Team project Handover Document

Data and AI TEAM

Trimester 1, 2023

https://github.com/redbackoperations/data-analysis

Projects

- User Ranking - Engagement
- FIT File Handling and Data Pipeline
- Corporate Reporting
- Sentiment analysis (language processing) and Community standards User/Community comments
- Performance Ranking (User)
- Workout Categorisation
- Data Warehouse
- Google Analytics/Hotter Analytics/MixPanel/App Analytics (Marketing and UX)
- Posture Analysis

Project Name: User Ranking - Engagement

Company: Redback Operations
Team Members: Saeed Alnaqeeb (Lead)

1. Project Overview

In general, this is a ranking system project that make use of the data produced from rides to evaluate riders' engagement and based on that, creates a competitive fun environment by ranking them. Each rider can either earn or lose points, depending on how they engage within the application and how regularly they ride. In this project, we aim to analyse the produced data, use that to build the system, and analyse users progress by building machine learning models. The deliverables of this project consist of a brief analysis extracting data that relates to users' engagement and a ranking system based on points.

2. User Manual

This project is currently being developed using Python and can be navigated by viewing the created files under project 15 of the Data/AI team repository on GitHub. There will be two files demonstrating the approach followed while working on this project, a python file containing analysis of the sample dataset, and another python file where the system's algorithm is developed. Also, a folder that holds the documentation of the project.

Available at:
https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2015%20User%20Ranking%20Engagement

3. Completed Deliverables

- Data analysis of users' activities and finding produced data do be used for evaluating a user's score.
- Ranking Criteria definitions.
- Algorithm that assigns points to users based on their engagement.
- Produced dataset of the users with their score and rank
- Post-ranking analysis to find patterns and trends for future ML implementation

All available at:
https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2015%20User%20Ranking%20Engagement

4. Roadmap

Roadmap for Trimester 2, 2023:

- More analysis on the produced dataset
- Search machine learning models to implement
- Build different ML models for user progress prediction and projected path

5. Open Issues

- The system needs more testing, as it was only tested on a sample dataset.
- Not yet implemented on the original dataset.
- Ranking criteria can be modified in the next trimester.

6. Lessons Learned

One of the main lessons learned is that the continuous use of Trello board would absolutely help organize both individual and team's work. Recommended technology for future machine learning models is skickit-learn library in Python.

7. Product Development Life Cycle

7.1. New Tasks

We do have two weekly meetings where we discuss the ongoing work. One of the meetings is at the start of the week which is a stand up meeting to present your plan for the week, and another meeting later for the progress and any encountered issues.

7.2. Definition of Done

Work is considered done if it is reviewed with the lead and committed successfully to GitHub.

7.3. Task Review

Tasks are generally reviewed by the team lead.

7.4. Testing

In this project, all the work and testing were on a sample dataset, so I runned different methods to test if the product works as it intended to work or not.

7.5. Branching Strategy

In GitHub, every member commits their work to a forked repository of the team's repo. With each update, we create a pull request so that it gets reviewed and branched to the main repo.

8. Product Architecture

8.1. Tech Stack

- Python: analysis, visualisation and algorithms
- SQL: Querying data
- GitHub
- Trello Board

9. Source Code

All available at:
https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2015%20User%20Ranking%20Engagement

10. Appendices

Documentation of the project and approach along with some important files are available to view on the project GitHub repository:
https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2015%20User%20Ranking%20Engagement

Showcase Video:
https://video.deakin.edu.au/media/1_blu17zp1

Sentiment Analysis

Redback Operations

11. Project Overview

SunCycle users have the option to leave comments on each other's activities as means of increasing engagement. To ensure that the comments are appropriate and to keep track of language usage, this project uses machine learning tools to analyse comment data and classify user comments.

The goal of the project is to build and maintain a safe, friendly environment by creating community guidelines, performing sentiment analysis, and monitoring users' comments. To help the company compete with other game developers in the market, it is very important to have a good environment and improve user experience from every aspects. More specifically, by preventing users from leaving toxic comments, this project helps protect users, foster inclusivity and diversity, and gather user feedback.

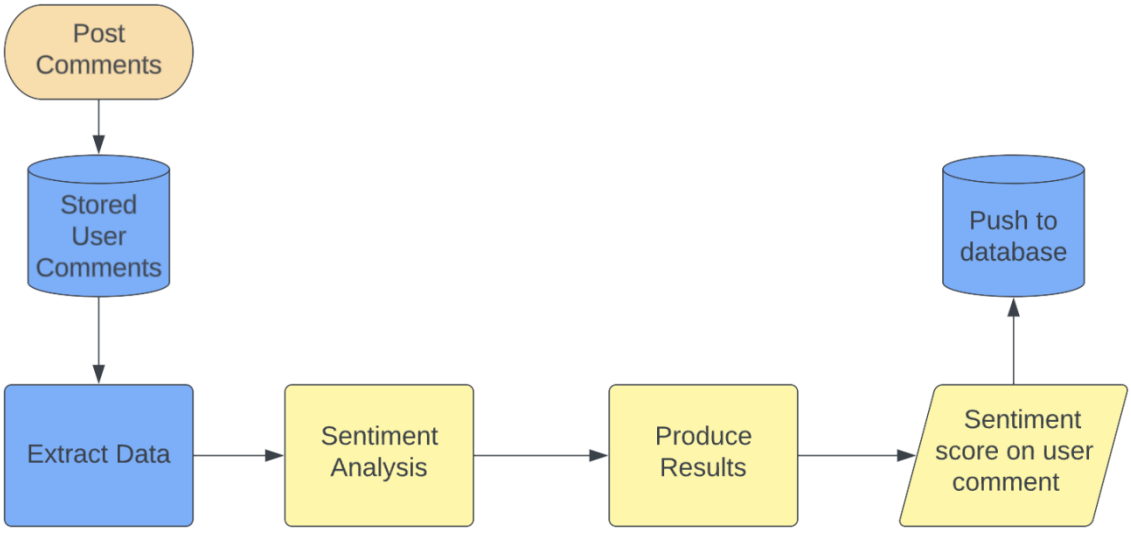
Deliverables include:

- Community guidelines.
- Data sets for training purposes.
- Machine learning models that can classify user comments.
- Showcase video

12. User Manual

This project is in the development phase. Currently the RNN model can classify a dataset with 1 label and 3 category values and achieve an accuracy of approximately 90%, and the DistilBert model can classify a dataset with 6 labels and achieve an accuracy of approximately 70%. Depending on the future requirement of Redback Operations, more accurate models might need to be constructed.

This project is fairly easy to use. When the model is finished, it will be added to SunCycle website as a new feature and will be used to process posted comments everyday after extracting them from the platform's database. The model is expected to analyze the data and produce a dataset with users' comments and their corresponding sentiment analysis result (positive/negative). Analysts can then use the dataset to monitor user comments.



13. Completed Deliverables

Deliverables:

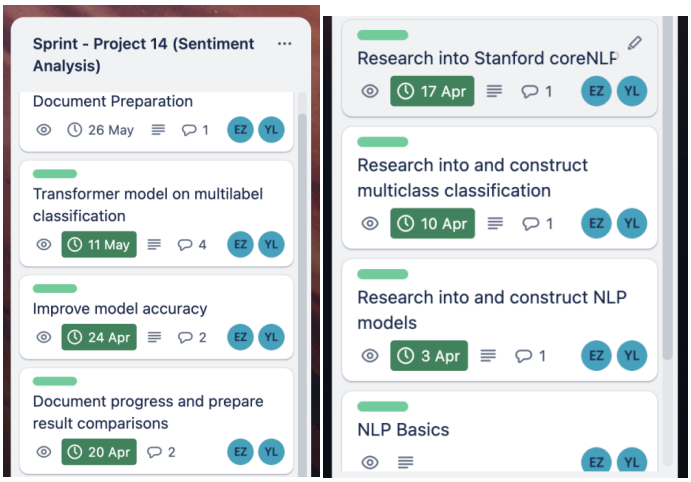
- RNN model:
  - Completed by Yvette Liang
  - A deep learning model that can classify positive, neutral, negative comments and achieve an accuracy of 90%.
  - Github link: https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2014%20Sentiment%20analysis/NLP_06_RNN-copy1.ipynb
  - Finished on May 11 (Trello)

- DistilBert model:
  - Completed by Yvette Liang
  - A deep learning model that can classify a toxic comment into one of six labels and achieve an accuracy of 70%.
  - Github link: https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2014%20Sentiment%20analysis/NLP_08_DistilBert-copy1.ipynb
  - Finished on May 11 (Trello)

- Community guidelines:
  - Competed by Yvette Liang
  - Simple community guidelines that can be posted on SunCycle's platform for users to follow.
  - Attached in the appendix.

14. Roadmap

1. NLP basics: Learned the basics of natrual language processing and constrcuted the first **Logistic Regression** model on a dataset downloaded from Kaggle.
2. Researched into and construct NLP models: Finished researching; Constructed two models (**Multinomial Naive Bayes** and **SVM**).
3. Researched into and construct multiclass classification: The **Random Forest** model is complete.
4. Researched into and constructed **Stanford CoreNLP.**
5. Documented process, compared results, tried to improve accuracy: **RNN**
6. Transformer model on multilabel classfication: **Bert** and **DistilBert**.



15. Open Issues

- Software compatibility issues: All the models are developed on Google Colab. Team members need to be careful when they use the code locally or run in a different system.
- Trello board: This board lists general topics that the project focused on in a certain period. If members want to check more specific details, they need to click on the cards and read the comments.
- The modeling process: The models are moving toward the field of deep learning. Members might need the corresponding skillset to develop more accurate models.
- Project requirement changes: Both the data (languages other than English) and the company requirement (classification threshold) may change in the future, members need to work on data preprocessing when they have new data come in.

16. Lessons Learned

1. Machine learning concepts: When I worked on NLP models, one of the biggest problems is the large dimension of data. It is very challenging to select the right corpus. It is also required the team member to master the machine learning concepts. For example, I should have tried other dimensionality reduction techniques other than PCA, especially when I realized that method required linear relationship of data.
2. Create fast models: Users' comments can sometimes be very lengthy, and the dataset can get very large, so it is very important to keep track of a model when it processes data. If members want to use a loop, make sure to print something at the end so that they know the amount of data left to be processed. Google Colab sometimes can stop automatically

if the user does not edit the webpage, so it is important to always keep an eye on the code.

3. Language issues: Members should familiarize themselves with people's language usage online to better preprocess data and tokenize them. For example, as a student whose English is not the first language, I could only create simple functions to preprocess user comments. I focused more on different machine learning models, and I realized I might have done a much better job if only I knew how people communicate online.

17. Product Development Life Cycle

The product development life cycle for this project is the agile process flow: moving through concept, inception, iteration and construction, release, production, and retirement.

Currently, as the model is still in the phase of iteration and construction, the introduction here is focused on the explanation of my workflow.

As there is limited computational capability in a personal laptop, I started from simple models such as Logistic Regression. As I learned more about machine learning, along the track of our unit progress, I could try more complex models and more feature engineering techniques such as Multinomial Naive Bayes, SVM, and Random Forest.

I created 4 models but then I realized their limit on improving accuracy. Therefore I started to try multi-label algorithms. I read about other people's articles online and tried Stanford CoreNLP model. I also tried deep learning models such as RNN, which was able to classify positive, neutral, negative comments and achieve an accuracy of 90%. I then moved toward multilabel classification. Bert was slow, so I constructed DistilBert, which could classify a toxic comment into one of six labels and achieve an accuracy of 70%.

Overall, the workflow during the construction phase is moving from simple to complex models, directed by the need of increasing accuracy.

17.1. New Tasks

In this project, tasks are created based on three rules: member's current skillset, model accuracy, and onTrack tasks (documents and presentations).

We have team meetings every week and each member would discuss about the current progress, ask questions, and offer help. For this project, tasks are goal-oriented. A team member needs to have adequate knowledge and skills to work on NLP models, so it is necessary to research and improve skill. The models should be effective, so accuracy is an important measurement to create tasks when necessary. OnTrack tasks are required in this unit, so they are also on the task list.

17.2. Definition of Done

- A classification model is done when it produces an accuracy score that is high enough. In this case, RNN and DistilBert can be called completed.
- A code file is done when it can be run all the way down, with comments, and with accuracy scores at the end. It is finished when other team members can understand it easily.
- A report is done when it is submitted and accepted by the instructor.

17.3. Task Review

As there is only one person working on this project this trimester, this person alone looks for dataset, wrties code, and reviews it before uploading to Github. Team leader of the data science team would review it before merging. The code uploaded have all been checked and made sure that there is no confusion. Comments are added so that future team members understand the documents.

17.4. Testing

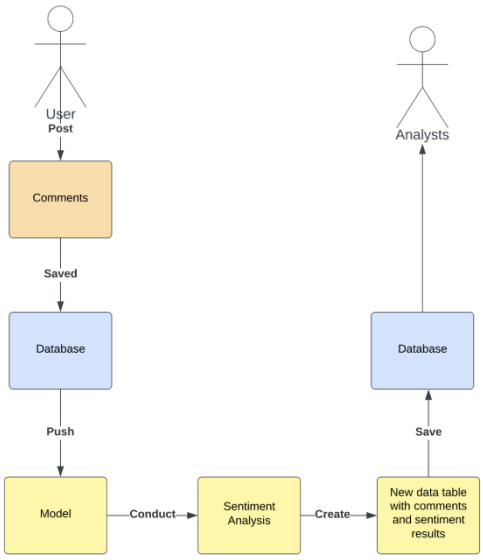
During the iteration and construction period, testing is done after training the model. In the future, the testing can be done by letting the model produce a data table, with users' original texts on the left-hand side and sentiment scores / label on the right-hand side. This way, our team members can take a sample and check how accurate the score / label is in terms of categorizing the corresponding texts. If the score is not correct, then they can improve and update the model.

17.5. Branching Strategy

Currently, there is only member pushing files to the origin. In the future, team members should use team chat to share their work and revise a final copy, and designate a person to submit the final copy and push it to the origin.

18. Product Architecture

18.1. UML Diagram



18.2. Tech Stack

Google Colab: It has Jupyter notebook and relevant packages. It is flexible and allows multiple person to edit.

19. Source Code

Github code:
https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2014%20Sentiment%20analysis

Dataset:
https://www.kaggle.com/datasets/kazanova/sentiment140
https://www.kaggle.com/datasets/charunisa/chatgpt-sentiment-analysis?select=file.csv
https://www.kaggle.com/datasets/julian3833/jigsaw-toxic-comment-classification-challenge?select=test.csv

Other helpful information:
https://scikit-learn.org/stable/modules/multiclass.html
https://colab.research.google.com/github/stanfordnlp/stanza/blob/master/demo/Stanza_CoreNLP_Interface.ipynb#scrollTo=xiFwYAgW4Mss
https://github.com/practical-nlp/practical-nlp-code/pull/38
https://huggingface.co/bert-base-uncased
https://huggingface.co/distilbert-base-uncased

20. Appendices

Showcase video link:
https://youtu.be/BOQfavLfvAQ

Community guidelines:
Maintaining a healthy and friendly online environment is essential for creating a positive and safe space for everyone. Here are some simple rules that the team can refer to when they draft community guidelines:

1. No hate speech: Any comments that are racist, sexist, homophobic, or discriminatory are not allowed.

2. No personal attacks: Personal attacks and insults towards other individuals are not allowed.

3. No bullying or harassment: Any form of bullying or harassment is not allowed, regardless it's directed towards another user or not.

4. No trolling: Comments that are posted with the intent of provoking an emotional or negative response from other users are not allowed.

5. Stay on topic: Comments should be relevant to the topic being discussed.

6. No spamming: Posting the same comment repeatedly or promoting products or services excessively is not allowed.

21. Posture Analysis

Mark Telley
Samuel Kamau

22. Project Overview

This project centers on a comprehensive, real-time posture analysis system for cyclists, designed to significantly reduce injuries and energy wastage resulting from incorrect form during cycling. Utilizing the power of Python with the OpenCV, PyQt, and MediaPipe libraries, the application encompasses three primary modules: pre-workout analysis, cycling analysis, and post-workout analysis.

The project aims to foster healthier cycling habits, enable more efficient training at home, streamline coaches' tasks in training clients, reduce posture-related injuries, and improve a cyclist's performance over time. It distinctively targets cyclists' needs, which sets it apart from more generalized posture analysis applications available in the market.

The key deliverables of the project are:

1. **Real-time pose estimation and cycling posture analysis module:** It provides real-time feedback on the cyclist's posture and form, allowing instant adjustments. It also includes a pedaling technique analysis and an aerodynamic analysis, which can be leveraged to optimize the cyclist's efficiency and performance.
2. **Pre-Workout and Post-Workout Demonstration videos with real-time user tracking:** This interactive feature empowers the cyclist to follow along with the demonstration videos while the system tracks and analyzes their form in real-time. This further facilitates more flexible training at home and aids in preparing for and winding down from cycling sessions.
3. **Data and video recording of sessions:** Each cycling session is recorded and stored both as video data and as quantitative data. This allows for tracking of progress over time, and helps to highlight areas of improvement.
4. **Data visualization tool:** A tool that presents the collected data from each session in an easily digestible format, such as graphs. This empowers cyclists and coaches to analyze the data and understand trends in performance and posture improvements.

23. User Manual

Set-Up and Activation

Refer to the ReadMe file for a guide

UI

The user interface has been designed for ease of use, providing only essential navigation options to streamline the user experience. Just run the User Interface.py file after the initial set-up and you’re good to go

24. Completed Deliverables

The product's successful deliverables at this stage include the following components:

- An analyzer for pre-workout and post-workout activities
- The feature to log both data and video during workout sessions
- A function to erase recorded data when necessary
- A tool for visualizing user data

All aforementioned deliverables can be accessed at the product's GitHub repository: [GitHub Repository](https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2020%20Posture%20Analysis)

25. Roadmap

The projected pathway for the product encompasses the following enhancements:

- Integration of AR features for immersive interaction
- Utilization of edge detection for precise spine and aerodynamics analysis
- Automation to switch from demonstration mode to live feed upon detection of a human and bike
- Machine learning algorithms to detect user performance trends and offer suitable advice
- NLP implementation for real-time audio feedback from users
- Facial tracking technology to monitor user energy levels and emotional states

26. Open Issues

One persistent issue hampers seamless project execution, specifically the Github Desktop's inability to access the project repository due to file naming compatibility issues. An interim solution has been implemented, utilizing Co-Pilot and a separate repository to track progress. Once significant modifications are complete and rigorously tested, the changes are then transferred to the main repository.

27. Lessons Learned

The project has underscored the importance of various aspects like efficient project management, time management, detailed task division, incorporation of feedback loops for iterative product enhancement, and meticulous project planning. It is advised for the future teams to adhere strictly to the project planning to eliminate unnecessary time expenditure. Nonetheless, retaining a degree of flexibility is beneficial, given the unpredictability of certain issues.

28. Product Development Life Cycle

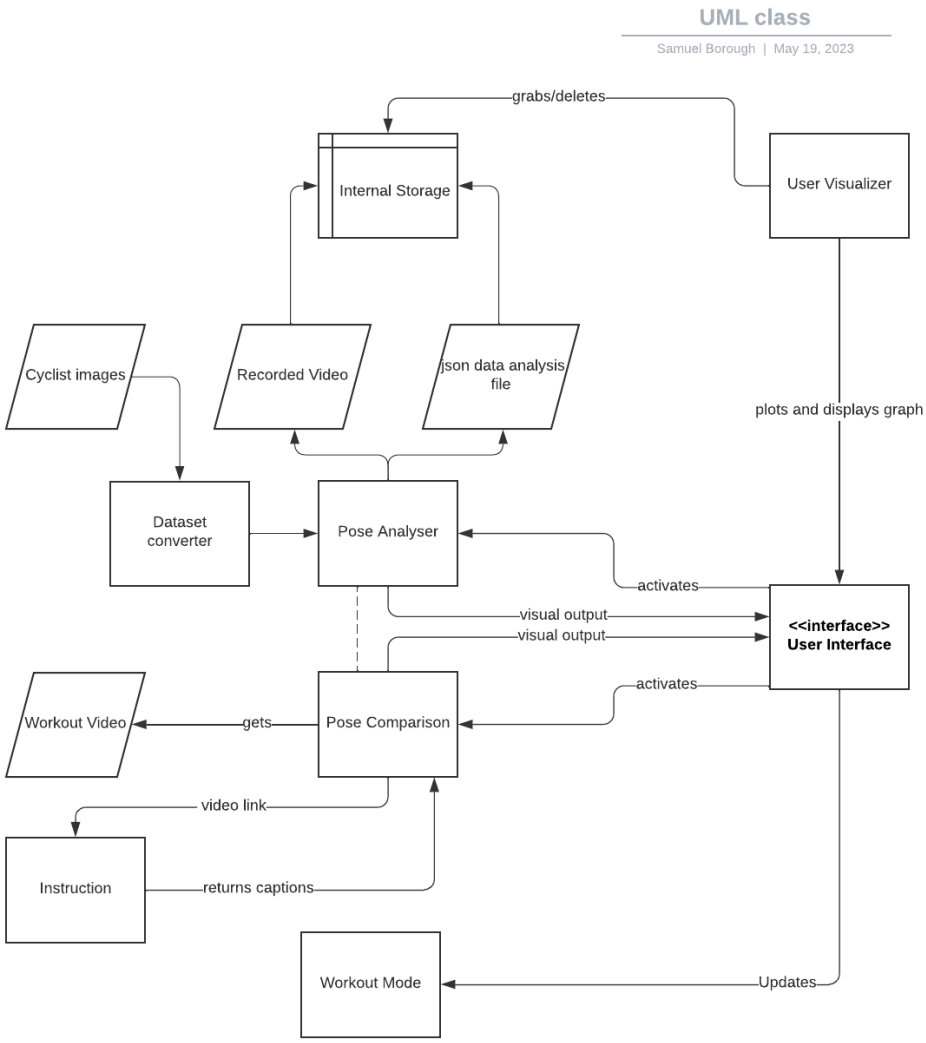
The creation of new tasks is managed via Trello cards and the associated checklists. A task is deemed complete when it is free from both syntax and logical errors, and the outcome aligns with the pseudocode initially outlined. The solo nature of the project means code reviews are conducted by the developer, focusing on runtime error checks and functional tests. For testing, live, recorded, or downloaded video footage is utilized.

29. Product Architecture

The project architecture comprises five primary classes:

- Pose Analyser: This class is responsible for the real-time analysis and logging of cycling data.
- PoseComparison: This class compares stretching workout videos with user live feed, incorporating methods from Pose Analyser.
- UserVisualizer: It handles the visualization of user data and facilitates data deletion when required.
- Instructions: This class overlays instructional captions onto stretching videos.
- WorkoutMode: This is used by the UI to interchange between the two analysers.

Additionally, a script named CDataset is present that converts an image folder into a standardized dataset of images, serving as a training set for the machine learning model.



The entire architecture utilizes Python with significant reliance on libraries including OpenCV, PyQt, and MediaPipe.

30. Appendices

30.1. Software Dependencies

The following libraries and software are required to run and develop the application. Each of these plays a crucial role in the functioning of different aspects of the project.

TensorFlow: TensorFlow is an end-to-end open-source platform for machine learning. It is utilized in this project for processing intensive tasks and machine learning models.

OpenCV: OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. In this project, it's used for image processing tasks.

MediaPipe: MediaPipe is a cross-platform framework for building multimodal applied machine learning pipelines. It's utilized in this project for real-time pose estimation.

PyQt5: PyQt5 is a set of Python bindings for The Qt Company's Qt application framework and runs on all platforms supported by Qt. It is used for creating the graphical user interface for this application.

PyDub: PyDub is a simple and easy-to-use Python library for audio manipulation. It is used for handling audio-related tasks in the project.

Seaborn: Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. It is used for data visualization tasks in this project.

Tkinter: Tkinter is Python's de-facto standard GUI (Graphical User Interface) package. It is used in this project for creating the user interface.

Pandas: Pandas is a software library written for data manipulation and analysis in Python. It is used in this project for data handling tasks.

Matplotlib: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter. It is used in this project for data plotting tasks.

31. Project 10 - Google Analytics/ Hotter Analytics/ MixPanel/ App Analytics (Marketing and UX)

1. Project Overview

We are using existing data from Google Analytics to create reports to study user's behaviour and get meaningful insights about their out-of-game engagement in order to create a feedback loop for product owners which will enable them to view and address product statistics, bounce rates and various other issues.

Aims for Trimester:

- To formalise report on key data insights
- Attempt to harmonise various data sources
- To capture the behavioural & consumption pattern
- To collect details of other products of same specification

Deliverables:

- To formalise report on key data insights
- Attempt to harmonise various data sources
- To capture the behavioural & consumption pattern
- To collect details of other products of same specification

Completed Deliverables

Used GA(Google Analytics) for a travel website to find the constitution of the in-flow of users, detailed statistics (acquisition, behaviour and conversions, etc.) of the site for various durations and their comparisons, demographic (Age, gender, etc.) of the users as well as the visualization of all the data from all the popular browsers.

https://github.com/ktripathi04/data-analysis/tree/main/Trimester_1_2023/Project%2019%20App%20Analytics

2. Roadmap

1. Define Objectives and Requirements:

- Clearly articulate the objectives and requirements of the custom tool.
- Identify the specific data needs and gaps in the existing data collection methods.
- Determine the key metrics and insights required to address the task effectively.

2. Research and Data Exploration:
    - Conduct research on available data sources and technologies that can provide the desired data.
    - Explore different APIs, databases, or data providers that offer relevant data.
    - Evaluate the feasibility, reliability, and cost implications of accessing and integrating the identified data sources.
3. Design Data Collection Strategy:
    - Develop a comprehensive data collection strategy that aligns with the project's objectives.
    - Determine the appropriate data collection methods, such as web scraping, API integration, data feeds, or data partnerships.
    - Consider data privacy and compliance requirements while designing the data collection strategy.
4. Build Data Collection Pipeline:
    - Implement the necessary infrastructure and tools to support data collection and storage.
    - Develop scripts or code modules to automate data retrieval from various sources.
    - Ensure the scalability, reliability, and security of the data collection pipeline.
5. Data Processing and Integration:
    - Clean, preprocess, and normalize the collected data to ensure consistency and accuracy.
    - Develop data integration processes to merge the new data with existing datasets, if applicable.
    - Implement data quality checks and validation mechanisms to identify and address any anomalies or errors.
6. Data Analysis and Visualization:

- Apply appropriate analytical techniques and algorithms to extract insights from the collected data.
- Develop data visualization components or dashboards to present the insights in a clear and meaningful manner.
- Enable interactive exploration and filtering of the data to facilitate deeper analysis.

7. Testing and Validation:
    - Conduct rigorous testing to ensure the accuracy, reliability, and performance of the custom tool.
    - Validate the results against known benchmarks or ground truth data, if available.
    - Seek feedback from relevant stakeholders to identify any areas of improvement or fine-tuning.
8. Deployment and Maintenance:
    - Deploy the custom tool in a production environment or integrate it into the existing workflow.
    - Establish proper monitoring and error handling mechanisms to ensure smooth operation.
    - Plan for regular maintenance and updates to accommodate evolving data sources or changing requirements.
9. User Training and Support:
    - Provide comprehensive training and documentation to users on how to effectively utilize the custom tool.
    - Offer ongoing support and address any user inquiries or issues promptly.
    - Encourage user feedback and suggestions for further enhancements or feature additions.
10. Continuous Improvement:

- Regularly evaluate the effectiveness and relevance of the data collected and insights generated.
- Incorporate user feedback and iterate on the custom tool to enhance its performance and usability.
- Stay updated with new data sources, technologies, and analytical methods to continuously improve the tool's capabilities.

3. Open Issues

- Data was preset as per the designers view point and therefore the project had to be tailored according to the designers views.

- Limitation on acquiring the data as it wasn't enough to come to a conclusion.

- Lack of a team since the project is relatively new and I'm the only one working on it.

7. Lessons Learned

Proper Planning and Scope Definition

- It's important to clearly define the project scope, objectives, and deliverables at the beginning. Without a clear plan, teams may face scope creep, lack of focus, or unrealistic expectations.

- Recommendation: Invest time in detailed project planning, including defining goals, identifying key metrics to track, and setting realistic timelines. Regularly reassess and refine the project scope to ensure alignment with the team's capabilities and available resources.

Effective Communication and Collaboration

- Communication breakdowns can hinder progress and lead to misunderstandings. Ineffective communication among team members, stakeholders, or during panel presentations can impact project outcomes.

- Recommendation: Establish regular communication channels, such as team meetings or project management tools, to ensure everyone is aligned, progress is shared, and challenges are addressed promptly. Practice clear and concise communication during panel presentations to effectively convey progress and insights.

Data Quality and Preprocessing

- Inadequate data quality or improper preprocessing can lead to unreliable insights and inaccurate conclusions. Failure to clean and validate data can introduce biases or skew results.

- Recommendation: Prioritise data quality assurance and establish robust preprocessing procedures. Implement data cleaning techniques, handle missing values, and ensure data integrity. Document data preprocessing steps to maintain transparency and reproducibility.

Training and Knowledge Enhancement

- Lack of familiarity with Google Analytics or inadequate training can limit the team's ability to leverage its full potential. Insufficient understanding of the tool's features and capabilities may result in underutilisation or misinterpretation of data.

- Recommendation: Invest in comprehensive training and knowledge enhancement on Google Analytics. Ensure team members have a solid understanding of the tool's functionalities, data interpretation, and

analysis techniques. Encourage continuous learning through online resources, tutorials, and hands-on practice.

Continuous Monitoring and Iterative Improvement

- Failing to monitor analytics data continuously and iterate on insights can limit the effectiveness of the project. Neglecting to review and act upon analytics insights in a timely manner may lead to missed optimisation opportunities.

- Recommendation: Establish a process for ongoing monitoring and analysis of analytics data. Regularly review key metrics, track performance against goals, and identify areas for improvement. Encourage an iterative approach to implement changes based on data insights and continuously optimize the website's performance.

8. Product Development Life Cycle

The team was tasked to find the footfall for the travel website .

Here we deployed the analytics tool to complete the given task.

The tool was deployed in the website to generate the data in a certain pattern.

We then compared the analytics report with that of another tool to ascertain the data's authenticity.

8.1. New Tasks

The team meets twice a week to discuss the progress and assign new tasks to the members.

8.2. Definition of Done

How does the team know when a task is done?

What are criteria for a successfully completed task?

This may seem obvious, but it in a software development project having a definition of done can ensure a certain standard of work that holds all team members accountable. For example, messy, clunky code that "just works" is very different to clean, well-commented code that works AND is easy to understand. Which would you prefer to be your team's definition of done?

8.3. Task Review

Who reviews a task once it's been marked as done?

How does the team ensure that all work is looked over before it's contributed to the main repository or working prototype?

If you don't currently have a system for reviewing tasks, make sure to flag this for next trimester's team to work on as soon as they begin.

8.4. Testing

How do you test your product to see if it does what it was originally planned to do?

If your product isn't heavily comprised of software, how can you build in testing to your team's product development life cycle to ensure that "stuff works as it should"?

8.5. Branching Strategy

How does your team currently use GitHub repository?

What rules for commits and pull-requests have been put in place so far?

How should new members use GitHub repository in a way that doesn't result in all commits being dumped in a messy Master branch?

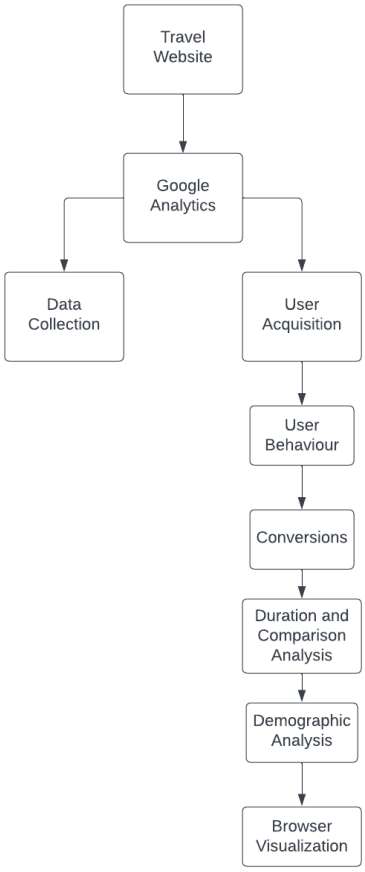
Again, if your team hasn't formally discussed a branching strategy, this a great opportunity to describe what your current system is and how it could be improved going forward.

For example, if you currently have all members of the team commit directly to the Master branch, can you recommend any tutorials for the future team to review that might lead to a cleaner, more organised and more efficient repository?

9. Product Architecture

9.1. UML Diagram

In the given diagram, the website is connected to Google Analytics, representing the integration between them. The "Google Analytics" component further connects different components representing various aspects of data analysis and tracking, such as "Data Collection", "User Acquisition", "User Behaviour", "Conversions", "Duration and comparison analysis", "Demographic Analysis" and "Browser Visualization".



9.2. Tech Stack

Google Analytics and a few scripts of python

10. Source Code

All source code should be found on your team's GitHub repository, unless your project has unique constraints that require you to store your code elsewhere. This includes any resources (e.g., wireframes, designs) that need to be transferred over to the new team as well.

Please provide all of the necessary instructions to accessing your source code. This includes URLs of online hosted repositories, links to any software dependencies, database components, or external libraries.

If your code is hosted on a server external to Deakin, make sure to also transfer digital copies of your code over to your client and the next team as a backup.

11. Login Credentials

Please provide all credentials (usernames and passwords) for any of the resources, websites, or platforms being utilised for this project. Please make sure that none of these credentials share passwords or usernames with any of your team's private credentials.

12. Other Relevant Information

This section is an invitation to add any additional information that you think will help to onboard new members. If you choose not to add any extra sections to this document, this section should be deleted.

Please edit this entire document as you see fit. If you think adding 5 extra sections that aren't listed here will help to communicate the nuances of your project to future members, go ahead! We want you to take full ownership of your handover and this document.

13. Appendices

Include all relevant artefacts delivered during the course of the project. Anything that will paint a clearer picture of your team's progress this trimester, the things that informed decisions, and the evolution of your product.

Please also include a link to your team's showcase video.

Project User Analysis

Project Leader: Miriam Llauce Cotrina

Team member & Database Responsible: Tejas Varun Baskar
Team member & Visualization Responsible: Miriam Llauce Cotrina

32. Project Overview

The User Analysis project's main goal is to provide access keep users engage with our product (Smart Bikes) by giving them access to their current and historical performance information. User's performance analysis will be available in real time and in multiple platforms; everyone who trains with our smart bikes will be able to analyse their performance evolution in a dynamic and user-friendly way.

The project was discussed on the first company meeting of this trimester. After the leaders explain what the company was about, we thought we needed to offer some sort of post-sale-experience to our users where they not only visualise the outcomes of their workouts but also it is a way to keep them engage with product by creating a competitive in-game environment.

During the development of this project, we used platforms such: MS Excel, Python, Big Query, Google Cloud, and Power BI. The reason we selected Power BI as the main platform to create the visualizations and final deliverables of this project is due to its property to turn unrelated sources of data into visually and interactive reports.

The final deliverables of this project will be dashboards where users can see their performance evolution and interact with its own data. Users can track their progress and see if it is going according to their personal goals. However, we also had to do research about cycling metrics and visualization, these metrics and why we selected them will be also detailed in this report (User Manual).

33. User Manual

The final deliverable of this Project contains four different dashboards in Power BI. These dashboards explore six different cycling metrics. It is important to resalt that the metrics were selected considering the research with have done and the data we had available.

Within each dashboard we have graphics that help the users observe their progress on a specific metric, in multiple and dynamic ways, each graphics was selected targeting the potential desire of our users. We aimed for these dashboards not to only look attractive but also to bring meaningful and valuable information that contributes to our user's fitness journey with our devices.

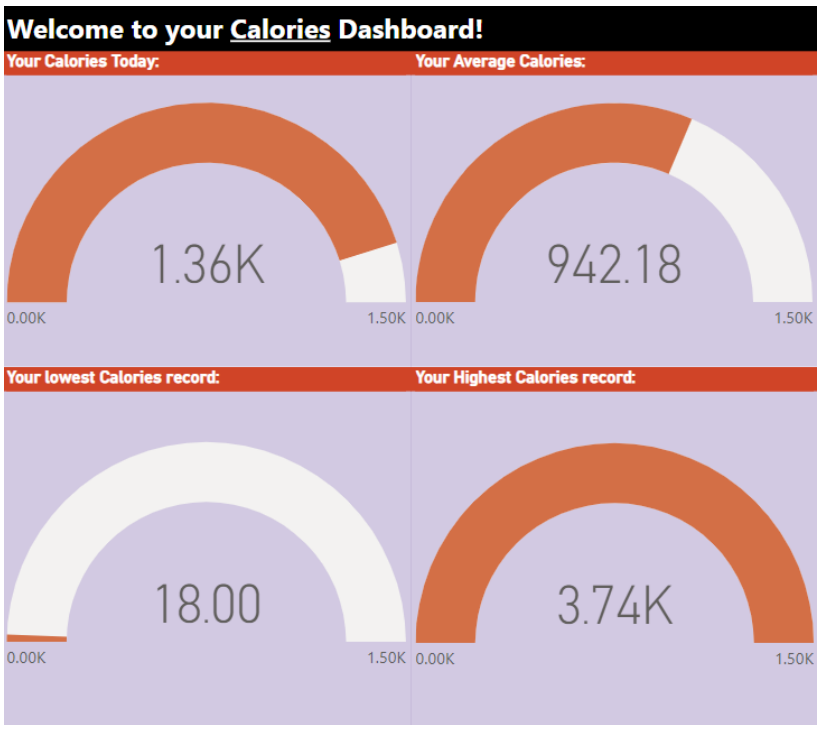
The Calories Dashboard:

This measure is noted to estimate the number of calories (unit of energy provided by food) that we are burning. Measuring this value is helpful while aiming for weight loss. The focus of weight loss is burning more calories and less intake of calories that burning. Most of the application and fitness devices use algorithms to calculate calories which is not exact most of the times. The better way to measure calories is by using power meter and heart rate sensors. The ration between power and calories is almost 1:1 with a 5% margin for error.

In this dashboard we can visualise four different graphics divided into two segments, the top two relates to Today's metrics compared to the daily average, and the other two at the bottom relates to the historical record (including today's).

The appropriate lecture of the below would be:

- "Today you have burnt 1,360 calories, the average calories you have burnt daily since you have joined is 942 Cal, your lowest in a day have been 18 Cal and your highest record is 3,740cal. – *This is an indicator that today our User have burnt more calories than their daily average, but not yet reached their maximum capacity".*

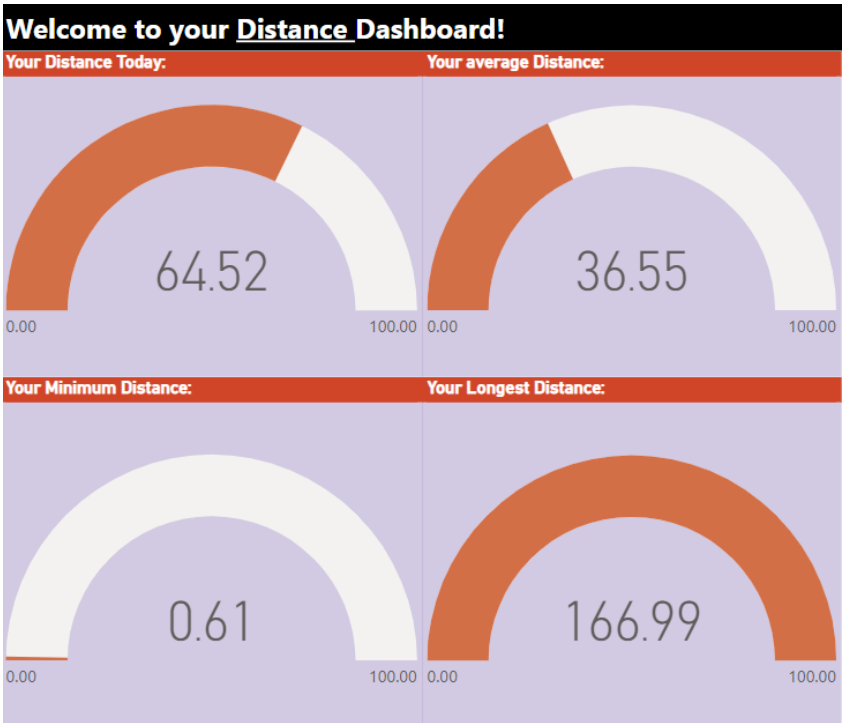


The Distance Dashboard:
Distance is used to measure the endurance of an individual. But calculating distance varies depending on the type of track/road they are riding on. Because riding on a flat track and riding on an elevated track is different because there is more effort put in when the track is a bit elevated. Consider a nominally fit person the average distance covered in kilometres is 20.

This dashboard also presents four different graphics divided into 2 segments, the top two relates to Today's metrics compared to the daily average, and the other two at the bottom relates to the historical record (including today's).

The appropriate lecture of the below would be:

- "Today you have ridden 64 kilometres, the average kilometres you have ridden daily since you have joined is 36.55 km, your lowest in a day have been 0.61 km and your highest record is 166km. – *This is an indicator that today, our User have ridden a distance longer than their daily average, but not yet reached their maximum capacity".*



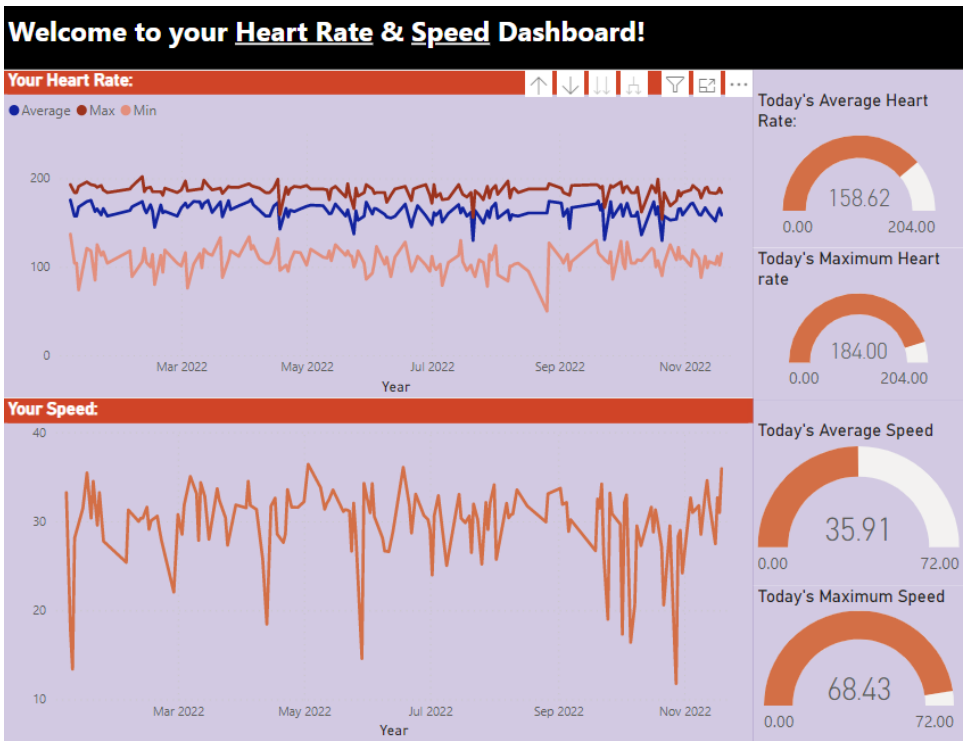
The Heart Rate & Speed Dashboard:

Measuring the heart rate is one of the important measures because it helps us to know about our pressure levels to the heart. When the heart rate is high it is a sign that we are pressurizing the heart too much in such cases we will have to reduce our work and give it a rest. Measuring high, low, and average is helpful as it helps us to analyse how we progress with our fitness. Maintaining lower and average heartrate is always better because it helps us avoid sudden heart problems.

Speed is majorly used for self-satisfaction. While see the speed and comparing it on a daily or monthly we can see the growth. Measuring the average speed is the best measure because while seeing a long-time review, we can consistently see the growth with average speed as current, min and max keeps varying a lot daily.

Both metrics were placed together due to their relation between each other, and this is probably one of the most complex dashboards that brings a lot of information to our users.

In this dashboard we can visualise the heart pulse and speed on a period of 1 year, and on the side graphics we can find most recently data such today's metrics and maximum reached since the user joined to the platform.

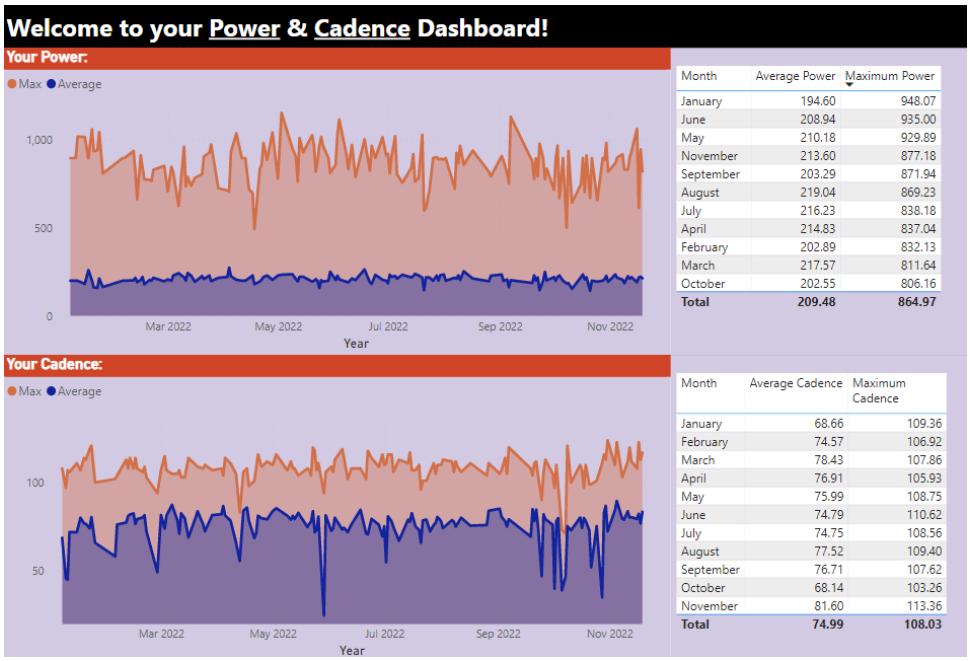


The Power & Cadence Dashboard:

Power is measured to determine how much effort you have put in to for the training session. Power is measured in watt. Measuring the average power is a better value considering a long-term analysis. As the average power increases, we can say that there a steady increase in performance and fitness. The average power for a beginner is around 75 – 100 watts.

Cadence is the number of revolutions per minute or RPM a person completes at a given speed while riding a bike. Generally, a good number to achieve in cycling cadence is between 80 – 100 rpm. Beginners' user will start pedalling as lower as 60 – 80 rpm and pro-users can do over 100rpm and 110 rpm during springs.

In this dashboard, we can visualise both the Power and Cadence. We have linear graphics with the last 12-month progression data, where the users can spot peaks and lowest points, and for a more specific data there is also tables with the numeric values per month.



Month	Average Power	Maximum Power
January	194.60	948.07
June	208.94	935.00
May	210.18	929.89
November	213.60	877.18
September	203.29	871.94
August	219.04	869.23
July	216.23	838.18
April	214.83	837.04
February	202.89	832.13
March	217.57	811.64
October	202.55	806.16
Total	**209.48**	**864.97**

Month	Average Cadence	Maximum Cadence
January	68.66	109.36
February	74.57	106.92
March	78.43	107.86
April	76.91	105.93
May	75.99	108.75
June	74.79	110.62
July	74.75	108.56
August	77.52	109.40
September	76.71	107.62
October	68.14	103.26
November	81.60	113.36
Total	**74.99**	**108.03**

34. Completed Deliverables

Data Cleaning & Handling
Primarily responsible: Tejas Varun Baskar Status: Fully Completed

Brief Description:
As far as the data cleaning is concerned the completed deliverables for this this trimester is processed data set of a single user of the wahoo devices. For this we converted the data that was given as each second's data and then it is now converted into a daily data that consists of metrics for a single day.

The other process done to the dataset is eliminating the NAN values and generalizing the given data set into 2 decimal points for better interpretation. Finally, after the selection of important metrics we find the minimum, maximum and the average for the appropriate metrics. Both, the code we use for data cleaning and the process ("How we did it") were stored on the below locations.

Locations:
Data Cleaning Procedure: https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2016%20Performance%20Ranking/Research%20on%20the%20important%20KPI%20and%20cleaning%20procedure..docx

Code for Data Cleaning: https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2016%20Performance%20Ranking/temp%20code%20for%20clearing%20the%20null%20values

Data Visualization & Dashboards
Primarily responsible: Miriam Llauce Cotrina Status: Fully Completed

Brief Description:
The main goal of this project was to create dashboards where users can visualise their fitness progress. At first, we needed to identify the appropriate cycling metrics, and then looked into our datasets to make sure we had stored the data we needed. After all of these was sorted, I uploaded the processed data into Power BI and created the dashboard.

The process of dashboard creation started before we jumped into Power BI, it started with Cycling metrics research as we needed first to understand what our Users would be interested to see. Both, the metrics research, and dashboards are part of the deliverables and they both can be found on the below locations.

Locations:
Cycling Metrics Research: https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2016%20Performance%20Ranking/Research%20on%20the%20important%20KPI.docx

Dashboards: https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2016%20Performance%20Ranking/RedBack%20Operations_Project%2016%20User%20Analysis%20FINAL%20PRODUCT.pbix

35. Roadmap

By the end of this trimester, all deliverables planned have been successfully finished, however there are plenty features that we identify during the process of creating our dashboards that could make our product more attractive and enhance the user experience by making our platform more useful.

Please note that the features will be described below were not planned to be developed during this trimester, hence we do not have them on our current Trello board, as they are features to be analyse and develop over the next trimesters.

- **User Data Handling**
  In terms of data cleaning the aim for next trimester deliverables are processing of data for all other user in the original dataset and finding a few more important metrics to analyse the performance in a much more efficient.

- **Data Visualization**
  It would be interesting if we could get feedback from user's based on their interaction with our product and improve the visualizations on our dashboard, and/ or incorporate new metrics for the user analysis.

- **User Interaction**
  One of the topics to work on next trimester is to incorporate interaction to the visualization. Users should be able to add their goals for the day, week, and month and track their current metrics with what they were expecting. For example, if a customer was expecting to burn 1,000 calories per day, we want them to input it into our system, and based on the data registered we should be able to tell them how far they are from achieving their goal. This feature will create an on-game competitive environment and will increase the engagement with our users.

- **Predictive Analytics**
  We believe this is the last goal for our dashboards. Based on user's previous data, we should be able to apply predictive analytics and forecast their metrics. This feature will help users to make a training plan to achieve their goals. For example, if x user would like to increase their speed from 25 km/h to 35 km/h based on their current

development how many hours should this user train per day and what their training should look like to achieve this goal in 30 days?

36. Open Issues

Data Access & Cleaning

- The major issue that we encounter at the start of the trimester is not having access to the actual dataset to extract the data and do the work. – *This was solved after a couple of weeks of having constant communication with one of our Company Leaders: Mark Telley.*
- Access to Google Cloud Platform – After the permissions were granted, one team member: Miriam Llauce had issues accessing to the google cloud platform. It was complicated to Log in into Google Cloud with the @deakin.edu.au account. Every time I tried to do it would take me back to my personal Google account. *I had to Log off and delete my personal account saved details from the browser and then log in into Google Cloud with my Deakin account.*

- Another issue was a confusion regarding the data we received, we had many cells filled as NaN and we did not know "how to eliminate the NaN values without affecting the dataset in large fashion" - *at the end we overcome this using Python collab.*

37. Lessons Learned

This trimester has been very insightful, and we both feel like we have accomplished a lot. Having our Company Team Leader: Mark and our Mentor: Ben support through this experience helped us to keep ourselves engage and motivated with the Company and Project.

Technology and Upskilling
In terms of technology, Power BI was the platform selected to develop our dashboards, and we both had very little knowledge. We both used it before, but we did not have the required experience, so we decided to upskill on it, we spend individual time upskilling and then we had weekly meetings where we both shared our learnings and put it into practise creating the dashboard.

We suggest to the next team members to dedicate time Upskilling if they think they need to, and to do so as soon as they start working on the project and not to leave it for the last minute, the better skilled they are the easier the work will be.

Teamwork
As a team, we believe that the major the major lesson learnt from this project in this trimester is the importance of teamwork. Since this project only consisted of two team members, there was a large amount of work that was split among us and most of these

tasks were dependent on the work of another team. We both showed responsibility, interest, and commitment with the project, we met every deadline, and we kept an open and constant communication which made this experience very pleasant.

We recommend to the next members to keep working as a team, to open different channels of communication and keep it ongoing.

Project Data Warehouse

Prastut Sapkota – Project Lead
Saransh Gupta – Developer
Ankit Mehta - Developer

38. Project Overview

The data for the Data and AI teams of Redback Company are temporarily stored in BigQuery environment. As BigQuery is a cloud-based environment and therefore has a limited database security option. As there is no infrastructure to manage and does not need a database administrator it can further expose the data into more vulnerabilities. So, we aim to provide an effective and long-term solution for the temporary measure that we have opted. The data will be pipelined into a Relational Database Management System (RDMS), and we aim to provide it through MSSQL. The data warehouse will consist of various layers from extracting Raw data to creating data marts for the business. In this project the primary focus would be providing an effective data warehouse architecture following data integration strategies, governance and security but not limited to modelling and analytics strategies.

38.1. Aims

- Defining an architecture for the warehouse which includes data ingestion methods, storage options, and software and hardware specification among others.
- Establishing data privacy, data security and data quality process.
- Development of logical models, data schemas and data marts.
- Establishing pipeline for the raw data connection and designing ETL pipelines.
- The ability of handling large chunks of data with continuous monitoring and optimizing.
- Development of master database.

38.2. Deliverables

38.2.1. Long-term

- A permanent solution to the extraction and storage of data.
- Integration of Extract, Transform and Load (ETL) workflows which extracts data from the source, transform as per the architecture and requirements of the data warehouse, and loads into the warehouse.
- An effective data quality and governance framework.

38.2.2. Trimester

- A visual representation of the data structure in the data warehouse.
- A detailed description of how data from various sources are integrated and transformed.
- A master documentation that consists of comprehensive documentation of the data in the warehouse along with system and user documentation.
- A beta version of the integration of ETL through the proposed data warehouse architecture.

39. User Manual

39.1. System Requirements

As we are mainly using major tools like Alteryx and MS SQL, we need to know their system requirements which are as follows:

39.1.1. Alteryx System Requirements:

- Operating System: Alteryx Designer is primarily available for Windows. As of my knowledge cut-off in September 2021, Alteryx Designer is not officially supported on Linux or macOS. However, you can use virtualization or containerization solutions like Parallels or Docker to run Alteryx on those platforms.
- Processor: 64-bit dual-core (x64) processor or higher
- RAM: 8 GB of RAM or higher (16 GB or more recommended for larger workflows and advanced analytics tools)
- Disk Space: 2 GB of available hard-disk space for installation
- Screen Resolution: 1280x800 or higher

39.1.2. Microsoft SQL Server System Requirements:

The system requirements for Microsoft SQL Server can vary depending on the specific version and edition you are installing. Here are the general system requirements:

- Operating System: Microsoft SQL Server is available for Windows, but there are also editions available for Linux and macOS.
- Processor: 64-bit processor with a speed of 1.4 GHz or faster (2 GHz or faster recommended)
- RAM: At least 1 GB (4 GB or more recommended)
- Disk Space: Minimum of 6 GB of available hard-disk space for installation
- Screen Resolution: Minimum of 1024x768 pixels

It's important to note that the system requirements can vary based on the specific version, edition, and workload you plan to run. It is recommended to refer to the official documentation provided by Alteryx and Microsoft for the most accurate and up-to-date system requirements for their respective products.

39.2. Downloading Alteryx

To download Alteryx, follow these steps:

- Visit the Alteryx website: Go to the Alteryx website at https://www.alteryx.com.
- Navigate to the Products section: Click on the "Products" tab in the top navigation menu.

- Choose Alteryx Designer: Select "Alteryx Designer" from the list of products. This is the primary tool for data preparation, blending, and advanced analytics. Follow the prompts to download Alteryx.

39.3. Setting up SQL Server

- Download SQL Server: Visit the official Microsoft website or the SQL Server product page to download the version of SQL Server that suits your needs. Choose the appropriate edition and ensure it is compatible with your operating system.
- Run the Installer: Once the SQL Server installation file is downloaded, run the installer by double-clicking on it. This will start the SQL Server Installation Centre.
- Choose Installation Type: In the SQL Server Installation Centre, select "New SQL Server stand-alone installation or add features to an existing installation" to begin the installation process.

39.4. Running the SQL Script

The link to the GitHub repository where the MS SQL script is located which is to be executed in the MS SQL located in the local machine.

39.5. Setting up Database Connections in Alteryx

To set up a database connection in Alteryx, follow these steps:

- Launch Alteryx: Open Alteryx Designer on your computer.
- Open Workflow: Create a new workflow or open an existing one where you want to set up the database connection.
- Drag Input Tool: From the "Connectors" tab in the toolbar, locate and drag the "Input Data" tool onto the workflow canvas.
- Configure Input Tool: Double-click on the Input Tool to open its configuration window.
- Select Database: In the configuration window, select the "Database" option from the left panel.
- Choose Database Type: Choose the appropriate database type from the drop-down menu.

39.6. Troubleshooting

39.6.1. Alteryx:

1. Verify Database Connection Details: Double-check the server's name, authentication mode, username, password, and database name you provided in the database connection configuration. Ensure they are correct and match the settings of your SQL Server.

2. Test the Connection: Use the "Test Connection" button in the Alteryx database connection configuration window to check if the connection can be established successfully. If the test fails, review the connection details and verify network connectivity to the SQL Server.

3. Firewall and Network Settings: Ensure that the necessary firewall ports are open to allow communication between your computer running Alteryx and the SQL Server. Check your network settings and consult with your network administrator if needed.

4. Driver Compatibility: Confirm that you have the appropriate database drivers installed for the version of SQL Server you are connecting to. Outdated or incompatible drivers can cause connection issues. You can usually download the required drivers from the database vendor's website.

5. Permissions and Credentials: Verify that the user credentials provided for the database connection have the necessary permissions to access the SQL Server and the specified database. Check with your database administrator to ensure the user has the required privileges.

39.6.2. Microsoft SQL Server:

1. SQL Server Service Status: Ensure that the SQL Server service is running on the server. You can check the service status using the SQL Server Configuration Manager or Services console.
2. Check SQL Server Error Logs: Examine the SQL Server error logs for any error messages or warnings that may indicate issues with the server. The error logs are typically located in the "Log" folder within the SQL Server installation directory.

3. Network Connectivity: Verify that your computer can reach the SQL Server by pinging the server's IP address or hostname. If there is no response, check your network configuration, firewall settings, and network connectivity.

4. Authentication Mode: If you are using SQL Server authentication, confirm that the provided username and password are correct. If using Windows authentication, ensure that the Windows account has the necessary permissions to access the SQL Server.

5. Check Database Availability: Ensure that the target database is online and accessible. Use SQL Server Management Studio (SSMS) or a similar tool to connect to the server and verify the database status.

If you encounter any specific error messages or issues, referring to the Alteryx and SQL Server documentation, online forums, or reaching out to their respective support channels can provide more detailed troubleshooting guidance.

40. Completed Deliverables

We have successfully completed the development of an ETL (Extract, Transform, Load) model for the three datasets provided to us. Each dataset was assigned to a team member, and by the end of this trimester, we have delivered the completed ETL model.
To begin the process, we imported the data from Google Big Query to our individual machines. Next, we established databases using MS SQL on our respective machines to store and manage the data. We utilized Alteryx, a powerful data analytical tool, to perform data cleaning and transformation tasks. By leveraging Alteryx and MS SQL, we created Raw, Staging, and Master databases for each dataset, ensuring an organized and structured data flow.
All the relevant files and folders associated with our project can be found on GitHub. We have committed our final changes and deliverables to the repository, allowing for easy access and version control.
In summary, the completed deliverables for this trimester include Alteryx Workflow for each of the dataset:

1. Bike Data:
    - Responsible Team Member: Saransh Gupta
    - Location: https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2018%20Data%20Warehouse/Development/Alteryx/Bike_Data
2. Review Data:
    - Responsible Team Member(s): Ankit Mehta
    - Location: https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2018%20Data%20Warehouse/Development/Alteryx/Review_data
3. Fitness Data:
    - Responsible Team Member(s): Prastut Sapkota
    - Location: https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2018%20Data%20Warehouse/Development/Alteryx/Fitness%20Data

41. Roadmap

41.1. Next Trimester:

- Enhanced ETL Workflows: Continuation of the development and improvement of the Extract, Transform, and Load (ETL) workflows. This involves refining and optimizing the existing ETL processes, incorporating additional data sources, and enhancing data transformation logic.
- Data Mart Development: Creation of specific data marts tailored to the needs of different business units or analytical requirements. Data marts provide subsets of data optimized for specific analysis or reporting purposes, enabling efficient and targeted data retrieval.

- Data Governance Implementation: Implementation of data governance practices and mechanisms to ensure ongoing data quality, compliance, and security. This includes establishing data stewardship roles, defining data standards, implementing data validation, and monitoring processes, and enforcing data privacy regulations.
- Advanced Analytics and Reporting: Development of advanced analytics capabilities, such as predictive modelling, machine learning, and data visualization tools. This will enable the Data and AI teams to derive valuable insights from the data warehouse and generate meaningful reports for decision-making purposes.
- Data Security Enhancements: Implementation of additional security measures to safeguard sensitive data within the data warehouse. This may include encryption, access controls, audit trails, and data anonymization techniques to ensure compliance with privacy regulations and protect against potential data breaches.
- Automated Data Pipeline Monitoring: Integration of monitoring and alerting systems to proactively identify and address issues within the data pipelines. This includes setting up automated alerts for data quality anomalies, performance bottlenecks, or pipeline failures, allowing timely intervention and minimizing potential disruptions.

41.2. In Progress (Incomplete Work Items):

- ETL Workflow Optimization: Currently, the ETL workflows are in progress and being optimized for improved performance, scalability, and reliability. This involves identifying and resolving any bottlenecks, fine-tuning data transformations, and ensuring efficient data loading processes.
- Data Governance Framework Development: The development of the data governance framework is ongoing. This includes the definition of data standards, establishment of data stewardship roles and responsibilities, and the implementation of data validation processes. The framework is being designed to ensure data quality and compliance.

These features and deliverables provide an overview of the planned work for future phases of the project, including those that are currently in progress. They demonstrate a progression towards a mature and efficient data warehouse solution that addresses Redback Company's data integration, governance, security, and analytics needs.

42. Open Issues

42.1. Potential issues and challenges the team is currently facing, along with progress made to address them:

- Data Quality and Availability: The team is encountering challenges with the data provided, such as null values and bugs in the data. This hampers their ability to clean and analyze the data effectively using Alteryx. To address this, the team has initiated data quality checks and validation processes. They are working on identifying and addressing null values and data bugs to improve the overall data quality. It is crucial to continue investing in data cleansing and validation techniques to ensure reliable and accurate data for analysis.

- Alteryx Compatibility and Platform Limitations: The team has identified that Alteryx does not work on macOS, which affects the ability of team members using Mac computers to utilize Alteryx for data processing. Additionally, different versions of Alteryx have different tools and configurations, and Windows versions may not support the latest versions of Alteryx or MS SQL. While progress may have been limited in addressing these issues, it is important for future teams to explore alternative data preparation and analytics tools that are compatible with macOS. They should also ensure that all team members are using compatible versions of Alteryx and have the necessary system requirements to support the desired functionality.
- Unclear Task Review Process: The team is facing challenges with an unclear process for reviewing completed tasks on Trello, resulting in a backlog of work that is between unfinished and finished. To address this, the team should establish a clear and well-defined task review process. This process should include criteria for task completion, designate responsible reviewers, and outline the necessary steps for closing tasks in Trello. Clear communication and regular updates on task statuses are essential to avoid confusion and efficiently manage the workflow.
- Team Member Availability: Availability of team members is a challenge that can impact project progress. If team members have conflicting schedules or other commitments, it can delay task completion and coordination. To address this, it is important to establish regular communication channels, set clear expectations for availability and responsiveness, and distribute tasks and responsibilities based on team members' availability and expertise. Regularly monitoring and adjusting workloads can help mitigate the impact of limited availability.

42.2. Recommendations for Future Teams:

- Data Quality Assurance: Establish data quality assurance processes from the early stages of the project. Implement data cleansing and validation techniques to identify and address data issues. Work closely with data providers to improve data quality and ensure accurate and reliable data for analysis.
- Compatibility and Platform Considerations: Assess the compatibility of tools and platforms with team members' systems and project requirements. Explore alternative software solutions that are compatible with different operating systems. Stay updated with the latest software versions and system requirements to ensure compatibility and access to necessary functionalities.
- Well-Defined Processes: Define and document clear processes for task management, including task review and closure. Ensure all team members are familiar with and follow these processes consistently. Regularly review and refine the processes based on feedback and evolving project needs.
- Resource Planning and Communication: Plan resources and workloads effectively, considering team members' availability and expertise. Establish regular communication channels and set clear expectations for responsiveness and availability. Proactively address conflicts and adjust workloads to ensure smooth project execution.
- System Security and Maintenance: Implement regular backups, system monitoring, and maintenance practices to prevent system corruption and ensure data integrity.

Develop a disaster recovery plan and invest in security measures to protect project data and minimize potential disruptions.

By addressing these challenges and following the recommendations, future teams can mitigate risks, enhance project efficiency, and create a more conducive working environment.

43. Lessons Learned

43.1. Key Lessons Learned:

- Clear Communication Channels: One important lesson learned is the significance of establishing clear communication channels within the team and with stakeholders. It is crucial to have regular and transparent communication to ensure everyone is aligned, progress is effectively conveyed, and any challenges or roadblocks are addressed in a timely manner. Clear communication helps in managing expectations and maintaining project momentum.
- Continuous Testing and Quality Assurance: A valuable lesson is the importance of incorporating continuous testing and quality assurance processes throughout the project lifecycle. This ensures that data transformations, ETL workflows, and system functionalities are thoroughly tested and validated, minimizing the risk of errors or issues in the production environment. Early and frequent testing helps in identifying and addressing issues promptly.
- Agile Project Management: Adopting an agile project management approach enables flexibility and adaptability to be changing requirements and evolving project needs. Agile methodologies, such as Scrum or Kanban, facilitate iterative development, frequent feedback cycles, and continuous improvement. This allows for better responsiveness to emerging challenges and ensures that the project stays on track.
- Documentation and Knowledge Sharing: Comprehensive documentation and knowledge sharing are essential for future reference, onboarding new team members, and facilitating seamless handovers. Maintaining updated documentation about the data warehouse architecture, data models, workflows, and system configurations helps in ensuring continuity and reduces dependency on specific individuals.

43.2. Recommendations for Future Teams:

- Emphasize Collaboration: Encourage a collaborative work environment where team members actively collaborate and share their knowledge and expertise. Foster a culture of open communication, regular stand-up meetings, and collaborative decision-making. This helps in leveraging the diverse skill sets within the team and promotes shared ownership of the project's success.
- Prioritize Data Governance from the Start: Establish a robust data governance framework early in the project to ensure data quality, security, and compliance. Define data standards, implement data validation processes, and establish clear roles and responsibilities for data stewardship. Proactively address data governance considerations rather than treating them as an afterthought.
- Test Automation: Invest in test automation tools and frameworks to streamline and expedite the testing process. Automated testing helps in reducing human errors,

ensuring consistent testing practices, and accelerating the feedback loop. It allows for more comprehensive test coverage and supports regression testing as the project evolves.

- Continuous Integration and Deployment: Implement continuous integration and deployment (CI/CD) practices to automate the build, testing, and deployment processes. This allows for rapid and reliable deployments, reduces the risk of introducing errors during deployment, and ensures a more efficient release cycle.
- Knowledge Transfer and Succession Planning: Plan for knowledge transfer and succession planning to ensure the smooth transition of project ownership and knowledge sharing when team members transition out. Encourage documentation, conduct knowledge sharing sessions, and assign mentors to facilitate the transfer of knowledge and expertise.

By incorporating these recommendations, future teams can overcome potential challenges and increase the effectiveness and efficiency of their project. Clear communication, agile practices, strong data governance, automation, and knowledge sharing will contribute to the project's success and ensure a more streamlined and sustainable data warehouse solution.

44. Product Development Life Cycle

Over the course of the trimester, as a team, our work methods, habits, and processes have evolved organically and while we may not be able to clearly defined Product Development Life Cycle, we have set up practices that would guide our work.

The first set of practices that we followed is planning and setting up the goals. We started the week from the planning phase itself and identified tasks that we are going to complete on those weeks. Furthermore, we set up weekly objectives and defined what can be the key milestones and deliverables.

Following Planning and goal setting, we do follow the task allocation and Responsibilities through the Trello platform. Once our goals and deliverables has been set, the project lead would assign themselves and other colleagues based on their expertise and availability.

Apart from the weekly meetings, we have a constant communication with the team where we would discuss if any hurdles that may arise during performing the tasks and find out solutions to solve those hurdles. We do follow the agile approach of product development, as the project that we had needs us to adapt to the changing requirements and iterate on our work.

44.1. New Tasks

The new tasks are created through teams collaborating process. Usually in our planning session we have a brainstorming and idea generation session where all the team members are encouraged to contribute their ideas and suggestion regarding the tasks. Once these ideas are finalized, we then evaluate them based on the impact and alignment to the project goals. Once we have the well-defined requirements, we break down the work into smaller, manageable tasks and then these tasks are assigned to the respective team members based on their weekly availability, workload, and skills. Some of the tasks may be reprioritized later and to tackle this requirement we have been involved in an iterative approach.

44.2. Definition of Done

We try to define the definition of done before starting the task itself which includes the specific criteria and expectations required for the task completion. The task should meet the objectives and delivers the expected outcomes. We set up the task deliverables to exhibit high level of quality and functionality following the global standard code, features, and components. The task should be free from significant bugs and should be listed in the bug section in Trello board.

44.3. Task Review

The tasks are reviewed by the team lead or the project manager who supervises the tasks and review process. The team themselves creates a unit testing environment to help the team lead to identify any bugs or error arising in the tasks and solves them before sending up for review.

The team lead performs a code review on the tasks that have been pushed into the fork repository. Once there is a green light from the team lead, the team members then prepare themselves for the next tasks. Since, we haven't had a proper designated Quality Assurance (QA) team or testing team we followed the process of unit testing environment. We are hoping for the Readback Operations to assign a designated team for QA and testing to have a quality product with bug free.

44.4. Testing

As the developer themselves must perform the testing phases, we opted for unit testing approach. Unit testing approach helps in assuring that the code and data from the source file to the databases are accurate.

As the team is growing in Redback Operation, once we have a designated QA or tester, we intend to follow the following steps:

- Creating a comprehensive test plan which outlines the testing approach, objectives, and scope.
- Based on the planned requirements, various test cases are to be developed.
- Checking any changes or additions made to the product do not impact existing functionalities.
- A bug tracking system that stores all the bugs and errors identified.

44.5. Branching Strategy

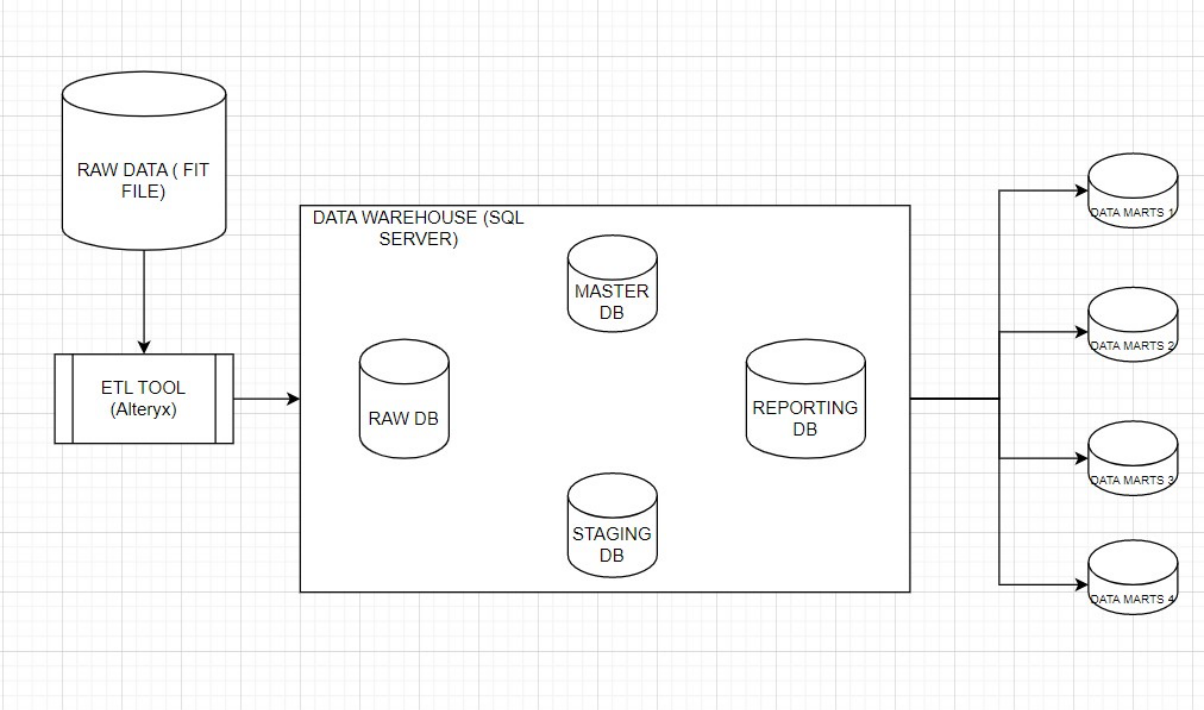
The current team has created a fork where they perform their development tasks. Once, the code has been developed the developer push their code into the designated folder inside the forked repository. Once it has been developed and approved by the project lead, the project lead performs pull request where the team lead perform a code review and merge it into the main branch.

The commits and pull-requests are only performed once the development is done, and unit testing is performed. Then, the project lead performs a code review and if only a green signal is given from the project lead the pull-requests is performed.

A good approach towards avoiding messy branch is to create branching by each task. The new member should perform regular committing and pushing into the branch. Also, team collaboration and communication can play an important role as any concerns are addressed properly by the team members regarding the GitHub repository.

45. Product Architecture

45.1. Data Architecture



45.2. Tech Stack

We have used two tools: Alteryx and SQL server. Here's is short description on both tools:

- Alteryx: Alteryx is a powerful data preparation and analytics platform that provides a range of functionalities for data blending, cleansing, and advanced analytics. It offers a visual interface that enables users to design and execute data workflows without the need for coding. Alteryx was likely chosen for this project because of its ability to handle complex data transformations and automate repetitive tasks, allowing for efficient data preparation and analysis.
- SQL Server: SQL Server is a relational database management system (RDBMS) developed by Microsoft. It provides a robust platform for storing, managing, and retrieving structured data. SQL Server supports the SQL (Structured Query Language) standard and offers features such as data integration, data warehousing, and business intelligence. SQL Server was likely chosen for this project due to its scalability, security features, and extensive toolset for data management and analysis.

46. Source Code

Please find the below link to the source code for the project:
Github Source Code Link

47. Login Credentials

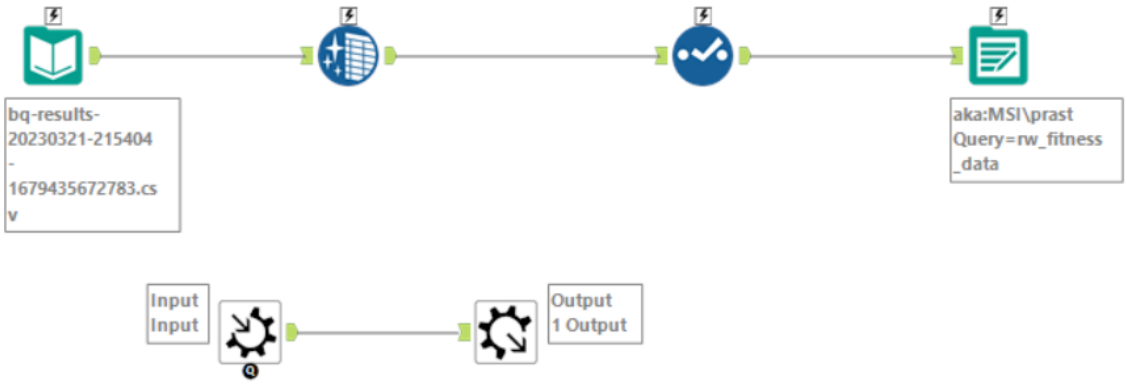
The project was developed and tested in local environments. Please download the following two tools:

- Alteryx
- MS SQL server

For running the Alteryx workflow, create a database connection to your local MSSQL server to the input and output of the workflow to run the workflow.

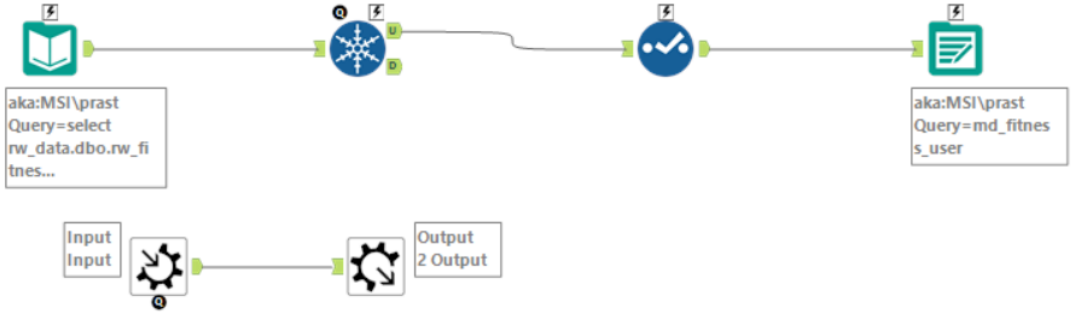
48. Raw Database

A raw database is referred to as the initial or source database that contains the original, unprocessed data collected from various sources. The main purpose of it is to provide reliable and persistent storage solution for the data before it goes through the Staging or ETL process.
Alteryx:



49. Master Database

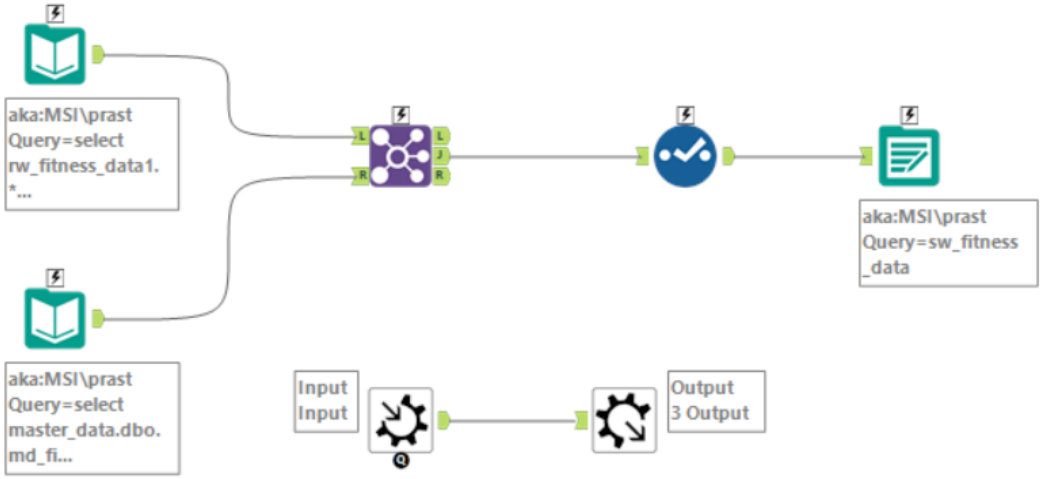
A centralized repository that stores accurate and comprehensive data about various entities is termed as Master Database. The master database contains key elements of the data which are vital and consistent across different applications. The main aim of master database is to maintain data integrity, eliminate redundancy, and promote data consistency.
Alteryx:



50. Staging Database

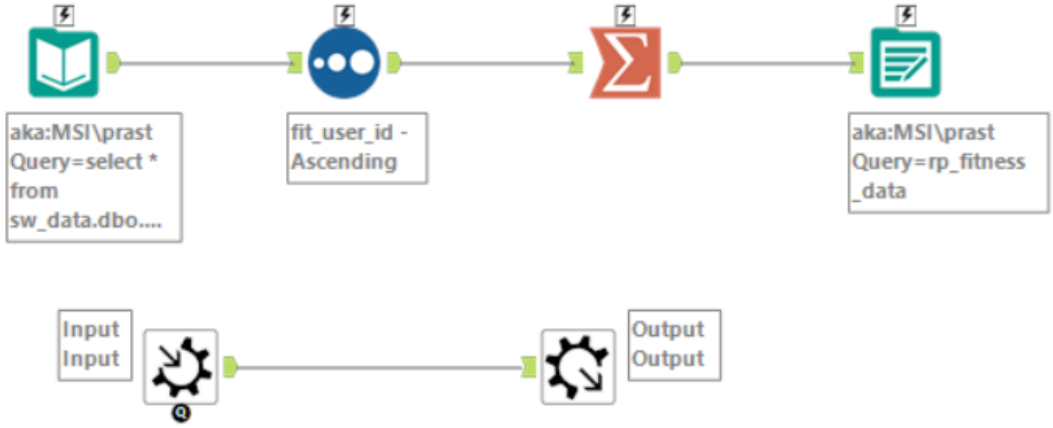
The purpose of staging data base is to provide data cleansing, transformation, and integration before loading the data into production stage and thus is referred as the intermediate step

between the source and the target data warehouse which in our case is Raw and Production database respectively.
Alteryx:



51. Production Database

The production database consists of live production environment data that contains aggregated and normalized data.
Alteryx:



52. Appendices

Project Video Link: Video Showcasing the project
Project Research Link: Github Data Warehousing Research
Project Trello Link: Trello

Workout Categorisation

Redback Operations

Team Member: Nicholas Manning (Project Lead)

53. Project Overview

Smart Bike users exercise for many different purposes: general fitness, fat loss, endurance etc. This project aims to develop a clustering model to categorise these workouts into groups of workout types to better tailor the user's experience and further gamify their training sessions. Each workout completed provides different outcomes in terms of distance covered, power generated, speed maintained, hills climbed etc. The model's goal is to identify the workout type of a particular workout so that we can decompose the benefits of the workout, offer workout suggestions or track their goals more accurately.

This project will use second by second breakdowns of user workout session data to develop an unsupervised machine learning clustering model to group workouts together with other similar workouts, then analyse those groups to determine what sort of similarities and features are identified. Once these labels are determined, they can be used for workout recommendations in the future.

The project is divided into four parts:
1. Research - Research and testing of clustering models and performance metrics.

2. Data Acquisition and Cleaning – Accessing and importing Smart Bike data for the model, followed by cleaning, filtering, standardising and extracting relevant features.

3. Modelling – Testing models and performance metrics to determine most successful outcome and applying those to identify appropriate clusters.

4. Cluster Analysis – Decomposing and visualising characteristics of clusters to identify similarities observed by the model and workout types.

Deliverables:
1. An aggregated dataset exemplifying standard user sessions.

2. An unsupervised clustering model trained and tuned on the dataset.

3. A set amount of workout types identified in the data.

4. A prediction model for assigning future sessions to groups.

5. Handover documentation for Trimester 2 – 2023.

54. User Manual

Access to files and scripts described can be found in the below:

GitHub:
https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2017%20Workout%20Categorisation

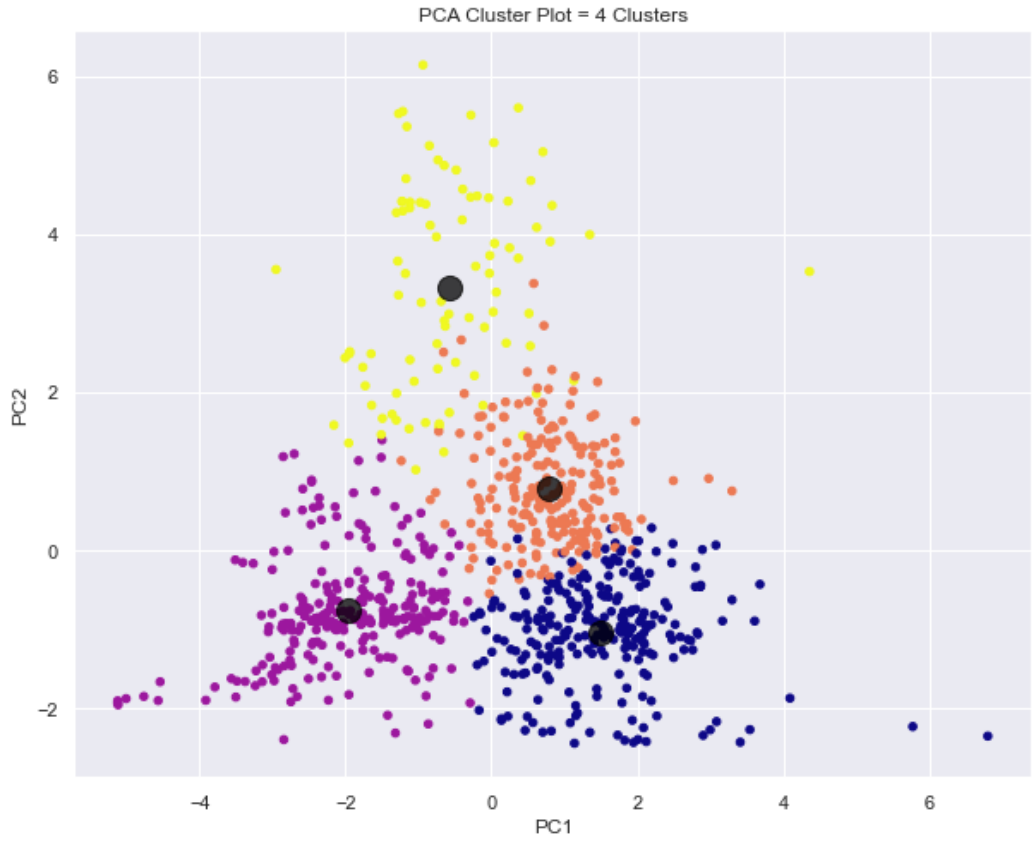
Google Cloud Console - BigQuery:
https://console.cloud.google.com/bigquery?project=sit-23t1-fit-data-pipe-ee8896e&ws=!1m4!1m3!8m2!1s891680982841!2sbb3a9d3a3feb4a9ea56a5c035945d280

Utilising the workout categorisation clustering model requires a Fit file dataset, made of workout sessions, which can be accessed through Deakin's Google Cloud Console - BigQuery environment and the Smart Bike fitness data:
Dataset ID = sit-23t1-fit-data-pipe-ee8896e.fitness_data.

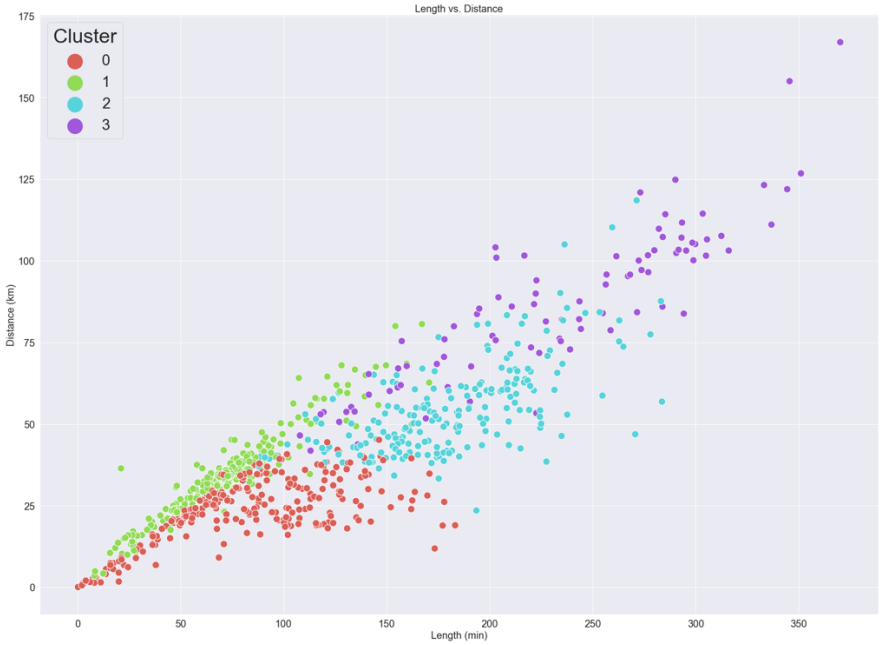
To retrieve the training dataset, I submitted the below SQL query:

```sql
SELECT
  tiemstamp_AEST, distance, ascent, enhanced_speed, heart_rate, cadence, power, userID
FROM
  `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data`
WHERE
    distance > 0
    AND power IS NOT NULL
    AND enhanced_altitude IS NOT NULL
    AND ascent IS NOT NULL
    AND heart_rate IS NOT NULL
    AND cadence IS NOT NULL
ORDER BY
  userID, tiemstamp_AEST ASC;
```

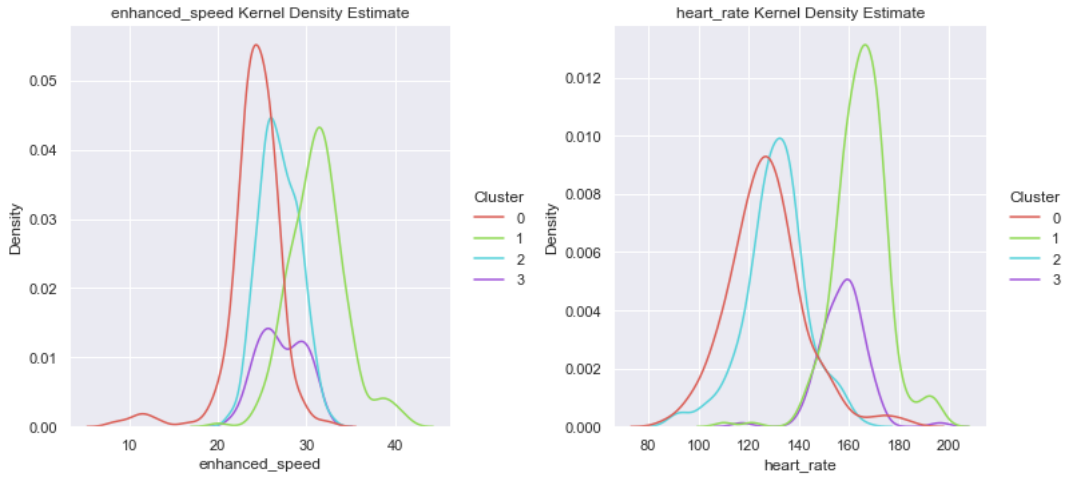
Once the dataset is exported to your local computer, it can be run through the *'Session-By-Second.ipynb'* python script. This script aggregates the second-by-second user data into individual workout sessions, cleans the data, standardises the distribution and removes any unnecessary datapoints, creating useful features for the final model, such as session length, total distance, total ascent, average heart rate, average cadence and average power.
The future user will then be able to recreate the results of the model and visualise the clusters. The whole dataset can be visualised using dimensionality reduction, turning the dataset 2D. Along with other metrics, the clusters can be assessed to confirm there are distinct workout types in the data. See below, the Principal Component Analysis (PCA) scatterplot with 4 clusters highlighted.



A future user may run the model and choose a different amount clusters if they feel there is justification in the data, for example, if new sessions in the training dataset show a new workout type has emerged. Currently there appears to be clear delineations in the data to justify 4 clusters, as shown by the session Length vs session Distance scatterplot below:



KDE plots, which highlight the distributions of individual features, also can be used to show a clear difference in clusters, example below of the Speed KDE plot and Heart Rate KDE plot.



Once a number of clusters is chosen, the model and the scaler must be saved down in *'.sav'* files for use in the prediction model. The current model files can be found in the project GitHub.

The prediction script, *'Import and Apply Clustering Model.ipynb',* takes a single second-by-second fitfile dataset and applies the same transformations required for entry into the model, It then runs the data through the model and assigns the new workout session to a cluster.

55. Completed Deliverables

Completed deliverables include:

1. BigQuery SQL query for exporting data for model training.
https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2017%20Workout%20Categorisation/BigQuery%20FitFile%20Master%20Query%20for%20Workout%20Categorisation.docx

2. Master Fit file aggregation, cleaning, feature extraction, modelling, data visualisation and analysis tools.
https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2017%20Workout%20Categorisation/Session-BySecond%20Pipeline.ipynb

3. Cluster prediction/allocation model.
https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2017%20Workout%20Categorisation/Import%20and%20Apply%20Clustering%20Model.ipynb

4. Model and Scaler files
https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2017%20Workout%20Categorisation/clustering_model.sav

https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2017%20Workout%20Categorisation/clustering_scaler.sav

56. Roadmap

The next step in the project requires the implementation of the model in a user interface:

1) Import recently completed or uploaded training session fitfile into the workout categorisation model to predict/allocate the session into a workout type.

2) Take the predicted workout type and determine what workouts to recommend.

3) Present the recommendation back to the user in a format they can utilise. For example, a pre-set training session focused on hills, sprints, average power etc.

57. Lessons Learned

From a process and technology perspective, the major lesson learnt from this project is to ensure that the data utilised in the machine learning model is in a format that correctly represents the action being modelled and is therefore able to be used in a model effectively.

When the trimester began our team did not have access to the BigQuery environment and could not get the full Fit file dataset from the Smart Bikes. Initially this was not an issue, given Mark Telley, the cohort leader was able to provide a useful test dataset, however, this dataset had some aggregation and feature extraction methods already applied to it, which skewed the data. It also meant that when access to the site was granted, a whole new approach was required and a new focus back on simply getting the data into the right shape.

This is not to say that the early part of the trimester was wasted, it just was not used as efficiently as it could have been. Future teams building machine learning models should prioritise the data because without the right data the model cannot learn and function properly.

58. Product Development Life Cycle

58.1. New Tasks

The idea for the project was part of a list of suggestions offered by the cohort lead at the start of the trimester, focusing on delivering new features to the Smart Bike project in the Data and AI space. As I was the only one who was interested in the workout categorisation project, I took on the task of outlining what could be done within the timeframe given and what broader expectations could be set for delivering insights to the user in the long term.

I bounced ideas and plans back and forth with cohort lead, Mark Telley, around what could be achieved and how the current data needed to be transformed to result in a deliverable product. I then set about working through the required steps to deliver the clustering model: researching, acquiring, and cleaning the data, modelling and analysing the results.

When problems were encountered, I referred to Mark for assistance and advice, but overall, I was able to work through problems and come up with new approaches to resolve issues on my own volition.

58.2. Definition of Done

This project had a simple yet complex goal. Build or extract a dataset reflective of the workouts and build a model based on the dataset. The difficulty in these two steps was not necessarily in the coding, but more so in the confirmation that the data was an accurate reflection of the workouts, and the model represented that fact. The task was deemed done when the visualisations and analysis performed on the results justified that there were distinct clusters that could then provide useful recommendations to the user in the future.

58.3. Task Review

Work was reviewed by the cohort lead Mark Telley when GitHub pull requests were submitted to merge updates with the main branch. Mark provided feedback on direction and offered support if needed. In future teams if there were more members working on this problem a different review process may be more beneficial.

58.4. Testing

The model testing involved analysis and visualisation of the clusters to ensure that the model grouped datapoints into distinct and relevant clusters. The analysis and visualisation highlighted what features the model deemed most important. It not only validated the model itself, but also the steps taken to manipulate, standardise and extract features from the data to create the dataset.

The final testing involved taking a single workout session Fit file from the source master data used to train the model, performing the requisite transformations, and inserting the session into the model. The final cluster allocation of the session matched the cluster allocated as part of the training set.

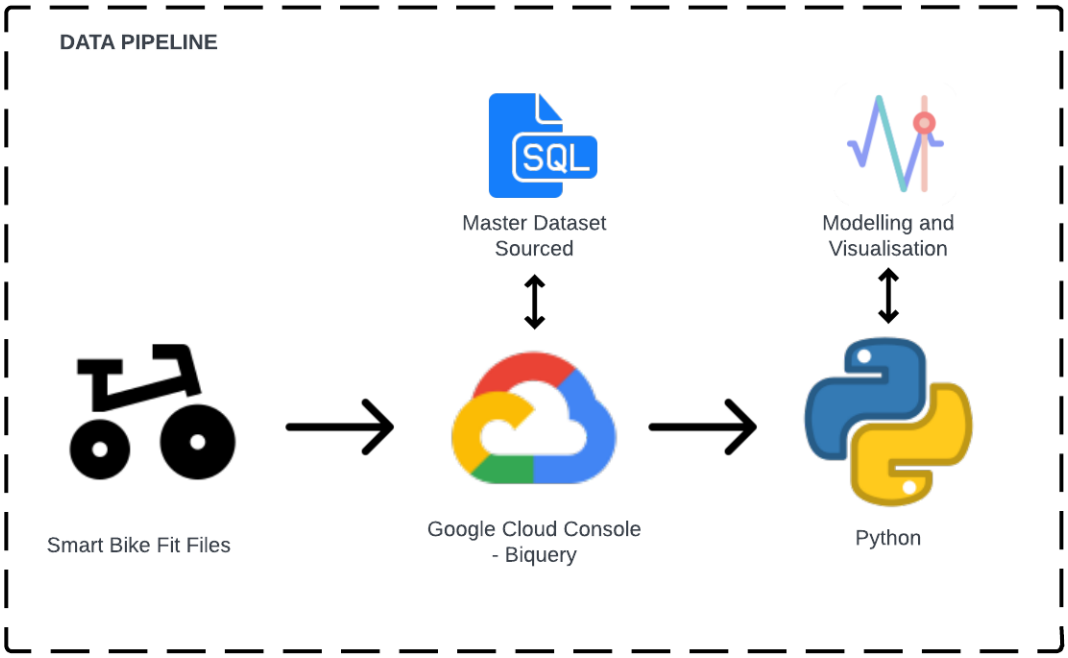
58.5. Branching Strategy

The Data and AI team GitHub had branches for each project. Each user created a fork of the main branch, but only needed to amend their branch. When pull requests were submitted to

merge the changes on a project only that branch was affected. Pull request were reviewed and approved by cohort leader Mark Telley.

59. Product Architecture

59.1. UML Diagram



59.2. Tech Stack

1. Google Cloud Console – BigQuery: Existing infrastructure used to house and query Smart Bike Fit file master dataset.

2. Anaconda3 – Jupyter Notebook 6.3.0: Python 3.8.8 selected for programming the model due to comprehensive machine learning libraries.

3. GitHub – Version control and web hosting service for the project.

5. MS Excel – Export/Import datasets from BigQuery into models.

60. Source Code

Workout Categorisation Project GitHub:
https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2017%20Workout%20Categorisation

Data and AI team Google Cloud Console - BigQuery environment:
https://console.cloud.google.com/bigquery?project=sit-23t1-fit-data-pipe-ee8896e&ws=!1m4!1m3!8m2!1s891680982841!2sbb3a9d3a3feb4a9ea56a5c035945d280

61. Login Credentials

Access to Google Could Console – BigQuery and Data and AI team GitHub Repository must be granted through the company lead.

62. Appendices

1. Data and AI Trello Board
https://trello.com/b/NSuF3z83/data-analytics

2. Workout Categorisation GitHub
https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2017%20Workout%20Categorisation

3. Project Showcase Video

Project 11 FIT File Handling and Data Pipeline

Project Team

Mark Telley

Company Redback Operations

63. Project Overview

The FIT File Handling and Data Pipeline project aims to handle FIT files from the Wahoo KICKR Live, convert them to CSV format, and upload the data to a database. It will aim to provide real-time performance metrics through a rudimentary user interface using basic JS, HTML, and CSS as an MVP (only data points). The project will offer guidance to the web/application team on integrating the data within the game experience. A Python script will communicate with the KICKR, download, and convert FIT files to CSV, and integrate the data into the data warehouse project for storage. The project will provide a comprehensive solution for handling KICKR Live FIT files, making the data easily accessible for analysis and real-time performance metrics.
What is the project about?

What problem is the project solve?

- Efficiently handles FIT files from the Wahoo KICKR Live.
- Converts FIT files to CSV format to support analysis and data manipulation.
- Unlocks the ability to upload the converted data to a database for storage and accessibility.
- Supports the ability to develop user interface to display real-time performance metrics (see all the other T2 projects)
- Offers guidance to the web/application team on integrating the FIT file data within the game experience.
- Streamlines the process of retrieving, converting, and analysing FIT file data.
- Eliminates the need for manual data extraction and conversion.
- Enables users to monitor their performance during workouts and make informed training decisions + allows the data science and AI team to conduct analysis etc.
- Provides a comprehensive solution for handling and integrating KICKR Live FIT files.

What are the aims of the project?
Aims for Trimester

- Develop a Python script for communicating with the Wahoo KICKR Live via Bluetooth connectivity and Wahoo API, downloading, and converting FIT files to CSV, and uploading data to a database.

- Create a rudimentary user interface using basic JS, HTML, and CSS to display real-time performance metrics as an MVP.
- Provide guidance to the web team on integrating the data within the game experience.

What are the deliverables?

Long-term Deliverables:

- A comprehensive solution for handling KICKR Live FIT files, ensuring easy accessibility for analysis and real-time performance metrics.
- Implementation of a scalable and secure data pipeline that seamlessly integrates with the game experience.

Trimester Deliverables:

- Development and completion of a Python script capable of downloading, converting, and uploading FIT file data utilising the Wahoo API and Bluetooth connectivity with the KICKR Live.
- Creation of a rudimentary user interface as a minimum viable product (MVP) to display real-time performance metrics.
- Documentation providing clear instructions on the usage and deployment of the Python script.
- Provision of guidance to the web team on effectively integrating the FIT file data within the game experience, including instructions on how to access the data through the implemented data pipeline.

64. User Manual

Three key elements make up the Project: Setting up a GCP and BigQuery environment, connecting to Bluetooth devices such as but not limited to Wahoo's Kickr bike trainer and connecting to Wahoo via API to retrieve information.

64.1. 1.1. GCP & BigQuery

The GCP project provides a sandbox environment and essential datasets to support various projects. From handling FIT files and corporate reporting to sentiment analysis and user ranking, the project covers a wide range of data-related activities. It also facilitates access to app analytics data, demographics analysis, fitness data, user data, and Wahoo Kickr data. Also note original fitness data is stored securely in Google Cloud Storage buckets. It's an opportunity for the team to explore and innovate while paving the way for future developments in data management and analytics.

Refer to the readme.MD for comprehensive instructions.
https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/GCP_Bigquery/GCP_BigQuery_Documentation.md

64.2. Direction Connection to Wahoo Kickr

The BLE Cycling Power Data Collection sub project is a python script that captures cycling power data from a Bluetooth Low Energy (BLE) device and saves it in a CSV file. Utilising libraries like asyncio, bleak, and pandas, the script provides a straightforward process for data collection. After setting up the necessary libraries, connecting to the BLE device (Wahoo Kickr), and specifying the session length, running the script captures data such as timestamps, power output, energy accumulation, pedal power balance, torque, wheel and crank revolutions, force and torque magnitudes, as well as dead spot angles. The collected data is stored in a pandas dataframe and saved in a CSV file. Documentation also focuses on the use of the PyCycling package as a valuable resource for cycling-related data analysis.

Refer to the readme.MD for comprehensive instructions.
https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Kickr_Connection/README.md

64.3. Direction Connection to Wahoo Kickr

The Wahoo API/FIT Handling is a comprehensive solution that utilises the Wahoo API to facilitate interactions with Wahoo products and services. It offers several functionalities to enhance the integration and handling of FIT files:

1. Authentication: The API allows developers to authenticate with Wahoo by providing the necessary client ID and client secret. These credentials can be obtained from the Wahoo Developer Portal.

2. User Details: The API enables the retrieval of user details, providing access to relevant information associated with the authenticated user.

3. Select Workout: Users can select specific workouts using the API, making it convenient to fetch and work with targeted exercise data.

4. Workout Summary / FIT File: The API allows users to retrieve workout summary information and obtain the associated FIT file. FIT files are widely used in the fitness industry to store detailed workout data, and accessing this information opens possibilities for analysis and further processing.

5. FIT Conversion and Handling: The script includes functionality to convert and handle FIT files. By using the "fitparse" and "csv" libraries, FIT files can be transformed into CSV format, allowing for easier manipulation and analysis of the data.

6. Data Warehouse / Table: The script also provides an **example** of integrating the FIT file data into a data warehouse or table. Documentation demonstrates connecting to a data warehouse, creating a table, inserting data, and closing the connection. This functionality streamlines the process of storing and organising the FIT file data for future analysis and retrieval.

7. API Endpoints: The Wahoo API offers various endpoints that facilitate communication with the Wahoo system. These endpoints include authentication, user details retrieval, workout listing, workout summary retrieval, and more.

8. Security Considerations: The script emphasises the importance of client secret security, ensuring that sensitive information is handled with care and stored in secure locations. Documentation discusses token refresh mechanisms to ensure seamless authentication without compromising user data.

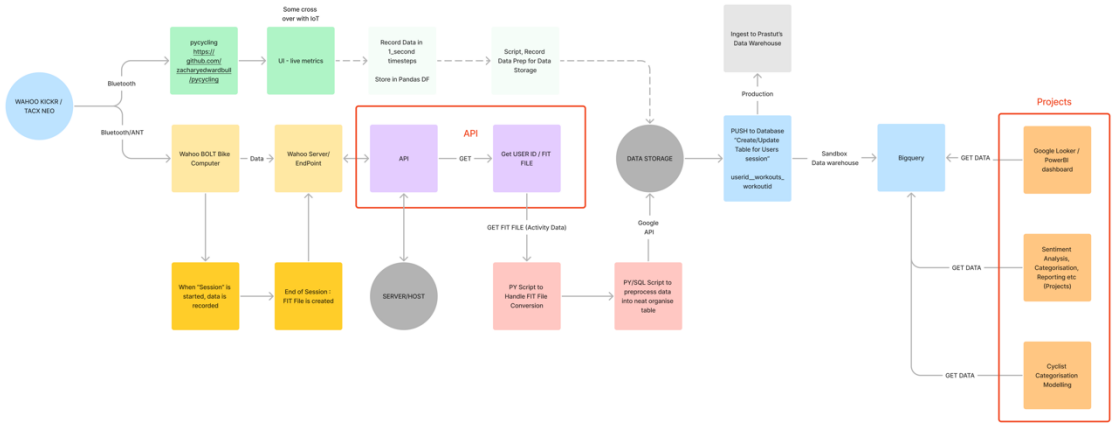
9. Database Handling: While providing an example of data insertion into a SQL Server database, the script acknowledges the need to address potential SQL injection issues and handle data types properly. Adhering to best practices, such as using parameterised queries and data type considerations, is crucial when interacting with databases to ensure data integrity and security.

Full Documentation: https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Wahoo_Cloud_API/readme.md

64.4. Pipeline Ideation

The below aims to provides an initial data pipeline that could service the Data/AI team:

- The Blue = GCP Environment with a future to switch to the data warehouse (3.1).
- The Green = Directly connecting to the Bluetooth device (3.2).
- The Yellow / Purple / and Red = FIT File and Wahoo API handling (3.3).



65. Completed Deliverables

GCP

- Set up a Google Cloud Account and established a sandbox BigQuery database for project use.

Wahoo Device Connection

- Successfully recorded data from a Wahoo Kickr during a live session
- Thoroughly documented the steps involved and tested code.

Wahoo API and FIT File Handling

- Obtained a Wahoo API development account for the data pipeline project.
- Developed a working Wahoo API to retrieve information.
- Finalised the FIT handling script, ensuring efficiency and error-free operation.
- Thoroughly documented the steps and tested code.

Refer to the GITHUB LOCATION FOR ALL UPDATES:
https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline

66. Roadmap

1. Collaboration with IoT Team:
    - Goal: Retrieve user-generated data from IoT devices.
    - Milestone: Enable the generation of FIT files upon exercise completion.
    - Action: Work closely with the IoT team to establish seamless data integration and mirror Wahoo's CLOUD API data model.

2. Integration with other Redback Teams and support data and analytical requirements:
    - Goal: Utilise the retrieved data from different teams to drive analytics and visualisation efforts across the entire company.
    - Milestone: Incorporate the IoT-generated data into the existing project framework Project 11)
    - Action: Align with the to identify the specific data requirements and adapt the project scope accordingly.

3. Completion and Deployment of Data Pipeline:
    - Goal: Establish a robust and efficient data pipeline using the Data Warehouse.
    - Milestone: Successfully deploy the pipeline for seamless data processing and storage.
    - Action: Collaborate with the other teams to design and implement the pipeline architecture, ensuring scalability, security, and efficiency.

4. Expansion of Data Sources:

- Goal: Capture additional data sources to enrich the project insights.
- Milestone: Incorporate data from app/website usage, corporate reporting, social interactions, and user ranking.
- Action: Engage with respective teams (e.g., UX, Cyber, Mobile) to define data collection mechanisms, integrate APIs or data connectors, and design data processing workflows.

5. Refinement of the Project Roadmap:
    - Goal: Align the project with Trimester 2 objectives and priorities.
    - Milestone: Evaluate the project roadmap and adapt it to future goals.
    - Action: Engage in strategic discussions with stakeholders to identify Trimester 2 project requirements, dependencies, and resources.

By following this roadmap, we can as a team and company aim to enhance our data capabilities by incorporating real user-generated data, expanding the scope of our analytics and visualisation project, and establishing a robust data pipeline that supports other teams in the Company. This roadmap promotes cross-team collaboration and ensures that the project remains aligned with the evolving goals and objectives of Redback.

67. Open Issues

Wahoo developer Account
. Formal application from Deakin needs to be confirmed/completed.
. Data model from IoT side of things needs to be established to mirror Wahoo's.

68. Lessons Learned

1. Data Model Agreement:
   Lesson: It is crucial to establish a clear and agreed-upon data model between IoT devices and external platforms like Wahoo or Garmin.
   Recommendation: Future teams should prioritise defining and aligning the data model early on to avoid compatibility issues and ensure seamless data integration.
2. Streamlined Communication Channels:
   Lesson: Effective communication channels are vital for project coordination and progress tracking.
   Recommendation: Future teams should establish streamlined communication channels to facilitate regular updates, feedback sharing, and efficient collaboration. Utilise tools like project management software and regular meetings to keep the team informed and aligned.
3. Thorough Documentation:
   Lesson: Comprehensive documentation is essential for knowledge transfer and project continuity.
   Recommendation: Future teams should prioritise thorough documentation of project scope, technical details, and any challenges encountered. Documenting processes, code repositories, and project resources will aid future team members and ensure smooth transitions.

4. Embrace Agile Methodologies:
   Lesson: Agile methodologies enable adaptability and iterative development.
   Recommendation: Future teams should consider adopting agile project management practices, such as scrum or Kanban, to promote flexibility, regular feedback, and continuous improvement. Agile approaches help teams navigate changing requirements effectively.
5. Testing and Quality Assurance:
   Lesson: Early testing and quality assurance are crucial for identifying and addressing issues promptly.
   Recommendation: Future teams should emphasise early testing and quality assurance throughout the development process. Implement automated testing frameworks, conduct regular code reviews, and involve stakeholders in the validation process to ensure high-quality deliverables.

69. Product Development Life Cycle

Team Collaboration: We prioritise working as a team to deliver tangible value to the project and the company.

Scheduled Stand-ups: We conduct two weekly stand-up meetings (start, and end of the week) to synchronise our progress and address any issues or additional tasks required for the project. Ad hoc discussions also take place via the Teams app for quick decision-making.

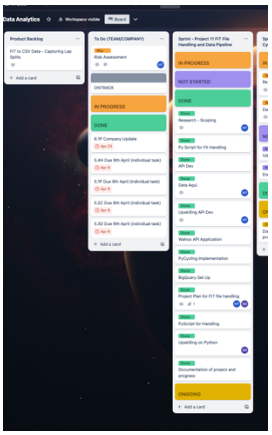
Task Planning and Progress Tracking: We utilise Trello boards to plan and track our tasks, updating our progress accordingly. We frequently create pull requests (PRs) and ensure timely merging by the team lead.

69.1. New Tasks

We come up with new tasks along the way while we are working on existing planned tasks or from each stand-up meeting time. Any new tasks will be created in the Trello board.

69.2. Definition of Done

A DoD list is normally clearly defined in each Trello card, so the card assignee will be able to know exactly when a task is treated as completed by meeting all the DoD items. Additionally, we also have different status labels on each task on the Trello board to indicate their completeness.



69.3. Task Review

All tasks' updates are reviewed by the Team Lead prior to being committed/merged in the Github Repo

69.4. Testing

Testing was conducted manually – this involved creating test cases and working through them. All task work provided has been tested, and retested.

69.5. Branching Strategy

We never directly push any changes into the company's `main` branch. To make any changes, we either create a new branch based off the latest `main` branch or fork the company's `main` branch into our own repo. After we've finished the changes, we create a PR against the company's `main` branch, and have it reviewed and merged by the team lead. We also ensure to resolve conflicts (if there's any) before merging back to the latest `main` branch.

70. Product Architecture

70.1. UML Diagram

Refer to point 3.4 Pipeline Ideation

70.2. Tech Stack

Google Cloud	. Google BigQuery . Google Storage Cloud (Buckets) . Google Looker Studio (Visualisation) . Google Colab (Python Coding)
wahoo DEVELOPER	. Wahoo Cloud API

	. Version control . Project management and documentation
Code Languages:	. SQL . Python . C (Curl request)

71. Source Code

Refer to the following links

FIT Handling and API:
https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Wahoo_Cloud_API/Wahoo_FIT_Handling.ipynb

Bluetooth Connection:
https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Kickr_Connection/Wahoo_Kickr_Connection
https://github.com/redbackoperations/data-analysis/blob/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline/Kickr_Connection/Wahoo_Kickr_Connection/main.py

72. Login Credentials

BigQuery / GCP refer to project documentation:
To gain access to the GCP Project / BigQuery, please contact your team lead. They will need to coordinate with:

- Scott Blackburn (Senior Technical Officer, Cloud Computing & AI, School of Information Technology)
- GCP Project: `SIT-23t1-fit-data-pipe-ee8896e`

Wahoo Developer:
Request Use of the Cloud API
The Cloud API uses the public Wahoo server and authorised user data. Because of this, Wahoo Fitness is currently limiting the use of the API to those who request it, as well as providing more information about the scopes involved and the purpose of the application.

When you apply request to Wahoo, it will show up on your Developer Portal as pending approval. Be sure to include as much information as you can about your application so Wahoo can be confident in approving your use of the Cloud API.

Refer to project documentation here

73. Appendices

Refer to the Github Repo for all project documentation.

https://github.com/redbackoperations/data-analysis/tree/main/Trimester_1_2023/Project%2011%20FIT%20File%20Handling%20and%20Data%20Pipeline

Corporate Reporting

1.Project Plan: Project Pan is designed and is uploaded at Trello board at Link
2.The Data fields of fitness data are studied and analysed to know their functionality and scope for reporting and below is the analysis:

Time Stamp: The FIT Profile defines the date_time type as an uint32 that represents the number of seconds since midnight on December 31, 1989 UTC. (UTC is AEDT + 10).
Position_lat/Position_long: Garmin GPS devices are set by factory default to lat/long DM. This means it is set to latitude and longitude in degrees and minutes, with decimal minutes.
i.e., it is represented as 156° 44', 72° 10'.
Distance: The Garmin automotive devices can show distance in either miles/feet (statute units) or kilometres/meters (metric units)
Enhanced_altitude/Altitude: The device will measure changes in air pressure to determine your elevation. This information is recorded during your activity and is used to report elevation related information in Garmin Connect. Elevation calibrated by GPS is accurate to +/-400 feet with a strong GPS signal. If the values of altitude are too large to be fit in Altitude, then enhanced altitude is used.
Ascent: a climb or walk to the summit of a mountain or hill/an instance of rising or moving up through the air. Total Ascent provides a total of all increases to elevation (also known as elevation gain). Average Ascent provides an average of all ascents recorded during an activity. Maximum Elevation provides the highest elevation achieved.
Grade: Data field for Garmin devices that calculates the slope (or grade) of the hill you are walking on. It publishes the grade value (in %) to Garmin Connect so you can have a timeline inside your activity.
Calories: This is the total of active and resting calories that are calculated during a recorded activity on your device (from the moment that you start the timer for the activity to the moment you stop the timer). Speed/Distance Algorithm: This is the most basic method of determining calories. It is represented in calories/Kcal.
Speed/Enhanced Speed: It is distance by total time sent on an activity. It is calculated in m/sec or m/h. If the values of speed are too large to be fit in speed, then enhanced speed is used.
Heart_rate: heart rate values can be set as absolute or relative values. Absolute values represent beats per minute (bpm) for heart rate, or watts for power.
Temperature: The Temperature widget will display the ambient air temperature near the barometric altimeter port. This reading can be affected by body heat.It is is represented in Fahrenheit.
Cadence: The cadence fields in a FIT file represent RPMs. For cycling 1 RPM equals one full rotation of the cranks. For running 1 RPM represents a step.
Power: Power values can be set as absolute or relative values. Absolute values represent watts for power.
Left_right_balance: It shows as a percentage the power separately put out by the left and right leg.
Gps_accuracy: It represent the drift with accurate gps values. GPS location accuracy is around 3 meters (10 feet), 95% of the time on Garmin devices. This means, at any given time, your device will save your location within 3 meters of your actual location.
Product_Name: It describes the product used for recording the activity.
Serial_Number: Most Garmin devices will have a unique serial number listed on the back or bottom of the device.

Age: Available on select Garmin watches, Fitness Age is an estimate of how fit you are compared to your actual age. Compatible Garmin watches will measure your Fitness Age differently, depending on which device you have. Fitness age is an estimate of how fit you are compared to your actual age.
Gender: It Shows gender of registered person of the device.
Weight: It gives the weight of the person in kgs.
FTP: Functional Threshold Power (FTP) is a measurement from power meters. It is the highest power level you can maintain for one hour without growing fatigued. FTP is beneficial because it provides an outlook on performance ability.
Session_ID: It is unique id generated for each session performed by user.
User_ID: It is the unique ID generated for every user.

3. Queries for Views are created to fetch out different fields:
All the Queries were done on `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy` table. A copy of `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data' is made and data is updated such that there is current year data to have visual output of all queries.

DISTANCE: Total Distance travelled by the user in last month, last week, previous day are fetched out by using the below queries.

MONTHLY_REPORT:

```
SELECT userID,sum(distance) AS TOTAL_DISTANCE
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE) - 1
and EXTRACT (YEAR from DATE_AEST) = EXTRACT (YEAR from CURRENT_DATE)
group by userID
order by userID;
```

WEEKLY_REPORT:

```
SELECT userID,sum(distance) AS TOTAL_DISTANCE
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (DAY from DATE_AEST) > EXTRACT(DAY FROM CURRENT_DATE) - 7 and
EXTRACT (DAY from DATE_AEST) <= EXTRACT(DAY FROM CURRENT_DATE)
and EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE)
and EXTRACT (YEAR from DATE_AEST) = EXTRACT(YEAR FROM CURRENT_DATE)
group by userID
order by userID;
```

DAILY_REPORT:

```
SELECT userID,sum(distance) AS TOTAL_DISTANCE
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (DAY from DATE_AEST) = EXTRACT(DAY FROM CURRENT_DATE) - 1 and
EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE)
and EXTRACT (YEAR from DATE_AEST) = EXTRACT(YEAR FROM CURRENT_DATE)
group by userID,date_AESt
order by userID;
```

CALORIES: Total Calories burned by the user in last month, last week, previous day are fetched out by using the below queries.

MONTHLY_REPORT:

```
SELECT userID,sum(Calories) AS TOTAL_Calories_Burned
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE) - 1
and EXTRACT (YEAR from DATE_AEST) = EXTRACT (YEAR from CURRENT_DATE)
```

```
group by userID
order by userID;
```

WEEKLY_REPORT:

```
SELECT userID,sum(Calories) AS TOTAL_Calories_Burned
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (DAY from DATE_AEST) > EXTRACT(DAY FROM CURRENT_DATE) - 7 and
EXTRACT (DAY from DATE_AEST) <= EXTRACT(DAY FROM CURRENT_DATE)
and EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE)
and EXTRACT (YEAR from DATE_AEST) = EXTRACT(YEAR FROM CURRENT_DATE)
group by userID
order by userID;
```

DAILY_REPORT:

```
SELECT userID,sum(Calories) AS TOTAL_Calories_Burned
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (DAY from DATE_AEST) = EXTRACT(DAY FROM CURRENT_DATE) - 1 and
EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE)
and EXTRACT (YEAR from DATE_AEST) = EXTRACT(YEAR FROM CURRENT_DATE)
group by userID,date_AESt
order by userID;
```

WEIGHT: Average weight maintained by the user in last month, last week, previous day are fetched out by using the below queries.

MONTHLY_REPORT:

```
SELECT userID,avg(weight) AS Weight
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE) - 1
and EXTRACT (YEAR from DATE_AEST) = EXTRACT (YEAR from CURRENT_DATE)
group by userID
order by userID;
```

WEEKLY_REPORT:

```
SELECT userID,avg(weight) AS Weight
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (DAY from DATE_AEST) > EXTRACT(DAY FROM CURRENT_DATE) - 7 and
EXTRACT (DAY from DATE_AEST) <= EXTRACT(DAY FROM CURRENT_DATE)
and EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE)
and EXTRACT (YEAR from DATE_AEST) = EXTRACT(YEAR FROM CURRENT_DATE)
group by userID
order by userID;
```

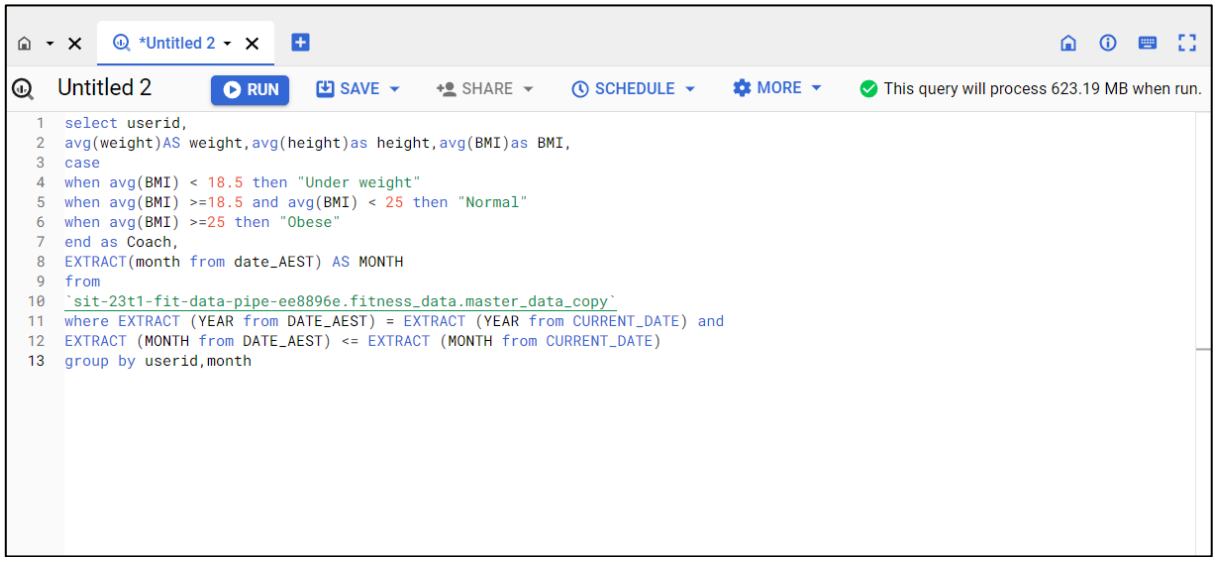
DAILY_REPORT:

```
SELECT userID,avg(weight) AS Weight
FROM `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (DAY from DATE_AEST) = EXTRACT(DAY FROM CURRENT_DATE) - 1 and
EXTRACT (MONTH from DATE_AEST) = EXTRACT(MONTH FROM CURRENT_DATE)
and EXTRACT (YEAR from DATE_AEST) = EXTRACT(YEAR FROM CURRENT_DATE)
group by userID,date_AESt
order by userID;
```

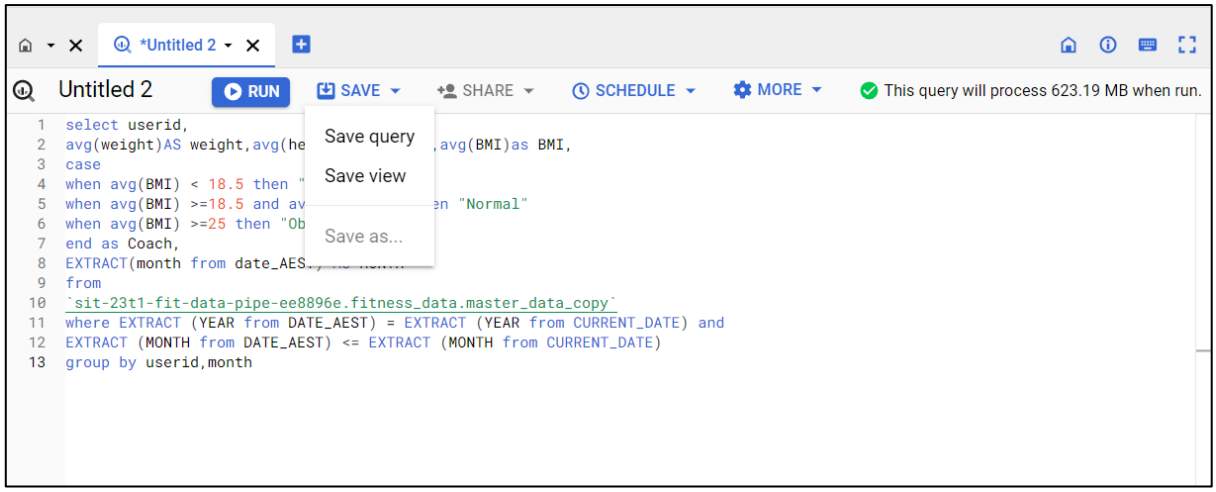
4.Views are created for the above drafted queries:

Steps to Create Views:

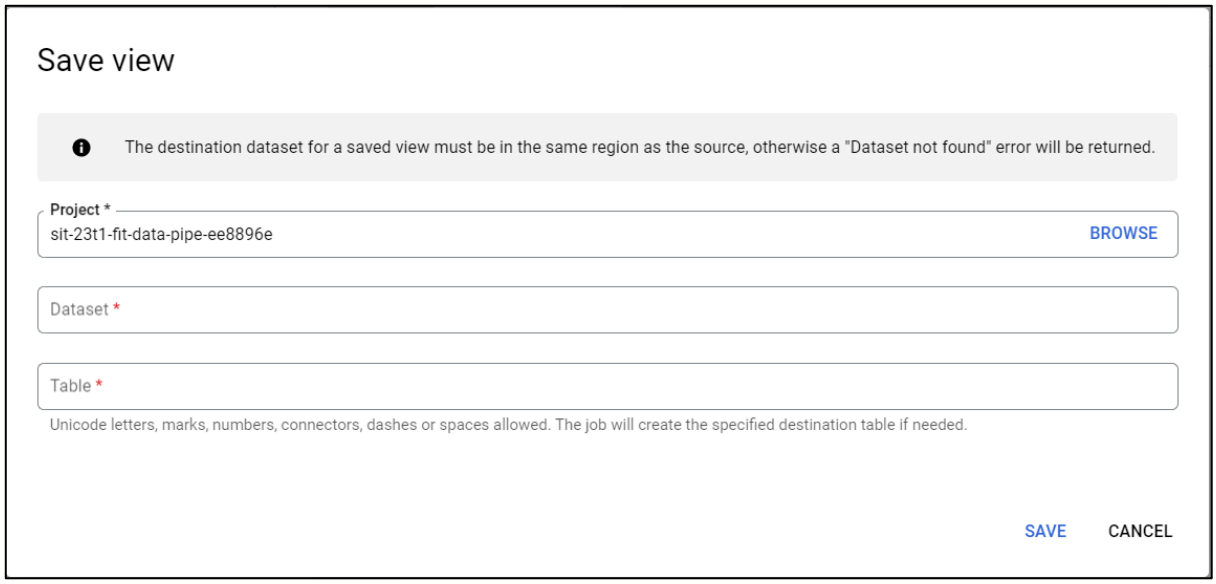
1. Copy the query on to new tab as below:



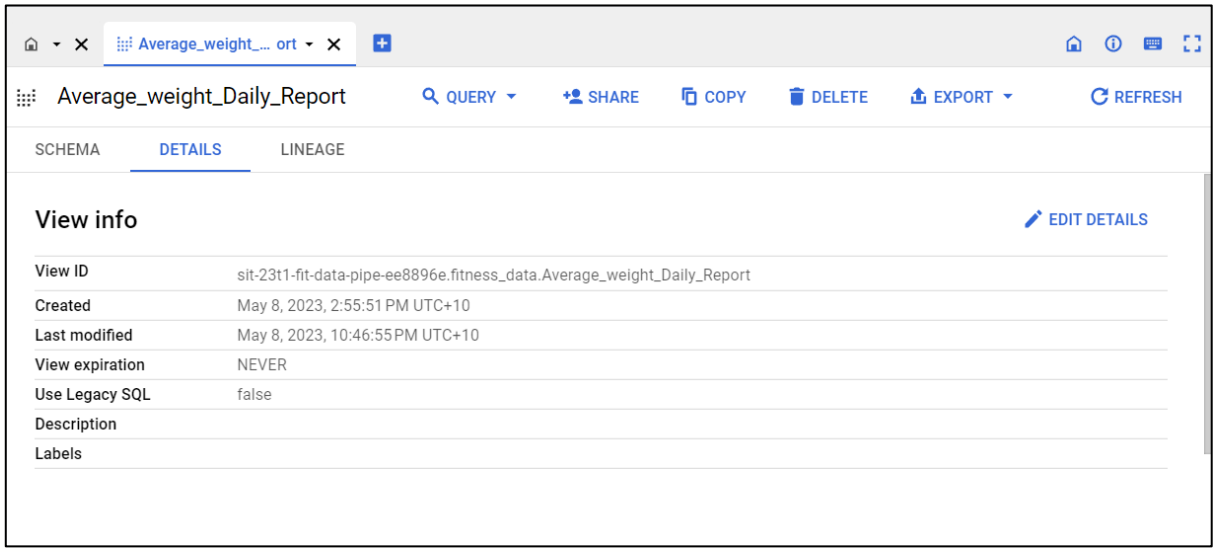
2. Click on Save view in the drop down next to save:

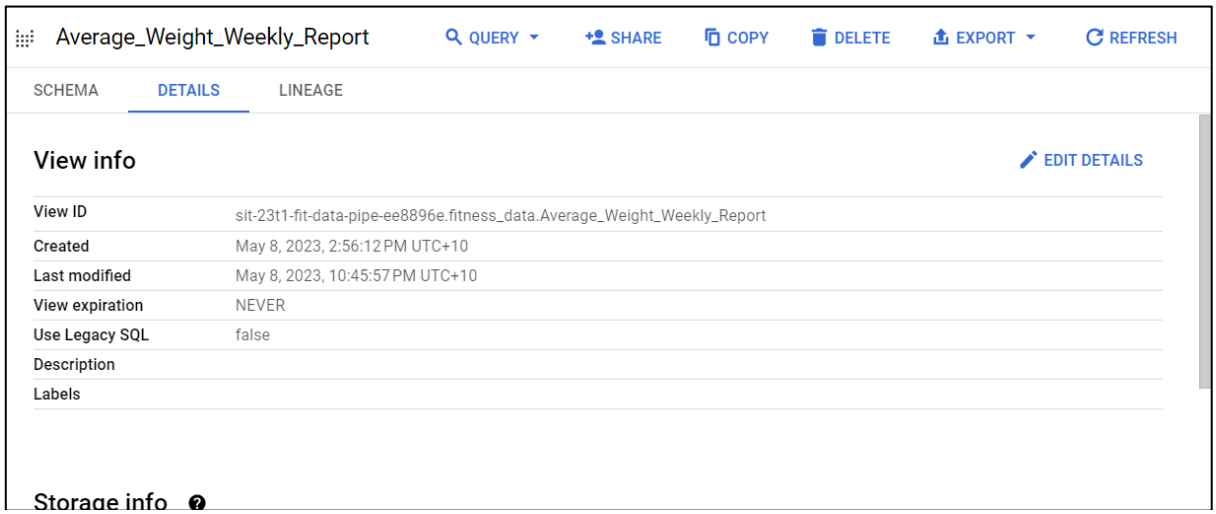


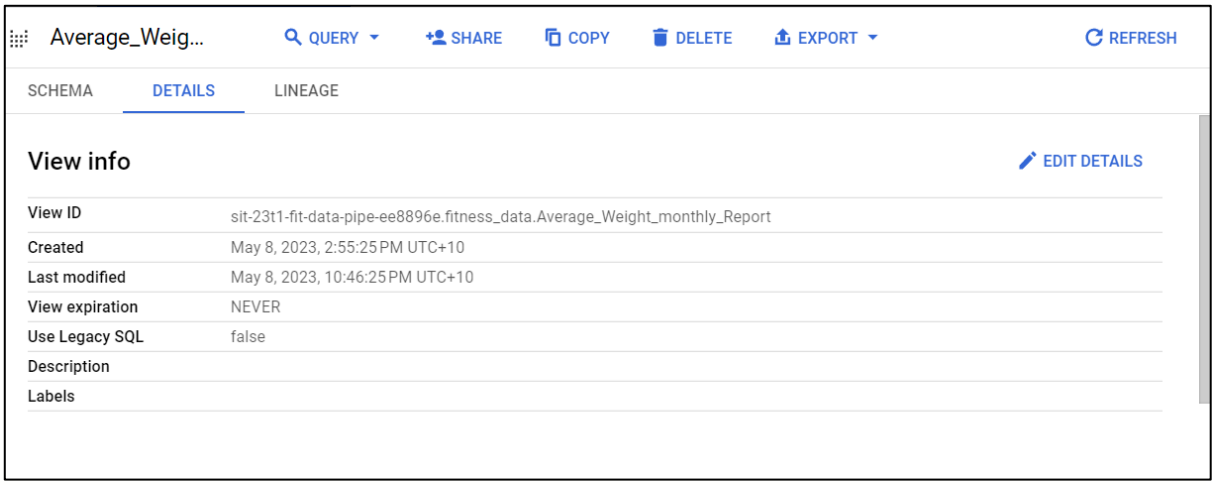
3. Now Give Dataset and View Name (as per choice) and click on save:

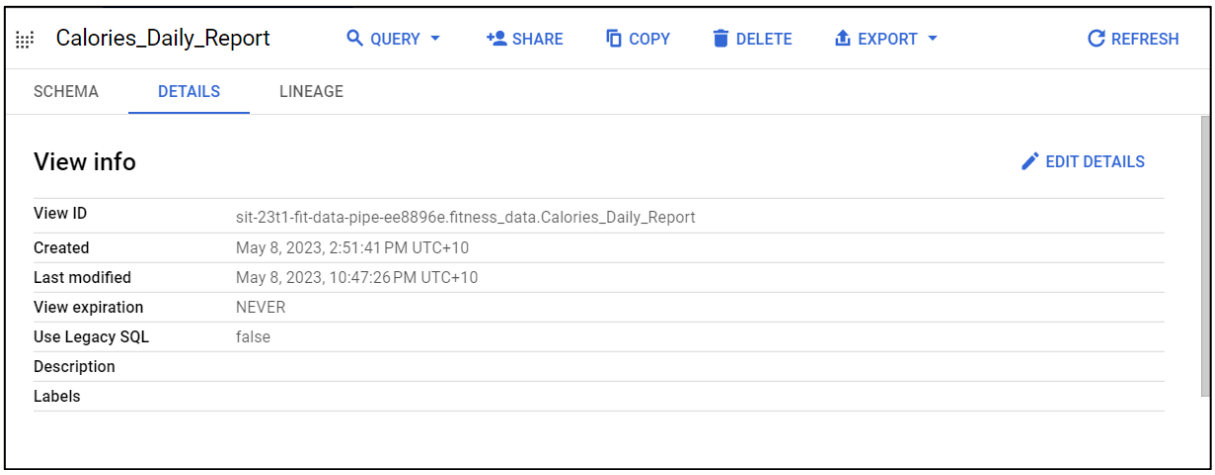


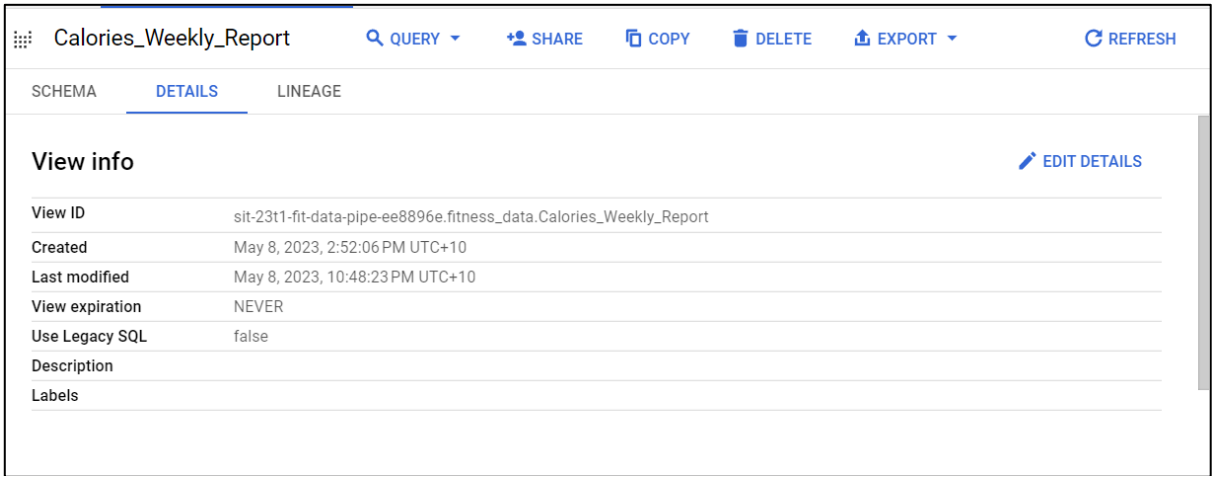
4. Now views are created, click on each view to see their details:

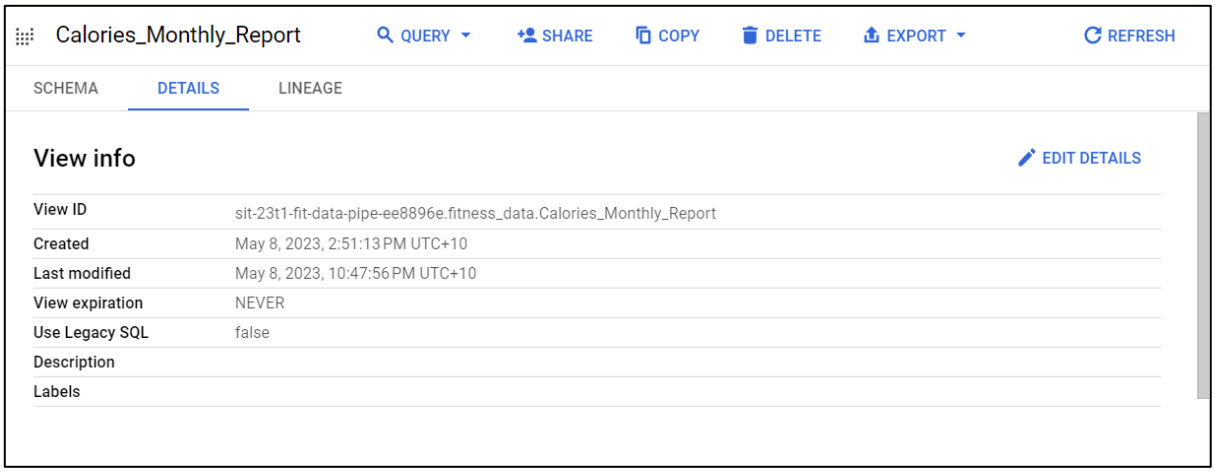


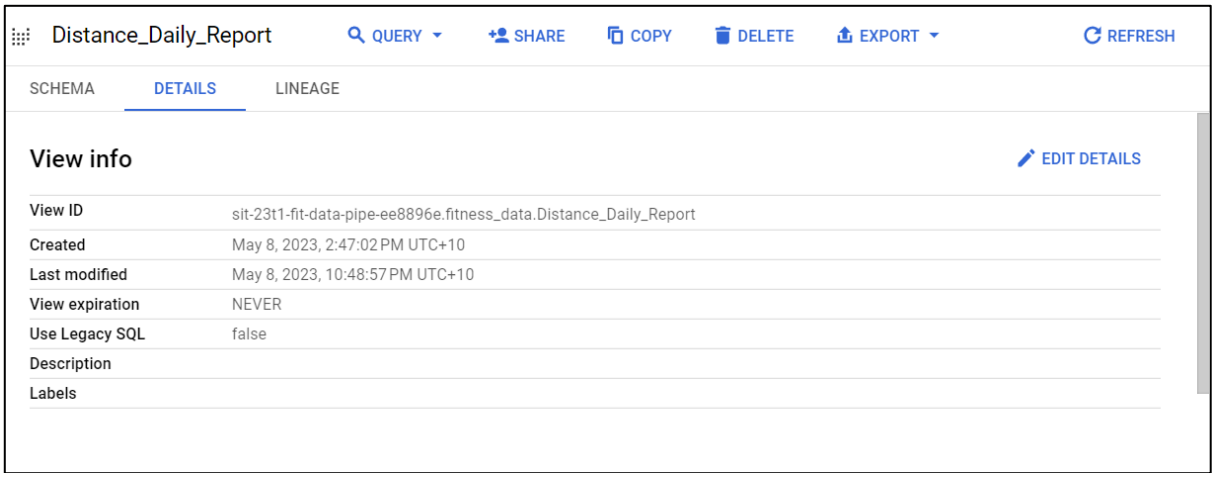


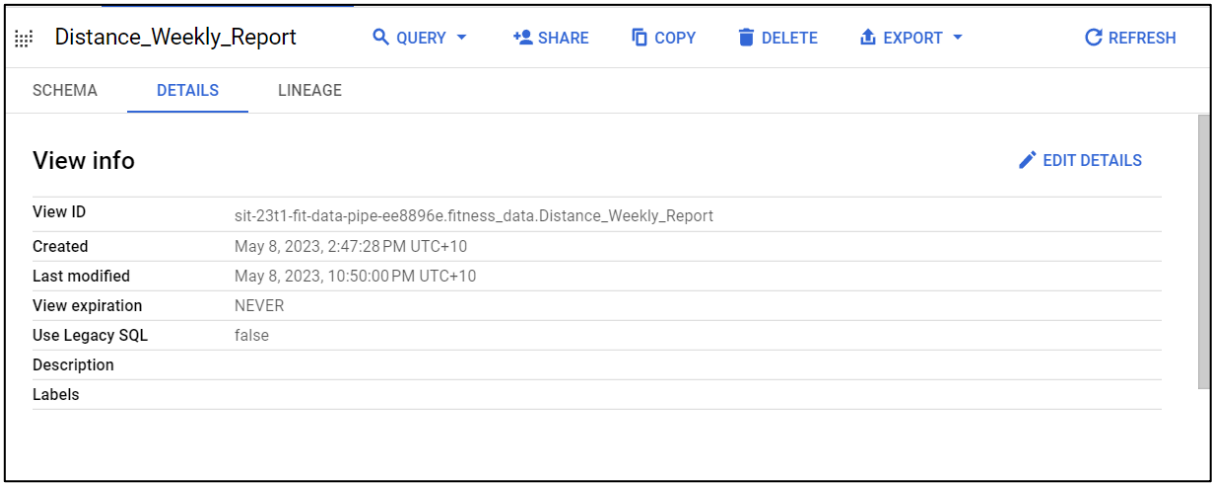

Average_Weig...
QUERY
SHARE
COPY
DELETE
EXPORT
REFRESH
SCHEMA
DETAILS
LINEAGE
View info
EDIT DETAILS
View ID
sit-23t1-fit-data-pipe-ee8896e.fitness_data.Average_Weight_monthly_Report
Created
May 8, 2023, 2:55:25 PM UTC+10
Last modified
May 8, 2023, 10:46:25 PM UTC+10
View expiration
NEVER
Use Legacy SQL
false
Description
Labels

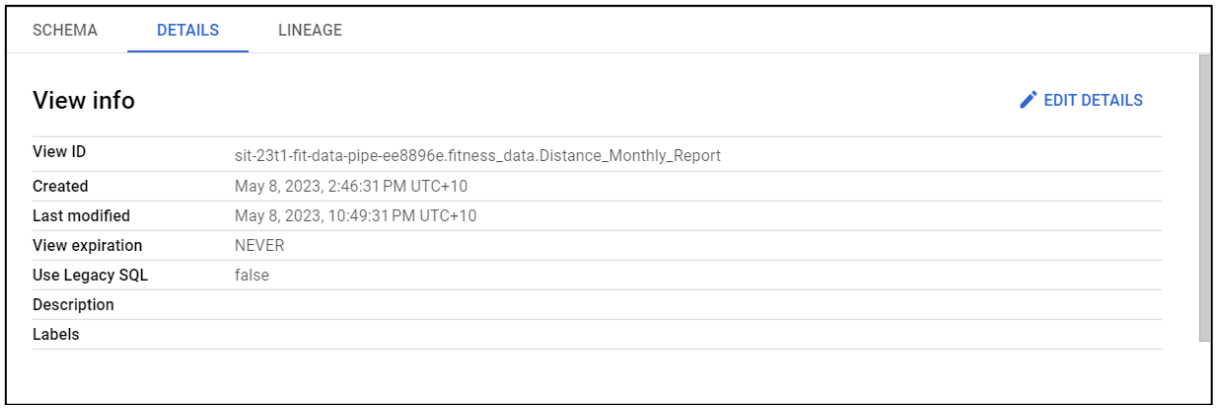

Calories_Daily_Report
QUERY
SHARE
COPY
DELETE
EXPORT
REFRESH
SCHEMA
DETAILS
LINEAGE
View info
EDIT DETAILS
View ID
sit-23t1-fit-data-pipe-ee8896e.fitness_data.Calories_Daily_Report
Created
May 8, 2023, 2:51:41 PM UTC+10
Last modified
May 8, 2023, 10:47:26 PM UTC+10
View expiration
NEVER
Use Legacy SQL
false
Description
Labels


Calories_Weekly_Report
QUERY
SHARE
COPY
DELETE
EXPORT
REFRESH
SCHEMA
DETAILS
LINEAGE
View info
EDIT DETAILS
View ID
sit-23t1-fit-data-pipe-ee8896e.fitness_data.Calories_Weekly_Report
Created
May 8, 2023, 2:52:06 PM UTC+10
Last modified
May 8, 2023, 10:48:23 PM UTC+10
View expiration
NEVER
Use Legacy SQL
false
Description
Labels

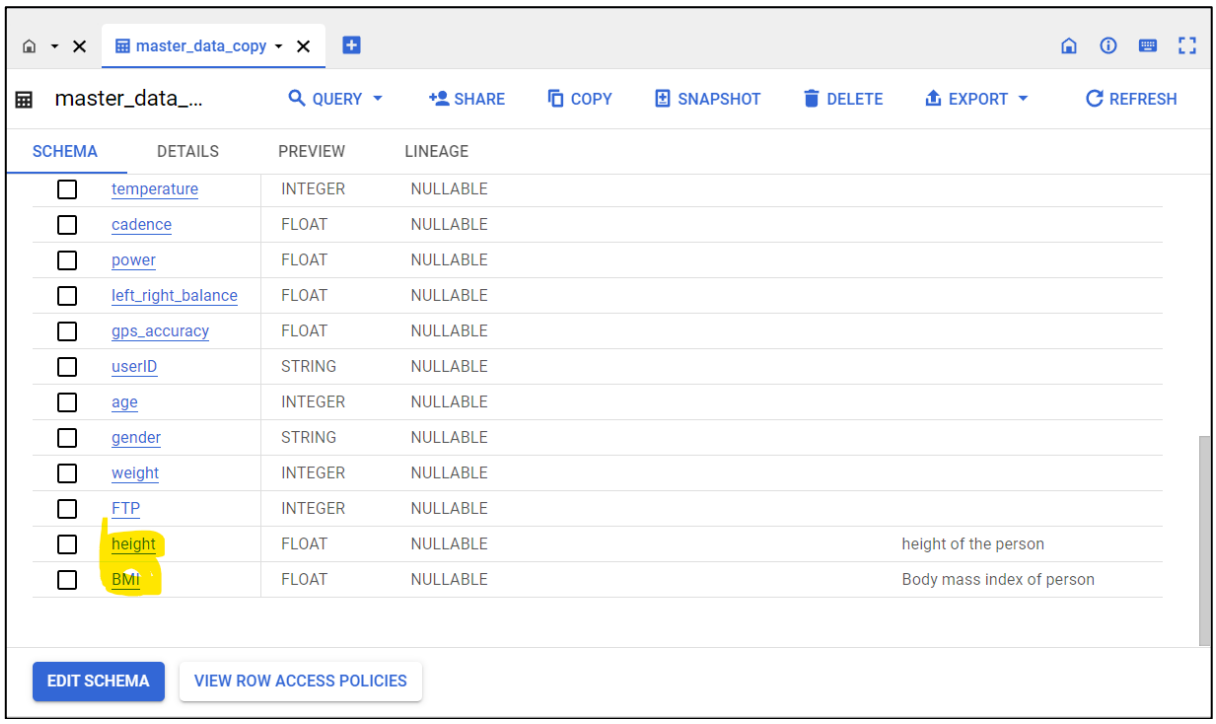

Calories_Monthly_Report
QUERY
SHARE
COPY
DELETE
EXPORT
REFRESH
SCHEMA
DETAILS
LINEAGE
View info
EDIT DETAILS
View ID
sit-23t1-fit-data-pipe-ee8896e.fitness_data.Calories_Monthly_Report
Created
May 8, 2023, 2:51:13 PM UTC+10
Last modified
May 8, 2023, 10:47:56 PM UTC+10
View expiration
NEVER
Use Legacy SQL
false
Description
Labels


Distance_Daily_Report
QUERY
SHARE
COPY
DELETE
EXPORT
REFRESH
SCHEMA
DETAILS
LINEAGE
View info
EDIT DETAILS
View ID
sit-23t1-fit-data-pipe-ee8896e.fitness_data.Distance_Daily_Report
Created
May 8, 2023, 2:47:02 PM UTC+10
Last modified
May 8, 2023, 10:48:57 PM UTC+10
View expiration
NEVER
Use Legacy SQL
false
Description
Labels


Distance_Weekly_Report
QUERY
SHARE
COPY
DELETE
EXPORT
REFRESH
SCHEMA
DETAILS
LINEAGE
View info
EDIT DETAILS
View ID
sit-23t1-fit-data-pipe-ee8896e.fitness_data.Distance_Weekly_Report
Created
May 8, 2023, 2:47:28 PM UTC+10
Last modified
May 8, 2023, 10:50:00 PM UTC+10
View expiration
NEVER
Use Legacy SQL
false
Description
Labels



5. BMI is calculated for all the users in the table. To do this there are 2 columns added to the table Height and BMI.



Height Column is updated with few random values for different users using below query:

```
update `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
SET height = 165
where USERID= 'U1000006'
```

Now BMI value is calculated as follows:

```
update `sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
SET BMI = weight*10000/(height*height)
where userid IN (SELECT distinct USERID from `sit-23t1-fit-data-pipe-
ee8896e.fitness_data.master_data_copy`)
```

6. Queries are created to fetch out views for Monthly and Yearly BMI:

BMI_MONTHLY_REPORT: (BMI of all users for previous month is generated)

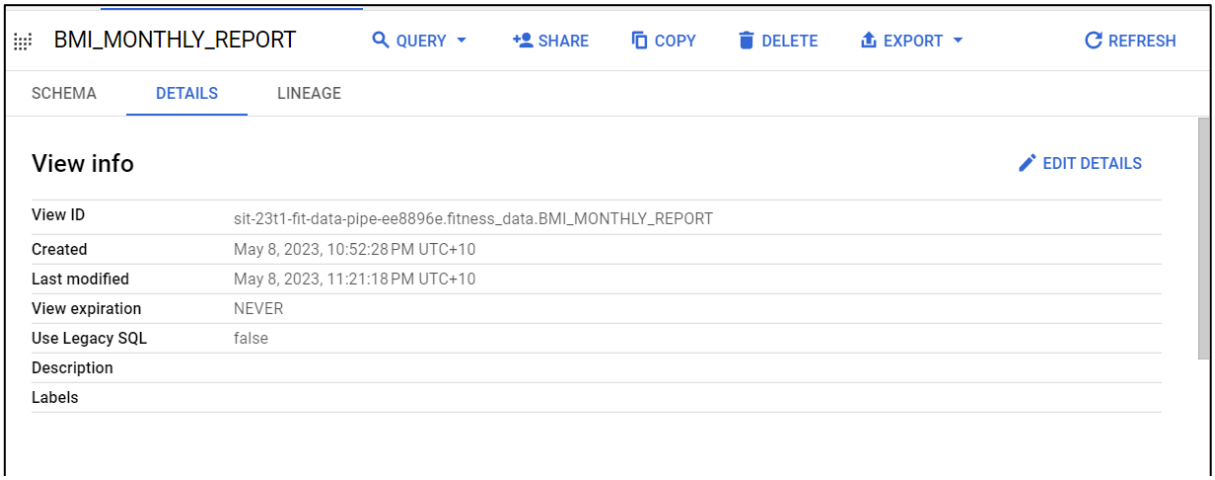
```
select userid,
avg(weight)AS weight,avg(height)as height,avg(BMI)as BMI,
case
when avg(BMI) < 18.5 then "Under weight"
```

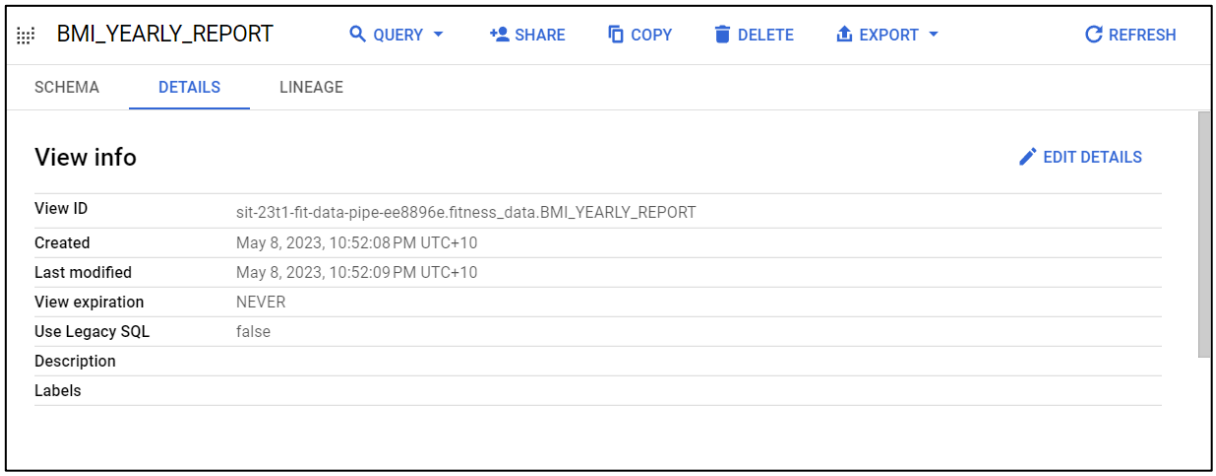
```
when avg(BMI) >=18.5 and avg(BMI) < 25 then "Normal"
when avg(BMI) >=25 then "Obese"
end as Coach,
EXTRACT(month from date_AEST) AS MONTH,
EXTRACT(YEAR from date_AEST) AS YEAR
from
`sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (YEAR from DATE_AEST) = EXTRACT (YEAR from CURRENT_DATE) and
EXTRACT (MONTH from DATE_AEST) = EXTRACT (MONTH from CURRENT_DATE)-1
group by userid,month,year
```

BMI_YEARLY_REPORT: (BMI of users for all months in the year so far is generated)

```
select userid,
avg(weight)AS weight,avg(height)as height,avg(BMI)as BMI,
case
when avg(BMI) < 18.5 then "Under weight"
when avg(BMI) >=18.5 and avg(BMI) < 25 then "Normal"
when avg(BMI) >=25 then "Obese"
end as Coach,
EXTRACT(month from date_AEST) AS MONTH
from
`sit-23t1-fit-data-pipe-ee8896e.fitness_data.master_data_copy`
where EXTRACT (YEAR from DATE_AEST) = EXTRACT (YEAR from CURRENT_DATE) and
EXTRACT (MONTH from DATE_AEST) <= EXTRACT (MONTH from CURRENT_DATE)
group by userid,month
```

7. Views are created using the above queries:

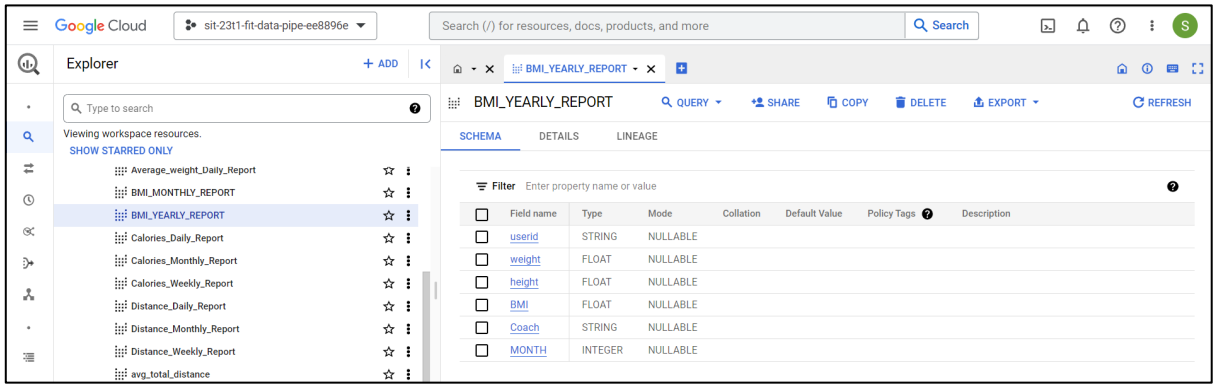




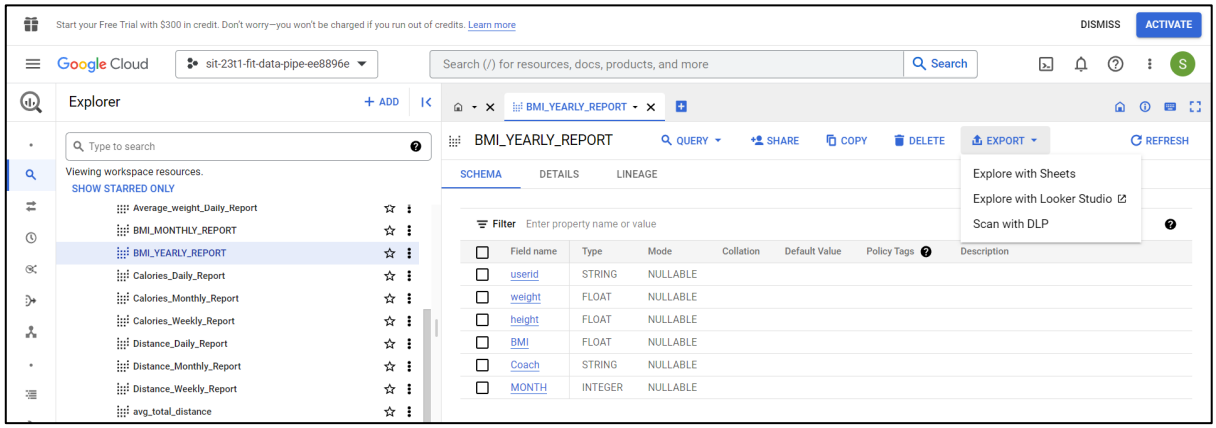
8.Visual Representation of BMI is Developed:

BMI_YEARLY_REPORT:

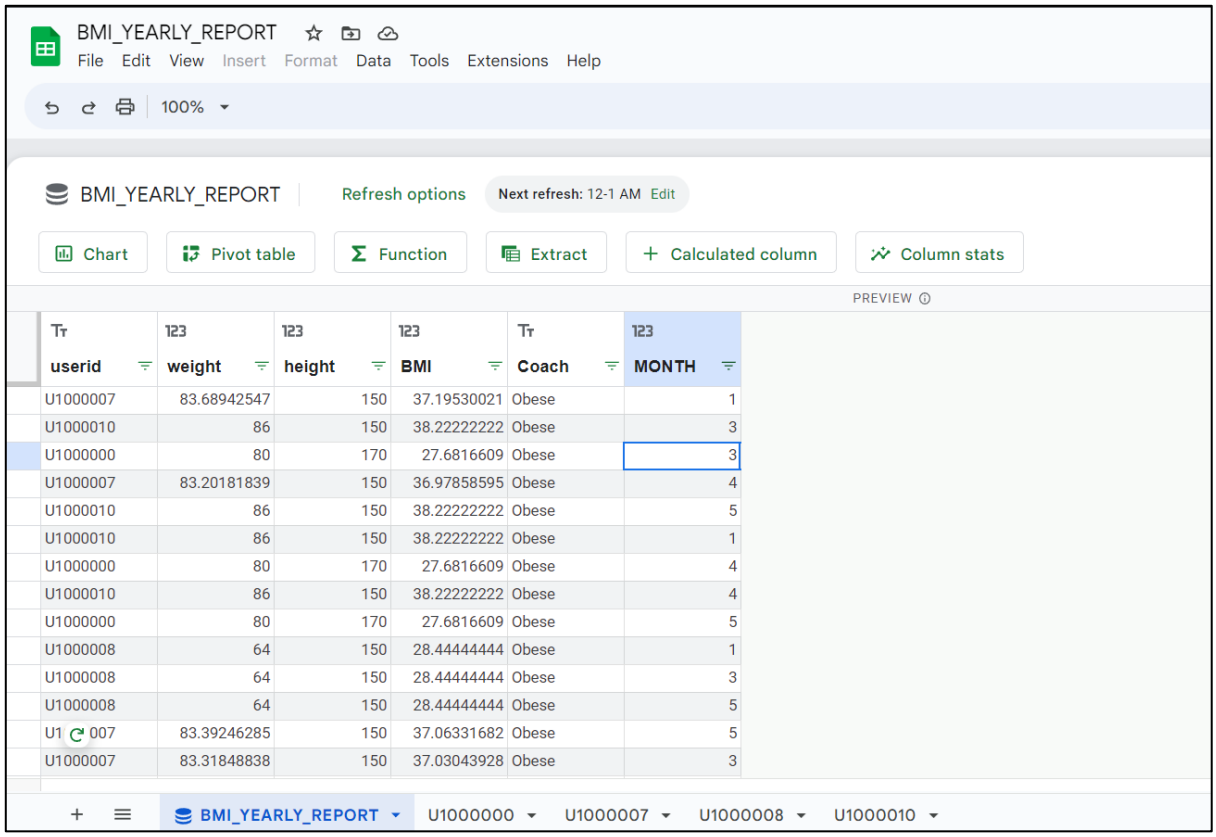
1. Click on BMI_YEARLY_REPORT view in the big query explorer and View opens in the right pane as below:



2. Click on drop down next to export label in the right pane and there will be option explore with sheets as below:

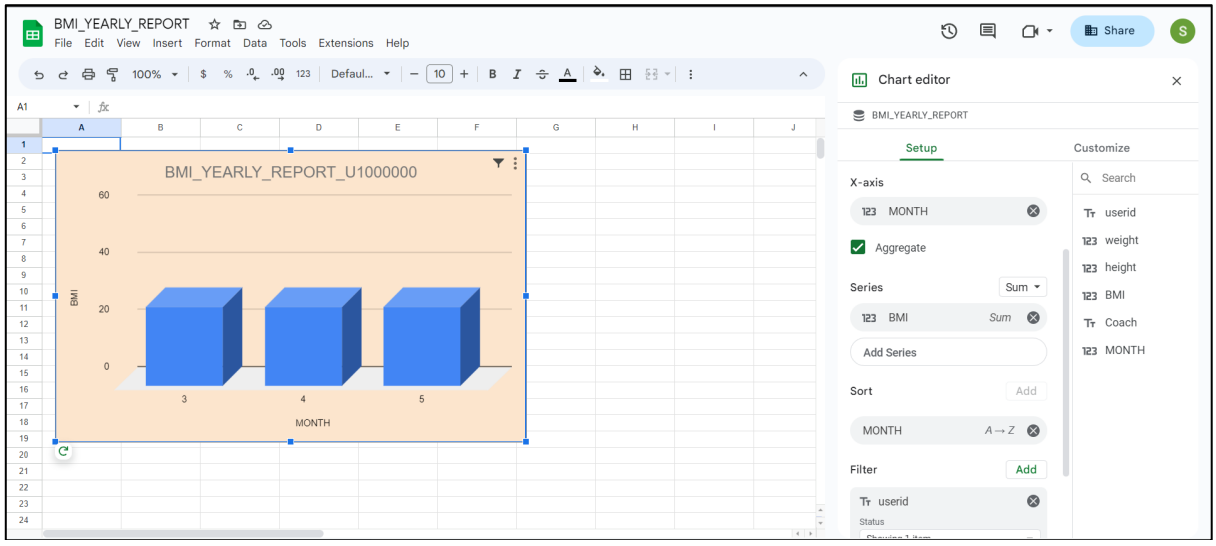


3. Click on Explore with sheets and new sheets opens with preview data from view result. Name that sheet to BMI_YEARLY_REPORT:

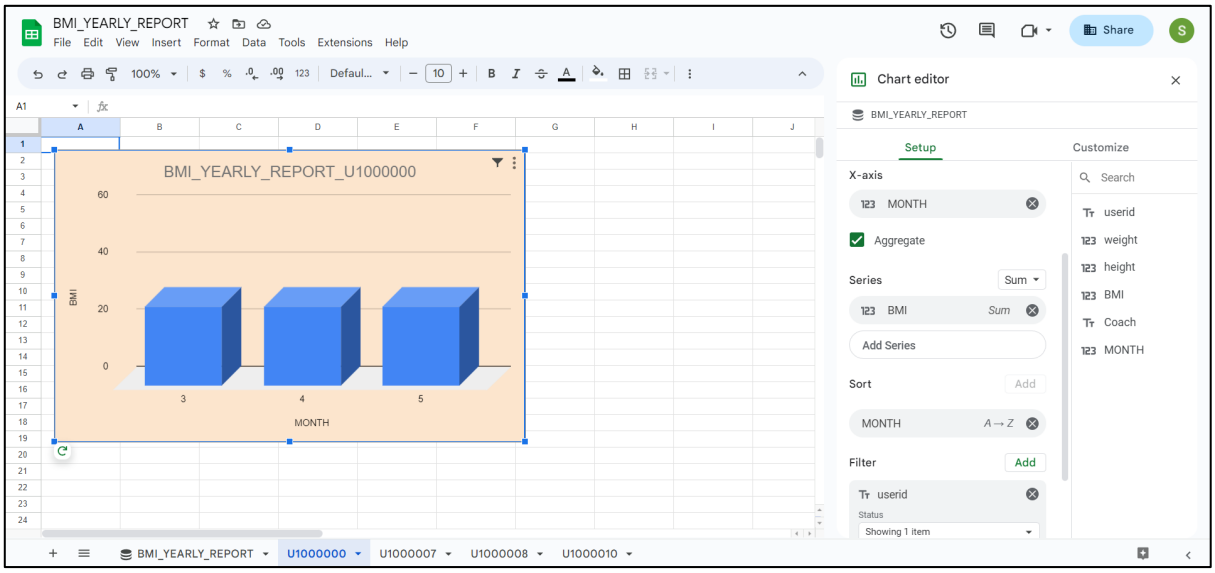


userid	weight	height	BMI	Coach	MONTH
U1000007	83.68942547	150	37.19530021	Obese	1
U1000010	86	150	38.22222222	Obese	3
U1000000	80	170	27.6816609	Obese	3
U1000007	83.20181839	150	36.97858595	Obese	4
U1000010	86	150	38.22222222	Obese	5
U1000010	86	150	38.22222222	Obese	1
U1000000	80	170	27.6816609	Obese	4
U1000010	86	150	38.22222222	Obese	4
U1000000	80	170	27.6816609	Obese	5
U1000008	64	150	28.44444444	Obese	1
U1000008	64	150	28.44444444	Obese	3
U1000008	64	150	28.44444444	Obese	5
U1 007	83.39246285	150	37.06331682	Obese	5
U1000007	83.31848838	150	37.03043928	Obese	3

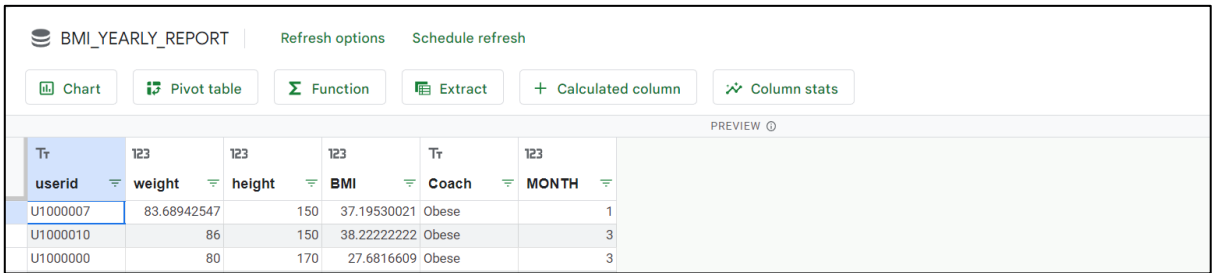
4. Now click on Chart Label to create charts and new sheet opens next to it:



5. Above Report is created by selecting userid in x-axis and BMI in series and sorted by MONTH and filtered by userid columns.
6. Select one user id from filter to fetch BMI of user for all months in the year.
7. Explore the customize options to style the dashboard.
8. Sheets for different users are created as below by selecting each user in each sheet:

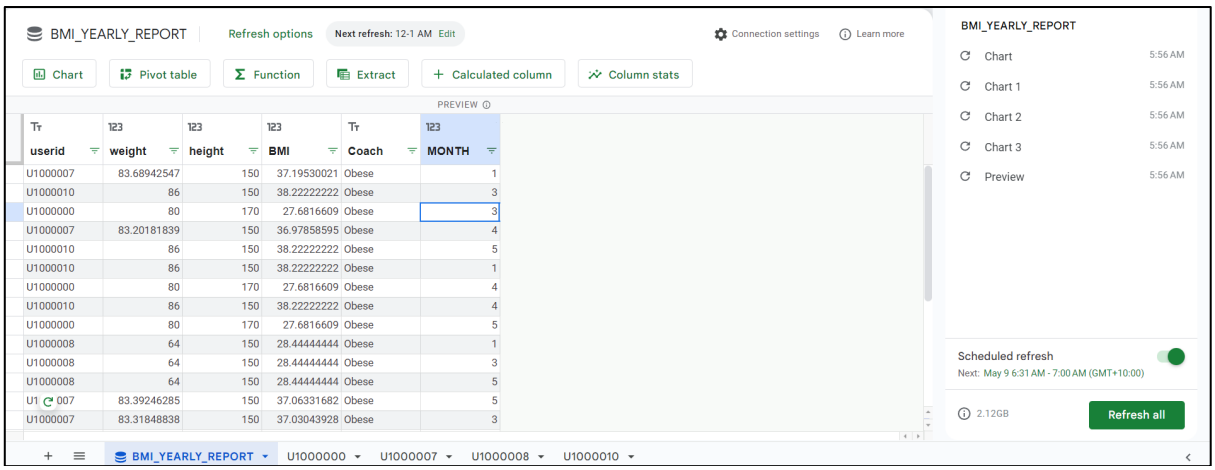


9. In the first sheet where data preview is present click on Schedule Refresh option on top to schedule the refresh of reports:



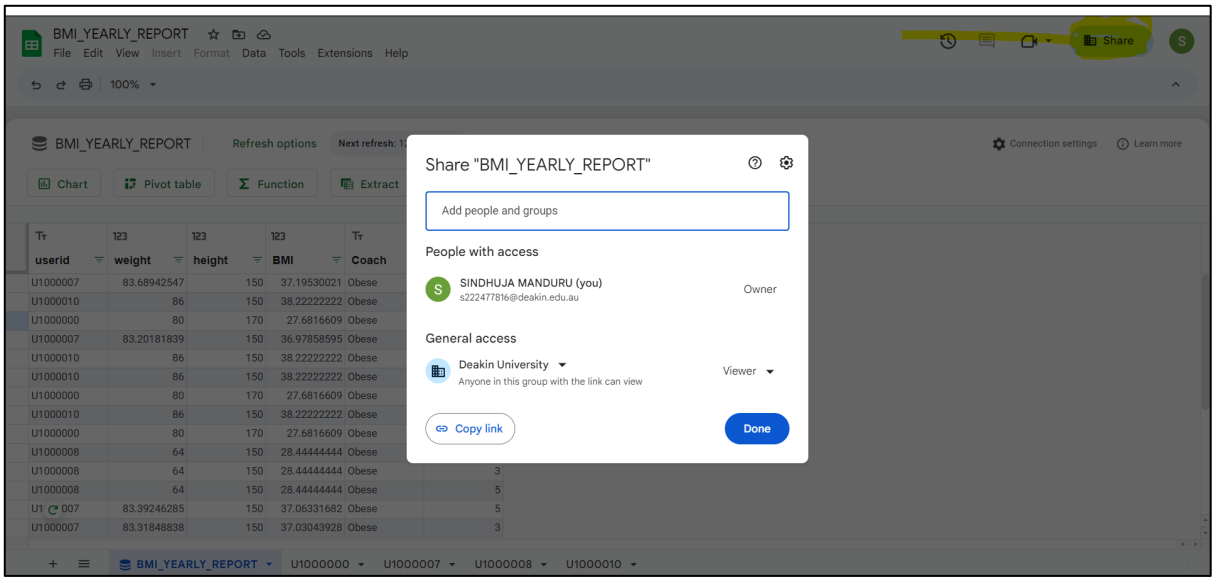
userid	weight	height	BMI	Coach	MONTH
U1000007	83.68942547	150	37.19530021	Obese	1
U1000010	86	150	38.22222222	Obese	3
U1000000	80	170	27.6816609	Obese	3

10. All the sheets in the report can be scheduled for refresh:



userid	weight	height	BMI	Coach	MONTH
U1000007	83.68942547	150	37.19530021	Obese	1
U1000010	86	150	38.22222222	Obese	3
U1000000	80	170	27.6816609	Obese	3
U1000007	83.20181839	150	36.97858595	Obese	4
U1000010	86	150	38.22222222	Obese	5
U1000010	86	150	38.22222222	Obese	1
U1000000	80	170	27.6816609	Obese	4
U1000010	86	150	38.22222222	Obese	4
U1000000	80	170	27.6816609	Obese	5
U1000008	64	150	28.44444444	Obese	1
U1000008	64	150	28.44444444	Obese	3
U1000008	64	150	28.44444444	Obese	5
U1000007	83.39246285	150	37.06331682	Obese	5
U1000007	83.31848838	150	37.03043928	Obese	3

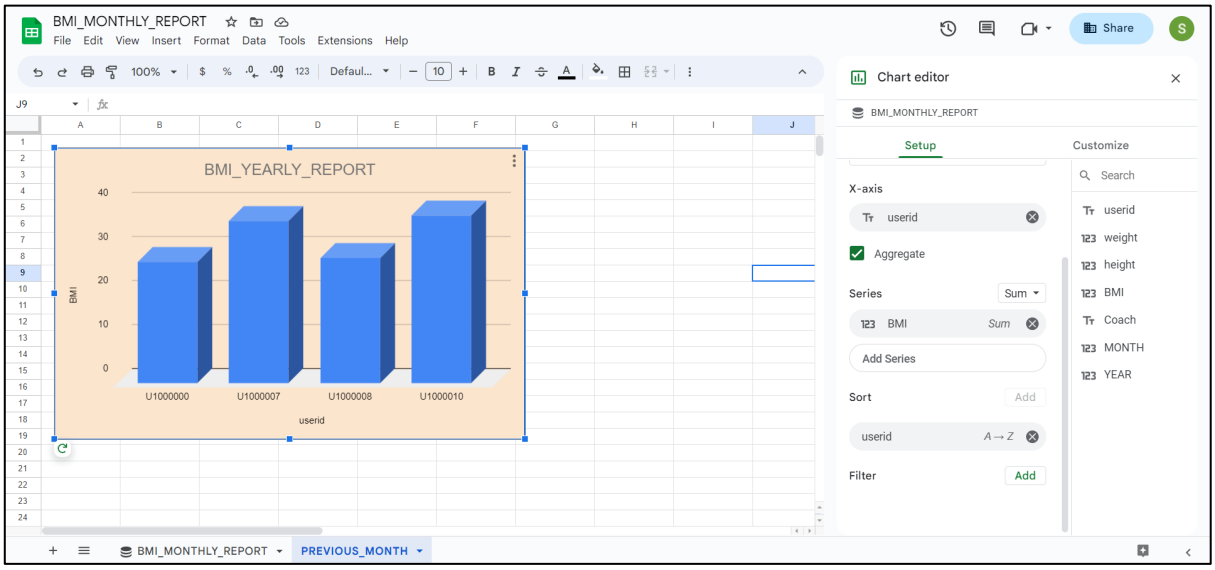
11. Reports can be published to audience by clicking on share on top right and giving access to Deakin group only in view of security:



12. Now this report can be accessed using URL: https://docs.google.com/spreadsheets/d/1xOI-TcMc_0_mIxODiJQKkvOOvC21zWWOXIEvgsvVZUo/edit?usp=sharing

BMI_MONTHLY_REPORT:

BMI_MONTHLY_REPORT is also visualised using the above steps. But here data of all users for previous month is shown.so only one chart is created with userid in x-axis and BMI in series and sort by user id:



This report can be accessed using: https://docs.google.com/spreadsheets/d/1Cg5NDlH2vKCdIj6nt2cYlkssPWfPipauB3Zjih1osLA/edit?usp=sharing