Sprint Report

PORTFOLIO TASK 2/3/4

Unit code: EAT40005
Unit Name: Engineering Project A
Submission date:

Student Name	Student Id	Statement of contribution to the report
Dale Bent	102567413	Equal contribution
Mohammed Barsat Zulkarnine	103801626	Equal contribution
Dang Khoa Le	103844421	Equal contribution
Sadman Tariq	103798780	Equal contribution

ACKNOWLEDGMENT OF COUNTRY

We, Dale Bent, Mohammed Barsat Zulkarnine, Dang Khoa Le, and Sadman Tariq, acknowledge the Traditional Custodians of the lands on which we lived and worked while completing this project. We pay our respects to their Elders past and present and extend that respect to all Aboriginal and Torres Strait Islander peoples.

We recognise their enduring connection to land, waters, and culture, and we honour their rich traditions and contributions to our shared community.

Acknowledgment of Country....1

1. Sprint plan 2

2. Sprint progress (Evidence)....8

3. Sprint Demonstration....19

I. Week 5 Demonstration: 19

II. Week 6 Demonstration: 20

III. Week 7 Demonstration: 20

4. Sprint review 21

I. What Was Demonstrated to the Client 21

iI. Feedback Client Provided 21

III. CRITICAL ANALYSIS OF PROGRESS AND FEEDBACK 22

5. Retrospect (Critical review of the process) 24

6. Lessons learned (Critical review of SPRINT one experience and future plan)....25

1. SPRINT PLAN

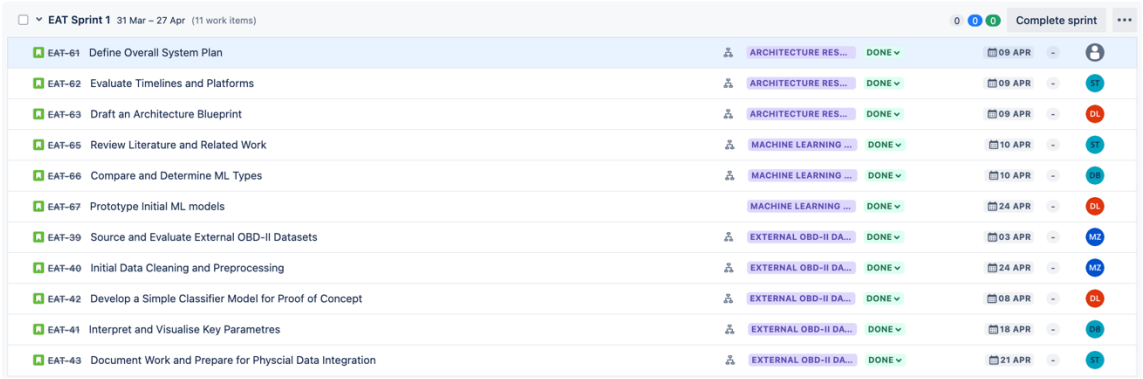
The goal for Sprint 1 was to transition from early project planning into actionable research and technical groundwork, following the Gantt chart created in the first two weeks. Initially, we worked under the assumption that the client would provide us with vehicle data. Based on this, our planned deliverables included background research, exploration of external OBD II datasets, and early experimentation with machine learning (ML) models using publicly available data to accelerate development.

As Sprint 1 progressed, we realised that client supplied data would not be immediately available. Consequently, our team adapted the plan to focus on collecting our own data using OBD-II devices. This adjustment shifted our Sprint 1 goals to include understanding OBD II data output, researching how OBD-II data is used for diagnostics, and exploring parsing systems to convert raw data into a readable format.

Other planned outcomes for Sprint 1 included defining a high-level architecture for the predictive maintenance system, completing a literature review of similar projects, investigating ethical considerations for data collection and consent (especially concerning GPS and vehicle data), and setting a clear project direction. Regular internal meetings (two stand ups per week) and weekly client and supervisor meetings were scheduled to track progress and coordinate team efforts.

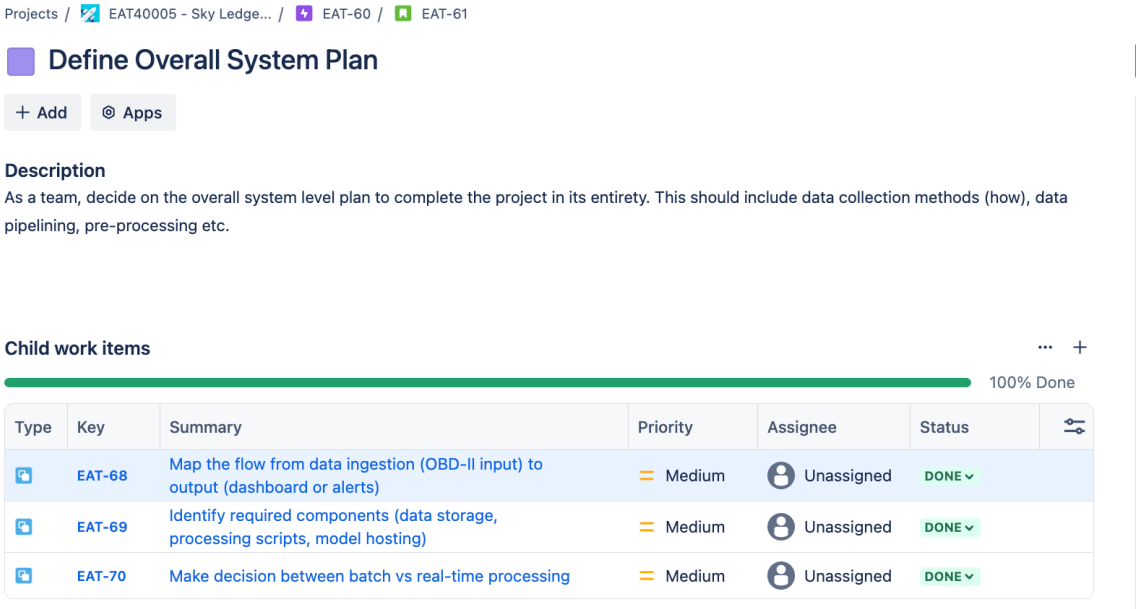
While we initially aimed to begin data collection during Sprint 1, the OBD-II device acquisition was delayed. However, we successfully adapted by shifting our focus to system preparation and deep technical research, ensuring we would be ready to start collecting and processing real world data early in Sprint 2.

Overall, Sprint 1 was completed successfully according to our updated plan, providing a strong foundation for the next phase of the project.



Assignees on the right (DB – Dale Bent, DL – Dang Khoa Le, MS - Mohammed Barsat Zulkarnine, ST – Sadman Tariq)

TASK 1



Projects / EAT40005 - Sky Ledge... / EAT-60 / EAT-61

Define Overall System Plan

+ Add Apps

Description

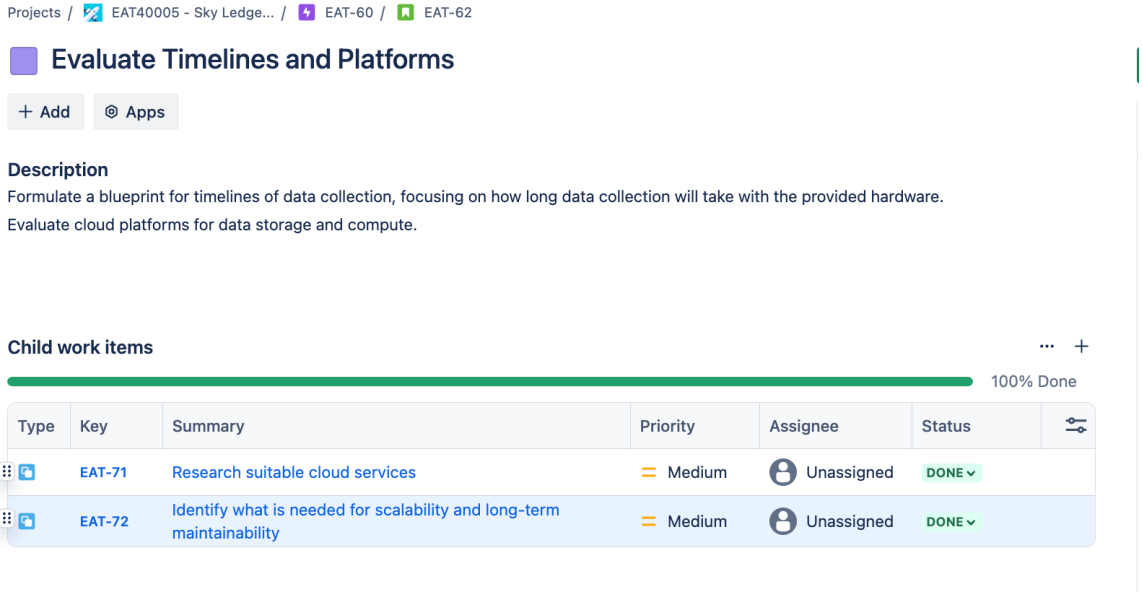
As a team, decide on the overall system level plan to complete the project in its entirety. This should include data collection methods (how), data pipelining, pre-processing etc.

Child work items

100% Done

Type	Key	Summary	Priority	Assignee	Status
	EAT-68	Map the flow from data ingestion (OBD-II input) to output (dashboard or alerts)	Medium	Unassigned	DONE
	EAT-69	Identify required components (data storage, processing scripts, model hosting)	Medium	Unassigned	DONE
	EAT-70	Make decision between batch vs real-time processing	Medium	Unassigned	DONE

TASK 2



Projects / EAT40005 - Sky Ledge... / EAT-60 / EAT-62

Evaluate Timelines and Platforms

+ Add Apps

Description

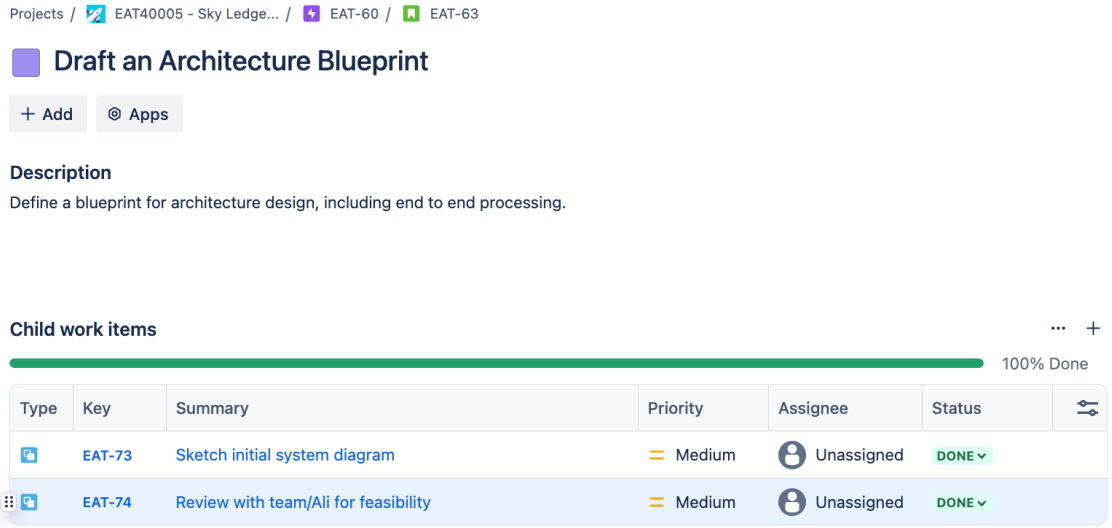
Formulate a blueprint for timelines of data collection, focusing on how long data collection will take with the provided hardware.
Evaluate cloud platforms for data storage and compute.

Child work items

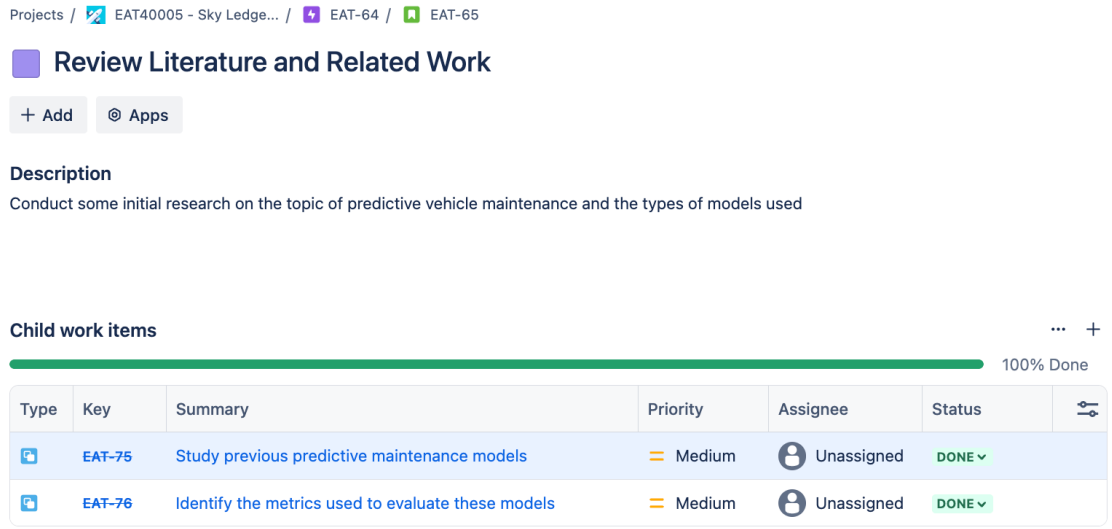
100% Done

Type	Key	Summary	Priority	Assignee	Status
	EAT-71	Research suitable cloud services	Medium	Unassigned	DONE
	EAT-72	Identify what is needed for scalability and long-term maintainability	Medium	Unassigned	DONE

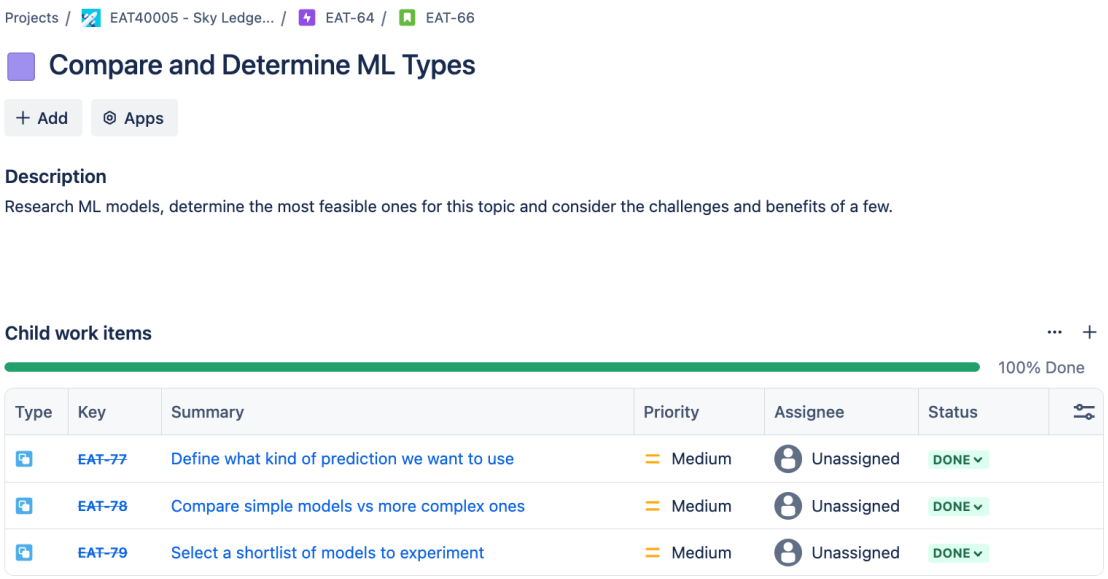
TASK 3


Projects / EAT40005 - Sky Ledge... / EAT-60 / EAT-63
Draft an Architecture Blueprint
+ Add
Apps
Description
Define a blueprint for architecture design, including end to end processing.
Child work items
100% Done
Type Key Summary Priority Assignee Status
EAT-73 Sketch initial system diagram Medium Unassigned DONE
EAT-74 Review with team/Ali for feasibility Medium Unassigned DONE

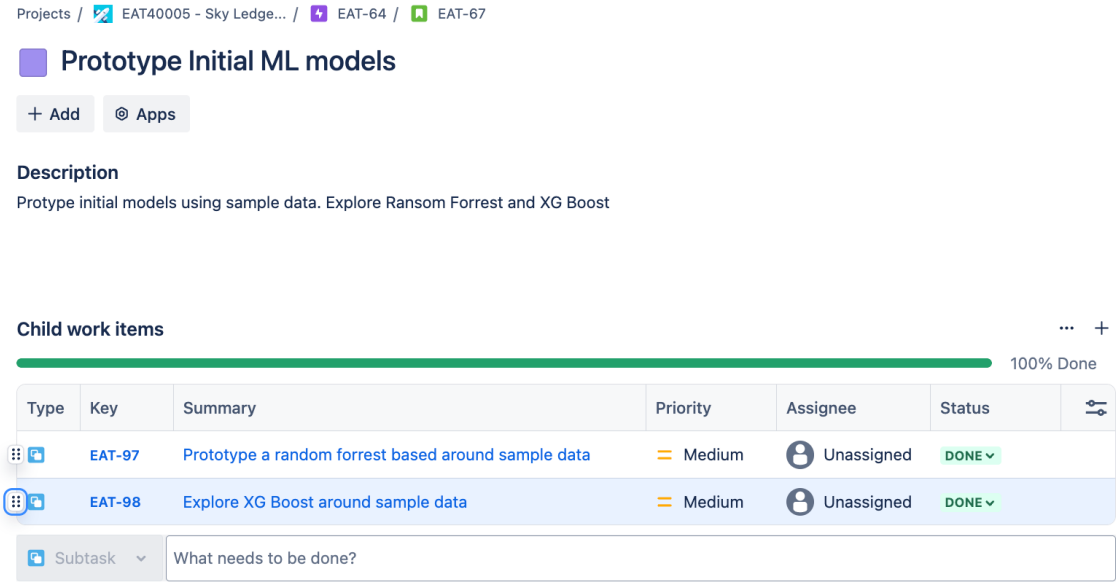
TASK 4


Projects / EAT40005 - Sky Ledge... / EAT-64 / EAT-65
Review Literature and Related Work
+ Add
Apps
Description
Conduct some initial research on the topic of predictive vehicle maintenance and the types of models used
Child work items
100% Done
Type Key Summary Priority Assignee Status
EAT-75 Study previous predictive maintenance models Medium Unassigned DONE
EAT-76 Identify the metrics used to evaluate these models Medium Unassigned DONE

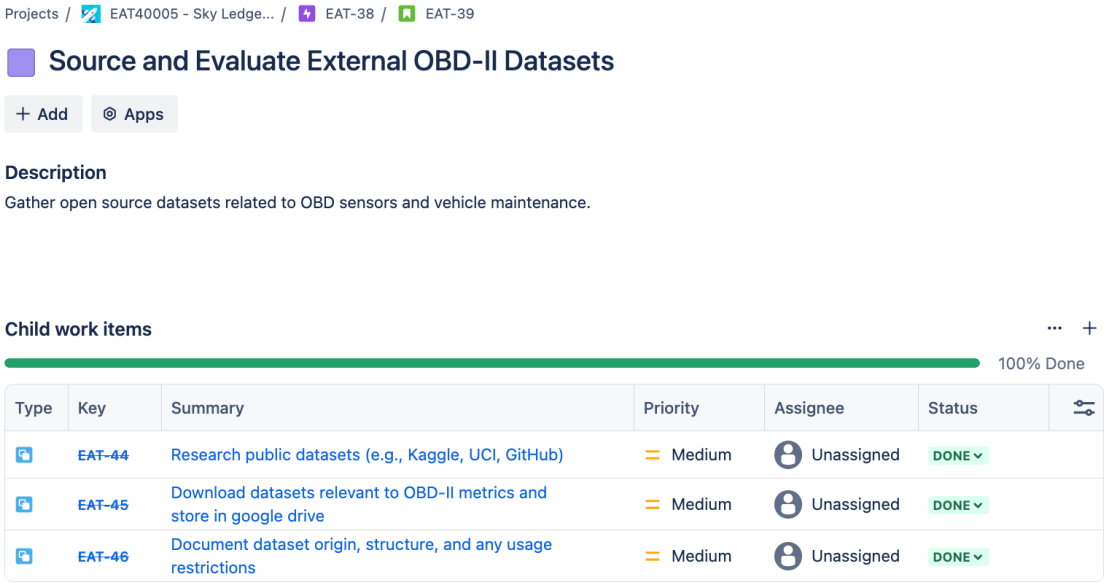
TASK 5


Projects / EAT40005 - Sky Ledge... / EAT-64 / EAT-66
Compare and Determine ML Types
+ Add
Apps
Description
Research ML models, determine the most feasible ones for this topic and consider the challenges and benefits of a few.
Child work items
100% Done
Type Key Summary Priority Assignee Status
EAT-77 Define what kind of prediction we want to use Medium Unassigned DONE
EAT-78 Compare simple models vs more complex ones Medium Unassigned DONE
EAT-79 Select a shortlist of models to experiment Medium Unassigned DONE

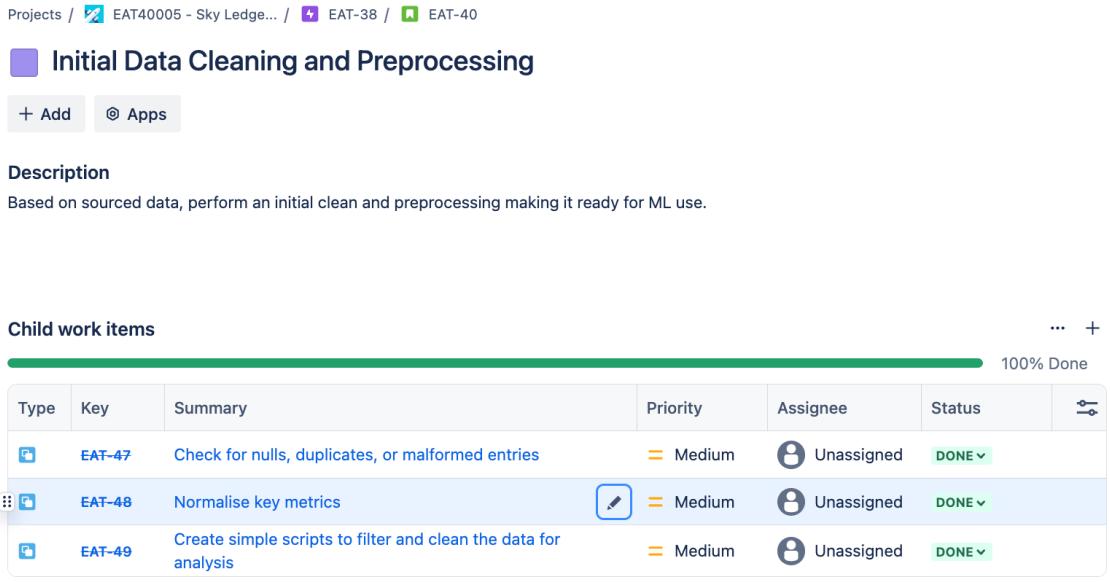
TASK 6


Projects / EAT40005 - Sky Ledge... / EAT-64 / EAT-67
Prototype Initial ML models
+ Add
Apps
Description
Protype initial models using sample data. Explore Ransom Forrest and XG Boost
Child work items
100% Done
Type Key Summary Priority Assignee Status
EAT-97 Prototype a random forrest based around sample data Medium Unassigned DONE
EAT-98 Explore XG Boost around sample data Medium Unassigned DONE
Subtask
What needs to be done?

TASK 7


Projects / EAT40005 - Sky Ledge... / EAT-38 / EAT-39
Source and Evaluate External OBD-II Datasets
+ Add
Apps
Description
Gather open source datasets related to OBD sensors and vehicle maintenance.
Child work items
100% Done
Type Key Summary Priority Assignee Status
EAT-44 Research public datasets (e.g., Kaggle, UCI, GitHub) Medium Unassigned DONE
EAT-45 Download datasets relevant to OBD-II metrics and store in google drive Medium Unassigned DONE
EAT-46 Document dataset origin, structure, and any usage restrictions Medium Unassigned DONE

TASK 8


Projects / EAT40005 - Sky Ledge... / EAT-38 / EAT-40
Initial Data Cleaning and Preprocessing
+ Add
Apps
Description
Based on sourced data, perform an initial clean and preprocessing making it ready for ML use.
Child work items
100% Done
Type Key Summary Priority Assignee Status
EAT-47 Check for nulls, duplicates, or malformed entries Medium Unassigned DONE
EAT-48 Normalise key metrics Medium Unassigned DONE
EAT-49 Create simple scripts to filter and clean the data for analysis Medium Unassigned DONE

TASK 9

Projects / EAT40005 - Sky Ledge... / EAT-38 / EAT-42

Develop a Simple Classifier Model for Proof of Concept

+ Add Apps

Description

Using Google Colab, we want to develop a proof of concept model based upon sourced datasets to help classification. The aim is to increase our knowledge on the topic of vehicle maintenance and the data we will be receiving.

Child work items

100% Done

Type	Key	Summary	Priority	Assignee	Status
	EAT-53	Define a basic binary or multi-class classification goal	Medium	Unassigned	DONE
	EAT-54	Engineer simple features	Medium	Unassigned	DONE
	EAT-55	Train a quick model	Medium	Unassigned	DONE
	EAT-56	Test and validate on a hold-out set or through cross-validation	Medium	Unassigned	DONE

TASK 10

Projects / EAT40005 - Sky Ledge... / EAT-38 / EAT-41

Interpret and Visualise Key Parametres

+ Add Apps

Description

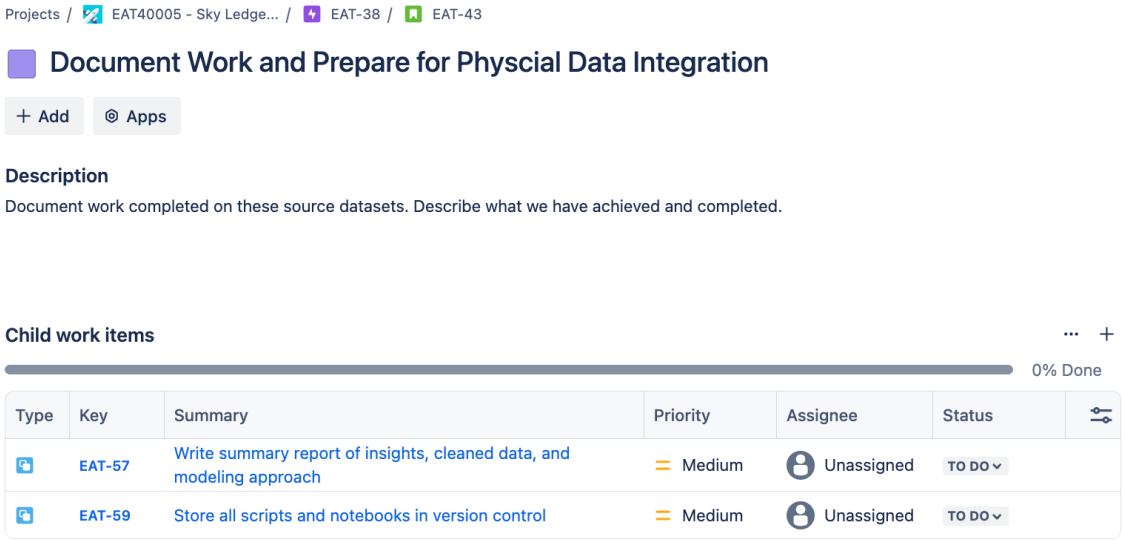
Using external datasets, visualise the data and key variables that contribute to vehicle maintenance

Child work items

100% Done

Type	Key	Summary	Priority	Assignee	Status
	EAT-50	Decode available fields into readable metrics	Medium	Unassigned	DONE
	EAT-51	Create quick visualizations	Medium	Unassigned	DONE
	EAT-52	Identify any anomalies, interesting trends, or gaps	Medium	Unassigned	DONE

TASK 11



2. SPRINT PROGRESS (EVIDENCE)

During the early part of the sprint (Weeks 3-4), we spent a lot of time looking at online datasets similar to what we hoped to get from the client. One team member started testing basic ML models with this data, while others focused on reading up and researching predictive maintenance systems. We also began creating the system layout and mapped out how the different parts would work together - data collection, cleaning, analysis, and prediction. This work matched our Gantt chart and seemed to be going well.

By Week 5, we realised we wouldn't get the client's data as soon as we thought. This meant we had to change our plans. After talking with our supervisor, we decided to collect our own data using an OBD-II scanner. We researched how the device works, how to get and read the data, and started planning how to clean it up. We shifted from jumping into ML right away to really understanding the data first. This step back was important for building a solid foundation.

Even with this change in direction, we kept up with weekly updates to both the client and supervisor. Our presentations covered what we'd accomplished, problems we ran into, and how our plans were changing. We finished an early version of the system design and agreed to take a slower but more thorough approach - starting with handling and studying the data manually before using automation or ML models.

Looking at our Jira board, we've completed all the tasks we planned for this sprint. The evidence for this is below, with details linking to each specific task.

TASK 1: Team Contribution

The development and completion of the system planning occurred on the 7th of April, in line with our submission for the project plan submitted on the same day. This plan entails all the subtasks within this task as well.

TASK 2 - Evaluate Timeline and Platforms: Sadman Tariq

The below Gantt Chart describes our initial plan for both sprint 1 and 2, but as clearly demonstrated above, we have slightly veered off plan and added a few more objectives.

2	**Research**			
2.1	Solution and AI Model Research	Dang + Dale	2 Weeks	
2.2	Project, Model, and Solution Constraints Investigation	Dang + Sadman	2 weeks	
2.3	Architecture Research	Sadman + Barsat	2 Weeks	
2.4	External Dataset Research	Dang + Dale	2 Weeks	
2.5	Ethical Data Collection and Consent Research (OBD-II + GPS)	Barsat	2 Weeks	
3	**Data collection**			
3.1	Physical Data Collection with OBD-II Sensor Device	All	4 weeks	
3.2	Data Preprocessing and Cleaning	Sadman	3 Weeks	
3.3	Exploratory Data Analysis (EDA)	Dang	3 Weeks	
3.4	Logging and Backup Strategy for OBD-II Data	Dale	1 Week	

Figure 1. Initial plan – Gantt chart

Based on this, we are still missing the physical data collection, but this objective has been moved to sprint 2.

The below links describe our platforms of choice for the project:

- **Communication:** Discord (general), Microsoft Team (weekly and occasion meetings)
- **Task organisation and sprint planning:** Jira (see above screenshots)
- **Cloud Solution:** Google Colab + Google Drive

Week 5 Colab session:
https://colab.research.google.com/drive/1UQpFgeMFV-7ARVjYEEcSADCKMGacBcJv
Week 7 Colab session:
https://colab.research.google.com/drive/1AtO5rNQdNKPHWwNsbAbPgQVXv9en1F3s

TASK 3 – Draft and Architecture Blueprint: Dang Khoa Le

The project's Architectural blueprint was conducted upon research on existing tools and integrated solutions:

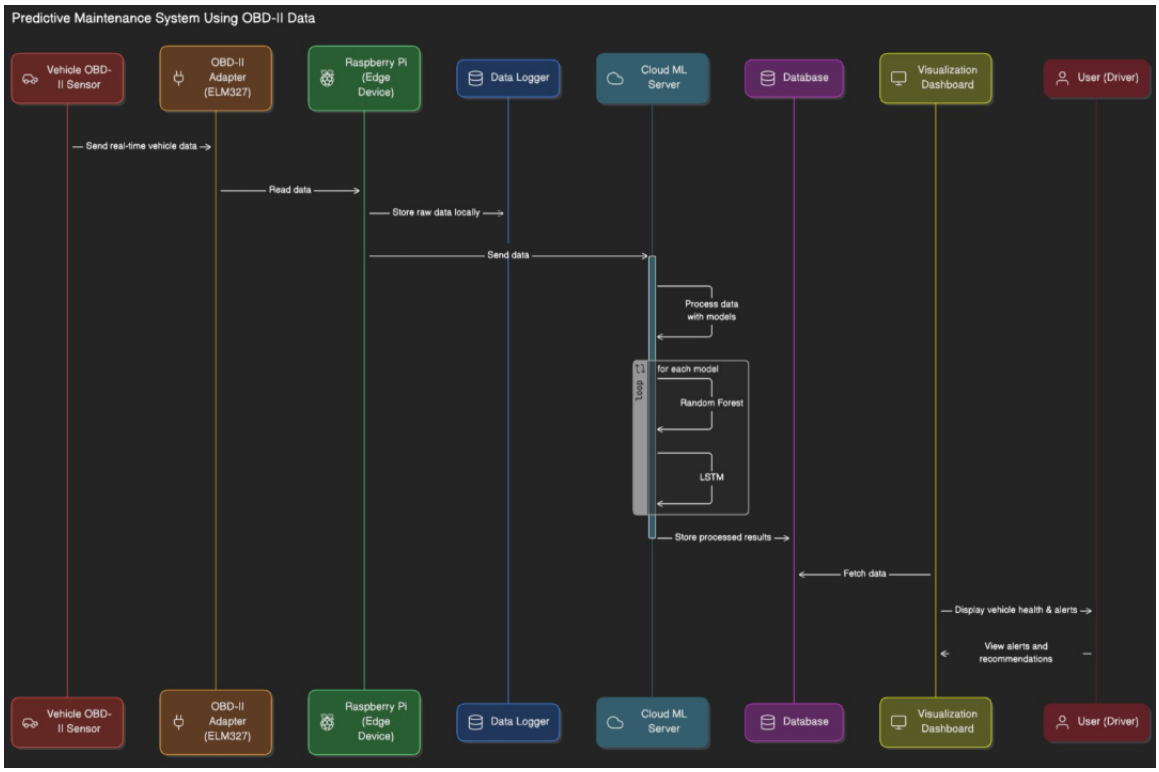


Figure 2. Predictive Maintenance Project - System Architecture

Which breakdown into:

- Vehicle OBD-II sensor sending real-time data to the OBD-II adapter (ELM327).
- Raspberry Pi (Edge Device) reads data, stores it locally, and forwards it to the Cloud-ML Server.
- In cloud, ML models (Random Forest and LSTM) process the data to detect issues and predict maintenance needs.
- Results saved in the Database, fetched by the Dashboard, and presented to the User with health status and maintenance alerts.

TASK 4 – Review literature and related works: Sadman Tariq

At the start of the sprint, a thorough literature review was undertaken, focused on predictive maintenance using OBD-II vehicle data. The research spanned multiple domains including:

- Predictive maintenance strategies in automotive systems
- Machine Learning (ML) techniques commonly used for fault detection (e.g., Random Forest, XGBoost, LSTM, Autoencoders)
- Challenges in raw OBD-II data interpretation and preprocessing
- Ethical considerations for data collection involving location (GPS) and personally identifiable information (PII)

The literature review provided key insights such as the importance of rigorous preprocessing to handle missing and corrupt sensor data, the trade-off between model complexity and interpretability, and the necessity of user consent for any live vehicle data collection.

These findings informed the initial system architecture and helped shape decisions regarding appropriate ML model selection and data handling protocols.

TASK 5 – Compare and determine ML types: Dale Bent

Based on a presentation to both the supervisor (Ali) and Harindu in week 6, the following was conducted:

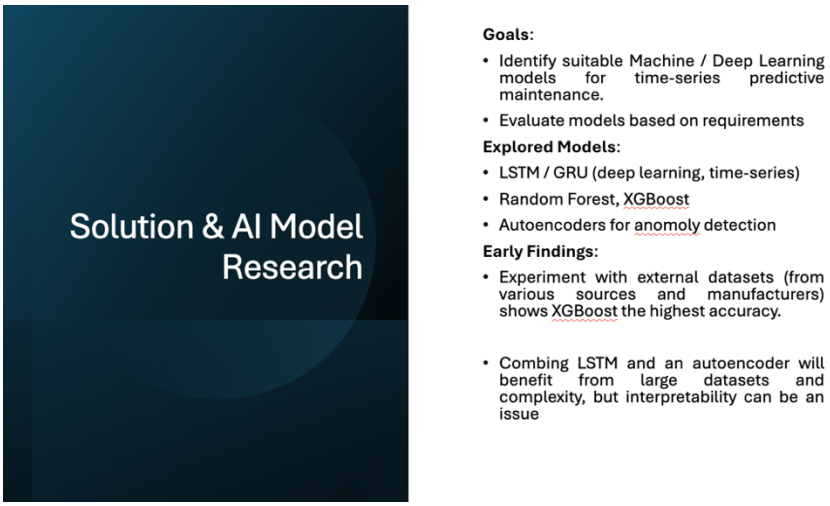


Figure 3. Comparison of ML models in Week 6 presentation slide

Using publicly available external datasets, a preliminary comparison of ML models suitable for predictive maintenance based on OBD-II data was conducted. The datasets included vehicle data from Hyundai, Automotive AVEH, and general maintenance logs.

ML models evaluation included:

- XGBoost
- Random Forest
- LSTM Networks
- Autoencoders (combined with LSTM)

TASK 6 – Prototype initial ML models: Dang Khoa Le

A Google Colab session was conducted and demonstrated in Week 5 and Week 6, highlighting the evaluations of different ML models and accuracy on vehicle predictive maintenance context (sourced from external datasets):

```python
# --- Step 7: Train Predictive Models (Classification + Optional Anomaly) ---
# 1. Random Forest Classifier
print("RandomForest starting")
from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
rf_preds = rf_model.predict(X_test)
print("RandomForest completed")

# 2. XGBoost Classifier
print("XGBoost starting")
from xgboost import XGBClassifier
xgb_model = XGBClassifier(use_label_encoder=False, eval_metric='logloss')
xgb_model.fit(X_train, y_train)
xgb_preds = xgb_model.predict(X_test)
print("XGBoost completed")

# 3. Logistic Regression
print("Logistic Regression starting")
from sklearn.linear_model import LogisticRegression
lr_model = LogisticRegression(max_iter=1000)
lr_model.fit(X_train, y_train)
lr_preds = lr_model.predict(X_test)
print("Logistic Regression completed")

# 4. Optional: Anomaly Detection Models (only for flagging “risky” profiles, not direct classification)
# 4.1. Isolation Forest
from sklearn.ensemble import IsolationForest
iso_model = IsolationForest(contamination=0.1, random_state=42)
iso_model.fit(X_train)
iso_scores = iso_model.predict(X_test)  # -1 = anomaly, 1 = normal
print("Isolation completed")
```

Figure 4. Python pipelines to train simple ML models (Classification + Anomaly optional)

The Python script lies in Week 5 Google Colab session and used to train simple ML models for sourced external predictive maintenance datasets.

Model Evaluation Insights: Predictive Maintenance on Hyundai Cars

📊 Performance Summary

Model	Accuracy	ROC AUC	F1 Score (Class 1)	Comment
XGBoost	0.568	0.567	0.54	⚠️ *Best performing model, but still modest results overall*
Random Forest	0.555	0.553	0.52	⚠️ *Performs slightly worse than XGBoost; better than logistic regression*
Logistic Regression	0.482	0.486	0.50	❌ *Below average performance; not recommended for this dataset*

✅ Model Recommendations

Recommendation	Justification
✅ **Use XGBoost for baseline prediction**	Slightly outperforms others in accuracy, F1, and AUC
⚠️ Consider tuning Random Forest further	May improve with hyperparameter tuning (e.g., max_depth, min_samples_split)
❌ Avoid Logistic Regression	Performance below 50%; lacks fit for this feature space
⚠️ Apply feature engineering or aggregation	Try combining or transforming `Temperature`, `Tire Pressure`, and `Brake Thickness`
✅ Save trained models for iteration/testing	Already done — useful for comparison and versioning
✅ Optional: Use anomaly detection models	Isolation Forest or One-Class SVM might uncover patterns missed by classifiers

Figure 5. Predictive Maintenance - Models Evaluation on Hyundai Cars dataset

Model Evaluation Insights: Predictive Maintenance on Automotive Vehicles Engine Health

📊 Performance Summary

Model	Accuracy	ROC AUC	F1 Score (Class 1)	Comment
XGBoost	0.640	0.589	0.73	⚠️ *Slightly better than other models but still modest performance*
Random Forest	0.635	0.582	0.73	⚠️ *Very close to XGBoost, with low class-0 recall*
Logistic Regression	0.650	0.575	0.76	⚠️ *Best accuracy, but recall on class 0 (failures) is poor*

✅ Recommendation

Recommendation	Justification
⚠️ **Use XGBoost for further tuning**	Slight edge over other models; can benefit from hyperparameter tuning or SMOTE
⚠️ Keep Random Forest for comparison	Simpler and equally performant; may serve as a robust baseline
⚠️ Use Logistic Regression only as reference	Despite best accuracy, poor performance on critical class (engine issues)
✅ Consider feature engineering or collection improvements	Add more high-resolution signals (e.g., vibration, torque, or historical faults)
✅ Try SMOTE or class weighting	To improve detection of under-represented class (class 0)
✅ Apply threshold tuning (ROC/PR curve)	May shift decision boundary to improve balance between precision and recall
⚠️ Avoid deploying without better class 0 performance	High false negatives may miss engine failure risk

Figure 6. Predictive Maintenance - Models Evaluation on Automotive AVEH dataset

Model Evaluation Insights: Predictive Maintenance on Vehicle Maintainance

Performance Summary

Model	Accuracy	ROC AUC	F1 Score (Class 1)	Comment
XGBoost	1.000	1.000	1.00	✅ *Perfect score on all metrics*
Random Forest	0.998	0.995	1.00	✅ *Nearly perfect; very high generalization*
Logistic Regression	0.921	0.862	0.95	⚠️ *Acceptable baseline, but underperforms compared to tree-based models*

✅ Recommendation

Recommendation	Justification
✅ **Use XGBoost as your production model**	Best overall performance, scalable, interpretable (via SHAP)
✅ **Keep Random Forest as a fallback**	Simpler, fast to train, useful for validation
⚠️ Use Logistic Regression for benchmarking only	Useful as a baseline, but not for deployment
⚠️ Tune Logistic Regression (optional)	Scale features, increase `max_iter`, or try different solvers (`liblinear`, `saga`)
✅ **Save trained models**	You've already saved them — great for deployment & reproducibility
✅ Visualize SHAP values (optional)	For explainability to stakeholders or clients

Figure 7. Predictive Maintenance - Models Evaluation on Vehicle Maintenance dataset

Early findings from 3 external datasets – ML models training found that:

- Experiments with external datasets (from various sources and manufacturers) show XGBoost the highest accuracy.
- Combing LSTM and an autoencoder will benefit from large datasets and complexity, but interpretability can be an issue

TASK 7 – Source and Evaluate External OBD-II Dataset: Mohammed Barsat Zulkarnine

In Sprint 1, external OBD-II datasets sourcing has been conducted 3 times to serve different purposes within ML-training contexts:

a) Week 5 & 6 – Sourcing cleaned datasets for simple ML-prototyping and EDA processing:

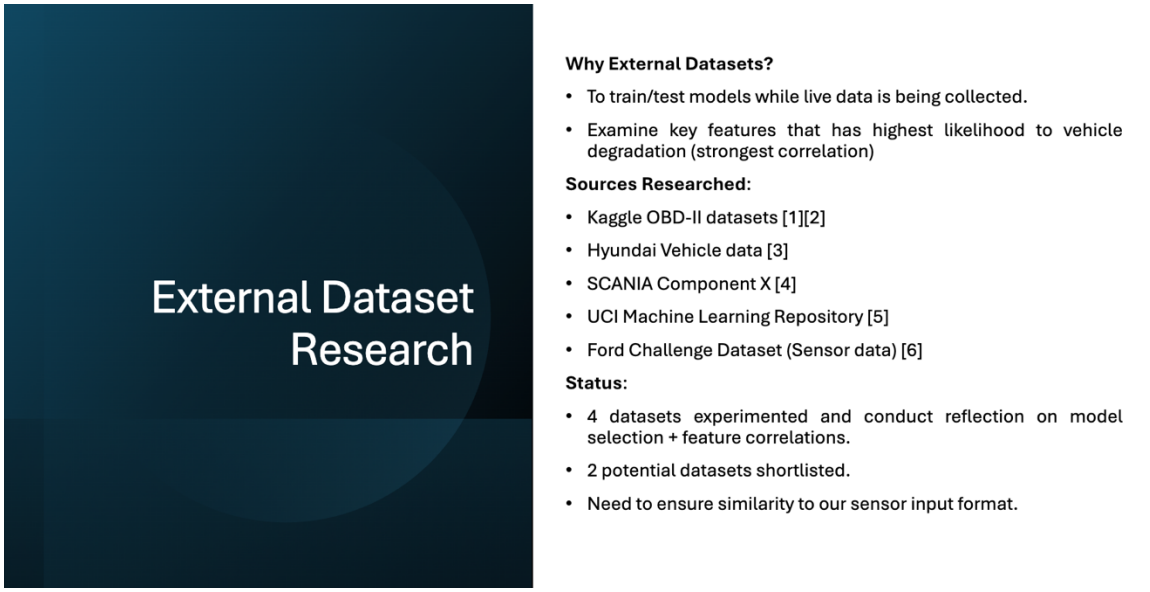


Figure 8. Cleaned datasets sourcing in Week 6 presentation slide

External datasets sources:

1. https://www.kaggle.com/datasets/chavindudulaj/vehicle-maintenance-data
2. https://www.kaggle.com/datasets/parvmodi/automotive-vehicles-engine-health-dataset
3. https://data.mendeley.com/datasets/sm45shp8s5/1
4. https://arxiv.org/abs/2401.15199

b) Week 7 - Collection of hexadecimal raw OBD-II sources and decoding tool research:

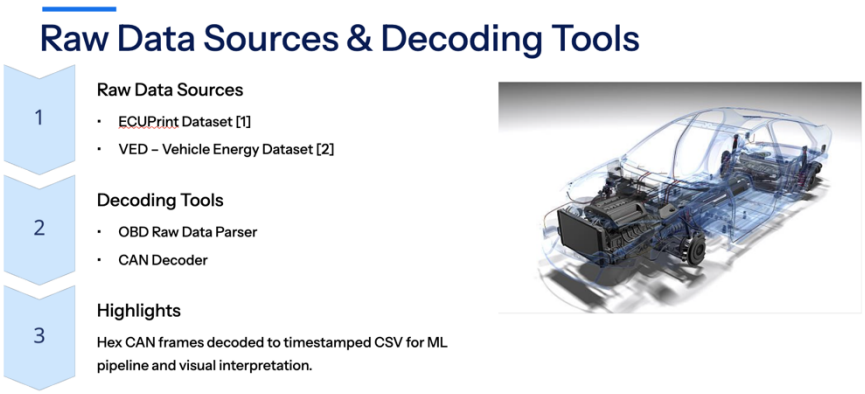


Figure 9. Raw hex-based sources and decoding tools in Week 7 presentation slide

External datasets sources:

[1] https://github.com/LucianPopaLP/ECUPrint
[2] https://github.com/gsoh/VED
[3] https://github.com/rakshitbharat/obd-raw-data-parser
[4] https://github.com/CSS-Electronics/can_decoder

c) Week 7 - Collection of CSV raw OBD-II sources and for data cleaning practices:

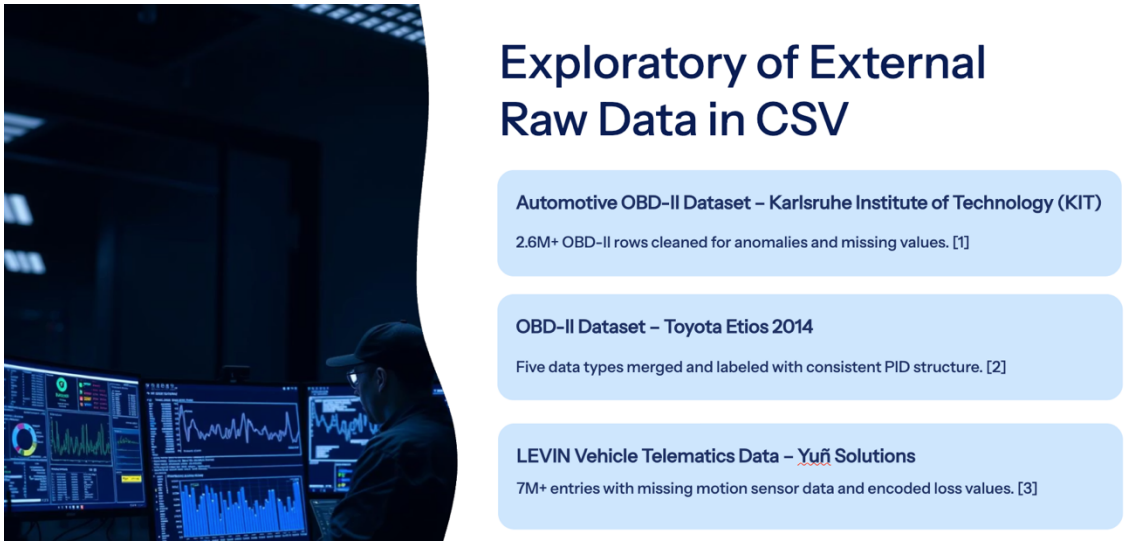


Figure 10. Raw CSV-based sources in Week 7 presentation slide

External datasets sources:

[1] https://radar.kit.edu/radar/en/dataset/bCtGxdTklQlfQcAq

[2] https://github.com/eron93br/carOBD

[3] https://github.com/YunSolutions/levin-openData

Task 8 - Initial Data Cleaning and Preprocessing: Mohammed Barsat Zulkarnine

Based on a dataset sourced in Sprint 1, some initial cleaning and EDA preprocessing of the datasets was conducted via the below Jupyter Notebook scripts:

```python
# --- Step 1: Initial Data Checks ---
df.info()
df.describe()
print(df.isnull().sum())


# --- Step 2: Encode / Map Categorical Features ---
df['Vehicle_Model'] = df['Vehicle_Model'].map({'Truck': 0, 'Van': 1, 'Bus': 2})
df['Maintenance_History'] = df['Maintenance_History'].map({'Poor': 0, 'Average': 1, 'Good': 2})
df['Fuel_Type'] = df['Fuel_Type'].map({'Petrol': 0, 'Electric': 1})
df['Transmission_Type'] = df['Transmission_Type'].map({'Manual': 0, 'Automatic': 1})
df['Owner_Type'] = df['Owner_Type'].map({'First': 0, 'Second': 1, 'Third': 2})
df['Tire_Condition'] = df['Tire_Condition'].map({'Worn Out': 0, 'Good': 1, 'New': 2})
df['Brake_Condition'] = df['Brake_Condition'].map({'Worn Out': 0, 'Good': 1, 'New': 2})
df['Battery_Status'] = df['Battery_Status'].map({'Weak': 0, 'Good': 1, 'New': 2})


# --- Step 3: Convert Dates & Create Features ---
df['Last_Service_Date'] = pd.to_datetime(df['Last_Service_Date'])
df['Warranty_Expiry_Date'] = pd.to_datetime(df['Warranty_Expiry_Date'])
# Convert date to datetime format
df['Days_Since_Last_Service'] = (pd.to_datetime('today') - df['Last_Service_Date']).dt.days
df['Days_Until_Warranty_Expiry'] = (df['Warranty_Expiry_Date'] - pd.to_datetime('today')).dt.days


# --- Step 4: One-Hot Encoding for Categorical Variables ---
categorical_cols = ['Vehicle_Model', 'Maintenance_History', 'Fuel_Type', 'Transmission_Type',
                    'Owner_Type', 'Service_History', 'Accident_History',
                    'Tire_Condition', 'Brake_Condition', 'Battery_Status']
df_encoded = pd.get_dummies(df, columns=categorical_cols, drop_first=True)
# Drop original datetime columns
df_encoded = df_encoded.drop(['Last_Service_Date', 'Warranty_Expiry_Date'], axis=1)
```

Figure 11. EDA and Data Processing on Week 5 Google Colab session

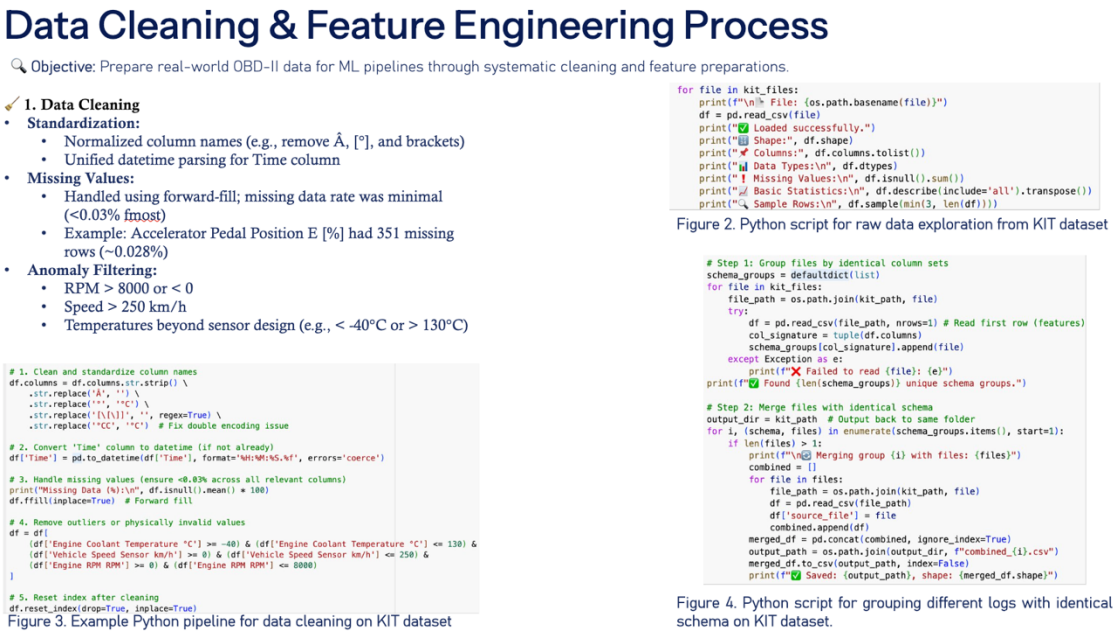


Figure 12. Data Cleaning and Feature Engineering conducted on Week 7 Colab session and presentation slide

Task 9 - Develop a Simple Model for Proof of Concept: Dang Khoa Le

Once again, sourced external datasets have allowed us to develop some models as a proof of concept, with performance summaries being detailed because of these:

Model Evaluation Insights: Predictive Maintenance on Vehicle Maintainance

Performance Summary

Model	Accuracy	ROC AUC	F1 Score (Class 1)	Comment
XGBoost	1.000	1.000	1.00	✅ *Perfect score on all metrics*
Random Forest	0.998	0.995	1.00	✅ *Nearly perfect; very high generalization*
Logistic Regression	0.921	0.862	0.95	⚠️ *Acceptable baseline, but underperforms compared to tree-based models*

Figure 13: Model insights on Vehicle maintenance

Model Evaluation Insights: Predictive Maintenance on Automotive Vehicles Engine Health

Performance Summary

Model	Accuracy	ROC AUC	F1 Score (Class 1)	Comment
XGBoost	0.640	0.589	0.73	⚠️ *Slightly better than other models but still modest performance*
Random Forest	0.635	0.582	0.73	⚠️ *Very close to XGBoost, with low class-0 recall*
Logistic Regression	0.650	0.575	0.76	⚠️ *Best accuracy, but recall on class 0 (failures) is poor*

Figure 14: Model development insights on engine health

Task 10 - Interpret and Visualise Key Parameters: Dale Bent

Visualization of key parameters and their relative features' important scores (features that most likely impact the car degradation and the need for maintenance) in the given datasets conducted in Week 5 Google Colab session:

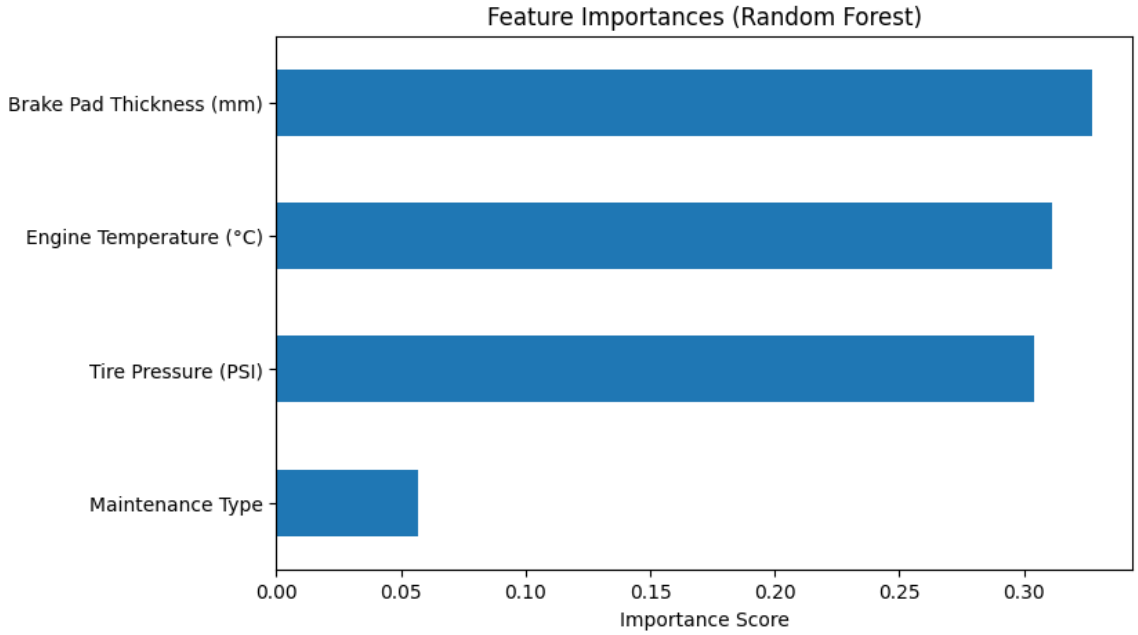


Figure 15: Random Forest feature importance scores on Hyundai Cars dataset

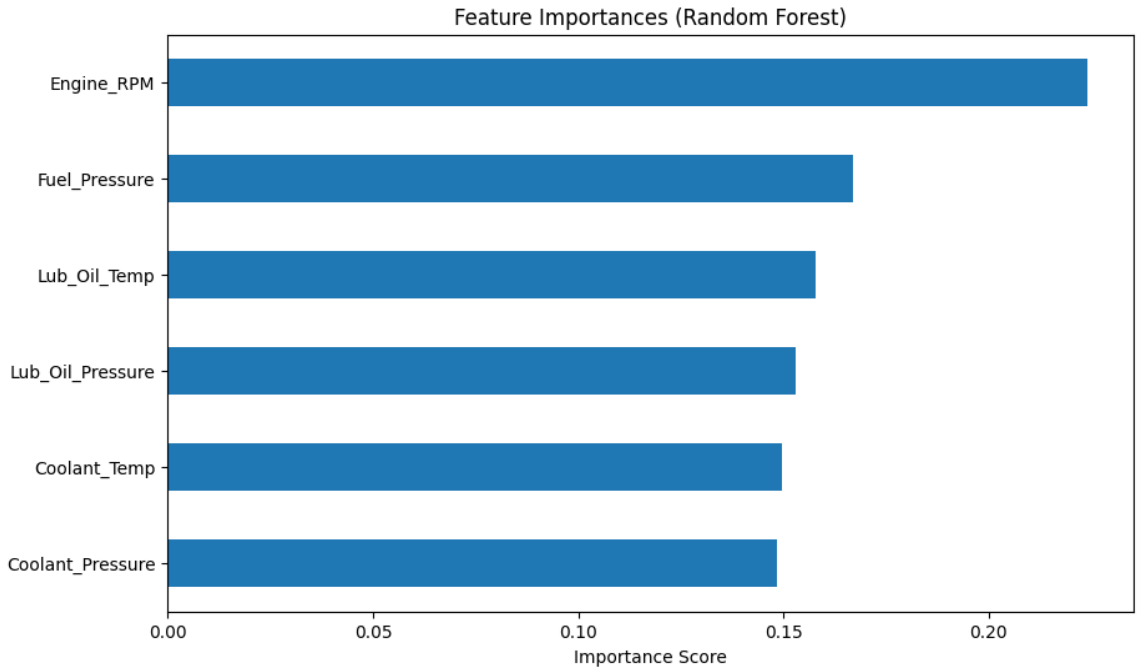


Figure 16: Random Forest feature importance scores on AEVH dataset

Task 11 - Document Work and Prepare for Physcial Data Integration: TEAM CONTRIBUTION

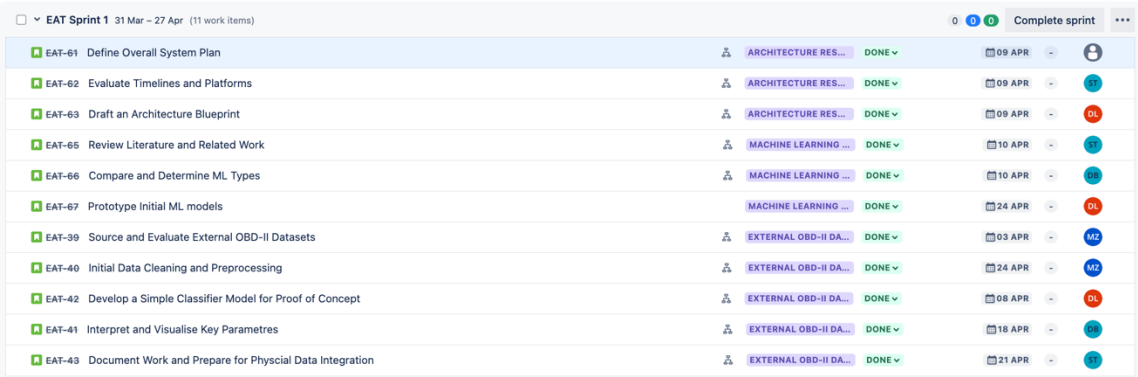


Figure 17. Jira Board displaying Sprint 1 tasks

Documentation of work completed, and pending tasks is meticulously maintained in Jira. We have fully mapped out Sprint 1 with comprehensive details, and the upcoming Sprint 2 is already structured with clear objectives and assignments to ensure seamless workflow continuation.

3. SPRINT DEMONSTRATION

I. WEEK 5 DEMONSTRATION:

- **Overview:**
  The team encountered initial setbacks due to miscommunication regarding presentation expectations. No formal PowerPoint presentation was prepared, leading the team to showcase their work directly through a Google Colab session (Jupyter Notebook).
- **Client Feedback:**
  The client emphasised the importance of clear, non-technical communication. It was advised that presentations should always begin with an introduction clearly outlining objectives, the purpose, and summarising key insights before delving into technical details. The audience, often unfamiliar with deep technical concepts, needs business-oriented summaries that clearly highlight the value and implications of the project.
- **Action:**
  The team proactively organised meetings to rectify miscommunications, clearly define deliverables, and improve overall planning and preparation. A structured sprint-by-sprint timeline was re-developed to ensure better project tracking moving forward.

II. WEEK 6 DEMONSTRATION:

- **Overview:**
  The team demonstrated significant improvement, presenting a detailed, structured PowerPoint covering introduction, project purpose, dataset insights, correlation analysis, and clear predictive model proposals. Additionally, budget and hardware (ELM327, Raspberry Pi setups) requirements were presented for client review.
- **Client Feedback:**
  While the client appreciated the improved professionalism and clear structure, it was highlighted that acquiring real-world vehicle data or raising the budget was challenging and might not be feasible immediately. The importance and complexity of data cleaning in the OBD-II data collection process, notably intrusive and complex in practical contexts, were emphasised.
- **Action:**
  Following the budget limitations highlighted by the client, Ali (Unit Supervisor) provided an alternative solution, offering to supply an OBD-II device directly for practical vehicle data collection by the team, thereby solving immediate hardware acquisition issues.

III. WEEK 7 DEMONSTRATION:

- **Overview:**
  Demonstration of thorough research and expertise in handling raw OBD-II data, specifically addressing hex-based data logs and decoding these into interpretable and ML-friendly CSV formats. Detailed data cleaning routines were also demonstrated, showing how to handle corrupted values, encoding errors, outliers, and missing data efficiently.
- **Client Feedback:**
- The client positively acknowledged the team's effort and preparation, especially their depth of understanding and handling of complex raw data. Appreciation was expressed for the clarity in data exploration, cleaning techniques, and comprehensive insight generation.
- **Action:**
  The team’s proactive effort in exploring multiple datasets (KIT, Toyota Etios, LEVIN), feature engineering, and creating insightful summaries marked significant progress and professionalism. These activities substantially prepared the team for upcoming physical OBD-II data logging in Week 8 when the device is obtained.

OVEERALL SRPINT 1 REFLECTION:

Despite a challenging start in Week 5, the team rapidly demonstrated growth, professionalism, and adaptability. Feedback from the client was integrated effectively in subsequent weeks, showing a marked improvement in the quality and professionalism of presentations, depth of research, dataset handling, and overall project execution. By the conclusion of Week 7, the team was exceptionally well-prepared, displaying a robust understanding of predictive maintenance data processes, data cleaning intricacies, and practical challenges associated with real-world data collection.

4. SPRINT REVIEW

We have made positive initial progress on the Sky Ledge predictive maintenance project. We've taken a proactive approach by researching existing datasets and methodologies available online before receiving the actual OBD data. This strategy was specifically commended by Harindu as it prepares us for what to expect with the real data and will streamline our workflow once we begin actual data collection.

I. WHAT WAS DEMONSTRATED TO THE CLIENT

Week 5:

- Demonstrated initial Exploratory Data Analysis (EDA) and preliminary data cleaning strategies via Google Colab notebook.
- Initial experiments with predictive models, specifically showcasing early trials with the XGBoost algorithm.

Week 6:

- A detailed, structured PowerPoint presentation was delivered, covering project introduction, clear objectives, dataset insights, and a comprehensive correlation analysis.
- Clearly outlined predictive model proposals (XGBoost, LSTM, Autoencoders).
- Presented practical considerations including budget proposals and hardware setup recommendations (ELM327 OBD-II adapter and Raspberry Pi).

Week 7:

- Thoroughly demonstrated expertise in decoding raw hexadecimal OBD-II data into interpretable and ML-ready CSV formats.
- Showcased detailed data cleaning routines addressing corrupted values, encoding errors, outliers, and missing data.
- Provided extensive exploration of external datasets (KIT, Toyota Etios, LEVIN), illustrating feature engineering and data processing techniques.

II. FEEDBACK CLIENT PROVIDED

Week 5 Feedback:

- Emphasised the necessity of clear, non-technical introductions in presentations.
- Advised that presentations should prioritise business-focused summaries, clearly communicating objectives, purposes, and key insights before technical details.

Week 6 Feedback:

- Acknowledged significant improvements in presentation structure and professionalism.
- Highlighted practical limitations of accessing real-world client data or raising additional project budgets.

- Reinforced the complexity and importance of meticulous data cleaning, particularly given the intrusive nature of real-world OBD-II data collection.
- Express the denial of real data access and budget proposal, insist on the team physical data collection with OBD-II device, ordered by Ali.

Week 7 Feedback (Sprint Review)

- Expressed happiness with team progress and how we have taken the initiative to deal with external data despite challenges of physical data collection.
- Noted we will need to move from Jupyter Notebooks to Python scripts for data acquisition via serial communication, Wi-Fi, or Bluetooth.
- Mentioned we need to develop strategies for handling missing values, inconsistent sample rates, and data resampling techniques when physical OBD data comes in.
- Encouraged recognition that 70-80% of our work will involve data cleaning rather than model development.
- Discussed focusing on end to end pipelining data ingestion through cleaning, modeling, and visualisation early on, to not cram in towards the end.

III. CRITICAL ANALYSIS OF PROGRESS AND FEEDBACK

Progress Against Objectives:

- Despite initial setbacks in Week 5 due to communication issues, the team swiftly adjusted, demonstrating remarkable improvement in subsequent weeks.
- Week 6 and 7 showed significant alignment with planned objectives, delivering structured presentations and sophisticated data handling techniques as per the original project roadmap.

Team's Adaptation and Improvement:

- Implemented clear role-based task distribution:
    - **Khoa:** Dataset processing (data cleaning, exploratory data analysis (EDA), important correlated feature exploratory, and data reflection).
    - **Dale:** Managed data backup and validation strategies, Jira task distribution.
    - **Barsat:** Conducted ethical data collection research and supported dataset exploration.
    - **Sadman:** Researched predictive model evaluations and supported dataset exploration.
- Effective integration of client feedback resulted in enhanced clarity, communication, and technical rigor in demonstrations.

Challenges Faced:

- Data availability was initially a major constraint, necessitating reliance on external datasets. This was mitigated by the supervisor providing physical OBD-II devices for real-world data collection.

- Technical challenges included decoding hex data, handling corrupted/missing timestamps, inconsistent sensor readings, and sensor saturation values (e.g., fixed at 255).
- Budget constraints limited hardware acquisitions, but were effectively resolved with supervisor support.

Key Achievements:

- Successfully validated predictive capabilities using external datasets, especially demonstrating XGBoost model effectiveness.
- Developed robust data cleaning and processing pipelines capable of handling complex real-world OBD-II data issues.
- Established comprehensive understanding and methods for decoding raw OBD-II hex data into machine-readable formats, crucial for upcoming practical data logging.

REFLECTION:

Our approach to research before data collection was well received, demonstrating good planning and initiative. However, we need to be mindful of the substantial technical challenges ahead, particularly in data processing and cleaning.

Harindu's advice to distribute work in parallel makes sense given the complexity of the project. We will need to coordinate effectively to ensure all components integrate seamlessly once we bring together the data collection, processing, and modeling elements.

The suggestion to incorporate GPS data offers an interesting opportunity to enhance our analysis, though it adds complexity to our data collection and integration strategy. We'll need to discuss with Ali whether this additional sensor is available or needs to be purchased, along with each other in future sessions to determine whether this is feasible.

Overall, while the supervisor expressed satisfaction with our progress so far, we recognise that the most challenging aspects of the project are still ahead. The next sprint should focus on beginning actual data collection and implementing the data cleaning strategies we've researched.

5. RETROSPECT (CRITICAL REVIEW OF THE PROCESS)

Throughout Sprint 1, the team maintained a strong and consistent rhythm, holding two internal stand-up meetings weekly and meeting outside these sessions as needed to plan, review work, and adjust tasks. Communication within the team was collaborative and open, enabling flexible reallocation of responsibilities when challenges arose.

Several technical-related challenges were identified and analy sed:

Client Data Availability Misalignment:

A critical bottleneck arose from an early assumption that client-supplied data would be available during Sprint 1. This misalignment caused initial efforts to be misdirected toward tasks dependent on unavailable data. The root cause was a gap in early project scope validation during the team planning stage. Mid-sprint, the team critically reassessed project dependencies, adjusted the sprint goals, and pivoted toward independent data collection. This demonstrated adaptability but also highlighted the need for earlier risk validation and contingency planning.

Informal Task Tracking Systems:

Initially, the team relied on shared documents and informal group chats for task tracking, which, while sufficient for small-scale coordination, lacked traceability and clarity as complexity grew. This weakness was identified as a risk during internal discussions and recorded informally. The absence of a centralised task management system like Jira limited formal progress tracking and created occasional uncertainties in task ownership and deadlines. Plans have been made to formally introduce Jira in Sprint 2, improving accountability, transparency, and alignment with agile project management best practices.

Communication and Presentation Planning:

Early sprint demonstrations revealed weaknesses in stakeholder communication. Presentations were initially technically heavy without business-context framing, which did not align with client expectations. This issue arose from incomplete stakeholder analysis during the communication plan phase. After client feedback, the team revised presentation strategies, ensuring future outputs prioritised business-focused narratives before technical details.

Quality Control and Validation:

Although quality expectations for code and documentation were established informally, the absence of a formal quality assurance plan meant that early experiment results (such as ML model comparisons) risked inconsistent documentation standards. This was recognised during sprint reviews, and a plan for implementing consistent documentation and peer-review processes for future technical outputs was agreed upon.

Ethical and Risk Considerations:

Ethical considerations, particularly around data collection involving GPS information and personal vehicle identifiers, were proactively researched. However, formal documentation of ethical risks and mitigation strategies was not fully integrated into the risk register during Sprint 1. Moving

forward, ethical risks will be explicitly recorded alongside technical and logistical risks to ensure a complete view of potential project challenges.

Overall, Sprint 1 highlighted the importance of rigorous project scoping, formalised task and risk management, and a stronger emphasis on stakeholder-focused communication. The team's ability to critically reflect and adapt processes rapidly positioned the project well for more structured and efficient execution in future sprints.

6. LESSONS LEARNED (CRITICAL REVIEW OF SPRINT ONE EXPERIENCE AND FUTURE PLAN)

Sprint 1 taught our team key lessons in both technical execution and project management. One major takeaway was the importance of validating assumptions early. We initially expected client-supplied data to be available immediately, but it was not. Early clarification would have allowed for better planning.

Adaptability proved crucial. When the original plan became impractical, we pivoted to sourcing external datasets and preparing for self-collection of OBD-II data. This strengthened our technical foundations and kept the project progressing. We also learned that a deep understanding of the data — especially raw OBD-II hex formats — is essential before building machine learning models, to avoid inefficiencies and rework.

Professional communication was another critical lesson. Our Week 5 demonstration showed the need to prioritise businessfocused summaries before diving into technical details, greatly improving our engagement with stakeholders in later weeks.

On the project management side, we recognised that informal task tracking (via chats and shared docs) was manageable short-term but unsustainable. We will transition to formal task management using Jira from Sprint 2 onward

Recommendations and Actions for Future Sprints:

- Begin structured OBD II data collection as soon as hardware arrives.
- Set up Jira for formal task and sprint tracking.
- Continue preparing client centric, business focused presentations.
- Fully document data cleaning, feature engineering, and collection processes.
- Update the risk register with data collection and hardware risks.
- Refine system architecture based on real world constraints.
- Maintain two internal stand ups weekly and conduct sprint reviews.
- Update ethical guidelines to cover live data acquisition