Work In Progress Disclaimer:

PROJECTS ARE FUNCTIONAL AND SLIDES ARE SUFFICIENT BUT ARE NOT FINALIZED; FUTURE UPDATES PENDING

Valorant Meta Analysis Series

ANALYSIS OF PREVAILING TRENDS AND STRATEGIES IN VALORANT

BY: BINARYXX SUNE

Table of Contents

Projects based on the Ranked Population

- Data collection through a REST API
- Interactive dashboard for agent prevalence
- Correlation between skill and performance metrics

Projects based on the Professional Population

- Data collection through web scraping
- Static visualization for agent prevalence

Changes to Implement

- Improvement for enhanced quality
- New features

Ranked Data Collection

IDENTIFYING, COLLECTING, AND STORING INFORMATION FROM THE RANKED PLAYER BASE

Use of Henrik-3’s Unofficial Valorant API

Henrik-3’s unofficial Valorant API provided the necessary data on player matches

- Included player ranks, team composition, match outcome, utility usage, and performance metrics

Implemented a Python script to collect a larger sample of data

- Henrik’s API only provided information on the player’s 5 most recent matches
- The script collected and stored new matches’ info
- Collected data for one month for a workable sample size

Agent Prevalence Dashboard

Allow users to select parameters to receive visualizations on agent prevalence

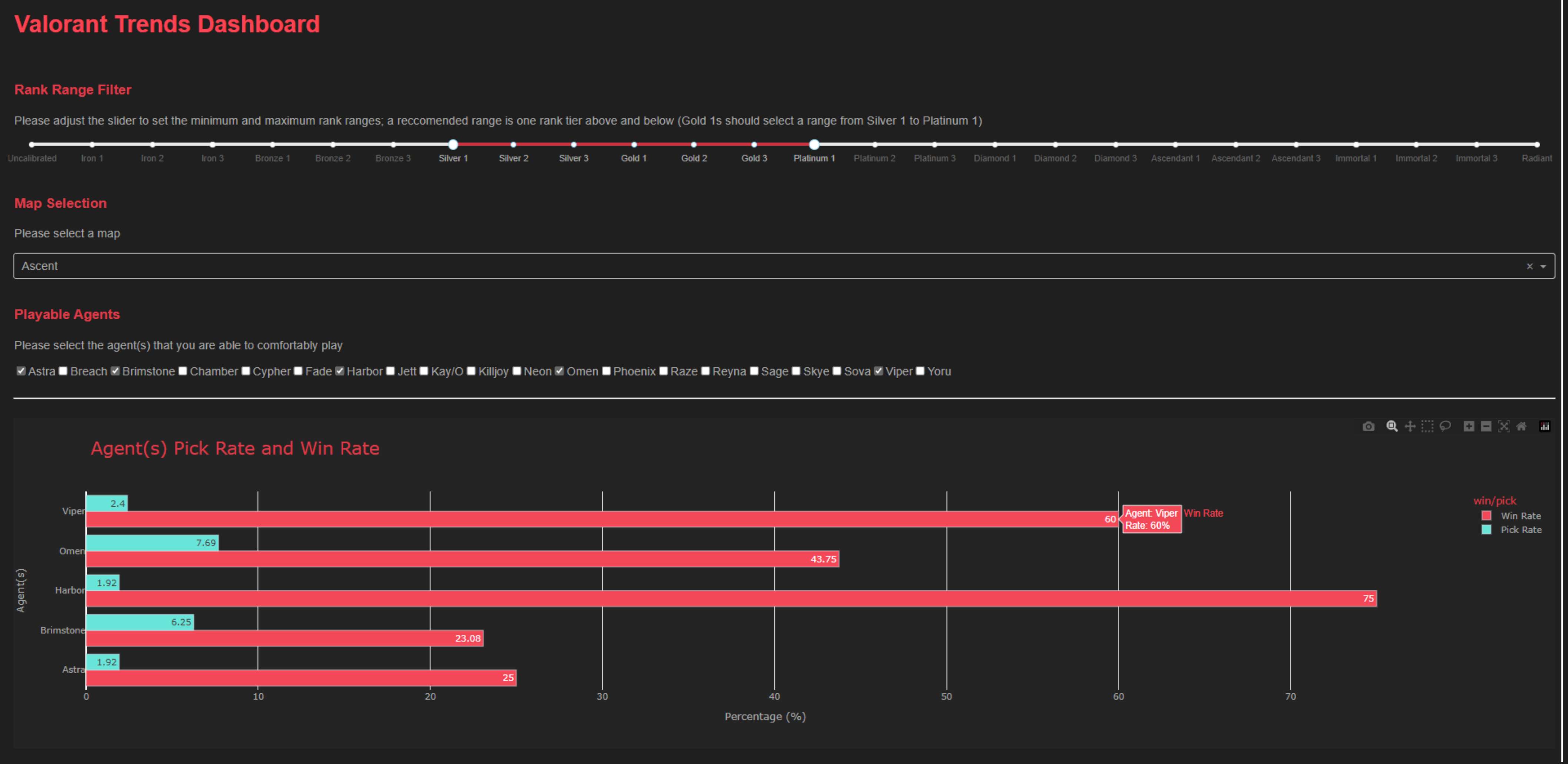
- Users can select the rank range, map, and playable agent pool
- Each selected agent’s pick rate and win rate graphed

The bar chart shows which agents are most popular and successful

- Another resource to utilize to aid decision making in team composition

Added an additional graph with the sample size

- Determine whether the stats provided are reliable
- Obsolete feature with larger data sets



Valorant Trends Dashboard

Rank Range Filter

Please adjust the slider to set the minimum and maximum rank ranges; a reccomended range is one rank tier above and below (Gold 1s should select a range from Silver 1 to Platinum 1)

Map Selection

Please select a map

Ascent

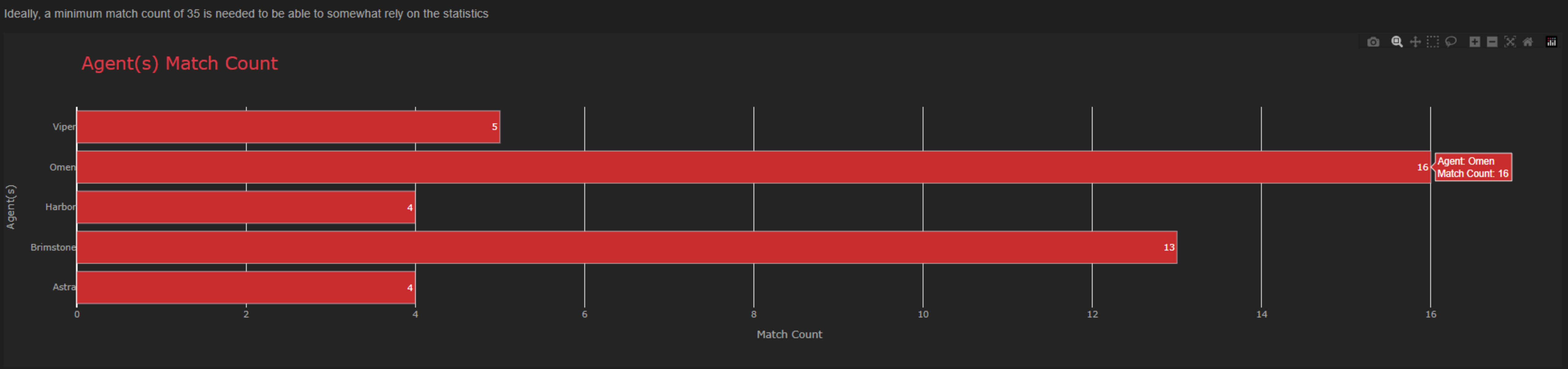
Playable Agents

Please select the agent(s) that you are able to comfortably play

☑ Astra ☐ Breach ☑ Brimstone ☐ Chamber ☐ Cypher ☐ Fade ☑ Harbor ☐ Jett ☐ Kay/O ☐ Killjoy ☐ Neon ☑ Omen ☐ Phoenix ☐ Raze ☐ Reyna ☐ Sage ☐ Skye ☐ Sova ☑ Viper ☐ Yoru

Agent(s) Pick Rate and Win Rate

Agent Prevalence Dashboard



Ideally, a minimum match count of 35 is needed to be able to somewhat rely on the statistics

Agent Prevalence Dashboard Add-On

Dashboard Interactivity in Action

Video link: https://youtu.be/oXSwWdHbggs

Performance and Rank Correlation

Identify which player metrics has a statistically significant correlation to rank

- Undermine toxic behavior by statistically backed claims

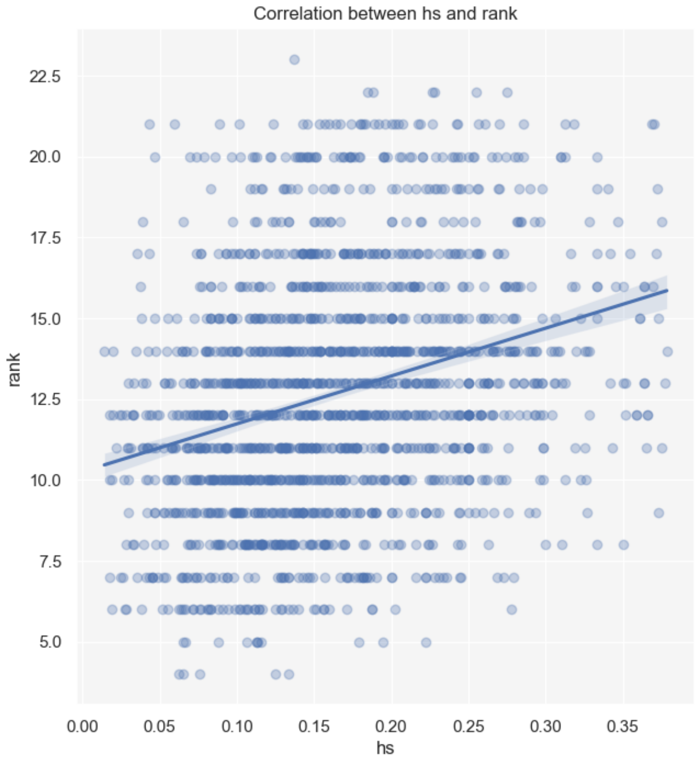
Most of the metrics were calculated from the stats provided

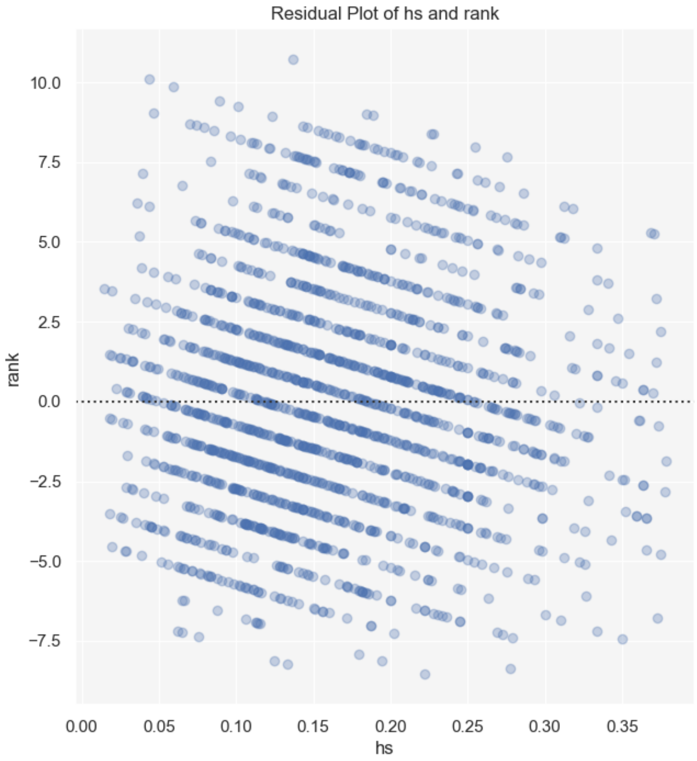
- Example: KAST (Kill/Assist/Survive/Trade) Score from event checks

Approaches used to prove statistical significance of correlation of rank and metric:

- Analyzing residual plots and checking p-values
- Comparing accuracy values of dummy classifier to machine learning algorithms
  - Dummy classifiers predict through random guesses
  - Machine learning algorithms predict through patterns

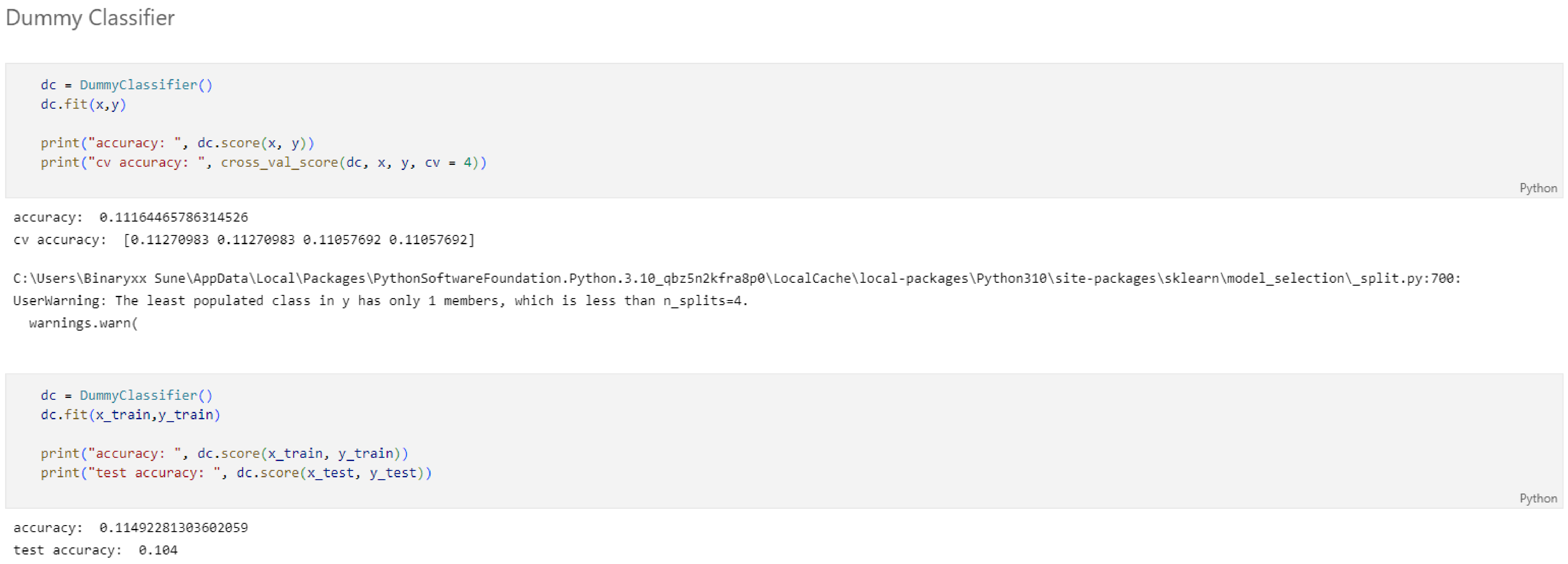
Used Residual Plots and Correlation Coefficients: P Value < α





```
The Pearson Correlation coefficient of hs and rank is 0.290755263718904  with a P-value of P = 8.118836630349276e-34
The Spearman Correlation coefficient of hs and rank is 0.30674675777016136  with a P-value of P = 1.24561263726667e-37
The Kendall Tau coefficient of hs and rank is 0.21177881972184628  with a P-value of P = 9.987456582216967e-36
Test Statistic:  13813.5542088365
P Value:  2.131279171769634e-31
```

Dummy Classifier as a Control Group Prediction: 10.4% Accurate Prediction



Dummy Classifier

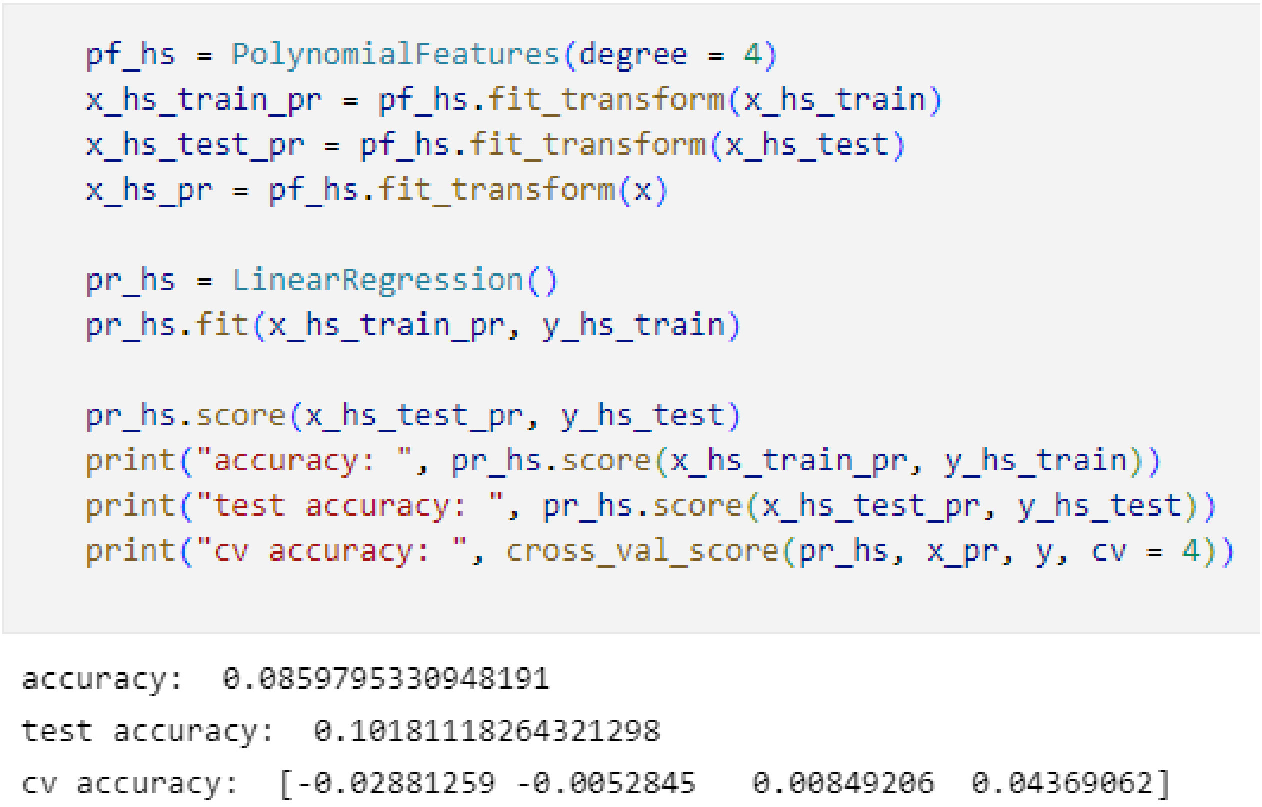
```python
dc = DummyClassifier()
dc.fit(x,y)

print("accuracy: ", dc.score(x, y))
print("cv accuracy: ", cross_val_score(dc, x, y, cv = 4))
```

```
accuracy:  0.11164465786314526
cv accuracy:  [0.11270983 0.11270983 0.11057692 0.11057692]

C:\Users\Binaryxx Sune\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-packages\Python310\site-packages\sklearn\model_selection\_split.py:700:
UserWarning: The least populated class in y has only 1 members, which is less than n_splits=4.
  warnings.warn(
```

```python
dc = DummyClassifier()
dc.fit(x_train,y_train)

print("accuracy: ", dc.score(x_train, y_train))
print("test accuracy: ", dc.score(x_test, y_test))
```

```
accuracy:  0.11492281303602059
test accuracy:  0.104
```

Polynomial Regression Predicts Rank with Headshot Percentage: 10.1% Accuracy



```python
pf_hs = PolynomialFeatures(degree = 4)
x_hs_train_pr = pf_hs.fit_transform(x_hs_train)
x_hs_test_pr = pf_hs.fit_transform(x_hs_test)
x_hs_pr = pf_hs.fit_transform(x)

pr_hs = LinearRegression()
pr_hs.fit(x_hs_train_pr, y_hs_train)

pr_hs.score(x_hs_test_pr, y_hs_test)
print("accuracy: ", pr_hs.score(x_hs_train_pr, y_hs_train))
print("test accuracy: ", pr_hs.score(x_hs_test_pr, y_hs_test))
print("cv accuracy: ", cross_val_score(pr_hs, x_pr, y, cv = 4))
```

```
accuracy:  0.0859795330948191
test accuracy:  0.10181118264321298
cv accuracy:  [-0.02881259 -0.0052845   0.00849206  0.04369062]
```

General Conclusion

THERE IS A CORRELATION BETWEEN HEADSHOT PERCENTAGE AND RANK, BUT ACCURATE PREDICTIONS ARE DIFFICULT

Pro Data Collection

IDENTIFYING, COLLECTING, AND STORING INFORMATION FROM PROFESSIONAL PLAYERS

Web Scraping for Tournament Results

Web scraped VLR.gg, a Valorant esports coverage website

- Provided match results and details for each tournament
- Limited matches collected to those with the current game patch

Focused on collecting team composition and match outcome of professional matches

- Provide insight in optimal strategies with skill as a nonfactor

Prevalent Pro Team Composition List

Create a script which lists the most popular team compositions for each map

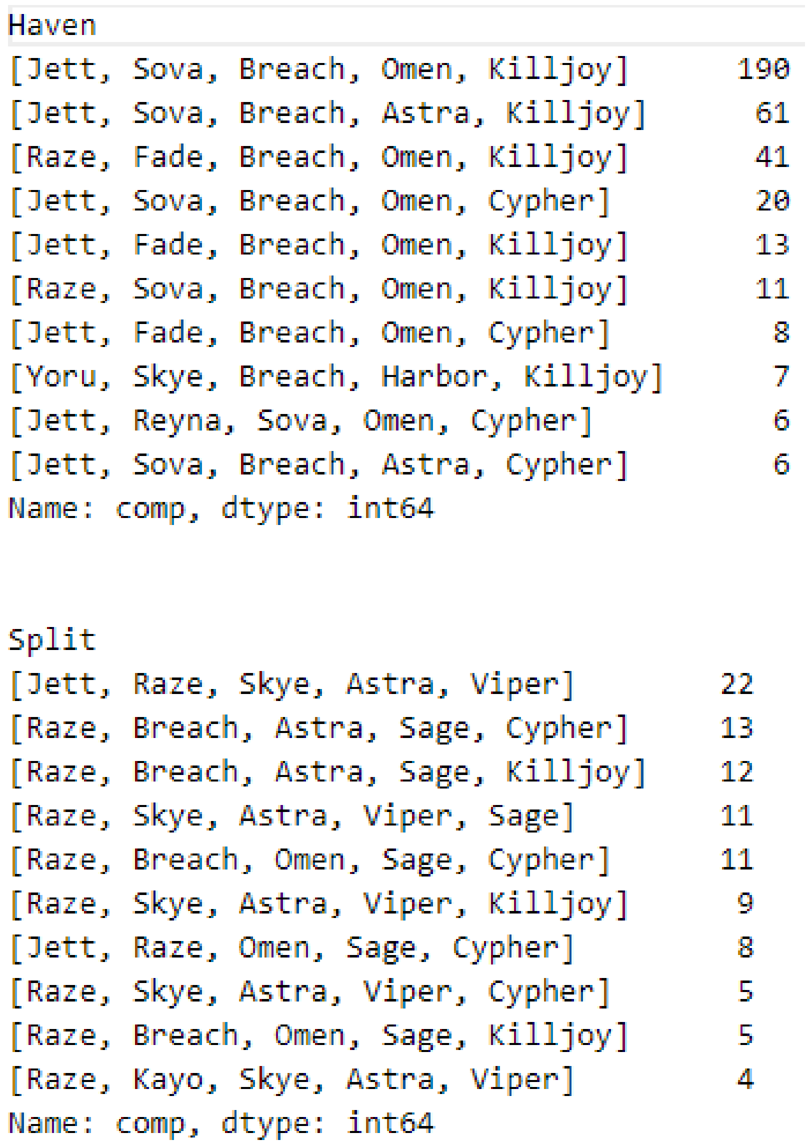
- Indicates key strategies implemented based on the most popular agents selected
  - Specific abilities are required for certain strategies to be implemented

Data collected does not distinguish agent order selection

- Teams with the same agents but in different order is considered different
  - Modify all entries to follow the same order to avoid duplicates

Prevalent Pro Team Composition List

Screenshot only includes two maps



```
Haven
[Jett, Sova, Breach, Omen, Killjoy]      190
[Jett, Sova, Breach, Astra, Killjoy]      61
[Raze, Fade, Breach, Omen, Killjoy]       41
[Jett, Sova, Breach, Omen, Cypher]        20
[Jett, Fade, Breach, Omen, Killjoy]       13
[Raze, Sova, Breach, Omen, Killjoy]       11
[Jett, Fade, Breach, Omen, Cypher]         8
[Yoru, Skye, Breach, Harbor, Killjoy]      7
[Jett, Reyna, Sova, Omen, Cypher]          6
[Jett, Sova, Breach, Astra, Cypher]        6
Name: comp, dtype: int64
```

```
Split
[Jett, Raze, Skye, Astra, Viper]        22
[Raze, Breach, Astra, Sage, Cypher]     13
[Raze, Breach, Astra, Sage, Killjoy]    12
[Raze, Skye, Astra, Viper, Sage]        11
[Raze, Breach, Omen, Sage, Cypher]      11
[Raze, Skye, Astra, Viper, Killjoy]      9
[Jett, Raze, Omen, Sage, Cypher]         8
[Raze, Skye, Astra, Viper, Cypher]       5
[Raze, Breach, Omen, Sage, Killjoy]      5
[Raze, Kayo, Skye, Astra, Viper]         4
Name: comp, dtype: int64
```

Agent Prevalence in the Pro Scene

Script which visualizations on the pick rate and win rate of an agent

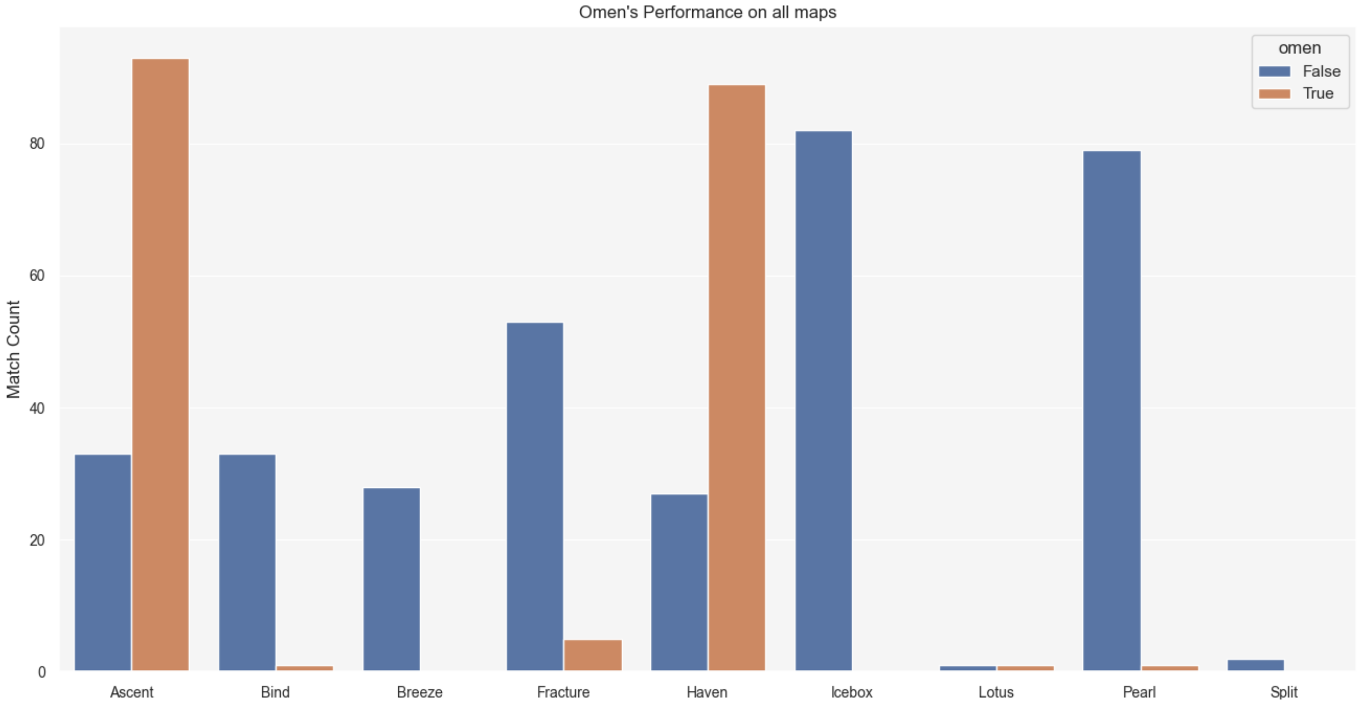
- Similar purpose as the dashboard but with pro players
- Also provides win rate if the agent was absent in a team
    - Implies the impact of an agent’s presence

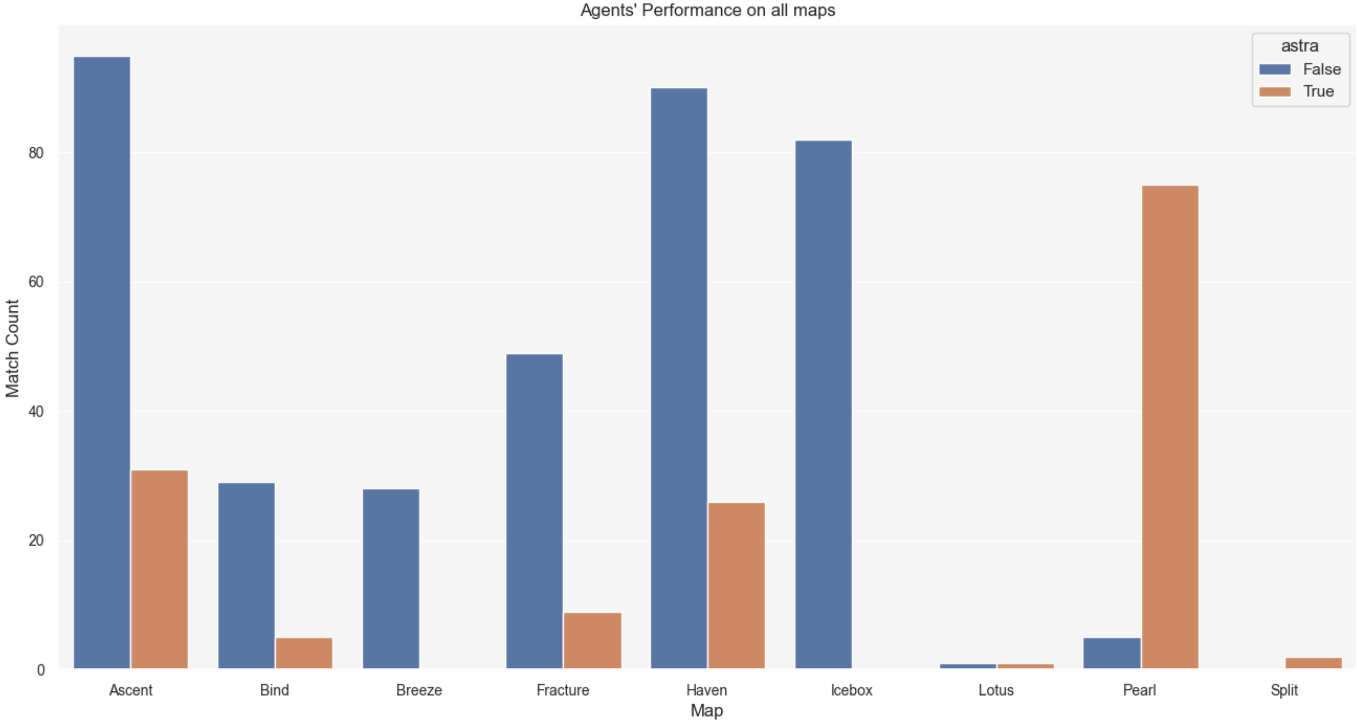
Currently not set up to be a dashboard

- Requires manual changes to the code to select other agents

Agent Prevalence in the Pro Scene

Screenshots of outputs from different scripts for different agents





Changes to Implement

IMPROVEMENTS FOR ENHANCED QUALITY AND NEW FEATURES

Improvements for Enhanced Quality

Improve the dashboard’s UX

- Better color palette for visibility
- Better composition for readability
  - More padding in elements, larger fonts, better space management

Detailed and elaborated walkthrough of rank and performance correlation

- Include necessary background for context
- More detailed explanations of the process

More aesthetically pleasing output for the team composition list

Concise yet detailed presentation slides

Larger sample size for better reliability on trends and observations

New Features for Added Value

Interactive dashboard for agent prevalence in the pro scene

- Similar to the preexisting dashboard for easy use

Breaking down the team composition further by abilities

- Agents are selected for their combination of abilities available
- Categories abilities for its function and list prevalence in each map
  - Allows greater freedom in agent selection without a rigid team line up

Automatic execution of data collection scripts

- Currently manually executed daily
- Capture accurate and up-to-date trends
  - Remove entries prior to balance updates and map/game changes for relevance