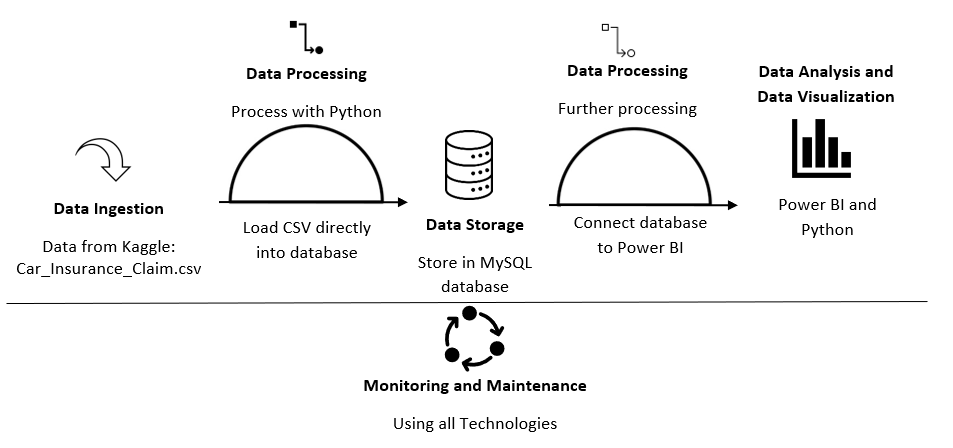
CAPSTONE PROJECT: CAR INSURANCE LOAN DATA

The dataset we chose to explore was car insurance data. The dataset identifies whether a customer claimed for a loan to pay for a car insurance claim.

### Data Pipeline

We created a data pipeline, through a set of data processing steps, to take data from its raw form, and put it through various stages of processing, to reach the end goal of generating analyses and insights.



**Data Ingestion:**

At the beginning of our data pipeline, the raw car insurance data in the form of a flat file (CSV file), was downloaded from kaggle.

**Data Processing:**

Once the data was **ingested**, data cleaning and data wrangling in Python was carried out in a JupyterLab Notebook to ensure data quality and relevance for analysis. This step involved filtering out irrelevant information, handling missing or erroneous data, and converting the data into a standardized format.

The data cleaning and wrangling was important in ensuring the accuracy, quality, and reliability of data before it had been imported into the database and then used in visualisations for analysis.

**Data Storage:**

It was then necessary to hold the data in a normalized structure to minimize the chance of data inconsistencies or duplications occurring when data is added or updated.

Our data already complied with Second Normal Form (2NF) and First Normal Form (1NF) but in order, so it complied with 3NF (Third Normal Form), we separated the data specific to vehicle into its own table along, while we gave the customer driving history and driver information its own tables. The fourth table was the original customer data table containing details on the customer.

We normalized the data into 3NF where we divided one large table into four smaller tables, linked to each other using relationships. I imported the processed data could into the chosen data repository, the MySQL database.

**Data Analysis and Data Visualization:**

Once I had stored the data in MySQL, we could then analyse the data using Power BI and furthermore in Python with Machine Learning. We then presented the results of data analysis in the form of visualizations and insights in Power BI and Python, making it easier to understand and interpret the findings.

**Monitoring and Maintenance:**

The data pipeline may require some monitoring and maintenance to ensure that it was operating correctly, by addressing data errors and quality issues.

During the data processing of the car insurance data, one of the data types was modified which ended up altering the data values of that field where they all showed as “True” instead of a mix of True and False. I noticed this when the data was imported into Power BI and so maintenance was required to alter the Python code and re-import the data in MySQL.

This issue emphasized the importance of data processing and particularly data cleaning and data wrangling was carried out fully and accurately with checks in place, as the time taken to go back a couple of steps to correct in Python where re-processing was required and re-importing data could cost time which could prove significant, especially in a real-world environment.

### Discuss Your Contributions

At the start Abey began data cleaning and data wrangling in Python and I was able to assist with any code where Abey and I was also involved in some of the decision making regarding changing datatypes, renaming fields and formatting values. Abey and I worked together on the database normalization while I authored the code in Python to output the four tables into four csv data files. I took ownership of creating the database schema and tables and then imported the data into the MySQL database.

I took ownership of the Data Analysis and Data Visualization which involved using Power BI, while Abey worked on visualisations using Python and Machine Learning. Using a combination of insights from both Power BI and Machine Learning was carried out to produce different insights.

### Personal Reflection

Regarding Project Management of the project, Abey and I met regularly to share our findings and we were able ask each other questions on each other’s Insights and results to check for accuracy and so that we both understood our own data and had an awareness of each other’s data.

One challenge I had when creating different visualisations, which looked at those who have submitted a claim for a loan or not, was not to just look at the data percentage for a particular datapoint but also to determine that datapoint along with the count of people for the same data point.

For example, if 100% of people claimed for a loan on their car insurance from the 16-25 age range for a particular state, it was important to understand how many people that was to keep the data analysis in proportion to the sample size, to gain a better understanding how representative it was, to ensure the validity and reliability of the results that were communicated.

On reflection at the setup of the database tables, we could have modified the database normalization where the relationship between the customer and vehicle was Many-to-Many (M:N), as many customers can insure one or more vehicles and many vehicles can be insure by one or more customers. However, the current dataset was based on one customer having one vehicle so this relationship so the decision to opt for 1:N between vehicle and customer sufficed.

Furthermore, on the data storage stage of the data pipeline, if I had more time, I would have opted for a NoSQL database such as MongoDB as opposed to the MySQL relational database. For the purposes of this dataset, MySQL was appropriate as it was structured data that required a small storage capacity. However, within this programme I have learnt that having a pre-defined schema can limit database flexibility. If the car insurance company changed the structure of how they collected data based on their customers, the NoSQL database is flexible so it can evolve as the data grows and evolves.

For example, Allstate Insurance allow customers to submit images and videos relating to a car insurance claim and a NoSQL database would be capable of images and video relating to that customer, which a SQL database would not be able to do.