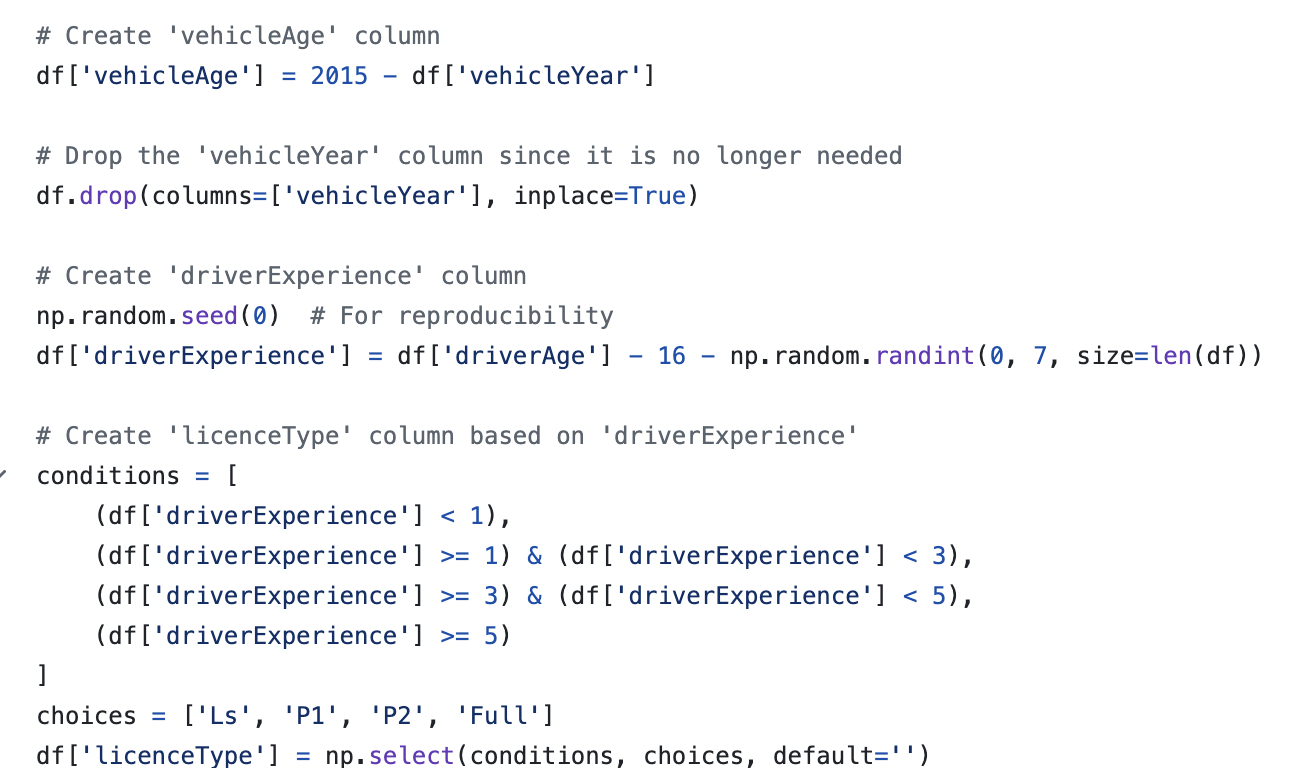
Columns such as ‘driverAge’, ‘driverExperience’, and ‘licenseType’ can give us information about the driver. These columns can also provide insights on the driver behaviour. For example, younger drivers, who are less experienced, are more likely to be driving rough and engage in speeding. Whereas, middle-aged drivers are more experienced and are more likely to drive safely (Keall & Frith). 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**

Fraud indicators must be considered because the dataset is designed to be used to train an AI model that would identify fraudulent claims. 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, synthesising and merging to get the combined dataset. We are using our dataset 1 as our foundational dataset. In data cleaning, the columns were renamed to match camel casing. All three datasets had the same column names and followed camel casing. For dataset 1, the missing values were handled. For dataset 2, ‘EducationLevel’ was filled in assuming the lowest degree which is ‘High School’. For dataset 3, the missing columns were filled in using various imputation methods such as KNN, Random Forest, GridSearchCV, etc.

In enrichment, ‘True’ ‘False’, ‘YES’ ‘NO’, ‘MALE’ ‘FEMALE’ were all converted to numeric 0’s and 1’s. Here, the categorical variables that could have been converted have been converted to numerical 0’s and 1’s to make it more understandable to the model. Here, the unwanted columns were also dropped in all three datasets. Some columns were generated such as the ‘vehicleAge’ column. As our dataset was based in 2015, we subtracted 2015 from the ‘vehicleYear’ column. The ‘driverExperience’ column was also generated using the ‘driverAge’ column. The ‘driverAge’ is subtracted by 16 as it is the minimum driving age and is further subtracted by a random number from 0 to 7 as we want some variability. The ‘licenseType’ column is also generated based on the ‘driverExperience’ column. An "L" licence was issued if the driver's experience was less than a year; a "P1" licence was issued if the driver's experience was more than a year but less than three years; a "P2" licence was issued if the driver's experience was more than three years but less than five years; and a "Full" licence was issued if the driver's experience was more than five years. The code to generate these columns is provided below:

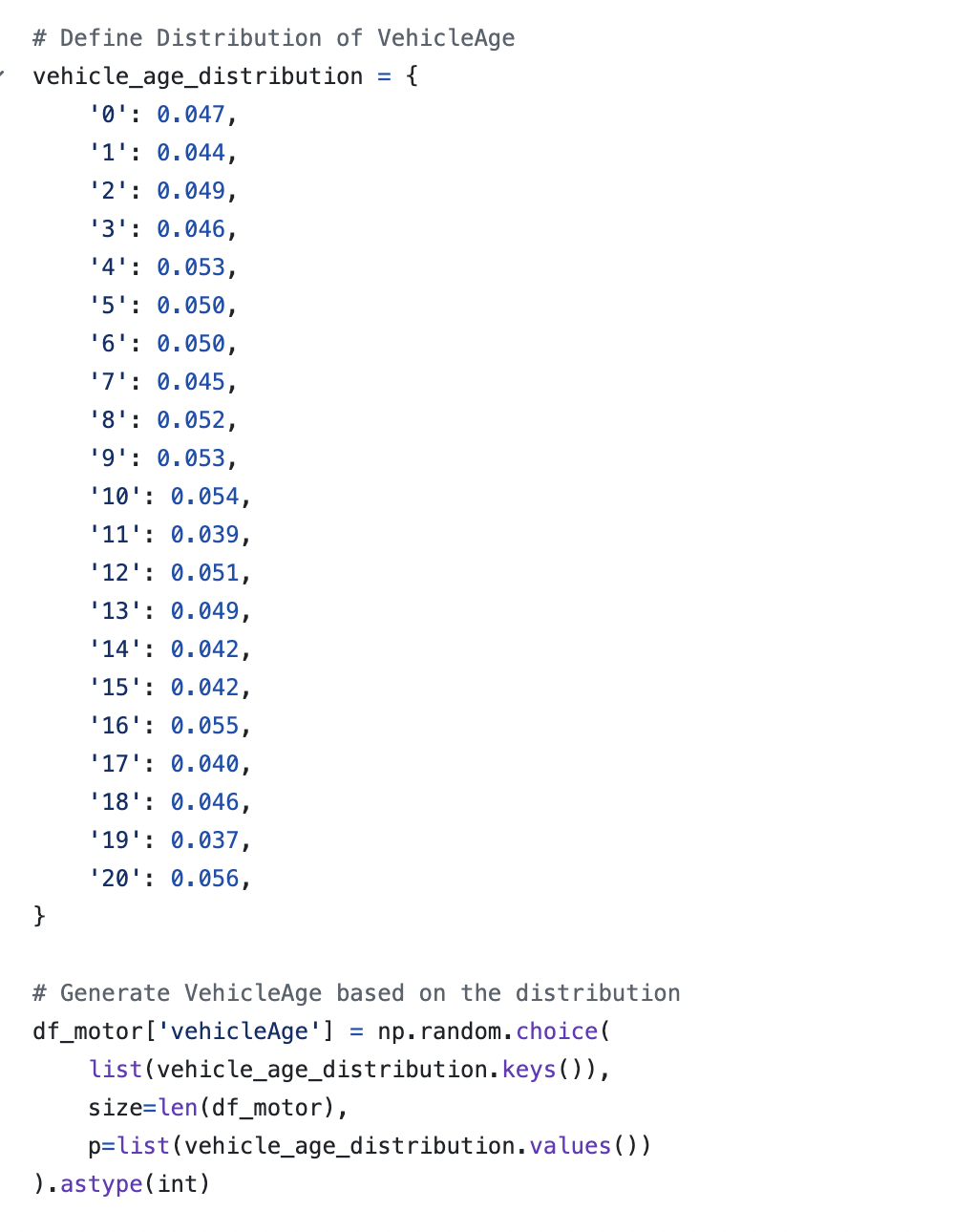


*Figure 1: Code to generate licence type*

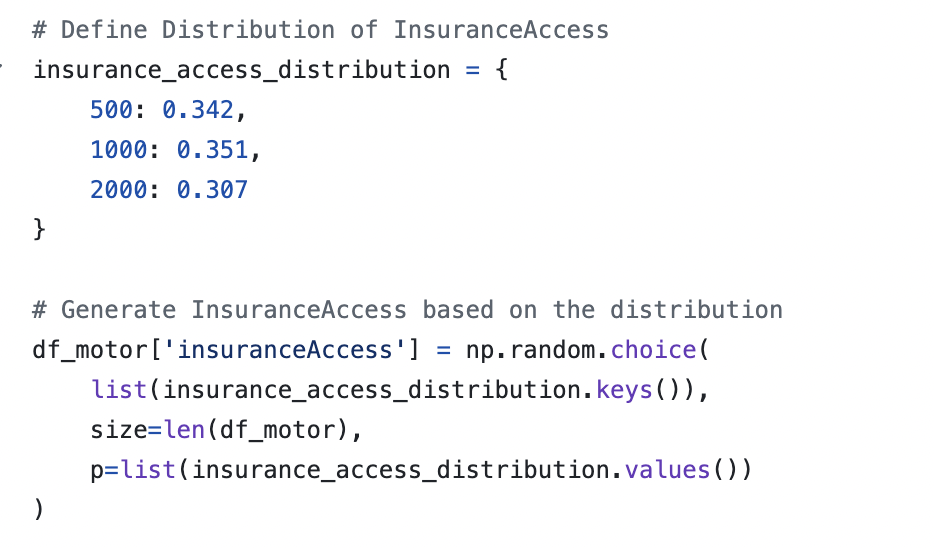
For dataset 2, a few columns had to be generated based on dataset 1 as dataset 2 did not have those columns. The ‘AccidentType’ column had to be enriched based on the distribution of that column in dataset 1. Using that distribution, ‘AccidentType’ was then randomised and generated. For ‘NumVehiclesInvolved’, the same method was used to generate it. If the ‘AccidentType’ was ‘Multi-Vehicle Collision’, the column was randomly generated based on the normal distribution of ‘Multi-Vehicle Collision’ in dataset 1. Otherwise, the column returned 1 as the rest of the accident types were involving a single vehicle. The ‘VehicleAge’ column was also generated based on the distribution of it in dataset 1. The weights were assigned to the different ages and were randomly generated once again. A similar process was also used to generate the ‘InsuranceAccess’ column. The code for the columns generated is provided below:



*Figure 2: Code to generate the Accident Type and Number of vehicles involved column*



*Figure 3: Code to generate the Vehicle Age column*



*Figure 4: Code to generate the Insurance Access column*

In all of the datasets, Fraud was converted to 1 and Non-Fraud was converted to 0. The same processes are used to generate the features in dataset 3. The 2 datasets were then merged and based on the merged dataset, 2 synthesization methods were used to generate 8000 rows of data.

## **1.3 Feature definition**

A lot of features can be created following the process we will discuss later on in this section. However, these are some of the important ones. For ‘driverExperience’, we can name a feature called ‘experienceCategory’ which will classify the driver experience into different categories. This will help to determine the information about the experience of the driver. Ranges for this feature can be: <2 years = ‘Low Experience’, 2-4 years = ‘Moderate Experience’, >4 years = ‘High Experience’.

The same is partially done for ‘licenseType’. A feature called ‘licenseCategory’ can be named which will classify the licence type of the driver. In our case, if the ‘driverExperience’ was less than 1 year, then the licence was a Learners or ‘L’. The licence was classified as a Provisional P1, or "P1," if the "driverExperience" was greater than one year but less than three. The licence was referred to as a Provisional P2 or "P2" if the "driverExperience" was greater than three years but less than five. The licence was a "Full" licence if the "driverExperience" was greater than five years. The ages for the driver can also be categorised into a feature called ‘driverAgeCategory’. The ranges for that category can be: <25 years = ‘Young’, 25-45 years = ‘Middle-Aged’, >45 years = ‘Old’.

The same can be done for ‘vehicleAge’. A new feature called ‘vehicleAgeCategory’ can be created. The ranges for that category can be: <2 years = ‘New’, 2-10 years = ‘Moderate’, >10 years = ‘Old’.

A feature called ‘timeCategory’ can be created to describe the time the incident took place. The range for that category can be, 0-6 = ‘Early Morning’, >6-12 = ‘Morning’, >12-18 = ‘Afternoon’, >18-24 = ‘Night’. In this case, 0 is 12AM and 23 is 11PM.

## **1.4 Semantic Features [NEW]**

Semantic features will allow us to convert the raw data to a more meaningful representation of our data. This will allow us to detect patterns and relationships which will be a positive while the AI model is being trained. The ‘licenseType’ column can be classified as a semantic feature as it uses the ‘driverExperience’ columns to formulate. We can create some semantic features to help represent the dataset better. For example, a semantic feature called ‘driverRiskProfile’ can be created which can determine the risk level of the driver. We can use ‘driverAge’, ‘driverExperience’, and ‘licenseType’ to assess the risk of the driver. Another semantic feature called ‘policyProfile’ can be created to identify the likelihood of a fraudulent claum. It can use ‘timeAsCustomer’, ‘insuranceAccess’, ‘insurancePremium’ to give an idea about the insurance policy of the driver. Another feature can be created called ‘fraudLikelihood’ which will use ‘totalClaimAmount’, ‘incidentSeverity’, ‘accidentType’, ‘numVehiclesInvolved’ to determine the likelihood of a fraudulent claim. In conclusion, these semantic features will help to analyse the risk of fraud and hence, will help to identify fraudulent claims.

# **2. Solution Architecture**

## **2.1 Current architecture**

### 

*Figure 5: Current Architecture Diagram*

**Cleaning Stage**

We have three raw data sources in CSV format from Kaggle, each provided by different organisations. As a result, these sources do not share the same columns and exhibit varying levels of quality. The initial task in the cleaning stage is to prepare these datasets by renaming similar columns to a predefined standard and addressing any missing values. We have developed several imputation methods for this purpose, ranging from simple techniques—such as using the median and mean for numerical data and the mode for categorical data—to more advanced algorithms like Support Vector Regression and Random Forest Classifier for imputing other columns. Given that the combined raw data consists of fewer than 15,000 rows and amounts to approximately 3MB, the computational resources required for this stage are minimal, allowing for efficient processing without the need for extensive infrastructure.

**Enriching Stage**

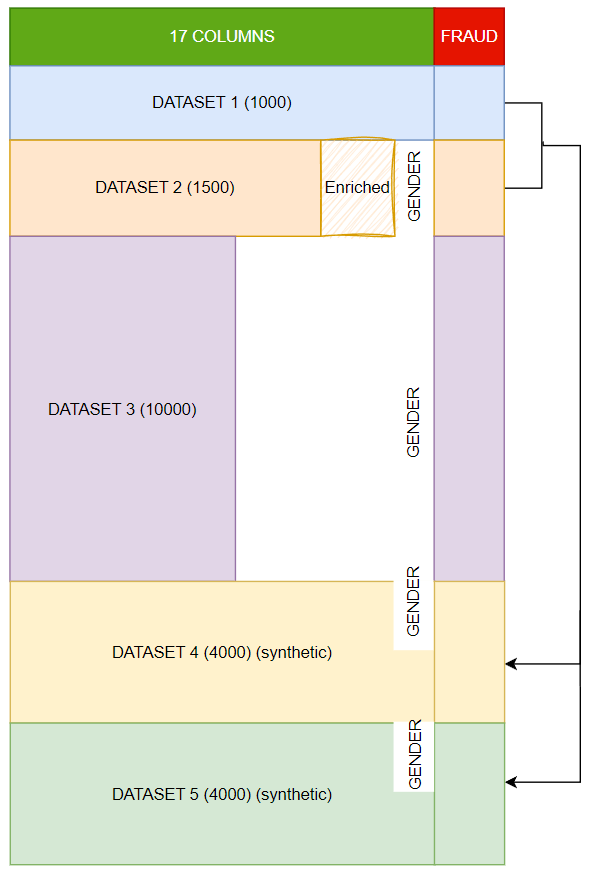
The enriching stage focuses on creating more meaningful columns within the datasets. One significant feature to be added is **Driver Experience**, which will be inferred from the distribution of the Australian legal driving age and the customer’s age. However, we face limitations in the amount of data available, which is insufficient for the AI team to effectively train their model. As a result, we have prioritised data enrichment over synthesis to maintain the reliability of our final dataset. This approach enhances the quality of the data, allowing our synthesis methods to learn from the enriched data's features and patterns. Notably, Dataset 3 has numerous missing columns necessary for enrichment. Therefore, we will focus on enriching only Datasets 1 and 2, ensuring that the enrichment process does not exceed one minute in running time due to the small size of the dataset. It is crucial that the enriched data remains realistic and not randomly fabricated, maintaining the integrity and reliability of the final dataset.

**Synthesising Stage**

The synthesising stage involves generating new data to address the limitations of our existing datasets. We will employ mathematical remapping techniques combined with sampling and Conditional Generative Adversarial Networks (CTGAN) to synthesise additional data. To ensure the reliability of the synthesised data, we will validate the output against known distributions and patterns from the enriched datasets. However, we recognize that CTGAN can be time-consuming, and Azure Functions have a time limit of approximately 10 to 15 minutes per execution. If the process exceeds this time frame, we will split the code into multiple runs to accommodate the time constraints.

**Merging Stage**

In the merging stage, we will integrate the synthesised datasets (1 and 2) with the enriched dataset (3). The merging will be conducted based on common keys or criteria defined during the cleaning and enrichment stages. Our expected output is a cohesive dataset with a reasonable number of null values, acknowledging that some null values are acceptable given the inherent imperfections of real-world data. However, we will prioritise creating a realistic dataset, as established in the enriching stage, ensuring it serves the intended purpose effectively.



*Figure 6: Final Dataset Diagram*

The final dataset is a combination of enriched dataset 1 and 2, synthesised dataset 4 and 5 from 2 methods and dataset 3.

### 

## **2.2 Cloud architecture (not yet deployed)**

### 



*Figure 7: Cloud Architecture Diagram*

**[NEW]** Due to concerns regarding costs, we have temporarily halted the deployment of the full pipeline on Azure. Although development and testing have been conducted extensively to ensure that both our code and configurations are fully compatible with Azure, we have decided to keep the codebase offline until the scheduled date of the client presentation. At that time, we will proceed with a live demonstration of the pipeline to showcase its functionality.

The pipeline's order closely resembles our current architecture. While we have successfully implemented the pipeline, it is currently not automated and operates solely on our local machines. The key distinction in this iteration is our use of Terraform for Infrastructure as Code. All data will be stored in Azure Data Lake Gen2, which will be organised into three distinct layers. The Bronze layer will exclusively contain our raw data, while the Silver layer will store data at various stages of transformation, including after cleaning, merging, synthesising, and further processing. Once the dataset is finalised, it will be moved to the Gold layer. These layers serve as an abstraction, facilitating efficient management of our data processes.

Additionally, we will utilise Azure Functions to execute Python scripts, automating processes whenever modifications are made to our scripts. Azure Functions is a cost-effective solution for small to medium-sized workloads, offering the first 400,000 GB/s of execution and up to 1,000,000 executions free of charge. Given our initial dataset, which comprises approximately 10,000 rows and around 5MB of data for the Bronze layer, costs will only be incurred if our data volume or execution frequency increases. Even in such cases, the projected expenses will remain minimal, considering the scale of this project.

Setting up and deploying Azure Functions is straightforward, making it a suitable choice for small-scale projects like ours. The pay-as-you-go model further enhances cost efficiency, as we only pay for the resources utilised. For our storage solution, we will employ Azure Data Lake Storage Gen2 (ADLS2) in the first MVP, leveraging its foundation on Azure Blob Storage. This option is not only economical but also capable of storing virtually limitless amounts of data, ensuring that both cost and scalability remain manageable.

# **3. Algorithms / models methods**

Cleaning the dataset is an important part of the project as the quality if the dataset directly impacts the accuracy and reliability of the outputs from the AI model. This project with NRMA involves collaborating with Team 13 to create an AI solution to detect fraudulent claims. Our team is responsible for delivering a dataset to Team 13 that is clean, complete and comprehensive so it can be used to train the AI model to make accurate predictions and generalise the data properly generating reliable predictions. To do this the raw dataset contains missing and inconsistent values using machine learning models to impute and standardise the values.

The main method of addressing the missing values was predictive supervised models due to their ability to make accurate predictions in relation to patterns in the data. Compared to other imputation methods like mean or mode, predictive supervised models are able to use the underlying relationships between features resulting in the imputed values being more context-aware. As the dataset contain both numerical and categorical data appropriate models were selected to handle each type of data

* **Numerical data:** we decided to use K-Nearest Neighbors (KNN) and Random Forest Regressor to predict missing numerical values. KNN is able to capture local relationships between data points, thus effective for data that can be predicted based on similar observations in the dataset. While Random Forest Regressor is able to capture complex relationships in the dataset and is robust against overfitting, this is effective for data that is more prone to overfitting and has more complex relationships.
* **Categorical data:** we decided to use Random Forest Classifier to predict the missing categorical values.

Through the combination of these predictive models, the dataset was clean so that the integrity and patterns of the original dataset is preserved and enables accurate training for the AI model.

Missing values:

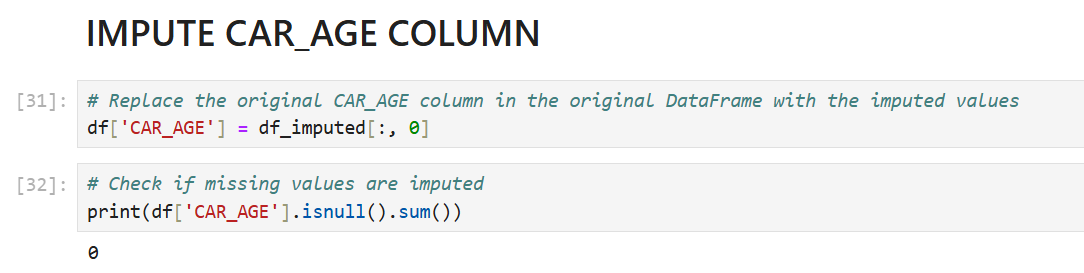
* Dataset 1: police\_report\_available, authorities\_contacted
* Dataset 2: address\_line2 (8505), city (54), customer\_education\_level (529), authorities\_contacted (1945), vendor\_id (3245)
* Dataset 3: age (7), yoj (548), income (570), home\_val (575), occupation (665), car\_age (639)

## **3.1. KNN Model**

KNN was used to impute the missing numerical values in the dataset. This is done by finding similar data on other columns and imputing the missing value based on the average of its nearest neighbour. KNN was used on columns like age, income, home\_val and car\_age in dataset 3 to impute the missing values from that column.



*Figure 8: Training the KNN model*



*Figure 9: Impute values into the ‘CAR\_AGE’ column*

### **3.1.1. Configuration Settings**

* n\_neighbours = 5: balances computations accuracy and complexity

### **3.1.2. Reasons for Model selection**

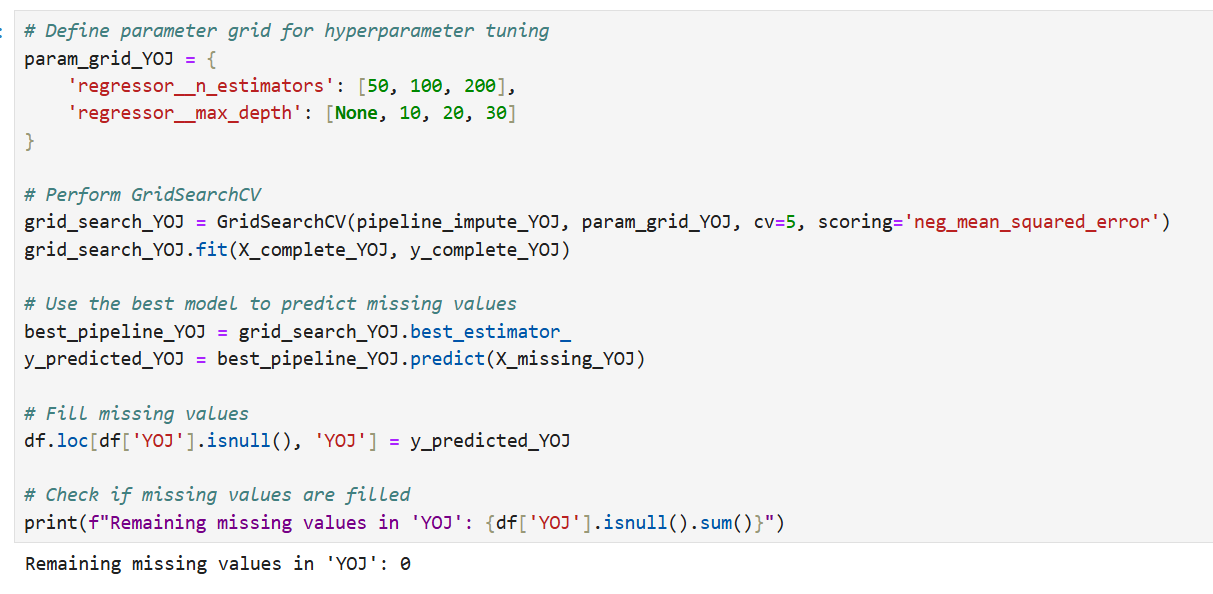
* Takes into account the similarity of records which is important when values relate to other columns.
* Ensures data completeness and standardisation
* Preserves integrity of the dataset by evaluating the relationship with other features.
* User friendly model as it is simple to implement and highly intuitive.
* Works with both numerical and categorical data thus is highly versatile
* Handles complex data with varied patterns as it is non-parametric hences, doesn’t make assumptions on current data distribution
* Fills in missing values based on the most common or average values of data points that are similar therefore, doesn’t need any complex pre-processing or assumptions.
* Efficient for moderate sized datasets due to it having distance calculations
* Captures non-linear relationships through feature space
* Hyper-parameters can be easily tuned to improve model performance.

## **3.2. Random Forest Models**

* **Random Forest Regressor:** was used to impute missing numerical values on columns that had a low accuracy score when using KNN such as yoj in dataset 3. This is done through the combination of predictions from numerous decision trees through the ensemble learning method to predict numerical values.
* **Random Forest Classifier:** was used to impute missing categorical values in the dataset such as customer\_education\_level in dataset 2 or occupation in dataset 3. This follows a similar method to Random Forest Regession however is used to assign classes to create a categorical result.

### **3.2.1. Configuration Settings**

* **GridSearchCV:** hyperparameter tuning used to improve accuracy
* **RandomisedSearchCV:** hyperparameter tuning used to improve accuracy
* **n\_estimators = 100:** number of trees was 100, which provided a good balance between both computation and performance time
* **Max\_depth = none:** allows the tree to fully grow increasing the ability of the model to capture complex data patterns



*Figure 10: Using GridSearchCV and Random Forest Regressor to impute YOJ column*

****

*Figure 11: Using RandomisedSearchCV and Random Forest Classfier to impute Occupation column*

### **3.2.2. Reasons for Model selection**

* Handles continuous target variables and categorical target variables thus is useful in datasets with mixed variables
* Can handle imbalance classes by adjusting class weights or oversampling the minority class thus, is good for datasets with skewed class distributions
* Performance is robust even in incomplete datasets
* Easy to interpret and straightforward to understand
* Non-parametric so it is able to model relationships that are non-linear and complex
* Can build multiple decision trees to simplify the preprocessing pipeline
* Can handle large datasets and is highly scalable, furthermore with its ensemble approach it ireduces variance and overfitting
* Hyperparameters can be tuned to improve performance and accuracy
* Captures non-linear data this the model is more robust
* Less chances of overfitting due to the creation of many decision trees improving generalisation

## **3.3. Why were these models chosen over other approaches?**

**3.3.1. Predictive Models**

* **Linear regression:** is a model that predicts missing values through linear relationships between target and independent variables, however, are not able to capture complex interactions in the data.
* **Decision Trees:** however, are prone to overfitting on small datasets like dataset 1 and dataset 2 and may not perform effectively with non-linear relationships.

**3.3.2. Supervised Learning Models**

* **Neural networks:** are able to predict missing values through the analysis of complex patterns in the dataset however, are computationally expensive and require significant tuning which, furthermore, they require large datasets for accurate training.
* **Gradient Boosting:** However, this is also computationally expensive and requires significant tuning while also having a slower performance.

**3.3.3. Unsupervised Learning Models**

* **K-means:** has the potential to group similar data points and impute missing values based on the mode or mean of that cluster however, it method mainly focuses on clustering hence not ideal for getting accurate prediction of missing values
* **Principal Component Analysis:** can be used to reduce dimensionality and use the dataset’s principal components to fill in missing values; however, this method will result in simple imputations failing to capture patterns in the data.

**3.3.4. Classifiers**

* **Logistic Regression: c**an be used to predict missing categorical values through modelling binary outcomes however, due to this it is not suited for imputing missing values.
* **Naive Bayes:** uses probability distributions to impute missing values thus, is not able to capt

**3.3.5. Chosen Models**

* **KNN:** was chosen for its ability to impute numerical values by identifying the closest data point based on similarity. The main reason it was chosen was its simplicity and ease of understanding and its ability to work well with small and large datasets.
* **Random Forest:** regressor and classifier models were chosen to impute numerical and categorical values. The main reason for being chosen include its ability to handle large datasets and capture a range of patterns. Also predictions from multiple decision trees ca help reduce the risks of overfitting hence is more robust than other models.

## **3.4. [NEW] Advanced Techniques**

When presenting our MVP to the client we were presented with the question of why we didn’t use AutoML to complete this problem and opted for a more difficult process. AutoML is an AI that automates the end-to-end process of machine learning application to increase efficiency. Some components of this include data preprcessinging, model selection, model training, model evaluation, and model deployment. Thus, for this deliverable we sought to explore Azure AutoML and if it would be beneficial to switch. AutoML provides many benefits such as accessibility lowering the learning curve and allowing non-experts to use machine learning, improves effciency by automating repetitive tasks and allows us to focus on higher-level problems, and improves model performance by exploring more algorithms and parameters. However, through this reaserch we found that AutoML would not be a good choice for this project. As our project requires deep knowledge on insurance claims it won’t be able catch the relationships in the dataset. Furthermore, as our datasets were of poor data quality the results may not be good as it relies on high quality data. Moreover, as some of our datasets are small AutoML may struggle to provide reliable results as result of overfitting. AutoML was also found to be computationally intensive thus, it was necessary to have time and hardware which was difficult to acquire with the scope of our project. Ultimately, we decided that we would not be able to use AutoML for our project to the need of specialised knowledge and data quality which were better found with KNN and Random Forest methods.

# **4. Detailed Data descriptions:**

## **4.1 Original Raw Datasets**

The raw datasets used in this project consist of three datasets sourced from Kaggle, each containing historical data on automobile insurance claims. These datasets were selected primarily because they include a column that explicitly indicates whether a claim was fraudulent, which is essential for training machine learning and AI models for fraud detection.

The original datasets are in a raw, unclean state, with many missing or non-standardized values. A thorough cleaning and enrichment process was applied to prepare them for analysis.

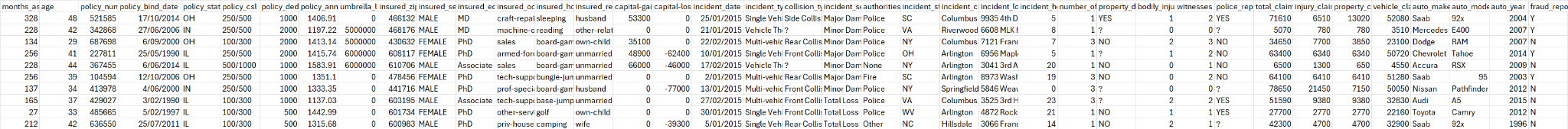
* Dataset 1: This dataset, containing 1,000 rows, is well-structured and widely available across various websites. It is frequently used in insurance fraud analysis due to its quality and completeness. The dataset was sourced from Kaggle, as it is an open-source platform. The dataset is hosted on GitHub at: [GitHub Link](https://github.com/mwitiderrick/insurancedata/blob/master/insurance_claims.csv), and the Kaggle source is available here: [Kaggle Link](https://www.kaggle.com/datasets/sumansuhag/insurance-dataset-csv).
* Dataset 2: This dataset is also quite well-structured and originally contains 10,000 rows but includes data on various types of fraud, not limited to automobile insurance. After filtering for automobile fraud, approximately 1,575 relevant rows were extracted. This dataset offers a broader range of fraud types but required focused extraction for the target use case of automobile insurance fraud detection. The dataset can be accessed on Kaggle: [Kaggle Link](https://www.kaggle.com/datasets/mastmustu/insurance-claims-fraud-data?select=insurance_data.csv).
* Dataset 3: This dataset, containing 10,303 rows, is the most challenging due to its high level of missing, blank, and non-standardized values. It required significant cleaning efforts to make it usable for analysis. Despite its issues, it provides valuable data for automobile insurance fraud detection once processed. The dataset can be found on Kaggle: [Kaggle Link](https://www.kaggle.com/datasets/xiaomengsun/car-insurance-claim-data).

These datasets form the foundation for the final dataset to be used by the fraud detection model. Providing this historical claims data allows the machine learning algorithms to learn patterns and anomalies indicative of fraudulent claims. The presence of a fraud indicator in each dataset ensures that they are particularly well-suited for supervised learning in fraud detection.

## 

## **4.2. Columns in Raw Datasets**

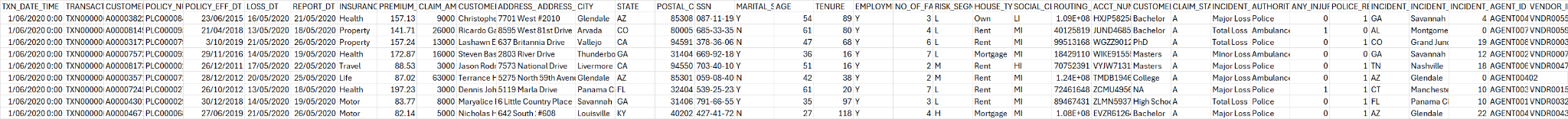
**Raw Dataset 1:**



*Figure 12: Raw Dataset 1*

| NO. | Column Name | Column Letter | Data Type | Description |
| --- | --- | --- | --- | --- |
| 1 | months\_as\_customer | A | Numerical | Number of months the individual has been a customer |
| 2 | age | B | Numerical | Age of the customer |
| 3 | policy\_number | C | Categorical | Unique identifier for the insurance policy |
| 4 | policy\_bind\_date | D | Date | The date when the policy was created/bound |
| 5 | policy\_state | E | Categorical | The state where the policy was issued |
| 6 | policy\_csl | F | Categorical | Combined Single Limit for the policy |
| 7 | policy\_deductable | G | Numerical | The amount of deductible in the policy |
| 8 | policy\_annual\_premium | H | Numerical | The annual premium paid for the insurance policy |
| 9 | umbrella\_limit | I | Numerical | The umbrella limit in the policy |
| 10 | insured\_zip | J | Categorical | Zip code of the insured person |
| 11 | insured\_sex | K | Categorical | Gender of the insured person |
| 12 | insured\_education\_level | L | Categorical | Education level of the insured person |
| 13 | insured\_occupation | M | Categorical | Occupation of the insured person |
| 14 | insured\_hobbies | N | Categorical | Hobbies of the insured person |
| 15 | insured\_relationship | O | Categorical | Relationship status of the insured person |
| 16 | capital-gains | P | Numerical | Capital gains of the insured person |
| 17 | capital-loss | Q | Numerical | Capital loss of the insured person |
| 18 | incident\_date | R | Date | Date of the incident |
| 19 | incident\_type | S | Categorical | Type of incident |
| 20 | collision\_type | T | Categorical | Type of collision |
| 21 | incident\_severity | U | Categorical | Severity of the incident |
| 22 | authorities\_contacted | V | Categorical | Whether authorities were contacted |
| 23 | incident\_state | W | Categorical | State where the incident occurred |
| 24 | incident\_city | X | Categorical | City where the incident occurred |
| 25 | incident\_location | Y | Categorical | Specific location of the incident |
| 26 | incident\_hour\_of\_the\_day | Z | Numerical | Hour of the day when the incident occurred |
| 27 | number\_of\_vehicles\_involved | AA | Numerical | Number of vehicles involved in the incident |
| 28 | property\_damage | AB | Categorical | Whether property damage occurred |
| 29 | bodily\_injuries | AC | Numerical | Number of bodily injuries |
| 30 | witnesses | AD | Numerical | Number of witnesses to the incident |
| 31 | police\_report\_available | AE | Boolean | Whether a police report is available |
| 32 | total\_claim\_amount | AF | Numerical | The total amount claimed |
| 33 | injury\_claim | AG | Numerical | Claim amount for injuries |
| 34 | property\_claim | AH | Numerical | Claim amount for property damage |
| 35 | vehicle\_claim | AI | Numerical | Claim amount for vehicle damage |
| 36 | auto\_make | AJ | Categorical | Make of the vehicle involved |
| 37 | auto\_model | AK | Categorical | Model of the vehicle involved |
| 38 | auto\_year | AL | Numerical | Year of the vehicle involved |
| 39 | fraud\_reported | AM | Boolean | Whether fraud was reported |

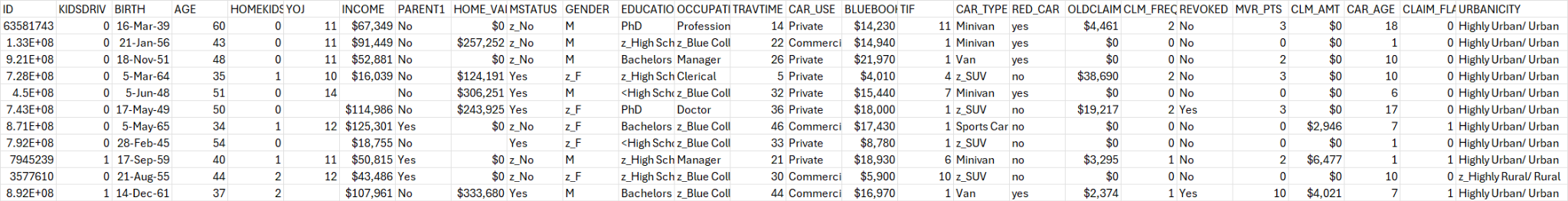
**Raw Dataset 2:**



*Figure 13: Raw Dataset 2*

| NO. | Column Name | Column Letter | Data Type | Description |
| --- | --- | --- | --- | --- |
| 1 | TXN\_DATE\_TIME | A | Date/Time | The date and time of the transaction |
| 2 | TRANSACTION\_ID | B | Categorical | Unique identifier for the transaction |
| 3 | CUSTOMER\_ID | C | Categorical | Unique identifier for the customer |
| 4 | POLICY\_NUMBER | D | Categorical | Unique identifier for the policy |
| 5 | POLICY\_EFF\_DT | E | Date | The policy effective date |
| 6 | LOSS\_DT | F | Date | Date when the loss occurred |
| 7 | REPORT\_DT | G | Date | Date when the incident was reported |
| 8 | INSURANCE\_TYPE | H | Categorical | Type of insurance |
| 9 | PREMIUM\_AMOUNT | I | Numerical | Premium amount for the insurance policy |
| 10 | CLAIM\_AMOUNT | J | Numerical | Amount claimed |
| 11 | CUSTOMER\_NAME | K | Categorical | Name of the customer |
| 12 | ADDRESS\_LINE1 | L | Categorical | Address line 1 of the customer |
| 13 | ADDRESS\_LINE2 | M | Categorical | Address line 2 of the customer |
| 14 | CITY | N | Categorical | City of the customer |
| 15 | STATE | O | Categorical | State of the customer |
| 16 | POSTAL\_CODE | P | Categorical | Postal code of the customer |
| 17 | SSN | Q | Categorical | Social Security Number |
| 18 | MARITAL\_STATUS | R | Categorical | Marital status of the customer |
| 19 | AGE | S | Numerical | Age of the customer |
| 20 | TENURE | T | Numerical | Number of years the customer has been with the insurance company |
| 21 | EMPLOYMENT\_STATUS | U | Categorical | Employment status of the customer |
| 22 | NO\_OF\_FAMILY\_MEMBERS | V | Numerical | Number of family members |
| 23 | RISK\_SEGMENTATION | W | Categorical | Risk category assigned to the customer |
| 24 | HOUSE\_TYPE | X | Categorical | Type of house the customer owns or rents |
| 25 | SOCIAL\_CLASS | Y | Categorical | Social class of the customer |
| 26 | ROUTING\_NUMBER | Z | Categorical | Customer’s bank routing number |
| 27 | ACCT\_NUMBER | AA | Categorical | Customer’s bank account number |
| 28 | CUSTOMER\_EDUCATION\_LEVEL | AB | Categorical | Education level of the customer |
| 29 | CLAIM\_STATUS | AC | Categorical | Status of the claim |
| 30 | INCIDENT\_SEVERITY | AD | Categorical | Severity of the incident |
| 31 | AUTHORITY\_CONTACTED | AE | Boolean | Whether the authorities were contacted |
| 32 | ANY\_INJURY | AF | Boolean | Whether any injury was reported |
| 33 | POLICE\_REPORT\_AVAILABLE | AG | Boolean | Whether a police report is available |
| 34 | INCIDENT\_STATE | AH | Categorical | State where the incident occurred |
| 35 | INCIDENT\_CITY | AI | Categorical | City where the incident occurred |
| 36 | INCIDENT\_HOUR\_OF\_THE\_DAY | AJ | Numerical | Hour of the day when the incident occurred |
| 37 | AGENT\_ID | AK | Categorical | Unique identifier for the insurance agent |
| 38 | VENDOR\_ID | AL | Categorical | Unique identifier for the vendor |

**Raw Dataset 3:**



*Figure 14: Raw Dataset 3*

| NO. | Column Name | Column Letter | Data Type | Description |
| --- | --- | --- | --- | --- |
| 1 | ID | A | Categorical | Unique identifier for the customer |
| 2 | KIDSDRIV | B | Numerical | Number of kids driving in the household |
| 3 | BIRTH | C | Date | Birth date of the customer |
| 4 | AGE | D | Numerical | Age of the customer |
| 5 | HOMEKIDS | E | Numerical | Number of kids living in the household |
| 6 | YOJ | F | Numerical | Years on the job |
| 7 | INCOME | G | Numerical (Currency) | Income of the customer |
| 8 | PARENT1 | H | Categorical | Whether the customer is a single parent |
| 9 | HOME\_VAL | I | Numerical (Currency) | Value of the home |
| 10 | MSTATUS | J | Categorical | Marital status |
| 11 | GENDER | K | Categorical | Gender of the customer |
| 12 | EDUCATION | L | Categorical | Education level of the customer |
| 13 | OCCUPATION | M | Categorical | Occupation of the customer |
| 14 | TRAVTIME | N | Numerical | Travel time to work |
| 15 | CAR\_USE | O | Categorical | Whether the car is used for private or commercial purposes |
| 16 | BLUEBOOK | P | Numerical (Currency) | Value of the car |
| 17 | TIF | Q | Numerical | Time in force of the insurance policy |
| 18 | CAR\_TYPE | R | Categorical | Type of car |
| 19 | RED\_CAR | S | Boolean | Whether the car is red |
| 20 | OLDCLAIM | T | Numerical (Currency) | Amount claimed in prior incidents |
| 21 | CLM\_FREQ | U | Numerical | Frequency of claims |
| 22 | REVOKED | V | Boolean | Whether the driver's licence has been revoked |
| 23 | MVR\_PTS | W | Numerical | Motor Vehicle Record points |
| 24 | CLM\_AMT | X | Numerical (Currency) | Amount claimed in the current incident |
| 25 | CAR\_AGE | Y | Numerical | Age of the car |
| 26 | CLAIM\_FLAG | Z | Boolean | Whether a claim was filed |
| 27 | URBANICITY | AA | Categorical | Urban/rural classification |

## **4.3. Cleaning & Enriching**

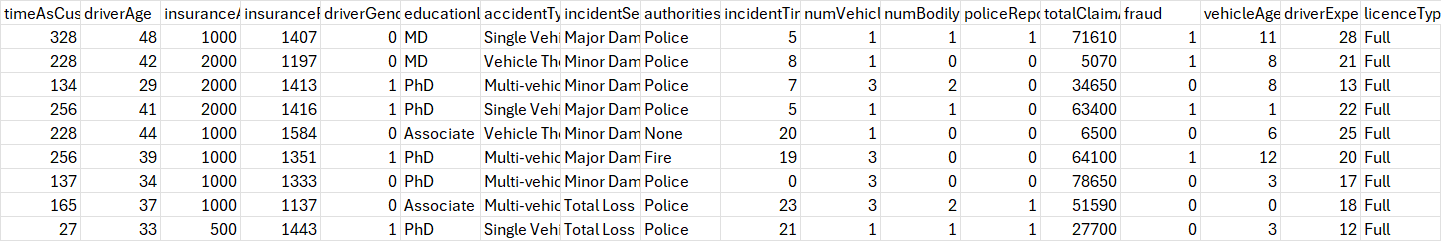
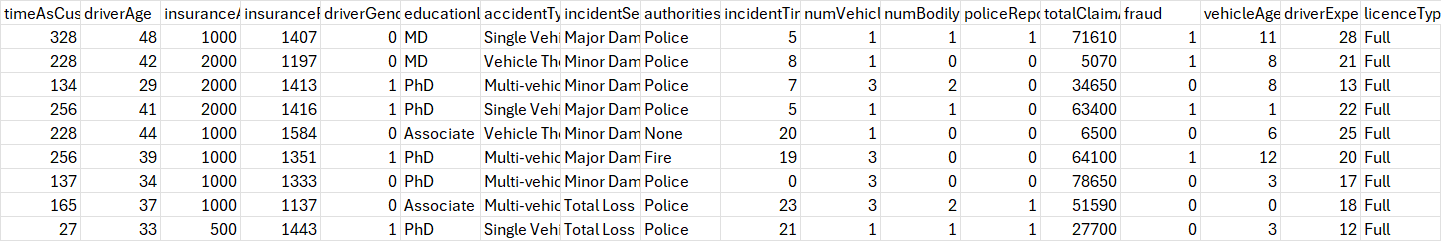
**Dataset 1 & Dataset 2**

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 licence, while the random variable represents the variability in the time drivers might wait before getting their licence.
* 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

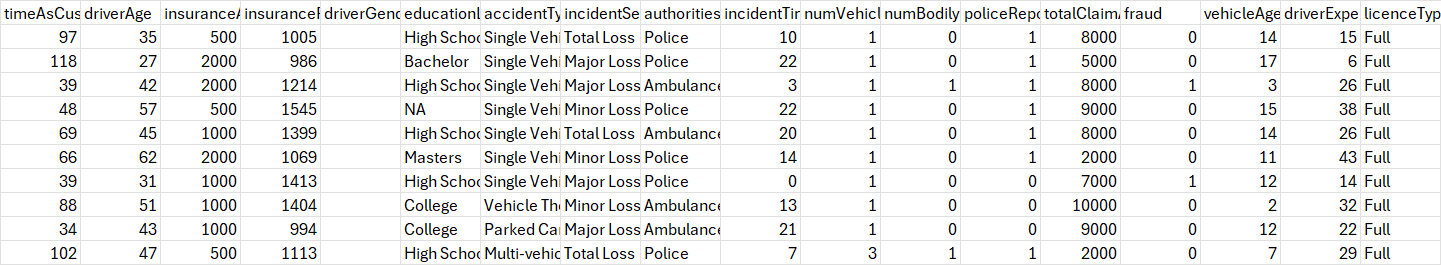
For Dataset 2, no gender data was available, so the driverGender column remains blank for that dataset, ensuring consistency without introducing any erroneous or bias inducing data.

**Cleaned & Enriched Dataset 1:**

*Figure 15: Cleaned and Enriched Dataset 1*

| NO. | Column Name | Column Letter | Data Type | Description |
| --- | --- | --- | --- | --- |
| 1 | timeAsCustomer | A | Numerical | Number of months the individual has been a customer |
| 2 | driverAge | B | Numerical | Age of the driver |
| 3 | insuranceAccess | C | Numerical | The level of access to insurance |
| 4 | insurancePremium | D | Numerical | The premium amount for the insurance policy |
| 5 | driverGender | E | Boolean | Gender of the driver (0 = Male, 1 = Female) |
| 6 | educationLevel | F | Categorical | Education level of the driver |
| 7 | accidentType | G | Categorical | Type of accident |
| 8 | incidentSeverity | H | Categorical | Severity of the incident |
| 9 | authoritiesInvolved | I | Categorical | Whether authorities were involved |
| 10 | incidentTime | J | Numerical | Time of the incident |
| 11 | numVehiclesInvolved | K | Numerical | Number of vehicles involved in the accident |
| 12 | numBodilyInjuries | L | Numerical | Number of bodily injuries |
| 13 | policeReportBool | M | Boolean | Whether a police report was available (0 = No, 1 = Yes) |
| 14 | totalClaimAmount | N | Numerical | The total claim amount |
| 15 | fraud | O | Boolean | Whether the claim was fraudulent (0 = No, 1 = Yes) |
| 16 | vehicleAge | P | Numerical | Age of the vehicle |
| 17 | driverExperience | Q | Numerical | Number of years of driving experience |
| 18 | licenceType | R | Categorical | Type of licence held by the driver |

### 

**Cleaned & Enriched Dataset 2:**  


*Figure 16: Cleaned and Enriched Dataset 2*

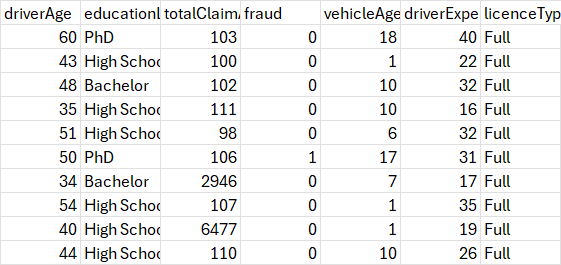
| NO | Column Name | Column Letter | Data Type | Description |
| --- | --- | --- | --- | --- |
| 1 | timeAsCustomer | A | Numerical | Number of months the individual has been a customer |
| 2 | driverAge | B | Numerical | Age of the driver |
| 3 | insuranceAccess | C | Numerical | The level of access to insurance |
| 4 | insurancePremium | D | Numerical | The premium amount for the insurance policy |
| 5 | driverGender | E | Boolean (0 = Male, 1 = Female, empty) | Gender of the driver (0 = Male, 1 = Female) EMPTY |
| 6 | educationLevel | F | Categorical | Education level of the driver |
| 7 | accidentType | G | Categorical | Type of accident |
| 8 | incidentSeverity | H | Categorical | Severity of the incident |
| 9 | authoritiesInvolved | I | Categorical | Whether authorities were involved |
| 10 | incidentTime | J | Numerical | Time of the incident |
| 11 | numVehiclesInvolved | K | Numerical | Number of vehicles involved in the accident |
| 12 | numBodilyInjuries | L | Numerical | Number of bodily injuries |
| 13 | policeReportBool | M | Boolean (0 = No, 1 = Yes) | Whether a police report was available (0 = No, 1 = Yes) |
| 14 | totalClaimAmount | N | Numerical | The total claim amount |
| 15 | fraud | O | Boolean (0 = No, 1 = Yes) | Whether the claim was fraudulent (0 = No, 1 = Yes) |
| 16 | vehicleAge | P | Numerical | Age of the vehicle |
| 17 | driverExperience | Q | Numerical | Number of years of driving experience |
| 18 | licenceType | R | Categorical | Type of licence held by the driver |

**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 biassed 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:**



*Figure 17: Cleaned and 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).

| No. | Column Name | Column Letter | Data Type | Description |
| --- | --- | --- | --- | --- |
| 1 | driverAge | A | Numerical | Age of the driver |
| 2 | educationLevel | B | Categorical | Education level of the driver |
| 3 | totalClaimAmount | C | Numerical | The total claim amount |
| 4 | fraud | D | Boolean | Whether the claim was fraudulent (0 = No, 1 = Yes) |
| 5 | vehicleAge | E | Numerical | Age of the vehicle |
| 6 | driverExperience | F | Numerical | Number of years of driving experience |
| 7 | licenceType | G | Categorical | Type of licence held by the driver |

## **4.4. Generating Data:**

The dataset consists of 4,000 rows of synthetic data generated using Method 1 and an additional 4,000 rows generated using Method 2. Method 1 involved using statistical remapping and random sampling techniques to synthesise data based on trends in the original dataset. Method 2 utlisied sophisticated machine learning methods, such as Generative Adversarial Networks (GANs), to create more realistic synthetic data by modelling the underlying patterns of the original dataset.

We used both methods to develop a more diverse and comprehensive dataset for analysis and machine learning tasks.

### **4.4.1. Method 1:**

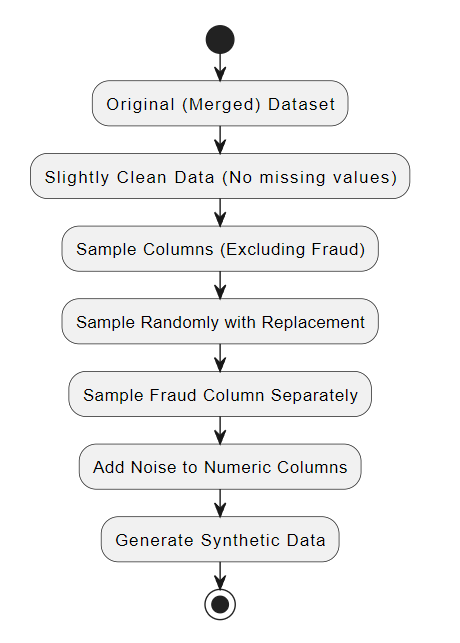
Synthetic Data Method 1 generates additional data by sampling from the original dataset, maintaining the same distribution and characteristics of the real data. This method is used to expand the dataset, especially when more data is needed for training machine learning models, but without introducing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

### **4.4.2. 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.

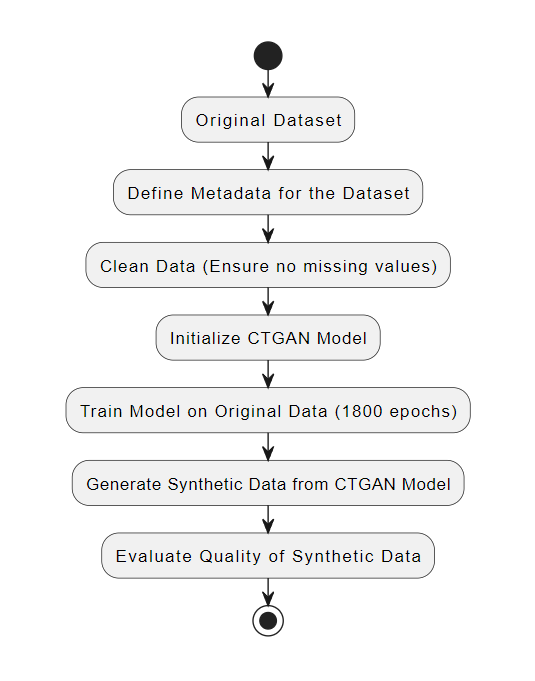
**Fraud Column Reliability:**

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.

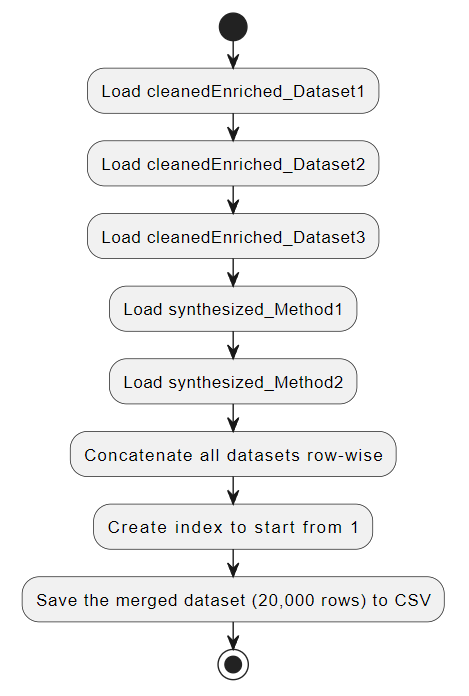
Steps:

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:** using the original dataset train the model so that it learns the underlying patterns and relationships in the dataset.
6. **Generate Synthetic Data:** After training the model, it generates new synthetic data that reflects the original data in terms of structure and distribution.
7. **Evaluate Quality:** The synthetic data that is generated is evaluated against the original data to ensure quality and reliability, particularly for crucial columns like fraud.

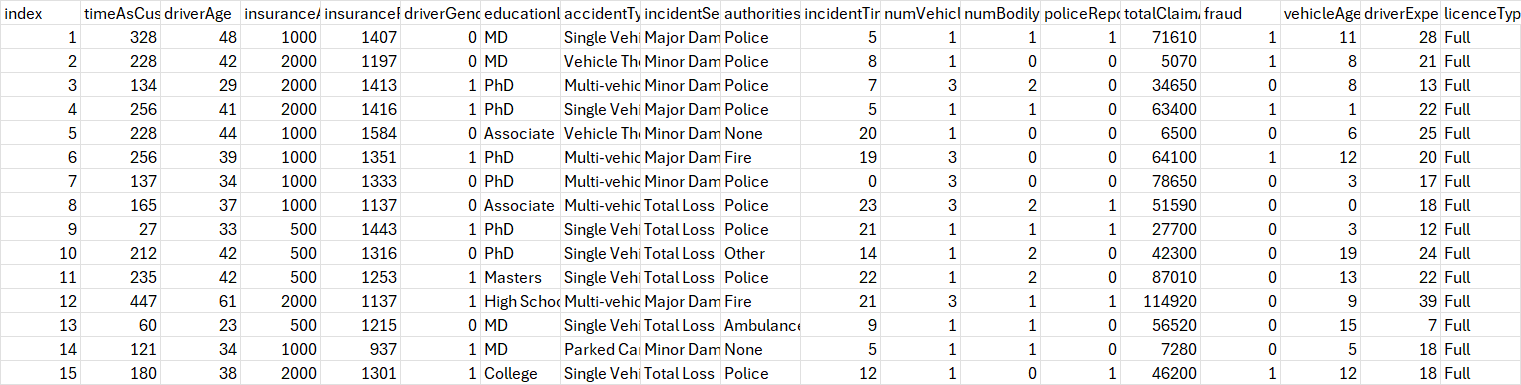
### **4.4.3. Merging Datasets:**

The process of merging datasets involving combining our three clean and enriched dataset with our two synthesised datasets to create a single merged dataset with ~20,500 rows, This ensures that the data is structured, consolidates and prepared for further analysis or machine learning activities.

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. **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.
4. **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.

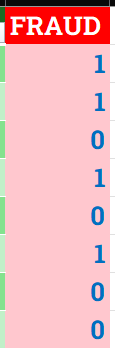
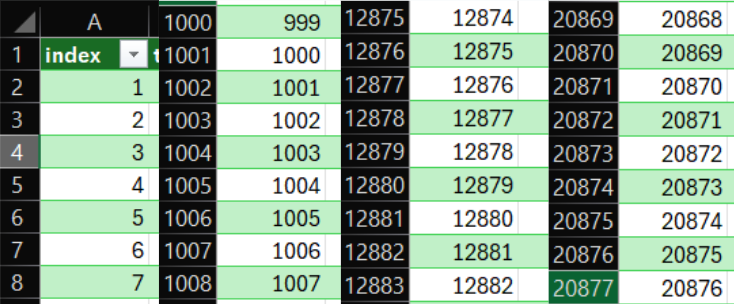
Merged Dataset (20877 rows):



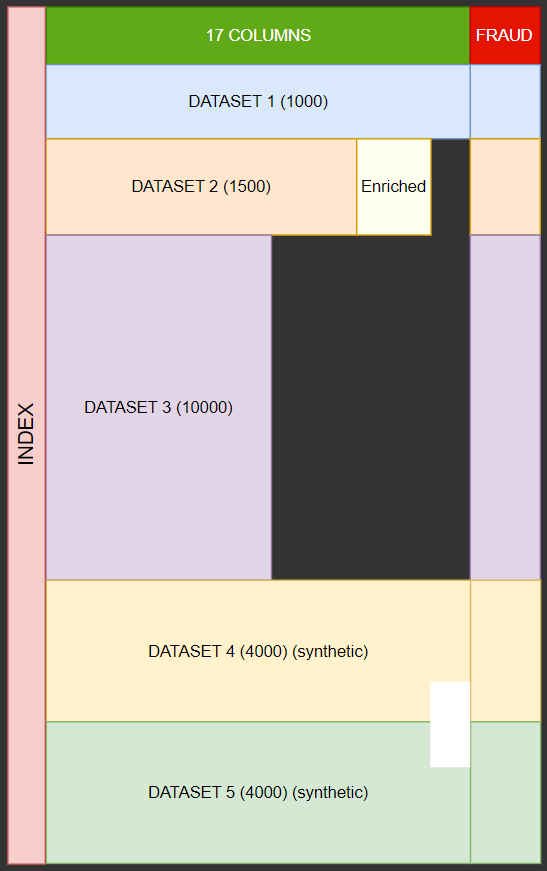
*Figure 18: Merged Dataset*

| NO. | Column Name | Column Letter | Data Type | Description |
| --- | --- | --- | --- | --- |
| 1 | index | A | Numerical | Row index for tracking purposes |
| 2 | timeAsCustomer | B | Numerical | Number of months the individual has been a customer |
| 3 | driverAge | C | Numerical | Age of the driver |
| 4 | insuranceAccess | D | Numerical | The level of access to insurance |
| 5 | insurancePremium | E | Numerical | The premium amount for the insurance policy |
| 6 | driverGender | F | Boolean | Gender of the driver (0 = Male, 1 = Female) |
| 7 | educationLevel | G | Categorical | Education level of the driver |
| 8 | accidentType | H | Categorical | Type of accident |
| 9 | incidentSeverity | I | Categorical | Severity of the incident |
| 10 | authoritiesInvolved | J | Categorical | Whether authorities were involved |
| 11 | incidentTime | K | Numerical | Time of the incident |
| 12 | numVehiclesInvolved | L | Numerical | Number of vehicles involved in the accident |
| 13 | numBodilyInjuries | M | Numerical | Number of bodily injuries |
| 14 | policeReportBool | N | Boolean | Whether a police report was available (0 = No, 1 = Yes) |
| 15 | totalClaimAmount | O | Numerical | The total claim amount |
| 16 | fraud | P | Boolean | Whether fraud was reported (0 = No, 1 = Yes) |
| 17 | vehicleAge | Q | Numerical | Age of the vehicle |
| 18 | driverExperience | R | Numerical | Number of years of driving experience |
| 19 | licenceType | S | Categorical | Type of licence held by the driver |

Index: Fraud:



*Figure 19: Merged dataset index and fraud column*



*Figure 20: Structure of resulting dataset after merging*

**TEST SPECIFICATION**

# **1. Model Evaluation**

## **1.1. Model Evaluation and Comparison**

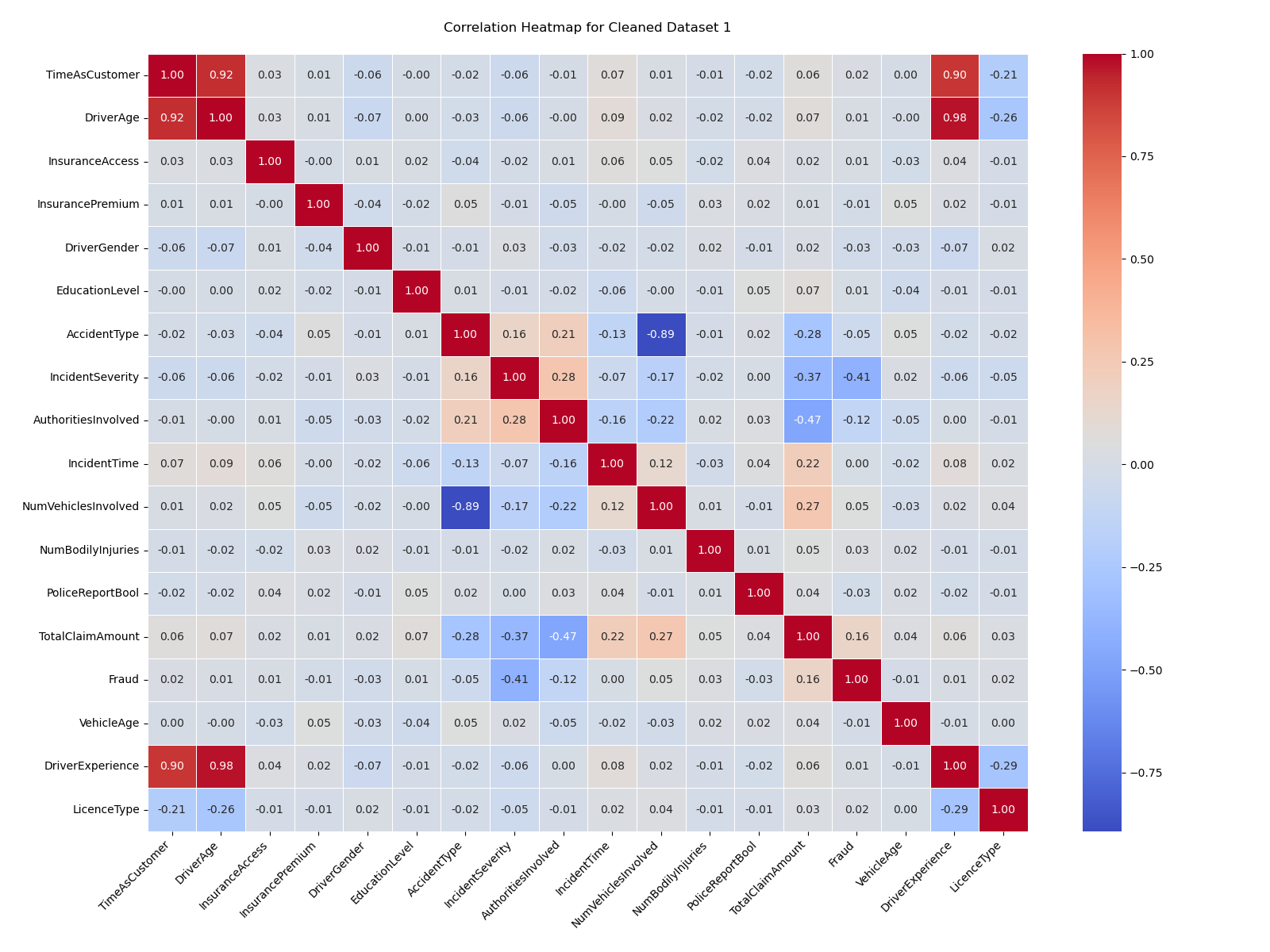
**Data Cleaning, Synthesisation, and Imputation:**

In order to confirm the precision and trustworthiness of our data, a comprehensive evaluation was conducted on the datasets, including comparisons between original, cleaned, and synthesised data. This step was essential in verifying the reliability and continuity of the data sets utilised in our research.

### 

### **1.1.1. Correlation Matrix and ANOVA Analysis:**

**Correlation Matrix Comparison:** Upon completing our data cleaning processes and synthesising data sets for analysis purposes. We examined correlation matrix comparisons to assess any alterations, in the strength of relationships among numerical features before and after cleaning our datasets and between the cleaned data and synthesised data sets. By scrutinising these correlation matrices for changes resulting from our data enhancement procedures and ensuring the consistency of variable relationships, across datasets remained intact. We were able to validate the effectiveness of our preprocessing steps effectively.



*Figure 1: Correlation Heatmap for clean dataset 1*

**ANOVA Tests:** We utilised ANOVA to compare the values across groups, in the datasets aiming to examine any variations present among them. Our team conducted ANOVA analyses to determine the discrepancies between the data distributions in the datasets compared to the initial dataset and to contrast the distributions between polished data and generated data. This statistical examination aided us in confirming that preprocessing procedures did not introduce any biases or notable alterations, in the distribution of data.

**Consistency Check:** In both synthesised datasets (generated using different methods),the overall count of fraudulent cases is tallied and matched up for comparison purposes. The number of fraud instances, in both scenarios hovered close to 400 points in line with the dataset figures. This alignment in the fraud case counts implies that both methods effectively mirror the distribution of activities. Validation through Distribution: Given the similarity in fraud counts across the synthesised datasets, it is highly unlikely that such results are coincidental. This suggests that the models are accurately capturing the underlying patterns of fraud found in the data set and confirming the effectiveness of the synthetic generation techniques.

### **1.1.2. Validation of Synthesised Data:**

**Fraud Count Consistency:** Both synthetic datasets generated approximately 400 instances of fraud, providing strong validation for the synthesis process. Given the rarity of fraud cases, it is improbable that both synthesised datasets would have this coincidence unless the process successfully replicated the underlying distribution patterns of the real data. This consistency was used as an additional checkpoint to ensure that the synthetic data reflected the real world occurrences correctly.

##### 

### **1.1.3. Imputation Techniques:**

Missing data presents a challenge that can significantly impact the quality and reliability of model predictions. We explored multiple imputation methods for both numerical and categorical columns. Given the complexity of our dataset and the need for more advanced handling of missing values, we decided to use K-Nearest Neighbors (KNN), Support Vector Regression (SVR), and Random Forest models for imputation. We used several key metrics to evaluate the accuracy and effectiveness of our imputation methods for both numerical and categorical columns.

**For Numerical Columns:**

The primary metrics used for evaluating imputation in numerical columns were:

* **Mean Squared Error (MSE):** MSE measures the average squared difference between the imputed values and the actual values (where known). Our target MSE was for it to be as low as possible ideally <0.10
* **Mean Absolute Error (MAE):** MAE is similar to MSE but focuses on the absolute differences between the imputed values and the actual values, making it less sensitive to large errors.Our target MSE was for it to be as low as possible ideally <0.05.
* **R-squared (R²):** R² measures how well the imputed values explain the variance in the dataset. Our target for R2 was to be as close to 1 ideally >0.80.

These metrics provided insight into how well the imputed values aligned with the true values in the dataset, allowing us to optimise and refine our approach.

##### 

**For Categorical Columns:**

For categorical columns, accuracy metrics are more appropriate for evaluating model performance. We focused on:

* **Accuracy:** Measures the percentage of correctly imputed categories compared to the total number of categories.Our aim for imputation accuracy was to >75%
* **Consistency Checks**: For instance, checking if the imputed values for columns like "Occupation" and "Education Level" logically fit the other related columns (e.g., if a person with a PhD is categorised under an occupation like "Clerical" might indicate poor imputation).

#### 

## **1.2. How we compared and evaluated Models and decided what to choose**

In our project, we used several models for both imputation and classification, including K-Nearest Neighbors (KNN), Support Vector Regression (SVR), and Random Forest classifier/ regressor. While the choice of model is essential, the performance evaluation metrics we used to assess these models played an even more significant role in guiding our final decisions. These evaluation techniques allowed us to assess the models' ability to accurately impute missing data and classify fraudulent and non-fraudulent claims. While also comparing and contrasting the models objectively, ensuring we selected the most suitable approach for each specific task.

### **1.2.1. R² (Coefficient of Determination)**

The R² metric is particularly useful for continuous variables. It measures how well the model explains the variability in the target variable. An R² closer to 1 indicates that the model captures most of the variability, whereas an R² closer to 0 means the model explains little of the variance.

**Impact on model choice:** For tasks like imputing continuous columns, we relied heavily on R² to understand how well the model was capturing the underlying patterns. A low R² suggested the model was underperforming, prompting us to try different models or adjust the features being used. On the other hand, a higher R² indicated the model was a good fit for the data, leading us to keep and refine those models.

For the YOJ (Years on Job) column, which is continuous data, R² became the most critical. The first model we applied gave us an R² score of 42%, indicating that the model explained only 42% of the variance in the target variable. This was relatively low, and it informed us that the model wasn't capturing the complexity of the data. Therefore we used it to refine the model. With such a low R², we explored other models (KNN, SVR, and Random Forest) to determine if we could achieve better performance. For continuous data like YOJ, a higher R² is crucial because it shows how well the model is capturing the real-world variations in the data. After hyperparameter tuning, we managed to improve the R² score to 99%, justifying our selection of the final model.

### **1.2.2. Accuracy**

Accuracy is a key metric for evaluating classification tasks. It simply measures the percentage of correctly predicted outcomes out of the total predictions made. While accuracy is often the most intuitive metric, it can sometimes be misleading, especially when dealing with imbalanced datasets.

**Impact on model choice:** In our tasks, accuracy helped us identify models that performed well overall. However, we also recognized its limitations, especially in cases where certain categories or classes were underrepresented. For example, if only a small portion of the data represented a specific outcome (e.g., fraudulent claims), a model might have high accuracy while still failing to predict the minority class effectively. As a result, we supplemented accuracy with other metrics (like precision and recall) to ensure the model was suitable for all parts of the data.

When imputing the 'Occupation' column, accuracy was used to evaluate how well each model predicted missing values. Initial attempts with one model resulted in low accuracy scores, indicating that it struggled to capture the relationships in the data. By analysing the accuracy, we identified that tweaking the hyperparameters and switching models improved performance.

For the 'Education Level' column, the accuracy score again played a crucial role. When accuracy was low, we either adjusted the model or added more features to improve predictions. The feedback from accuracy scores ensured that we did not settle on a model unless it provided reasonable confidence in the imputed values. While accuracy was a useful initial measure, we avoided making decisions just based on it, because we didn't always capture the nuances in our dataset.

#### 

### **1.2.3. Mean Squared Error (MSE) and Mean Absolute Error (MAE)**

Both MSE and MAE are fundamental metrics for evaluating the performance of models, especially for tasks like imputing missing values or handling continuous data. These metrics help quantify the prediction errors, guiding us toward selecting models that minimise these errors.

**MSE**: Measures the average squared difference between actual and predicted values. Because it squares the differences, it penalises larger errors more heavily. This makes MSE especially useful when outliers need to be handled with care. Impact on model choice: In cases where larger errors are undesirable, we prioritised models with lower MSE scores, knowing they would provide more reliable predictions, particularly for continuous data.For example, our results for the HOMEVAL column was an MSE of 0.179 suggests that, on average, the value is relatively low meaning the imputation predictions are close to the true values, with small errors.

**MAE**: Represents the average of the absolute differences between actual and predicted values. Unlike MSE, it doesn’t overly penalise larger errors, making it more robust to outliers. Impact on model choice: When MSE indicated that certain models were too sensitive to outliers, we referred to MAE to see if a model could still offer stable performance. Models with consistently low MAE scores were prioritised for their robustness in handling diverse data points. By comparing MSE and MAE, we balanced error sensitivity and the need for stability, selecting models that provided the best trade-off between the two metrics.For example, our results for the HOMEVAL column was a MAE of 0.075 suggests that, on average, the predictions deviate from the actual values by about 0.115 units. This is a low error, signifying good predictive performance.

For the YOJ (Years on Job) column MSE and MAE were used for Error Minimisation. Alongside R², we used MSE and MAE to measure the precision of our predictions. The models that minimised these errors were retained, while those with high MSE or MAE were discarded. Random Forest, in particular, performed well in minimising these errors, making it a suitable choice for YOJ imputation. R² was a crucial indicator for assessing models’ ability to generalise and provide accurate predictions for continuous data, influencing which models we selected for such tasks.

### 

## **1.3. Overall**

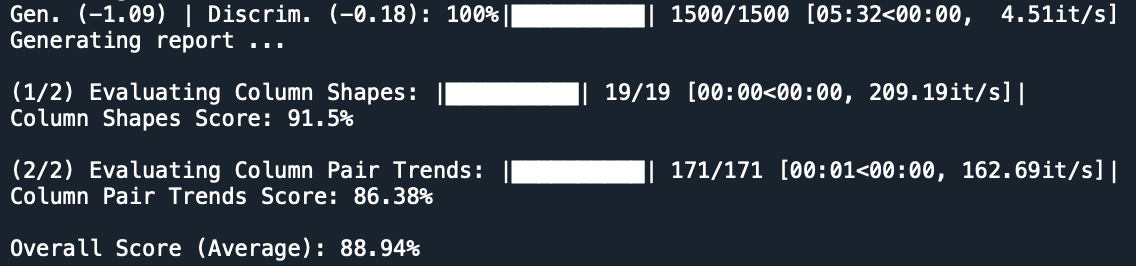
Ultimately, the evaluation metrics like MSE, MAE, accuracy, and R² guided our decision-making far more than the models themselves. These metrics provided detailed insights into the strengths and weaknesses of each model, allowing us to choose models that performed best for specific tasks.

Instead of focusing solely on which model to use, we focused on how these evaluation techniques informed us about model performance. This approach ensured that we selected models based on how well they fit the data and how effectively they minimised errors or maximised predictive power, rather than relying purely on the theoretical strengths of each model.

# **2. Performance evaluation results:**

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.



*Figure 2: Evaluation metric for the quality of synthesised data for method 2*

The screenshot above shows the results of a test which was run to evaluate the quality of the synthetic data generated using the synthetic method 2 which is Conditional Tabular Generative Adversarial Networks (CTGAN). This test is run using a function in the Synthetic data vault which is a library in python called ‘evaluate\_quality’.

The column shapes score indicates how closely the distribution of each column in the synthetic data matches with the dataset that it was trained on. A score of 91.5% indicates that the distribution of the synthetic data was similar to the real data.

The column pair trends score evaluates the relationships between the columns in the synthetic dataset compared to that of the dataset that it was trained on. This score will indicate the pairwise trends. A score of 86.38% indicates that most column pairs show a similar pattern to that of the dataset it was trained on.

The overall score indicates the actual accuracy of the synthetic method. A high score of 88.94% indicates that this method is good and most of the data synthesised closely mimics the actual dataset which is exactly what we want.

**[NEW]** The positive evaluation results of the synthesised method indicates high accuracy and reliability of the synthesised data. This will mean that the model will be more accurate while making a prediction even in a situation where the data is not real. Moreover, as the synthesised data closely mimics the actual dataset we base it on, this will reduce the risk of the model overfitting. This will mean that the outcome from the model will be accurate and this will contribute to the success of the project.

# **3. References**

Keall, M.D. and Frith, W.J. *OLDER DRIVER CRASH RATES IN RELATION TO TYPE AND QUANTITY OF TRAVEL*. Australasian Transport Research Forum.

<https://australasiantransportresearchforum.org.au/wp-content/uploads/2022/03/2003_Keall_Frith.pdf>

# 

| **SCRIPTS / MODEL EXECUTION**  **Team 14**  Revolutionise Claims Management: Unleash the  Power of GenAI for Peak Efficiency in the Insurance  Sector |
| --- |

**TABLE OF CONTENTS**

[**1. How is our solution delivered 1**](#_z69jq6f8h50k)**46**

[**2. How to run our code using Github 1**](#_e0f9joahez4k)**46**

[Step 1: Clone the Repository 1](#_6mmq7kf3n4)46

[Step 2: Run the Code](#_9iztjvhzsxr9) 147

[Method ONE: Run All Scripts with a Single Command (in development)](#_rmdoamxo7eqh) 147

[Method TWO: Run Each Script Individually](#_ma7byk11zocu) 147

[2.1. Script description:](#_xepnhn3lrstw) 148

[**3. How to run our code on Azure**](#_xg3ghqr1k8ru) **149**

[**4. Troubleshooting/ Where to get help ?**](#_lzkwx1hwt23d) **149**

[4.1. General Setup Issues](#_2wuxuzomdp9y) 149

[4.2. Local Code Execution Issues](#_ups520hswc8u) 150

[4.3. Azure Deployment or Execution Issues](#_px02angu82ke) 150

[4.4. Data-Related Issues](#_xj2vxkaeg7xc) 151

[4.5. Where to Get Help?](#_yflvauguqiu) 151

[**5. Website**](#_fsj1l118itlm) **151**

[5.1. How the website works (back end & front end)](#_aqha6s82q4j3) 151

[5.2. How to Use the Website](#_hwl446gj5dr2) 152

[5.3. Functionality](#_v3li46ruq7nz) 154

[5.4. Troubleshooting (In Case the Website Doesn't Function)](#_d94qs7mxjfvo) 155

[**6. End-Users 1**](#_2m0dnztto7bj)**56**

[6.1. Claims Agent 1](#_lumgnc1env9)57

[6.2. Supervisors/Managers 1](#_77itj9yu6ado)58

[6.3. Business Analyst 1](#_xmgxttwpfyqe)58

[6.4. System Administrators 1](#_w3oayg8wba7l)59

[6.5. Data Engineers 1](#_99jt7kg2u)59

[6.6. AI Engineers 1](#_8506krltm4xa)60

[**7. Frequently Asked Questions (FAQs) 1**](#_m98df3yqu4l1)**61**

# **1. How is our solution delivered**

Our solution is delivered through a combination of GitHub and Azure services to ensure seamless development, deployment, and execution. Here’s how the process works:

1. GitHub as the Central Repository:

* We use GitHub to host both the codebase and documentation. It acts as the central repository for our development team.
* GitHub enables version control, allowing our team to track changes, collaborate efficiently, and maintain an organised workflow.
* GitHub plays a crucial role in automating the deployment of our infrastructure and scripts to Azure, making the process streamlined and efficient.

1. Development and Testing:

* While Azure serves as the remote environment where we run our code, all development and testing of the Python scripts are carried out locally. This allows for a faster development cycle, ensuring that we refine the scripts before deployment.
* Once the scripts are thoroughly tested and ready, they are hosted on Azure to interact with the data stored in Azure Data Lake Gen2.

1. Azure as the Execution Environment:

* Azure is utilised as a remote server where we run our finalised scripts.
* The raw data is hosted on Azure Data Lake Gen2, and Azure Functions are employed to run Python scripts that clean, enrich, and synthesise the data.
* After the pipeline is executed, the final processed dataset is produced and stored on Azure.

In summary, GitHub serves as the foundation for development, documentation, and deployment, while Azure provides the execution environment to process and generate the final dataset. This approach ensures that both code and data are handled securely and efficiently, with proper version control and automation in place.

# **2. How to run our code using Github**

To run the code from our GitHub repository, follow these steps:

## **Step 1: Clone the Repository**

First, you will need to clone the repository to your local machine. Use the following command in your terminal:

*`git clone https://github.com/AlanDataPortfolio/ey-azure-fn-pipeline.git`*

This will download the entire codebase to your local environment.

## **Step 2: Run the Code**

There are two methods to execute the code:

### **Method ONE: Run All Scripts with a Single Command (in development)**

We have provided a master script that automates the process by executing all the other scripts in the correct sequence. This is the simplest and fastest method to run the pipeline.

To run the master script, navigate to the project directory and execute the following command:

*`bash run\_all.sh` for Unix*

*`pwsh run\_all.ps1` for Windows*

This script will automatically run all the necessary scripts to set up the infrastructure, clean the data, enrich it, synthesise the results, and merge them into the final dataset.

### 

### **Method TWO: Run Each Script Individually**

Alternatively, you can run each script manually if you prefer more control over the execution process. To do this, navigate to the src folder and run each script in the order below:



This method gives us the flexibility to review and monitor each step individually. This method will run some of the scripts in parallel. We can refer to them as stages as seen in the flowchart. In stage 1, the cleaning scripts for the 3 datasets can run at the same time. After cleaning the datasets, we can move on to stage 2 where we enrich the 3 datasets. The enriching process is essentially refining the cleaned datasets. The enriching scripts can also run in parallel. Then, we move on to stage 3 where we merge dataset 1 and 2. The merging is done so that data can be synthesised based on datasets 1 and 2. In this stage, dataset 1 and dataset 2 is combined together. This script cannot run parallel to other scripts. Stage 4 is where data synthesisation takes place. Data is synthesised based on the merged dataset. Two synthesisation methods are used and the scripts for these two methods can run parallel to each other. Finally, we move on to the last stage which is stage 5. Here, all of the datasets are merged together to have dataset consisting of 20,900 rows approximately. In this stage, the 3 enriched datasets and the 2 synthesised datasets are combined together to create a comprehensive dataset.

## **2.1. Script description:**

Our scripts will run in stages. In stage 1, the raw datasets are loaded and are being cleaned. Universal paths are used in all of the scripts to ensure that the code runs in any environment. In stage 1, missing values are filled in using logic and imputation methods. Thecolumns named are renamed using camelCase for consistency across the datasets. In this stage, it is also ensured that ‘None’ is not treated as null. After data cleaning is complete, three csv files are created which will then be used in stage 2.

In stage 2, our data enrichment process takes place. The cleaned datasets are loaded in from stage 1 and the datasets are refined. In this stage, data transformation takes place such as feature engineering and data distribution modelling. Key columns are adjusted as well like the insurancePremium columns are multiplied by 12 to convert it to an annual amount. Binary categorical columns such as Yes and No are converted to 1 and 0 respectively. For example, policeReportBool which has ‘YES’ and ‘NO’ outputs are converted to ‘1’ and ‘0’ respectively. In this stage, some columns are created based on the distribution and logic. For example, the vehicleAge column was created by subtracting the vehicleYear from 2015 which was the year the dataset was based on. Finally, this stage also filters out the columns that we deemed were unnecessary for our model. This script will create 3 csv files and hence will complete our enriching process.

In stage 3, we merge enriched datasets 1 and 2 to combine them for data synthesisation. For synthesisation, we do not use dataset 3 as it has a lot of columns missing compared to our base dataset which is dataset 1. This script essentially loads both dataset 1 and 2 and uses the concat function from pandas to combine both the datasets and creates a csv file containing 2575 rows of data. An index is also created to keep track of the specific rows.

In stage 4, our data synthesisation takes place. The synthesisation scripts can run parallel to each other. In our synthesisation process, we use datasets 1 and 2 to synthesise more rows. We use two methods to synthesise the data. The first method directly samples the data from the original dataset’s distribution and some noise is added to ensure variance. The second method uses two neural networks to synthesise the data. This model is called a CTGAN model where synthetic data is created by learning the underlying distribution of the datasets. The two methods creates 4000 rows of data each which adds up to 8000 rows of synthesised data. The two scripts will create two csv files also with an indexed column to keep track of the rows.

In our final stage, all of our datasets are combined together. Enriched dataset 1, 2, and 3 are combined with synthesised methods 1 and 2 to create 20877 rows of data which is our final dataset. Like stage 3, we also use the concat function from pandas to combine the five datasets. This script also handles any potential issues regarding indexing. It ensures that the index starts from 1 and does not reset. This script will create a csv file named 'merged\_20000\_rows.csv' and this is the final dataset that will be used to train the AI model.

# **3. How to run our code on Azure**

Our scripts are hosted on Azure Functions and can be triggered by making a POST request to a private endpoint secured with an authorisation key. The way you can run the scripts via Azure:

**Method ONE: Make a Pull Request to Activate GitHub Actions (in development)**

Another way to run the code on Azure is to make a pull request (PR) to the GitHub repository. This will automatically trigger a GitHub Actions workflow that will make the POST request to Azure Functions on your behalf. Here's how you can do it:

Make changes to the code (or leave a comment) and create a pull request in the repository.

Once the PR is submitted, GitHub Actions will be activated, and the necessary POST request will be made to trigger the Azure Functions pipeline.

This method allows for fully automated script execution through GitHub's continuous integration workflow.

# 

# **4. Troubleshooting/ Where to get help ?**

## 4.1. General Setup Issues

* **Erorr in Github cloning:** 
  + **Issue:** User has difficulties in cloning the Github repository.
  + **Solution:** Ensure the Github is installed on your machine. Utilise the command *‘git–version’* to verify the installation. Additionally, verify you have stable internet access and correct permissions for the repository,
* **Python or Module Not Found Error**:
  + **Issue**: User receives an error relating to the absence of Python packages such as *pandas* or *numpy*
  + **Solution**: Verify that correct Python is installed and can be accessed using the command line *python–version* to confirm. Follow-up steps include:
    - Instal missing packages with *pip install -r requirements.txt*
    - Handle permission issues with *pip install --user -r requirements.txt*

## 4.2. Local Code Execution Issues

* **Master Script *(run\_all.sh or run\_all.ps1)* Errors**:
  + **Issue**: User has issues as scripts are terminating early or not running correctly.
  + **Solution**:
    - Ensure you have the correct permissions to run shell or PowerShel scripts. Use *chmod +x run\_all.sh* on Unix.
    - Confirm that the required python modules are installed.
    - Examine for any typos in file paths and ensure that the paths match your operating system’s format. For example forward or backward slashes.
* **Missing Permissions for PowerShell Scripts on Windows**:
  + **Issue**: By default, Powershell scripts are disabled.
  + **Solution**: Enable script execution with the following command:
    - Powershell Code: *Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser*
* **Parallel Execution Errors**:
  + **Issue**: During parallel execution some datasets arent processed correctly.
  + Some datasets aren't processed correctly during parallel execution.
  + **Solution**: Ensure that all input files are properly formatted and located in the correct directories. If needed, execute each stage separately to pinpoint the source of the error.

## 

## 4.3. Azure Deployment or Execution Issues

* **Azure Function Not Triggering**:
  + **Issue**: user is unable to execute Azure Functions after a pull request or POST request.
  + **Solution**:
    - Inspect the logs on Azure Functions to check for any errors that occured when using the Azure portal
    - Verify that authorisation key is accurately incorporated into the post request
    - Ensure that the private end-point is correctly configured and the function app service is functioning properly.
* **GitHub Actions Not Triggering**:
  + **Issue**: User’s pull request does not initiate the GitHub Action workflow
  + **Solution**:
    - Verify that the workflow file can be found in ‘*.github/workflows/’* and follows correct syntax.
    - Esure that the repository settings that authorise actions and triggers are properly configured.

## 

## 4.4. Data-Related Issues

* **Data Not Found or Incorrect Paths**:
  + **Issue**: Scrips fail to locate the processed or raw datasets.
  + **Solution**: Make sure the data is located in the correct directory, and that the paths align with those specified in the scripts. Whenever possible, use relative paths to prevent OS-related issues.
* **Unexpected Values in Processed Data**:
  + **Issue**: The synthesised or enriched data contains unexpected values.
  + **Solution**: Review the data cleaning and enrichment logic. Verify that missing values are appropriately addressed and categorical columns are consistently converted.

## 4.5. Where to Get Help?

* **Documentation**:
  + Refer to the README file and documentation in the GitHub repository for comprehensive instructions and usage guidelines.
  + This documentation will ensure that you have some examples and troubleshooting steps for common issues.
* **GitHub Issues**:
  + If the problem continues, open an issue in the GitHub repository. Include detailed information, such as error messages and screenshots, if relevant.
* **Azure Logs**:
  + When issues relating to runtime mistakes and misconfigurations are found utilise the Azure Portal to access logs for Azure Functions.
* **Stack Overflow / Online Forums**:
  + When general programming issues or errors that are package-related arise, look on the Stack Overflow or seek information from the Python package documentation

# **5. Website**

## **5.1. How the website works (back end & front end)**

The website is built using Next.js, a React framework that allows for server-side rendering and API integration. The back end of the website consists of two main components:

* Serverless API endpoints that handle the fetching and updating of claim data from a CSV file.
* Front-end user interface implemented with React, which displays the form and handles user interactions.

Back End (Server-Side Logic)

* Data Storage: The data is stored in a CSV file named claims.csv, which includes fields for claim information such as claim status, claim outcome, description, and other details.
* APIs:
  + /api/getClaim: This API fetches the first open and pending claim from the CSV file and returns it to the front end for display. It uses the csv-parser library to read the CSV file.
  + /api/updateClaim: This API updates the claim status and outcome in the CSV file. It first reads the CSV file to locate the specified claim, modifies the fields, and writes the updated data back to the file using the json2csv library.

Front End (React Components)

* The front end is built with React components that include form elements for claim information, claim outcome, fraud evaluation, and related actions.
* State Management: The application uses React's useState for managing the state of form fields such as claim details, fraud risk score, and claim description.
* Button Functionality:
  + "Get Claim" Button: Fetches the first open and pending claim using the /api/getClaim endpoint.
  + "Close Case" Button: Updates the claim outcome to "approved" or "denied," and sets the claim status to "closed" via the /api/updateClaim endpoint.
  + "Escalate to Manager" Button: Updates the claim outcome to "escalated" while leaving the claim status as "open."
  + "Check Fraud" Button (Not Functional Yet): Intended to run a fraud detection check on the current claim using the /api/checkFraud endpoint, returning a fraud risk score and analysis summary.

## 

## **5.2. How to Use the Website**

1. Accessing the Website:
   * Navigate to the website URL or open the local development server (http://localhost:3000) when running locally.
2. Fetching a Claim:
   * Click on the "Get Claim" button to load the first open and pending claim from the database (CSV file). The claim details will automatically populate the form fields.



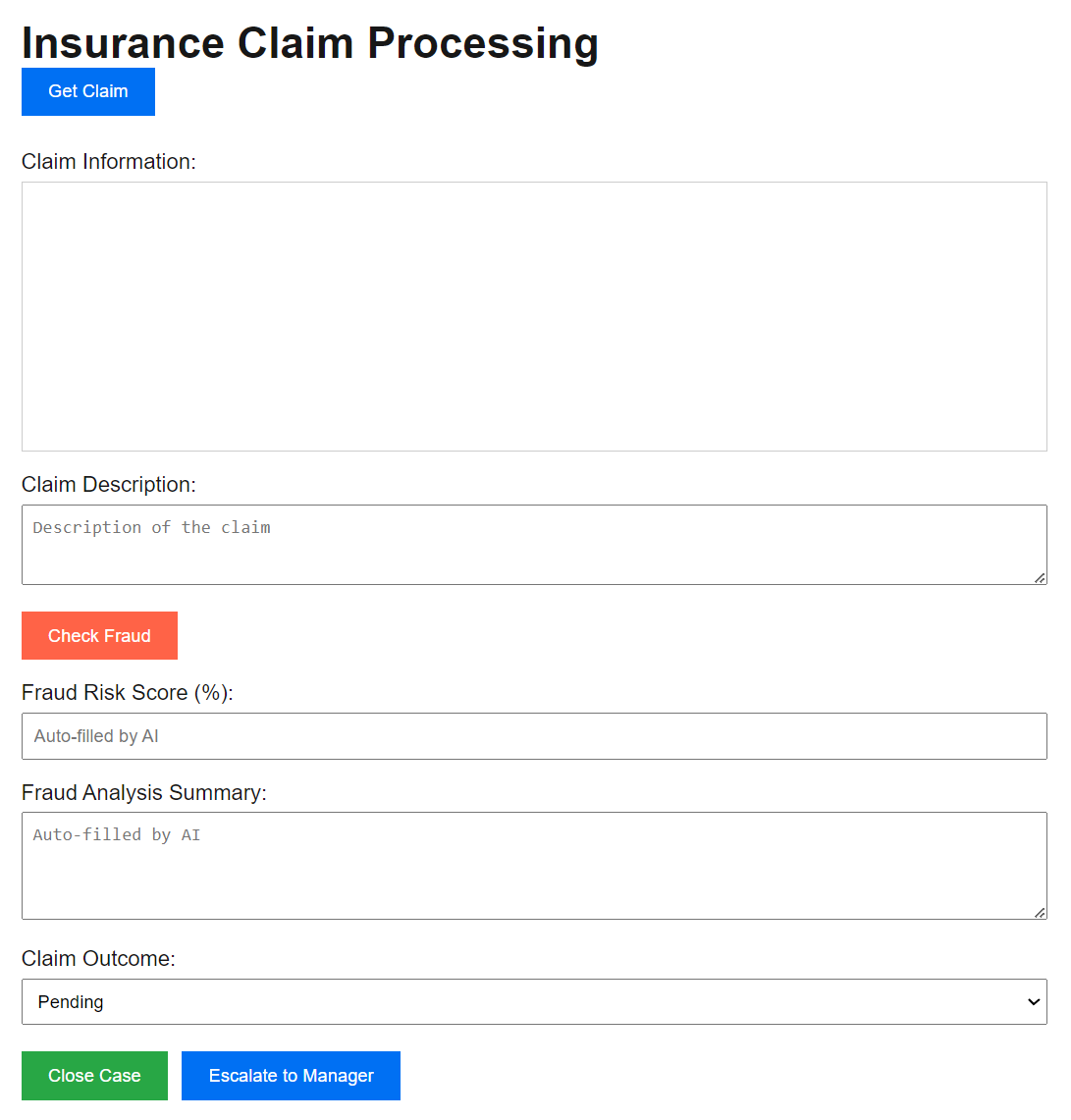
1. Viewing Claim Information:
   * The fetched claim information will be displayed in a formatted text area, including details such as Claim ID, customer information, and claim description.
   * The Claim Outcome dropdown will be populated based on the fetched claim.



1. Checking for Fraud (Not Functional Yet):
   * Click the "Check Fraud" button to run a fraud detection check on the current claim.
   * This will return a fraud risk score (%) indicating the likelihood of fraud, along with a brief analysis summary explaining the key factors influencing the score, such as unusual claim patterns or inconsistencies.

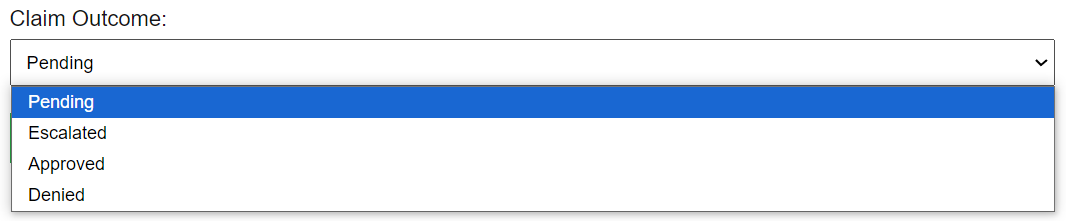


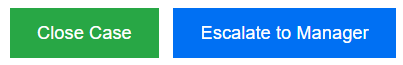
1. Updating the Claim:
   * Choose an option from the Claim Outcome dropdown (Pending, Approved, Denied, or Escalated).
   * Click on "Close Case" to mark the claim as resolved, or "Escalate to Manager" to flag the claim for further review.
2. Clearing the Form:
   * After updating the claim (closing or escalating), the form fields will be automatically cleared for the next operation.



## **5.3. Functionality**

* Claim Management: The website allows users to manage insurance claims by updating claim outcomes and statuses.
* Automated Field Population: The form auto-populates when a claim is fetched, reducing manual input and potential errors.
* Data Persistence: Changes to the claims are saved back to the CSV file, ensuring data consistency.
* Fraud Detection: A "Check Fraud" button is included to eventually allow automated fraud risk analysis. (This feature is currently under development).
* Flexible Claim Actions:
  + Closing the Case: Automatically sets the claim to "closed" if the outcome is "approved" or "denied."
  + Escalation: Marks the claim as "escalated" while keeping it open for further processing.





## **5.4. Troubleshooting (In Case the Website Doesn't Function)**

1. Website Not Loading:
   * Ensure the development server is running by executing npm run dev in the project directory.
   * Confirm that Node.js is properly installed and configured on your machine.



1. Buttons Not Responding:
   * Check the console for any JavaScript errors that might indicate a problem with the button's event handler.
   * Verify that the API endpoints (/api/getClaim and /api/updateClaim) are reachable by testing them directly in the browser or using a tool like Postman.
2. Claim Not Fetching or Updating:
   * Make sure the CSV file (claims.csv) is located in the correct directory and is not open in another program that may lock the file.
   * Ensure the CSV file has the correct format with all required columns (ClaimID, ClaimStatus, ClaimOutcome, etc.).
   * Restart the development server to clear any temporary issues related to file reading or writing.
3. Data Not Displaying Correctly in Form:
   * Check if the state variables (e.g., claimDetails, claimOutcome) are being correctly set by logging their values.
   * Confirm that the API responses contain the expected data format, and adjust the front-end code accordingly.

Claim.csv data description

| Column Name | Data Type | Description |
| --- | --- | --- |
| ClaimID | String/Integer | A unique identifier for each insurance claim. |
| timeAsCustomer | Integer | The number of years the claimant has been a customer. |
| driverAge | Integer | The age of the driver involved in the claim. |
| insuranceAccess | Integer | The monetary amount or level of access to the insurance coverage. |
| insurancePremium | Integer | The annual cost of the insurance policy. |
| driverGender | String | The gender of the driver, typically "M"/"F" or "0"/"1". |
| educationLevel | String | The highest level of education attained by the claimant (e.g., "High School", "Bachelor"). |
| accidentType | String | The type of accident or incident (e.g., "Collision", "Theft"). |
| incidentSeverity | String | The severity of the incident (e.g., "Minor", "Major", "Severe"). |
| authoritiesInvolved | String | Indicates if authorities were involved (e.g., "Yes" or "No"). |
| incidentTime | String | The time of the incident (e.g., "14:30" for 2:30 PM). |
| numVehiclesInvolved | Integer | The number of vehicles involved in the incident. |
| numBodilyInjuries | Integer | The number of bodily injuries reported. |
| policeReportBool | Boolean/String | Indicates if a police report was available (e.g., "Yes"/"No" or true/false). |
| totalClaimAmount | Integer | The total amount claimed in the insurance request. |
| vehicleAge | Integer | The age of the vehicle in years at the time of the incident. |
| driverExperience | Integer | The number of years the driver has had a valid license. |
| licenseType | String | The type of driver's license (e.g., "Full", "Provisional", "Learner"). |
| firstName | String | The first name of the person submitting the claim. |
| lastName | String | The last name of the person submitting the claim. |
| claimStatus | String | The current status of the claim ("open" or "closed"). |
| claimOutcome | String | The outcome of the claim ("pending", "approved", "denied", or "escalated"). |
| claimDescription | String | A description provided by the claimant, explaining the details of the incident. |

# **6. End-Users**

**User Manual Overview**

| **End User** | **Key Responsibilities** | **Primary Tools** | **Where to Get Help** |
| --- | --- | --- | --- |
| **Claims Agent** | * Analyse pending claims and decide on claim status * Ensure data accuracy and consistency | * Fraud detection website * Claim management page | * User manual (5.1.) * System administrators (5.4.) * Website Manual () |
| **Supervisor/Manager** | * Review escalated claims * Monitor trends in claims for fraud patterns * Ensure proper outcomes are decided for claims | * Fraud detection website * Escalated claims page | * User manual () * System Administrators () * Website Manual () |
| **Business Analyst** | * Interpret insights for Power BI and fraud reports * Using data advise NRMA on strategic decisons | * Power BI dashboard * Fraud detection website | * User manual () * Data engineering team () * System Administrators () |
| **System Administrators** | * Monitor the technical infrastructure * Oversee APIs and maintain the functionality of the system * Search for and issues and errors with the system | * node.js * API management tools * Log monitoring systems | * User manual () * Data engineering team () * Website Manual () |
| **Data Engineers** | * Monitor and improve the data pipeline * Troubleshoot error to ensure the integrity of the data | * GitHub * Azure | * User manual () * GitHub Documentation () * Troubleshooting () |
| **AI Engineers** | * Enhance and maintain the AI model * Update model * Monitor performance metrics of the model | * CI/CD tools * API endpoints | * User manual () * Data engineers () * Troubleshooting () |

## 

## **6.1. Claims Agent**

**Purpose**:

The claims agent central role is to analyse the received claims and determine whether they should be approve, deny or escalate for further review by a supervisor.

**How to Use the Website:**

1. **Accessing the System:**
   * Open the website by clicking on the provided URL or it can be opened locally using (http://localhost:3000).
2. **Fetching and Reviewing a Claim:**
   * Click "Get Claim" to retrieve the next pending claim from the database
   * Review details of the claim such as, claim ID, claim amount, type of accident, etc. match the accident description.
3. **Managing Claim Status:**
   * Click "Check Fraud" to prompt the AI to analyse the claim
   * Examine the given ‘Fraud Risk Score’ and ‘Fraud Analysis Summary’
   * Select the appropriate outcome from the ‘Claim Outcome’ dropdown which includes, Pending, Approved, Denied, and Escalated.
   * Click “Escalate to Manager” is the claim requires further review and notify the respective manager
   * Click "Close Case" to conclude the claim analysis and progress onto the next claim.
4. **Clearing the Form:**
   * After the claim has been completed, the form with reset to the default empty form and once "Get Claim" is clicked a new pending claim will be received.
5. **Data Entry Guidelines:**
   * Review updates to the data are consistent and accurate to prevent issues that arise with inconsistencies in data.

## **6.2. Supervisors/Managers**

**Purpose:**

The main role of the Supervisor is to manage escalated claims and ensure they are resolved appropriately.

**How to Use the System:**

1. **Accessing Escalated Claims:**
   * Log in to the system and retrieve escalated claims by clicking the “Get Claim” button.
2. **Reviewing and Updating Claims:**
   * Examine the claim details and decide on the appropriate resolution. Choose an option from the Claim Outcome dropdown, then click “Close Case” to finalise the review.
3. **Monitoring Claim Trends:**
   * Monitor claim statuses and identify any suspicious patterns that may need further investigation.

## **6.3. Business Analyst**

**Purpose:**

The main role of the Business Analyst is to interpret high-level insights from the fraud detection system to advise the company on strategic decisions with the support these insights.

**How to Access and Use the Data**

1. **Where to Find the Data:**
   * The website will have a page for reports on insights made by the fraud detection model thus, the navigation bar can be used to find the page called ‘Fraud Insights’
   * Clicking on this page will lead to a Power BI dashboard that can be used to find accessible and digestible insights.
2. **How to Use the Data for Decision-Making:**
   * Review the dashboard and reports for key indicators, trends, and summary metrics.
3. **Requesting Additional Data:**
   * If there is a need for more specific reports or information they can create new Power BI dashboards or raise a ticket with the Data Engineering Team to create a more precise dataset.

## **6.4. System Administrators**

**Purpose:**

After the project is handed over to NRMA the System Administrators will oversee the technical infrastructure for the fraud detection website, APIs, and data files.

**How to Maintain the System:**

1. **Setting Up the Website:**
   * Verify that Node.js is installed and the server is running by executing *npm run dev*
   * Ensure the CSV file (Claims.csv) is correctly located in the directory.
2. **Managing APIs:**
   * Test the/api/getClaim and /api/updateClaim endpoints to confirm smoot data transfer between the website and backed.
3. **Troubleshooting Issues:**
   * Monitor system logs for errors and take corrective action to address server downtime or connectivity problems.
   * Check permissions to guarantee that the website can read from and write to the CSV file.
4. **Data Backup and Security:**
   * Ensure that regular backups are scheduled of claims.csv and verify that it is compliant with data security policies.

## **6.5. Data Engineers**

**Purpose:**

Once we hand this project over to NRM the data engineers will have to ensure they maintain, troubleshoot, and enhance the data pipeline while overseeing the infrastructure.

**Pipeline Execution Guide:**

1. **Pre-requisites:**
   * Ensure you have access to the Github repository and Azure portal.
   * Install the necessary packages using:
   * *pip install -r requirements.txt*
2. **How to Run the Code Locally:**
   * Clone the repository:
   * *git clone* [*https://github.com/AlanDataPortfolio/ey-azure-fn-pipeline.git*](https://github.com/AlanDataPortfolio/ey-azure-fn-pipeline.git)
   * Execute the **master script** to run the entire pipeline:

*bash run\_all.sh # Unix*

*pwsh run\_all.ps1 # Windows*

1. **How to Monitor the Azure Functions:**
   * Log in to the **Azure portal**.
   * Go to the Function App and check the monitor tab for logs and errors.
   * Ensure that all the appropriate authorisation keys are set for triggering the function.
2. **Handling Failures:**
   * If any script encounters an error, run an individual stages manually to troubleshoot:
   * *python src/clean\_dataset1.py*
   * Check the Azure activity log and Github Actions logs for any comprehensive error messages.

## **6.6. AI Engineers**

**Purpose**: After the project is handed over to NRMA, the role of AI engineers is crucial in maintaining, troubleshooting and enhancing the machine learning models integrated into the fraud detection system, They ensure the models continue to operate effectively , adapt to changing data patterns and provide valuable insights to claim analysis.

**Model management Guide:**

1. **Pre-requisites:**
   * Ensure you have access to the Github repository and Azure portal.
   * Install the necessary packages using:
   * *pip install -r requirements.txt*
2. **How to Run the Code Locally:**
   * Clone the repository:
   * *git clone* [*https://github.com/AlanDataPortfolio/ey-azure-fn-pipeline.git*](https://github.com/AlanDataPortfolio/ey-azure-fn-pipeline.git)
   * Execute the **master script** to run the entire pipeline:

*bash run\_all.sh # Unix*

*pwsh run\_all.ps1 # Windows*

1. **Monitoring model performance**
   * Log into the Azure portal.
   * Navigate to the relevant Azure Machine Learning Service.
   * Check performance metrics and logs for anomalies or degradation in model accuracy
2. **Handling model failures:**
   * If a model fails to execute, run individual components to diagnose the issue:
   * *Python src/run\_model.py*
   * Review azure activity logs and Github action logs for detailed error messages.
3. **Model Retraining:**
   * Locate fresh datasets in the Azure Data Lake storage account
   * Ensure data is in expected format for retraining
   * Schedule retraining by setting up a schedule using Azure Data Factory or a cron job.
   * Execute the retraining script:
   * *python src/retrain\_model.py*
   * Validaate the updated model against a holdout dataset
   * Use the following command to deploy the updated model:
   * *az ml model deploy --name <model\_name> --model <model\_path>*

# **7. Frequently Asked Questions (FAQs)**

1. **How can we deploy the fraud detection system if new data sources are added in the future?**

In the documentation we have provided instructions for deployment which you can find above. To accommodate the addition of new data the ingestion pipeline can altered by updating scripts or functions in Azure Data Factory or Functions. Additionally, you can refer to the AI Team’s user manual for the AI model for retraining steps due to the addition of new datasets

1. **What do we do if the model starts showing lower accuracy over time?**

If you observe a decrease in performance, it could signal concept drift, which refers to shifts in fraud patterns. Refer to the training manual for guidelines on retraining the model with the latest data. Additionally, you can utilise the automatic retraining process to address this issue.

1. **How do we troubleshoot if the data pipeline fails during execution?**

Azure will generate comprehensive error logs. We suggest reviewing the logs in Azure functions or Data factory for specific error codes. For guidance on restarting the pipeline and trouble shooting common issues, please consult the user manual. If necessary, reach out to our team during the designated support period.

1. **Can the fraud detection system be integrated with our existing CRM or claim processing system?**

Absolutely, the fraud detection model can interface with other systems through API endpoints included in the deployment. Additionally, your team should coordinate with the CRM team to ensure proper alignment of the data flow.

1. **What steps should we follow if Azure services experience downtime?**

If the Azure services experience downtime then make sure the Data Engineering team is monitoring the Azure tools that provide alerts. Next, the user should follow the Azure recovery plan in the documentation provided. Furthermore, remember to set-up a back-up storage system to prevent the event of data loss if this instance occurs.

1. **How do we update libraries or dependencies for automation scripts?**

All provided scripts come with a requirements.txt file listing the essential libraries. We recommend using a virtual environment or containerization (Such as Docker) to handle dependencies effectively.

1. **How frequently should the AI model be retrained?**

We would suggest retraining the model every 3 to 6 months, based on the influx of new claims and evolving fraud trends. Additionally, regular performance assessments such as accuracy and recall should be performed to evaluate whether retraining is needed sooner.

1. **What support will be available post-handover?**

After the handover we will be providing minimal support for any issues and troubleshooting as we expect that the team over at NRMA or a contracted third party will provide support.

1. **How do we ensure data security and privacy when using this system?**

All data utilised in this system adheres to NRMA’s security protocols and Azures data protection policies. Role-based access control (RBAC) has been implemented to manage access. It is advisable to regularly review roles and monitor data access logs to ensure compliance.

1. **What tools are available to interpret and visualise the model’s outputs?**

For the handover period no visualisation model will have been built due to constrictions with the deadline. Thus, will the agreement with NRMA it has been decided that they will continue the development of the fraud detection system to include the existence of visualisation

1. **What if fraudulent claims are missed by the model?**

Although the model is engineered to reduce false negatives, no system can achieve perfection. We suggest conducting regular manual reviews of both flagged and non flagged claims to uncover any gaps. Enhancements to the model can be achieved by retraining ot with additional fraudulent data as more cases are discovered.

1. **How is data security ensured on this website?**

To ensure the security of the website as it contain private information that should only be view by authorised individuals we have include a authentication system and access controls that prevent unauthorised individuals from accessing sensitive information