Retrospective Sprint 3 of Group RTMST1

Real-Time Wine Sensing Tool

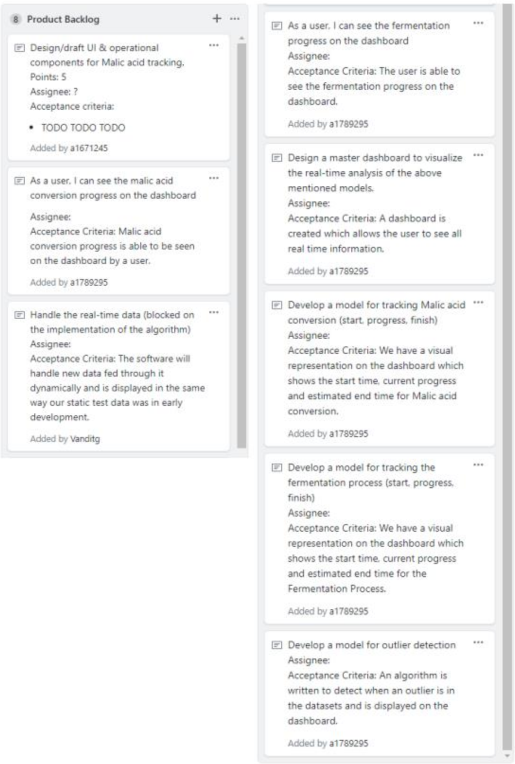
by

a1724222	Andrew Grace
a1783911	Harnoor Bandesh
a1716640	Harrison Greven
a1670268	Jay Hansen
a1720561	Joshua Lennon
a1671245	Ross Pickett
a1785307	Ruonan Fu
a1608613	Ryan Swiggs
a1779153	Vandit Gajjar

Snapshots (Group):

First Snapshot:

Product Backlog and Task Board:



8 Product Backlog

Design/draft UI & operational components for Malic acid tracking.
Points: 5
Assignee: ?
Acceptance criteria:

- TODO TODO TODO

Added by a1671245

As a user, I can see the malic acid conversion progress on the dashboard

Assignee:
Acceptance Criteria: Malic acid conversion progress is able to be seen on the dashboard by a user.

Added by a1789295

Handle the real-time data (blocked on the implementation of the algorithm)
Assignee:
Acceptance Criteria: The software will handle new data fed through it dynamically and is displayed in the same way our static test data was in early development.

Added by Vanditg

As a user, I can see the fermentation progress on the dashboard
Assignee:
Acceptance Criteria: The user is able to see the fermentation progress on the dashboard.

Added by a1789295

Design a master dashboard to visualize the real-time analysis of the above mentioned models.
Assignee:
Acceptance Criteria: A dashboard is created which allows the user to see all real time information.

Added by a1789295

Develop a model for tracking Malic acid conversion (start, progress, finish)
Assignee:
Acceptance Criteria: We have a visual representation on the dashboard which shows the start time, current progress and estimated end time for Malic acid conversion.

Added by a1789295

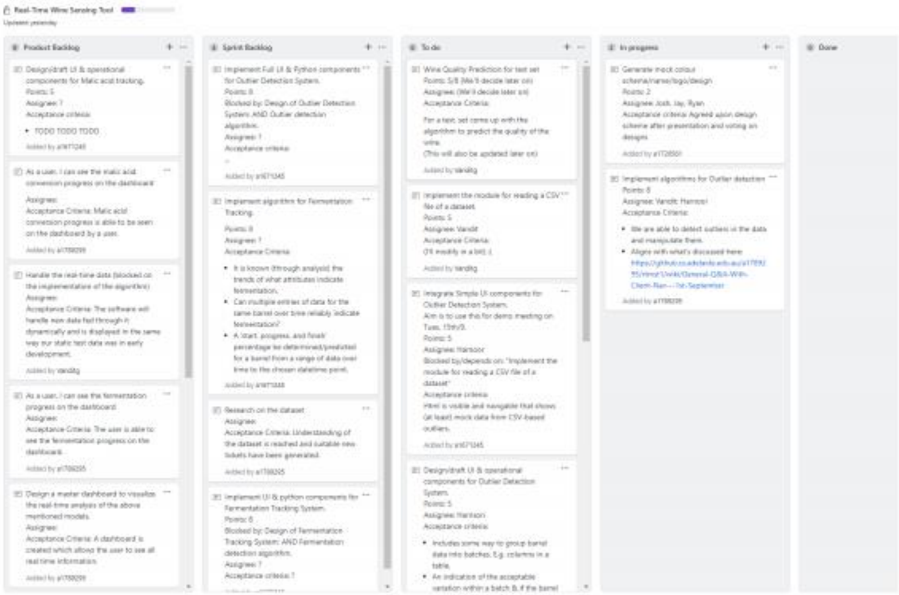
Develop a model for tracking the fermentation process (start, progress, finish)
Assignee:
Acceptance Criteria: We have a visual representation on the dashboard which shows the start time, current progress and estimated end time for the Fermentation Process.

Added by a1789295

Develop a model for outlier detection
Assignee:
Acceptance Criteria: An algorithm is written to detect when an outlier is in the datasets and is displayed on the dashboard.

Added by a1789295

Figure 1. Product Backlog



Real-Time Wine Sensing Tool
Updated yesterday

Product Backlog

Design/draft UI & operational components for Malic acid tracking.
Points: 5
Assignee: ?
Acceptance criteria:

- TODO TODO TODO

Added by a1671245

As a user, I can see the malic acid conversion progress on the dashboard

Assignee:
Acceptance Criteria: Malic acid conversion progress is able to be seen on the dashboard by a user.

Added by a1789295

Handle the real-time data (blocked on the implementation of the algorithm)
Assignee:
Acceptance Criteria: The software will handle new data fed through it dynamically and is displayed in the same way our static test data was in early development.

Added by vanditg

As a user, I can see the fermentation progress on the dashboard
Assignee:
Acceptance Criteria: The user is able to see the fermentation progress on the dashboard.

Added by a1789295

Design a master dashboard to visualize the real-time analysis of the above mentioned models.
Assignee:
Acceptance Criteria: A dashboard is created which allows the user to see all real time information.

Added by a1789295

Sprint Backlog

Implement Full UI & Python components for Outlier Detection System.
Points: 8
Blocked by: Design of Outlier Detection System AND Outlier detection algorithm.
Assignee: ?
Acceptance criteria:
-

Added by a1671245

Implement algorithm for Fermentation Tracking.

Points: 8
Assignee: ?
Acceptance Criteria:

- It is known (through analysis) the trends of what attributes indicate fermentation.
- Can multiple entries of data for the same barrel over time reliably indicate fermentation?
- A 'start, progress, and finish' percentage be determined/predicted for a barrel from a range of data over time to the chosen datetime point.

Added by a1671348

Research on the dataset
Assignee:
Acceptance Criteria: Understanding of the dataset is reached and suitable new labels have been generated.

Added by a1789295

Implement UI & python components for Fermentation Tracking System.
Points: 8
Blocked by: Design of Fermentation Tracking System AND Fermentation detection algorithm.
Assignee: ?
Acceptance criteria: ?

To do

Wine Quality Prediction for test set
Points: 5/8 (We'll decide later on)
Assignee: (We'll decide later on)
Acceptance Criteria:

For a test set come up with the algorithm to predict the quality of the wine.
(This will also be updated later on)

Added by vanditg

Implement the module for reading a CSV file of a dataset.
Points: 5
Assignee: Vandit
Acceptance Criteria:
(I'll modify in a bit) :)

Added by vanditg

Integrate Simple UI components for Outlier Detection System.
Aim is to use this for demo meeting on Tues. 13th/9.
Points: 5
Assignee: Harrison
Blocked by/depends on: "Implement the module for reading a CSV file of a dataset"
Acceptance criteria:
Html is visible and navigable that shows (at least) mock data from CSV-based outliers.

Added by a1671245

Design/draft UI & operational components for Outlier Detection System.
Points: 5
Assignee: Harrison
Acceptance criteria:

- Includes some way to group barrel data into batches. E.g. columns in a table.
- An indication of the acceptable variation within a batch & if the barrel

In progress

Generate mock colour scheme/name/logo/design
Points: 2
Assignee: Josh, Jay, Ryan
Acceptance criteria: Agreed upon design scheme after presentation and voting on designs.

Added by a1728501

Implement algorithms for Outlier detection
Points: 8
Assignee: Vandit, Harrison
Acceptance Criteria:

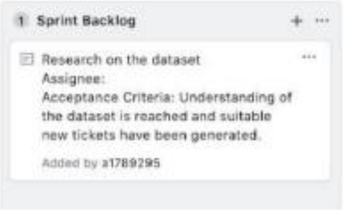
- We are able to detect outliers in the data and manipulate them.
- Aligns with what's discussed here: https://github.cs.adelaide.edu.au/a178929 5/rimst/wiki/General-Q&A-With-Chem-Ran---7th-September

Added by a1789295

Done

Figure 2. Task Board

Sprint Backlog and User Stories:



1 Sprint Backlog

Research on the dataset
Assignee:
Acceptance Criteria: Understanding of the dataset is reached and suitable new tickets have been generated.

Added by a1789295

Figure 3. Sprint Backlog

- **Wine Quality Prediction for test set:**

Design and implement an algorithm that determines the quality of the wine based on test set data.

Acceptance Criteria: We have an algorithm that can take input data and return a value for the quality of the wine.

- **Implement the module for reading a CSV file of a dataset**

Create a python module for reading input CSV files.

Acceptance Criteria: We have a code module that can parse input files in CSV format.

- **Integrate Simple UI components for Outlier Detection System**

Implement User Interface components that connect to the Outlier Detection algorithm. The aim for this task is to present it in the Demo meeting on the 15th September.

Acceptance Criteria: We have a simple partial interface for the outlier detection system.

- **Design UI & operational components for Outlier Detection System**

Design a user interface that contains the necessary functional components for detecting outliers in the reading of a barrel.

Acceptance Criteria: We have a design of the user interface that includes a method of grouping barrel data, indicates acceptable variance within a batch, presented in the dashboard as a chart, and includes an explanation of how to implement it into code.

- **Design UI & operational components for Fermentation Tracking**

Design a user interface that contains the necessary functional components for tracking fermentation of a barrel.

Acceptance Criteria: The design of the User Interface includes a method of tracking fermentation of the same barrel over time, tracking the average of a batch of barrels.

- **Implement frontend for the dashboard**

Create HTML and CSS code to implement the User Interface as specified in the designs.

Acceptance Criteria: The front end of the dashboard has been coded to fit the designs.

- **Generate mock colour scheme/name/logo/design**

Create some drafts of a User Interface design. Create a name and a logo for the project.

Acceptance Criteria: As a group we have decided on a name, logo, and design for our project User Interface which would best engage and interest the user

- **Implement algorithms for Outlier Detection System**

The data will have outliers that will skew the results of our machine learning programs, these outliers need to be removed to provide more precise analysis of the data.

Acceptance Criteria: We have a definition of an outlier. We have an algorithm that detects outliers which can reliably inform the user when a wine is at risk.

Definition of Done:

The Definition of Done for the sprint, and the sprint is ‘done’ when:

- The sprint was planned and mutually understood, reviewed, and accepted by all members of the team.
- Coding and designed have been completed.
- As many sprint items as possible (described in the previous ‘User Stories’ section) have met their respective acceptance criteria.

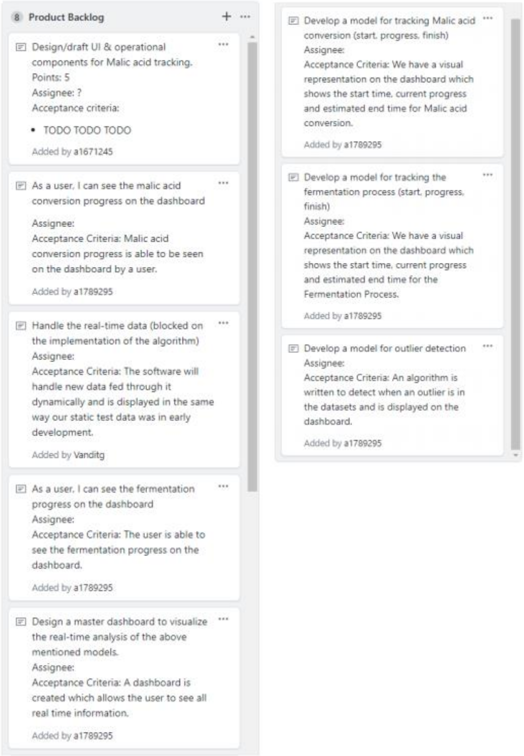
- All sprint items that have met their acceptance criteria have been verbally or explained in writing and understood by all.
- All sprint items that have met their acceptance criteria have been moved out of the In Progress or TODO stages.
- All sprint items that have not met their acceptance criteria have been moved into the next sprint or back to the backlog.

Summary of changes:

We concluded Sprint #2 last week and it was quite productive. The team was able to finish many tasks including: finalizing the tech stack for Infra, implementing ML algorithms on similar datasets, creating a folder structure and skeleton of the code base, and furthering the research into the wine data attributes. This week we have officially started with Sprint #3 and the team has once again proactively picked up many challenging tech tickets like: Implementing module for reading the CSV file for dataset, implementing algorithms for outlier detection, and designing UI components for outlier detection and fermentation tracking system, etc. In the last sprint the team tested different algorithms on the data set and out of those, three algorithms were returning promising results. The team will be selecting one of them to move ahead with. The team will also try to implement a UI for outlier detection system. Broadly, it will be like a dashboard that will show the data of the outliers per batch, which will be generated by our ML algorithm. This is expected to be shown to the clients in our first DEMO of this project, on 22nd September 2020.

Second Snapshot:

Product Backlog and Task Board:



8 Product Backlog

Design/draft UI & operational components for Malic acid tracking.
Points: 5
Assignee: ?
Acceptance criteria:

- TODO TODO TODO

Added by a1671245

As a user, I can see the malic acid conversion progress on the dashboard
Assignee:
Acceptance Criteria: Malic acid conversion progress is able to be seen on the dashboard by a user.

Added by a1789295

Handle the real-time data (blocked on the implementation of the algorithm)
Assignee:
Acceptance Criteria: The software will handle new data fed through it dynamically and is displayed in the same way our static test data was in early development.

Added by Vanditg

As a user, I can see the fermentation progress on the dashboard
Assignee:
Acceptance Criteria: The user is able to see the fermentation progress on the dashboard.

Added by a1789295

Design a master dashboard to visualize the real-time analysis of the above mentioned models.
Assignee:
Acceptance Criteria: A dashboard is created which allows the user to see all real time information.

Added by a1789295

Develop a model for tracking Malic acid conversion (start, progress, finish)
Assignee:
Acceptance Criteria: We have a visual representation on the dashboard which shows the start time, current progress and estimated end time for Malic acid conversion.

Added by a1789295

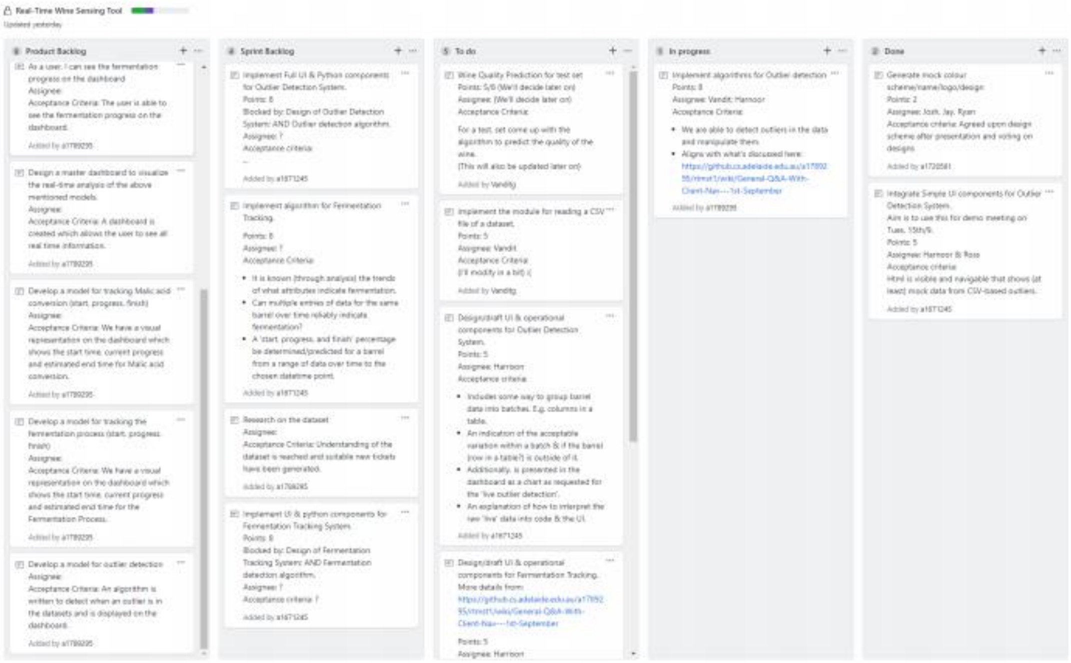
Develop a model for tracking the fermentation process (start, progress, finish)
Assignee:
Acceptance Criteria: We have a visual representation on the dashboard which shows the start time, current progress and estimated end time for the Fermentation Process.

Added by a1789295

Develop a model for outlier detection
Assignee:
Acceptance Criteria: An algorithm is written to detect when an outlier is in the datasets and is displayed on the dashboard.

Added by a1789295

Figure 4. Product Backlog



Real-Time Wine Sensing Tool
Updated yesterday

Product Backlog

As a user, I can see the fermentation progress on the dashboard
Assignee:
Acceptance Criteria: The user is able to see the fermentation progress on the dashboard.

Added by a1789295

Design a master dashboard to visualize the real-time analysis of the above mentioned models.
Assignee:
Acceptance Criteria: A dashboard is created which allows the user to see all real time information.

Added by a1789295

Develop a model for tracking Malic acid conversion (start, progress, finish)
Assignee:
Acceptance Criteria: We have a visual representation on the dashboard which shows the start time, current progress and estimated end time for Malic acid conversion.

Added by a1789295

Develop a model for tracking the fermentation process (start, progress, finish)
Assignee:
Acceptance Criteria: We have a visual representation on the dashboard which shows the start time, current progress and estimated end time for the Fermentation Process.

Added by a1789295

Develop a model for outlier detection
Assignee:
Acceptance Criteria: An algorithm is written to detect when an outlier is in the datasets and is displayed on the dashboard.

Added by a1789295

Sprint Backlog

Implement Full UI & Python components for Outlier Detection System.
Points: 8
Blocked by: Design of Outlier Detection System: AND Outlier detection algorithm.
Assignee: ?
Acceptance criteria:
--

Added by a1671245

Implement algorithm for Fermentation Tracking.

Points: 8
Assignee: ?
Acceptance Criteria:

- It is known (through analysis) the trends of what attributes indicate fermentation.
- Can multiple entries of data for the same barrel over time reliably indicate fermentation?
- A 'start, progress, and finish' percentage be determined/predicted for a barrel from a range of data over time to the chosen datetime point.

Added by a1671245

Research on the dataset
Assignee:
Acceptance Criteria: Understanding of the dataset is reached and suitable new tickets have been generated.

Added by a1789295

Implement UI & python components for Fermentation Tracking System.
Points: 8
Blocked by: Design of Fermentation Tracking System: AND Fermentation detection algorithm.
Assignee: ?
Acceptance criteria: ?

Added by a1671245

5 To do

Wine Quality Prediction for test set
Points: 5/8 (We'll decide later on)
Assignee: (We'll decide later on)
Acceptance Criteria:

For a test set come up with the algorithm to predict the quality of the wine.
(This will also be updated later on)

Added by Vanditg

Implement the module for reading a CSV file of a dataset.
Points: 5
Assignee: Vandit
Acceptance Criteria:
(I'll modify in a bit) :(

Added by Vanditg

Design/draft UI & operational components for Outlier Detection System.
Points: 5
Assignee: Harrison
Acceptance criteria:

- Includes some way to group barrel data into batches. E.g. columns in a table.
- An indication of the acceptable variation within a batch & if the barrel (row in a table?) is outside of it.
- Additionally, is presented in the dashboard as a chart as requested for the 'live outlier detection'.
- An explanation of how to interpret the new 'live' data into code & the UI.

Added by a1671245

Design/draft UI & operational components for Fermentation Tracking.
More details from:
https://github.cs.adelaide.edu.au/a178929 5/rtmst1/wiki/General-Q&A-With-Client-Nav---1st-September

Points: 5
Assignee: Harrison

In progress

Implement algorithms for Outlier detection
Points: 8
Assignee: Vandit: Harnoor
Acceptance Criteria:

- We are able to detect outliers in the data and manipulate them.
- Aligns with what's discussed here: https://github.cs.adelaide.edu.au/a178929 5/rtmst1/wiki/General-Q&A-With-Client-Nav---1st-September

Added by a1789295

Done

Generate mock colour scheme/name/logo/design
Points: 2
Assignee: Josh, Jay, Ryan
Acceptance criteria: Agreed upon design scheme after presentation and voting on designs.

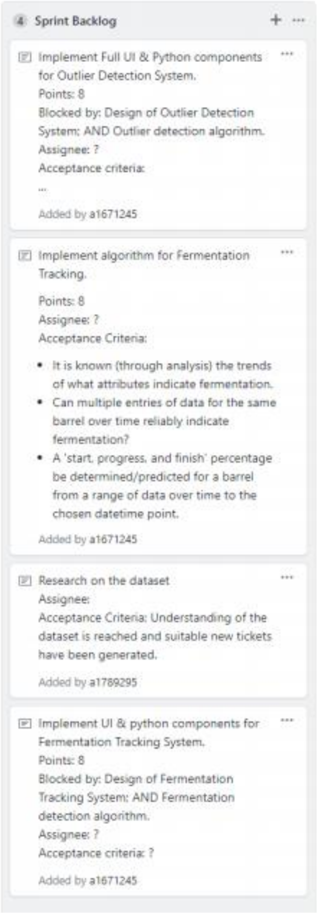
Added by a1720581

Integrate Simple UI components for Outlier Detection System.
Aim is to use this for demo meeting on Tues, 15th/9.
Points: 5
Assignee: Harnoor & Ross
Acceptance criteria:
Html is visible and navigable that shows (at least) mock data from CSV-based outliers.

Added by a1671245

Figure 5. Task Board

Sprint Backlog and User Stories



4 Sprint Backlog

Implement Full UI & Python components for Outlier Detection System.
Points: 8
Blocked by: Design of Outlier Detection System; AND Outlier detection algorithm.
Assignee: ?
Acceptance criteria:
...
Added by a1671245

Implement algorithm for Fermentation Tracking.

Points: 8
Assignee: ?
Acceptance Criteria:

- It is known (through analysis) the trends of what attributes indicate fermentation.
- Can multiple entries of data for the same barrel over time reliably indicate fermentation?
- A 'start, progress, and finish' percentage be determined/predicted for a barrel from a range of data over time to the chosen datetime point.

Added by a1671245

Research on the dataset
Assignee:
Acceptance Criteria: Understanding of the dataset is reached and suitable new tickets have been generated.
Added by a1789295

Implement UI & python components for Fermentation Tracking System.
Points: 8
Blocked by: Design of Fermentation Tracking System; AND Fermentation detection algorithm.
Assignee: ?
Acceptance criteria: ?
Added by a1671245

Figure 6. Sprint Backlog

- **Wine Quality Prediction for test set**

Design and implement an algorithm that determines the quality of the wine based on test set data.

Acceptance Criteria: We have an algorithm that can take input data and return a value for the quality of the wine.

- **Implement the module for reading a CSV file of a dataset**

Create a python module for reading input CSV files.

Acceptance Criteria: We have a code module that can parse input files in CSV format.

- **Integrate Simple UI components for Outlier Detection System**

Implement User Interface components that connect to the Outlier Detection algorithm. The aim for this task is to present it in the Demo meeting on the 15th September.

Acceptance Criteria: We have a simple partial interface for the outlier detection system. Task shifted into Done state.

- **Design UI & operational components for Outlier Detection System**

Design a user interface that contains the necessary functional components for detecting outliers in the reading of a barrel.

Acceptance Criteria: We have a design of the user interface that includes a method of grouping barrel data, indicates acceptable variance within a batch, presented in the dashboard as a chart, and includes an explanation of how to implement it into code.

- **Design UI & operational components for Fermentation Tracking**

Design a user interface that contains the necessary functional components for tracking fermentation of a barrel.

Acceptance Criteria: The design of the User Interface includes a method of tracking fermentation of the same barrel over time, tracking the average of a batch of barrels.

- **Implement frontend for the dashboard**

Create HTML and CSS code to implement the User Interface as specified in the designs.

Acceptance Criteria: The front end of the dashboard has been coded to fit the designs.

- **Generate mock colour scheme/name/logo/design**

Create some drafts of a User Interface design. Create a name and a logo for the project.

Acceptance Criteria: As a group we have decided on a name, logo, and design for our project User Interface which would best engage and interest the user Task shifted into Done state.

- **Implement algorithms for Outlier Detection System**

The data will have outliers that will skew the results of our machine learning programs, these outliers need to be removed to provide more precise analysis of the data.

Acceptance Criteria: We have a definition of an outlier. We have an algorithm that detects outliers which can reliably inform the user when a wine is at risk. Aligns with what's discussed here: https://github.cs.adelaide.edu.au/a1789295/rtmst1/wiki/General-Q&A-With-Client-Nav---1st-September

Definition of Done:

Below states our Definition of Done for the sprint. The sprint is 'done' when:

- The sprint was planned and mutually understood, reviewed, and accepted by all members of the team.
- Coding and designed have been completed.
- All or as many sprint items as possible (described in the previous 'User Stories' section) that were in the state of To Do or In Progress have met their respective acceptance criteria.
- All sprint items that have met their acceptance criteria have been verbally or explained in writing and understood by all.
- All sprint items that have met their acceptance criteria have been moved to Done status.
- All sprint items that have not met their acceptance criteria have been moved into the next sprint or back to Sprint Backlog.

Summary of changes:

A functional UI for outlier detection system was implemented for the aim of demonstrating it during the meeting with the client on the 15th of September. Attendance was impossible however, due to all our group members being occupied with other study and work commitments. Additionally, a logo, design and color scheme were voted upon to allow the progression of another dependent task. These two tasks were moved into Done.

Declaration: I have attended sprint 3 planning meeting on 9th September 2020 and sprint 3 retrospective meeting/sprint 4 planning on 7th October 2020 with the tutor Mr. Navpreet Singh Ahuja.

What went well in the sprint (Individual):

Our third sprint was about to come up with the demo version of our product that we did so far such as User Interface for our product application, CSV file loader, and Outlier Detection from the given CSV input file. The last sprint was one of the best sprints so far as we performed and completed our previous tasks, communicated very well with the team regarding the implementation of the tech stack infrastructure of Front-End and Back-End communication. These implementations were completely based on the user stories that our group members selected for the third sprint backlog. One of the most important things to note here is that we finally designed our Outlier Detection algorithm using the Standard Deviation approach [1] in which based on the past and current trends if any faulty data comes from the input, the system triggers and let the user know that the provided data is faulty and needs to be checked again. Considering the Web-development part, I believe that the team did tremendous progress as well by finalizing the logo and dashboard designs and the implementation is undergoing.

What could be improved (Individual):

As we progressed to the fourth sprint, there were certain limitations I figured out during the third sprint retrospective.

- There was a major issue, I found till this 3rd sprint is that considering the documentation part, version control, and update to the team. Even though we communicated a lot regarding documentation and provide updates to the team, probably after every commit to our GitHub repo, there was no one on the team who updates the wiki page and provides updates to the team about the commit immediately rather procrastinated till the very end. So I believe these areas, require improvement.
- I am focusing on the Outlier Detection part and because of my very small mistake in finalizing the algorithm, our team's User Interface work was delayed by 2-3 days which lost our track a bit. However the team helped during these difficult times and all the team members have approached me and helped me to finalize the algorithm, which eradicated our previous issue

of computation cost. So, I need to ensure from the next sprint that if I find difficulties in the product development and am not be able to come up with the solution, I should update the team immediately and ask for the help debugging it alone and waiting for the miracle to happen. This was the learning lesson for me and the team.

What will be the group commit to improve in the next sprint(Individual)?

In our last daily scrum meeting of the third sprint, we discussed several issues and decided to follow certain rules and strategies in the implementation phase for successful product development. Also, we got a sample of the data from the client in order to get an idea of how the actual data will look like.

- During our mid-semester break, we received the sample data from the client to get an overall idea about how the actual data will look like when we complete the development of our product. The data was challenging but thanks to the effort of the team, we were already working on a similar dataset. However, the sample data had some extra/different features which seem to be troublesome if we did not check our current Outlier Detection algorithm on sample data. So this is the area where our group needs to commit most of the time because if the Outlier Detection does not work properly, the follow-up tasks on User Interface and Web-development would be affected.
- The last sprint was one of the best sprints of all the sprints though we need to make sure that we are approaching the final deadlines of our timeline and the time is less to complete the tasks as well. So effective and efficient communication, completion of assigned tasks in a timely manner, and updating the wiki documentation is very crucial and our team needs to devote a majority of the time in the fourth sprint.

References:

[1] Kriegel, Hans-Peter, Peer Kröger, and Arthur Zimek. "Outlier detection techniques." Tutorial at KDD 10 (2010): 1-76.

.