Project Title
ValoFoDummies.gg

Project Summary
It should be a 1-2 paragraph description of what your project is.
We aim to build a web app to help players improve by providing players easily accessible information and recommendations. We plan to do this by making stats more understandable through visual formats, and also analyzing those stats to understand play habits. Using a combination of individual player stats and meta trends, the web app will give recommendations for things such as certain areas to focus on, or agents to play. Utilizing LLMs, the web app will be capable of giving advice for a particular match. It is also a fun site that allows users to draw comparisons between themselves and their favorite pros. They can also create an account to track their progress over time.

Description
State as clearly as possible what you want to do. What problem do you want to solve, etc.?

We want to give new and casual players an easy way to learn more about the game and how to improve their gameplay. Decision making will be less overwhelming as the app will recommend a few agents for a given map along with playstyle recommendations using data from pro players. Users will have an easy GUI to understand what agents and abilities are best for what map, a common question frequently asked.

On the other hand, all the operations for data analytics should be efficient, so the users get instant results. The game is very complex and has many variables, very difficult for casual players to sift through the mountain of data on sites like blitz.gg and vlr.gg and come to their own conclusions. Using data from pro-player matches as well as casual matches will allow us to analyze patterns and trends for best performing agents and abilities and present this information in a simple and easy to understand format for our users.

We will also be displaying information of performance based on Agents, maps and other relevant variables so players who are more versed with the game can use the information.

Creative Component

What would be a good creative component (technically challenging function) that can improve the functionality of your application? (What is something cool that you want to include? How are you planning to achieve it?)

- Interactive Visualizations: create interactive graphs of relevant statistics such as player, agent and map statistics
    - Complex frontend and backend code to achieve this
- Machine Learning: predict live match win rate using the calculated stats of the other players on the team and the enemy team
    - Need to understand best ML algorithm to achieve this, train and test the ML
- LLM: we can fine-tune an open-source LLM to give specific Valorant insights, like creating live strategies against the opposing team or recommendations on playstyle per agent and per map.
    - Will need to understand LLM API calls, how to train the model, how to interpret output
- Player-lookalikes: A player can find a pro player with similar stats/play-styles as them. If the lookalike is a streamer, they would be able to get better at their preferred playstyle by watching them.
    - Have to tag players as Pros/Streamers/Casual

Usefulness

Explain as clearly as possible why your chosen application is useful. What are the basic functions of your web application? (What can users of this website do? Which simple and complex features are there?). Make sure to answer the following questions: Are there any similar websites/applications out there? If so, what are they, and how is yours different?

Everyone wants to improve their gameplay and analyzing pro stats can help them get insights on how to better play certain maps, agents, understand team comps and so on.
Features

- View own stats
    - Players can link RIOT accounts so we can use Riot API to get data
    - Combination with other data will allow for valuable insights
- User-Friendly GUI for important META information
    - GUI allows users to easily locate information for FAQ such as META team combinations for maps for team agents and roles
    - Personal Agent recommendations based on map/ desired role
    - More detailed information such as match or player-specific also displayed but not as prominent to avoid clutter

    - Display Attacker/ Defender sided maps
- Generate match based predictions for win loss
    - Use data on agent, maps and players to predict win loss %

Similar websites like tracker.gg or blitz.gg give insights into player stats, but they don't give direct tips for their users to improve. They are mainly a clutter of data one has to dig through to gain any real benefit. We offer recommended agents based on stats, desired role and map which are easy to see and interpret. We also display information that these sites do not like attacker/defender sided maps.

Realness

We want you to build a real application. So, make sure to locate real datasets. Describe your data sources (Where is the data from? In what format [csv, xls, txt,...], data size [cardinality and degree], what information does the data source capture?). It would be hard to satisfy stage 2 requirements with one dataset. Thus, we strongly recommend identifying at least two different data sources for your project.

- vlr.gg

This dataset would serve as the basis for recommendations as it is the match data of pro players. It can be scraped directly from the site or we can use past data through Kaggle (https://www.kaggle.com/datasets/visualize25/valorant-pro-matches-full-data). Based on Kaggle, the dataset is in the .sqlite format and has 4 tables. The Game_Rounds table has 15531 rows and 4 columns, the Game_Scoreboard table has 157939 rows and 28 tables, the Games table has 15888 rows and 36 columns, and the Matches table has 7818 rows and 12 columns. If our team were to scrape new data from the vlr.gg site, it would likely follow a similar format to the Kaggle dataset, but with a greater number of rows.

- blitz.gg

https://github.com/IronicNinja/valorant-stats

This dataset is scraped from blitz.gg, which contains data about the entire Valorant player base across different ranks. The data is very granular and split by multiple levels. The top level is split by abilities, agents, maps, and weapons. Within each folder, there are separate folders for each of the maps and another folder for combined data across maps. Within those folders, there are individual CSV files, which are split by the ranks of players. To use this data, we would reformat it into tables, so that there would be relationships between the abilities, agents, maps, and weapons.

Functionality

This is where you talk about what the website delivers. Talk about how a user would interact with the application (i.e., things that one could create, delete, update, or search for). Read the requirements for stage 4 to see what other functionalities you want to provide to the users. You should include:

Our website seeks to allow users to understand the Valorant meta and use relevant information to improve their gameplay. For example, they may be able to:

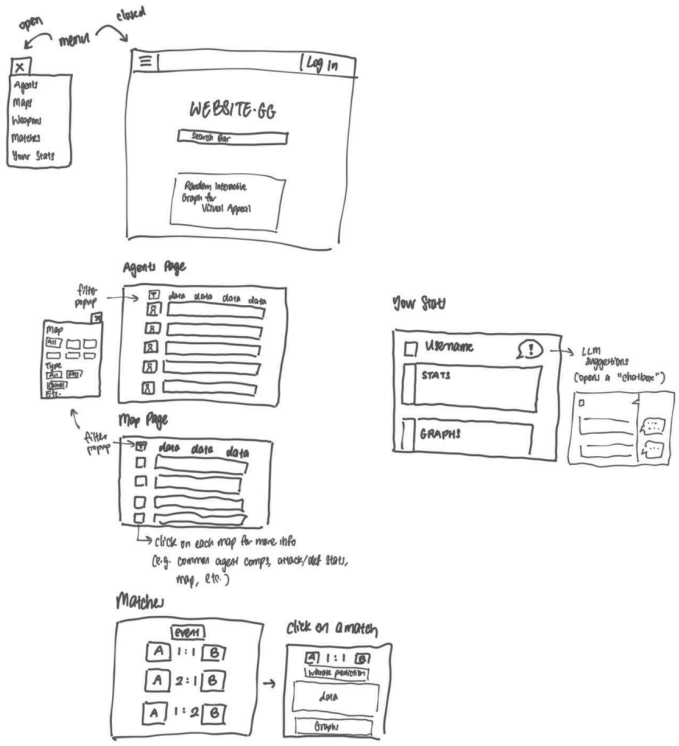
- find popular agent picks for certain maps
- whether the map is defender or attacker sided
- compare themselves to pro players that have a similar playstyle/role to the user.
- See general match/ map/ player specific data

They would start interacting with the website by logging in with their Riot account, so we can get their match history. The API also allows us to see if they are currently in a live game or not, and we can use the match data to generate recommendations and predictions. If they are just browsing, past match history would be visible and it would be correlated to specific improvements based on the role of pro players.

- Duelists should have higher first bloods -> need to be more aggressive
- Support agents generally have higher assists -> Play more backline
- If player has anomalies in stats, could suggest role closest to player's natural playstyle based off of stats and swap

Besides the overall functionalities of the site, the site would also have an interactive feature that visualizes statistics and aims to utilize machine learning and LLMs to provide win-rate predictions and suggestions to improve player performance

A low-fidelity UI mockup
What do you imagine your final application's interface might look like? A PowerPoint slide or a pencil sketch on a piece of paper works!



Project work distribution
Who will be responsible for each of the tasks or subtasks?
Explain how backend systems will be distributed across members. Be as specific as possible as this could be part of the final peer evaluation metrics.

Frontend:
Andrew: React JS, Python, Java, Firebase
Daniel: React, Python, GCP

We will be responsible for making the layout based on agreed mockups and making API calls to the backend. We will primarily use React.js and data visualization libraries like D3.js for interactive datasets.

Backend:
Kris: Python ML & Visualizations
Acelynn: Python, LLM

The backend will be written in Python using the FastAPI library. We will use sqlite to access and query the local datasets and integrate Riot SSO so users can log in using their Riot accounts.

We also plan to train and integrate an LLM to give recommendations that are tailored to Valorant. If any data needs to be cleaned and transformed into a different format (CSV to SQL), we will also handle that.

We plan to split the work of the actual app, but we will work on the design components (Stage 2 and Stage 3) together.