Beyond Usage Statistics: Predicting Top-Cut Performance in Pokémon VGC Regulation H Through Ensemble Learning

Final report for MATH 8710 - Introduction to Machine Learning

Jacob Manning

December 11, 2025

1 Introduction

In this project, we will try to be the very best, like no one ever was. To reach this goal, we will use machine learning techniques in order to predict what would be the six best Pokémon that you could bring to a tournament in order to give you the best chance of winning. The Pokémon Company (TPC) every year holds a competitive circuit called the video game championship (VGC) with various tournaments ranging in size from "locals", to the world champion tournament that happens annually in August. The most popular format by far is the masters level (generally 18 years and older) double battles, which is a best of three which allows trainers to bring four out of six Pokémon to each battle. We will denote "the best" teams as those that reach the top-cut of these tournaments. Throughout each battle, two out of the four Pokémon that each trainer brought are on the field at a time. This adds a lot of complexity that is not seen in a single battle format. Additional complexity is added when you consider the fact that each Pokémon has been trained in completely different ways. This is seen in move choice, item choice, abilities, effort values, and individual values and this is not even considering any synergy that the team itself contains. This problem has much of the complexity, and to a lesser extent, reward that is present in problems such as market predictions or gambling forecasts. However, I find this problem much more interesting as it is a niche that I personally enjoy following.

Usually TPC develops new games every few years and the VGC cycle starts with Regulation A where the only legal Pokémon are the basic non-legendaries, then gradually increasing the power level until nearly every Pokémon in the game is legal. During this cycle however, after the 2024 World VGC tournament, there was one format that reduced the number of legal Pokémon (which reduced the power level). Many players said that this format, called Regulation H, felt like the first regulation that happened in the newest games. Later, after the 2025 World VGC tournament, Regulation H was revisited for official tournament play. As Regulation H is so similar to Regulation A, and has had far more time to be played than any other regulation, it is unique in the fact that many of the strategies have been cemented and had enough time to be established as "the best" in the current meta. This gives a huge amount of data to be analyzed, far more than for any other regulation that has been played in any other Pokémon game.

In this project, we have sourced the usage data of Pokémon used during Regulation H in 2024 and 2025 until October, from a fan site called "Smogon"[10] where trainers practice for official tournaments. This usage data is based off of 1,384,370 best of three battles that has been weighted toward current usage to reflect meta shifts that have occurred. We will also only be considering

Pokémon that had more than a five percent usage stat in any given month during the regulation (43 Pokémon in total). Additionally, tournament results will be taken from "nimbasacitypost.com"[6] where our training data will be every Regulation H tournament that occurred before October 31st 2025. The actual information used to analyze how viable a Pokémon is, its moves, base stats, type, etc., was sourced from "pikalytics.com"[7] and the "PokéAPI"[8]. As this particular API was not updated to reflect all of the changes made to the current generation of games, we have supplemented this information from a fan site called "Serebii"[9]. It should be noted that this analysis is also a simplification of the real game as there are many choices that each trainer has that have been ignored. For example, in the past, trainers have intentionally reduced a Pokémon's stats to under-speed their opponents. These complexities will not be explored.

As we are analyzing "the best" teams, our goal is to try to predict what a top-cut team would look like in a tournament setting. This is a classification problem of how far a team reached in a tournament. As we mentioned before, the way that a Pokémon is trained has a big impact on the performance. This causes the problem to be highly dimensional by nature. Additionally the problem is suspected to be highly non-linear as many binary categorizations along with numeric classifications are made in rating any given team. The model that was chosen to tackle this problem is a very famous machine learning ensemble approach, namely random forests. This was chosen to try to tackle the high dimensionality as well as the non-linearity. One of the most interesting results we suspect will come from looking at feature importance. As some of the data will come from analyzing top-cuts of tournaments, there will be a class imbalance. To this point, we will use metrics that are robust to class imbalance such as the F1 score.

In the end, this project will be an example of the machine learning pipeline from concept to data acquisition, handling, model building, and finally evaluation and interpretation. Even though this problem is fairly unimportant to the world's collective knowledge, it does contain a lot of complexity that is similar to constructing a profitable ETF or choosing which players to put on your fantasy sports team. This work could also give insights to VGC players through results such as feature importance and might develop connections that were previously undiscovered.

2 Related Work

Competitive games have always caught interest of humans and there has always been an interest in predicting the winner of tournaments. This has been seen in sports and in the newly popular e-sport tournaments. A common genre that has been studied is what is called a MOBA, we can see in "Win prediction in multiplayer esports: Live professional match prediction" by Hodge et. all[4] that match prediction can be made from performance and information gathered before a match has started. This is no different than stats that are used for betting on games from the MLB, NBA, or NFL. In fact predictive analytic model forecasting such as machine learning techniques, namely random forest classification, was used in "A predictive analytics model for forecasting outcomes in the National Football League games using decision tree and logistic regression" by Matt Gifford and Tuncay Bayrak.[3]

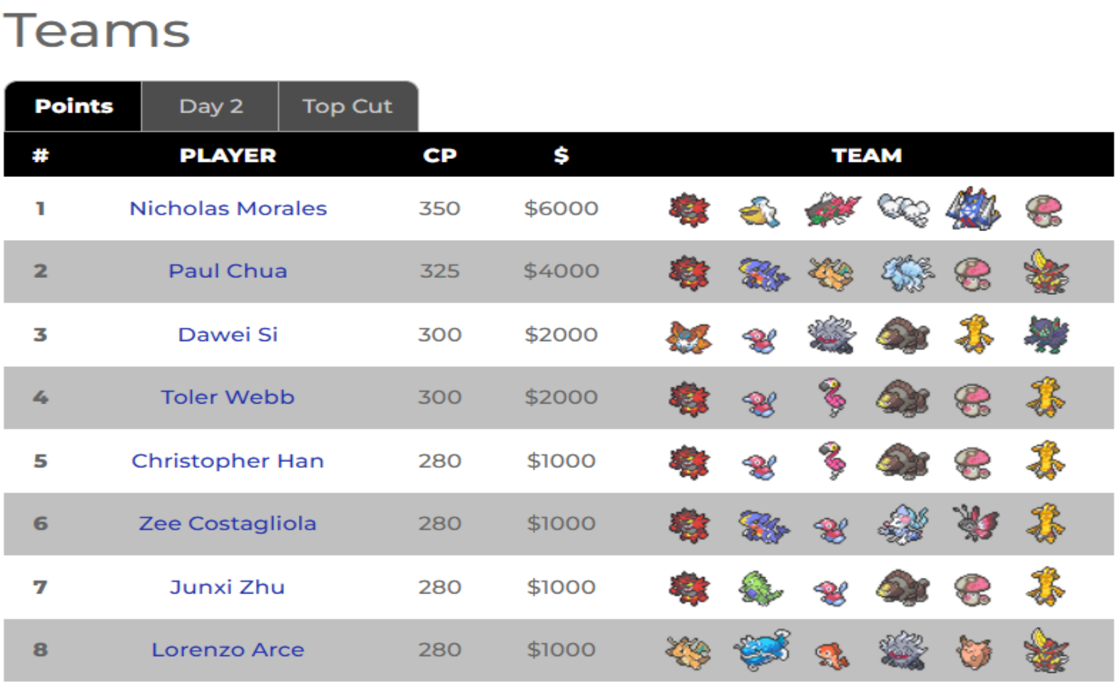
Some of the limitations that we expected to encounter are related to player performance and luck. There are many parts of Pokémon that cannot be quantified at the team building stage. This can be seen in "Analysis of video game players' emotions and team performance: An esports tournament case study" by Abramov et. all where they note how a player's emotions can have a huge impact on the performance of their team. In a single game like Pokémon, this is all more pronounced as it is entirely on them to perform. These emotions can be varied such as confidence or anger either from past performance or something they saw as unlucky. Luck is actually a major issue in

these predictive model. As noted in “A Multi-Agent Pokemon Tournament for Evaluating Strategic Reasoning of Large Language Models” by Tadishetty, Sai Yashwanth and Dhatri, C.[11] “Pokémon battles offer [...] a blend of deterministic relationships and stochastic elements.” This inherently places a limit on how well our model will perform as we are trying to predict tournament placements without any of these stochastic or lucky elements accounted for such as critical hits or high/low damage roles.
On the other hand, we also tried to frame this in terms of a portfolio selection. This is a novel idea that tries to pick which Pokémon would give the best return. To rephrase Harry Markowitz’s work on portfolio management in his landmark paper on the topic[5] in terms of Pokémon team building, “One type of rule concerning choice of [a team] that the [player] does (or should) maximize [is] the [...] value of future returns. Since the future is not known with certainty, it must be ”expected” or ”anticipated’ returns which we [value].” In layman’s terms, we should pick the best Pokémon that will help us maximize the chance of winning a tournament. This gives a guiding principle in building this model. We will always choose the smartest decision that we could to give us the best expected return.

3 Methods

3.1 Data

Sourcing
Almost all of the data for this project was scraped from “nimbasacitypost.com”[6] as this website had most of the information about the tournaments. On each page corresponding to the tournament, there was a link to a cite called “pokepaste.es”[2] which is the format used by each competitor to relay information about their team. The quality of the pokepastes varied widely. All pokepastes had each Pokémon’s item, tera type, and moves. However, some pokepastes had specific IV and EV sets. There was another fan project that had additional pokepastes with EV sets listed that supplemented the training data. When this information was not given we assumed that the sets used were common sets that were found on “pikalytics.com”[7].



Teams

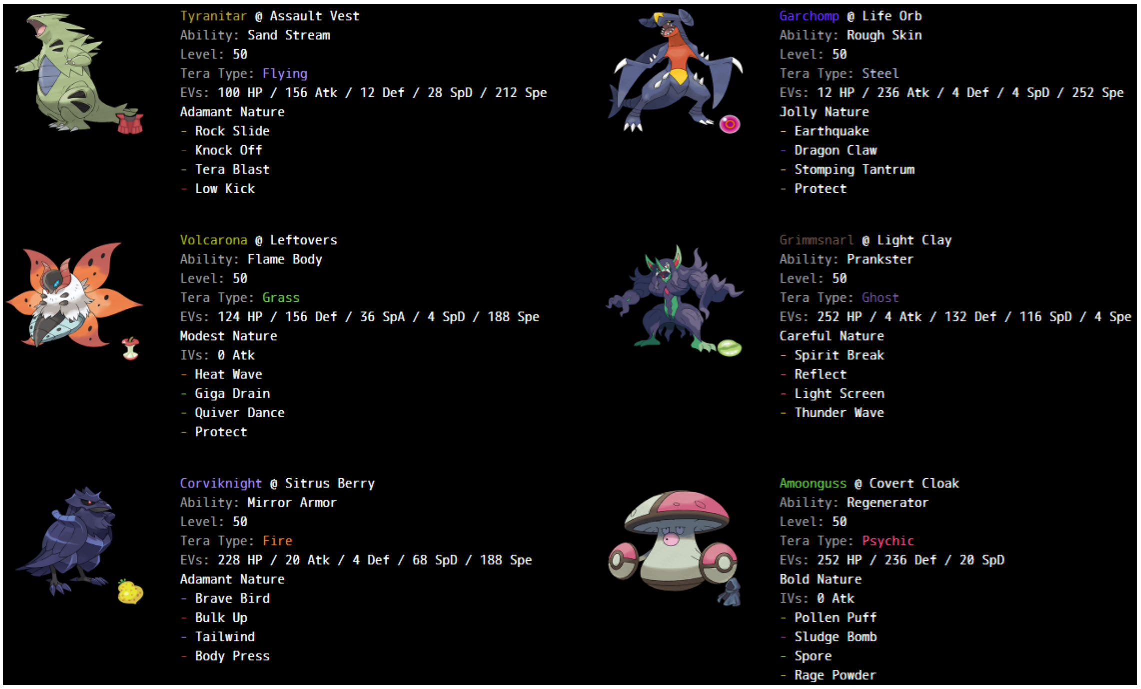
#	PLAYER	CP	$	TEAM
1	Nicholas Morales	350	$6000	
2	Paul Chua	325	$4000	
3	Dawei Si	300	$2000	
4	Toler Webb	300	$2000	
5	Christopher Han	280	$1000	
6	Zee Costagliola	280	$1000	
7	Junxi Zhu	280	$1000	
8	Lorenzo Arce	280	$1000	

A snapshot of the Top 8 from Nimbasa City Post of the tournament results for the 2025 Baltimore Regionals.

There were two sources that were used to get information about a Pokémon that were used in this model. The first is from a cite called Pikalytics, specifically from the Regulation H Best of 3 section. This cite has the stats of each Pokémon, a list of the common items, moves, abilities, and natures with EV spreads. In the event that a pokepaste was not found that listed their EVs, spreads were chosen by the weights on this cite. The other font of information was from the PokeAPI[8]. This API is outdated and needed to be supplemented with information from “serebii.net”[9] for moves and abilities that were not present in the ninth generation of the games.

Train test

In total, the training set had 3,319 teams and how many championship points they received in the tournament, which is a common metric between regional, special event, and international tournaments. We assumed that a similar point structure would be applied to the premier league tournaments and assigned championship points for each placement in these tournaments. We used our model to predict the placement of each team that was reported for the Latin American International Championship that occurred in November of 2025. The size of the test series was 250 teams that were reported for this tournament.



```
Tyranitar @ Assault Vest
Ability: Sand Stream
Level: 50
Tera Type: Flying
EVs: 100 HP / 156 Atk / 12 Def / 28 SpD / 212 Spe
Adamant Nature
- Rock Slide
- Knock Off
- Tera Blast
- Low Kick
```

```
Garchomp @ Life Orb
Ability: Rough Skin
Level: 50
Tera Type: Steel
EVs: 12 HP / 236 Atk / 4 Def / 4 SpD / 252 Spe
Jolly Nature
- Earthquake
- Dragon Claw
- Stomping Tantrum
- Protect
```

```
Volcarona @ Leftovers
Ability: Flame Body
Level: 50
Tera Type: Grass
EVs: 124 HP / 156 Def / 36 SpA / 4 SpD / 188 Spe
Modest Nature
IVs: 0 Atk
- Heat Wave
- Giga Drain
- Quiver Dance
- Protect
```

```
Grimmsnarl @ Light Clay
Ability: Prankster
Level: 50
Tera Type: Ghost
EVs: 252 HP / 4 Atk / 132 Def / 116 SpD / 4 Spe
Careful Nature
- Spirit Break
- Reflect
- Light Screen
- Thunder Wave
```

```
Corviknight @ Sitrus Berry
Ability: Mirror Armor
Level: 50
Tera Type: Fire
EVs: 228 HP / 20 Atk / 4 Def / 68 SpD / 188 Spe
Adamant Nature
- Brave Bird
- Bulk Up
- Tailwind
- Body Press
```

```
Amoonguss @ Covert Cloak
Ability: Regenerator
Level: 50
Tera Type: Psychic
EVs: 252 HP / 236 Def / 20 SpD
Bold Nature
IVs: 0 Atk
- Pollen Puff
- Sludge Bomb
- Spore
- Rage Powder
```

An example of pokepaste. In this report, all information to reproduce the team was given including EVs, natures, and IV adjustments.

Features

As there are many aspects of Pokémon that cannot be predicted before a tournament like which teams someone will face or the inherent luck in the game, we focused on features that would reflect aspects of team building that are important to consider. In this section we will talk about how different features were selected and the way they are used in the model. Each feature could fall into three different groups, Pokémon specific features, inter-team features, and team categorizing features. In total fifteen features were designed to try to describe how well the team was built. We will start by talking about the Pokémon specific features.

Each Pokémon has its own typing for a maximum of two types per Pokémon. The way that these types play together can have a big impact on how a Pokémon plays. Offensively, if a Pokémon uses a move that is of the same type as itself, the power of the move receives a bonus called a same type attack bonus or STAB. Defensively, each type can either be immune, resist, or be weak against each other type (check the type chart for more details). Thus, the amount of types a Pokémon can resist or hit super effectively with STAB is of utmost importance when building a team. These are the first two features that were introduced into the model. In the current generation of Pokémon, terastalization was introduced to spice up offensive and defensive play. This feature was hard to balance as there is a lot of nuance that cannot be put into a program that is behind which tera type was chosen. The next Pokémon specific feature is related to the item that it is holding. Each item has a (usually) unique effect that we do not have time to go over in this report, but this feature gives bonuses for strong items that are held by a Pokémon. Lastly, a Pokémon can learn a maximum of four moves. These moves can either attack or support. As such, the move score could be viewed as a inter-team or Pokémon specific feature. Moves were rated by either how much damage they could deal to the opposing team, or how well they support their own team. As the items feature, we cannot go over every move. However, notable move categories would be pivoting moves, screen moves, speed control moves, spread moves, and status dealing moves.

The inter-team specific features are at the same time more basic and more involved. Some of the basic features would be things like average base stat total, mean and standard deviation of the speed stat. All of these features give a measure of how strong a Pokémon is and when they go in the turn order. The standard deviation of the speed stat is notable as having a wide range of speed stats can be used to defend against a common stat that reverses the turn order. Another feature considered are game mechanics that control the speed of each team. As a team is built, it is important to make sure that each member is not weak to the same type so that you won't be bulled over by one Pokémon. This is commonly referred to as core synergy. This feature shows measures how many Pokémon on a team cover the weaknesses of their team members. Another important aspect to consider is status prevention, specifically to sleep. Putting your all of your opponent's Pokémon to sleep can severely reduce the amount of answers that your opponent has to counter act your team. Thus sleep prevention is one of the important features to consider when building a team.

Lastly, the team categorizing features are more broad features. These things are usually related to the presence of weather or terrains on a team as their presence can limit the ways a team is built. We also looked at how many Pokémon were popular in the meta and which ones were off-meta picks, each having their own features. The last feature was added to try to show the importance of specific combinations that could not be shown in the other features. For example, Tatsugiri and Dondozo are two Pokémon that when used together get a large bonus that could not be shown in any of the other features. **Preprocessing**

Ultimately, we chose to try to categorize each team into four groups being top 256, top 128, top 64, and top 32. As the size of each group halves at every step, our training data is severely imbalanced as the majority of the teams do not get to the higher rounds of tournament play. To counteract this issue, we used SMOTE to try to add artificial resampling of higher placing teams. Additionally, when running the model, we used RFE to narrow the features used in training the model down to ten features. This was in an effort to avoid interdependence in the features. These steps occurred within the pipeline in order to try and limit data leakage.

3.2 Models and Algorithms

The machine learning model that was chosen for this project is a common ensemble method known as random forests. This was chosen as the problem being approached is high dimensional and has binary and continuous features. We decided to use scikit-learn and a package called imbalanced-learn for their pipeline and SMOTE functions. The pipeline was constructed to try to avoid data leakage and SMOTE was used as previously mentioned to help with the large class imbalance inherently present in a tournament setting. There were a few hyper-parameters that were adjusted in the model. These were mainly the number of trees in the forest, max depth, minimum sample split, minimum sample leaves, and class weight. We did sort the data into differing bins depending on their placement, we found that four classes with a k-means strategy provided the best results. Other hyper-parameters will be shown for comparison. There were a few features that were not being used consistently and thus we decided to use RFE to increase the metrics of the model.

3.3 Evaluation

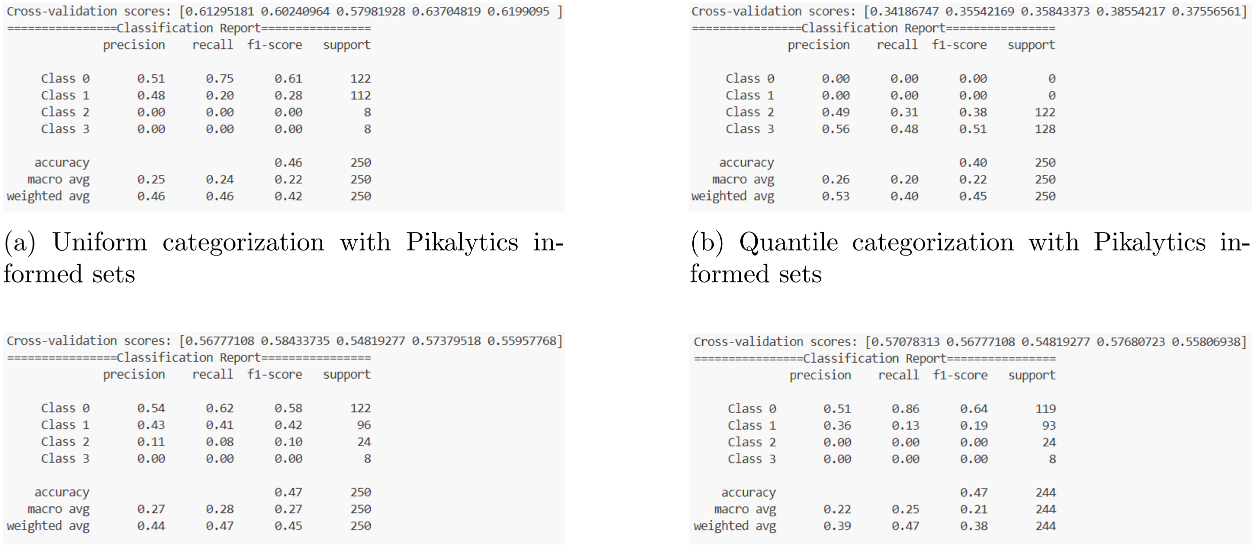
As the model is based on a random forest classification method, we decided to look at accuracy, precision, recall, and F1 metrics. These are common metrics that are used when looking at classification problems. In particular the F1 score is the most important for these problems. We also used stratified cross validation to look for consistency across the model. The scores were very similar and similar to the overall reported accuracy of the model.

Our training set consisted of tournaments in Regulation H that occurred before October 31st 2025 and the test data came from the Latin American International Championship 2026 tournament that occurred in November of 2025. To mitigate data leakage, a pipeline was used for preprocessing and validation.

We used a naïve comparison of randomly guessing categories as well as randomly choosing sets without any prior information about usage statistics from Pikalytics for comparison. We then compared the categorization methods and the summary statistics from the classification report as well as how well distributed the support of the problem was.

4 Results

In Figure 1 are the cross-validation (CV) scores along with the classification report for each model. The CV scores were used to try and show that the accuracy score was roughly similar across multiple runnings of the model. Consistently, the CV scores were higher than the accuracy that was in the classification report. We tried to mitigate this by using a pipeline to avoid data leakage. The four results showed are three different categorization strategies being, uniform, quantile, and kmeans while using pikalytics informed sets and the last being kmeans with randomly chosen sets. In Figure 2 the respective confusion matrices are also presented. The best performing overall model had an overall accuracy of 47% with the recall and F1 scores for the first two classes being the largest. Each class represents different tiers of placement in the tournaments.



```
Cross-validation scores: [0.61295181 0.60240964 0.57981928 0.63704819 0.6199095 ]
===============Classification Report===============
              precision    recall  f1-score   support

     Class 0       0.51      0.75      0.61       122
     Class 1       0.48      0.20      0.28       112
     Class 2       0.00      0.00      0.00         8
     Class 3       0.00      0.00      0.00         8

    accuracy                           0.46       250
   macro avg       0.25      0.24      0.22       250
weighted avg       0.46      0.46      0.42       250
```

(a) Uniform categorization with Pikalytics informed sets

```
Cross-validation scores: [0.34186747 0.35542169 0.35843373 0.38554217 0.37556561]
===============Classification Report===============
              precision    recall  f1-score   support

     Class 0       0.00      0.00      0.00         0
     Class 1       0.00      0.00      0.00         0
     Class 2       0.49      0.31      0.38       122
     Class 3       0.56      0.48      0.51       128

    accuracy                           0.40       250
   macro avg       0.26      0.20      0.22       250
weighted avg       0.53      0.40      0.45       250
```

(b) Quantile categorization with Pikalytics informed sets

```
Cross-validation scores: [0.56777108 0.58433735 0.54819277 0.57379518 0.55957768]
===============Classification Report===============
              precision    recall  f1-score   support

     Class 0       0.54      0.62      0.58       122
     Class 1       0.43      0.41      0.42        96
     Class 2       0.11      0.08      0.10        24
     Class 3       0.00      0.00      0.00         8

    accuracy                           0.47       250
   macro avg       0.27      0.28      0.27       250
weighted avg       0.44      0.47      0.45       250
```

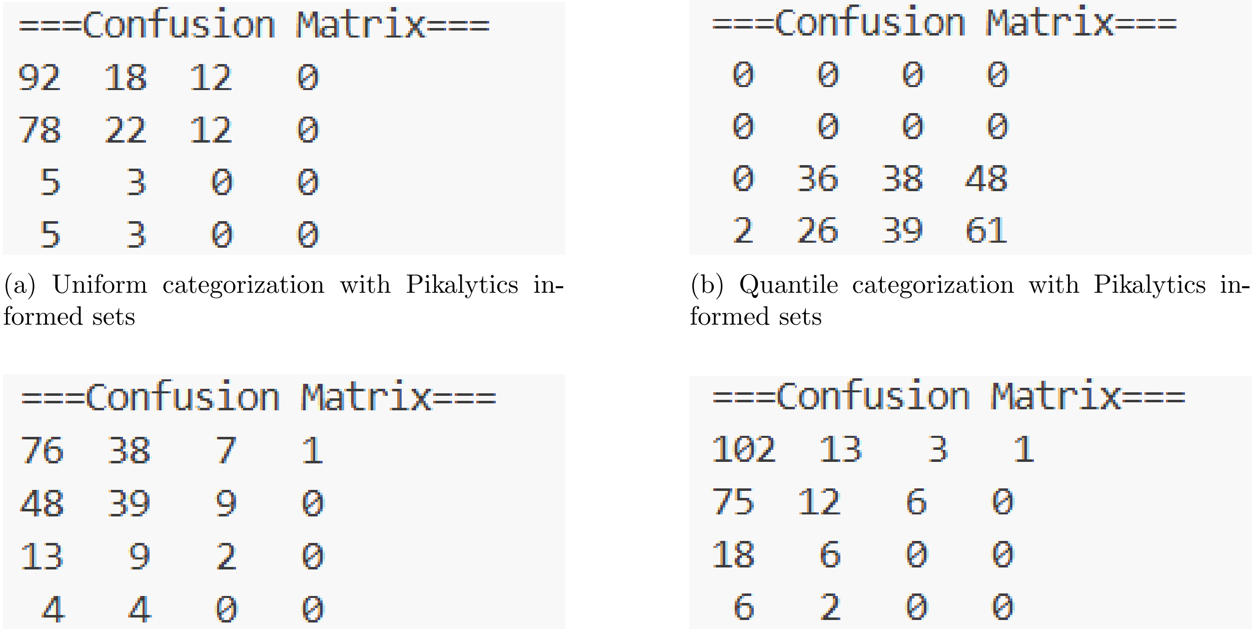
(c) kmeans categorization with Pikalytics informed sets

```
Cross-validation scores: [0.57078313 0.56777108 0.54819277 0.57680723 0.55806938]
===============Classification Report===============
              precision    recall  f1-score   support

     Class 0       0.51      0.86      0.64       119
     Class 1       0.36      0.13      0.19        93
     Class 2       0.00      0.00      0.00        24
     Class 3       0.00      0.00      0.00         8

    accuracy                           0.47       244
   macro avg       0.22      0.25      0.21       244
weighted avg       0.39      0.47      0.38       244
```

(d) kmeans categorization with random sets

Figure 1: Four different results with different hyper-parameters

Below in Figure 3 is the feature importance for the overall best performing model, the kmeans categorization with Pikalytics informed sets. These features were chosen using RFE out of 15 total possible features.



```
===Confusion Matrix===
92  18  12   0
78  22  12   0
 5   3   0   0
 5   3   0   0
```

(a) Uniform categorization with Pikalytics informed sets

```
===Confusion Matrix===
 0   0   0   0
 0   0   0   0
 0  36  38  48
 2  26  39  61
```

(b) Quantile categorization with Pikalytics informed sets

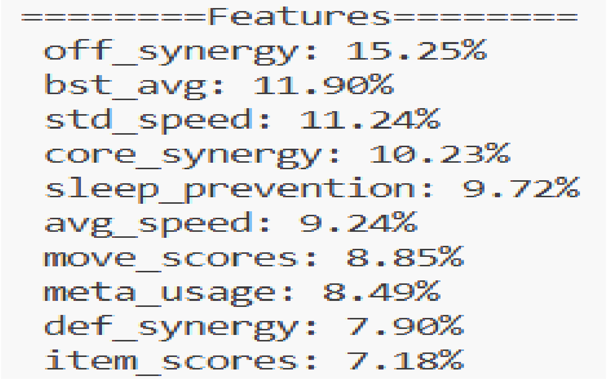
```
===Confusion Matrix===
76  38   7   1
48  39   9   0
13   9   2   0
 4   4   0   0
```

(c) kmeans categorization with Pikalytics informed sets

```
===Confusion Matrix===
102  13   3   1
75  12   6   0
18   6   0   0
 6   2   0   0
```

(d) kmeans categorization with random sets

Figure 2: Four different confusion matrices with different hyper-parameters



```
========Features========
 off_synergy: 15.25%
 bst_avg: 11.90%
 std_speed: 11.24%
 core_synergy: 10.23%
 sleep_prevention: 9.72%
 avg_speed: 9.24%
 move_scores: 8.85%
 meta_usage: 8.49%
 def_synergy: 7.90%
 item_scores: 7.18%
```

Figure 3: Feature importance for the kmeans Pikalytics informed sets.

5 Discussion

There are a few interpretations that one could make with the results. The first and more optimistic interpretation would be that the teams that were misidentified in a higher class were built appropriately, but got unlucky and were eliminated earlier than anticipated. Conversely, those identified as lower classes that actually placed higher got very lucky this tournament. Unfortunately, this aspect of the game cannot be quantified and so it is difficult to say that this is the correct interpretation of the data. The more pessimistic interpretation is that there were many characteristics of the problem, like luck or player skill, that could not be analyzed in the model. This is the way that we will approach the results as we cannot say much more about the first perspective analytically.

Limitations and Looking Forward

In this project, we did not have all of the information about each team that was brought to every tournament. Information about the teams was supplemented by the most likely sets that were retrieved from Pikalytics. Although this is the best we could do, competition thrives on innovation and this approach stifles these possible mix ups that could give a competitor an edge against the rest of the players. Unfortunately, this limitation cannot be helped; as with any dataset, we live in an imperfect world with imperfect data.

As previously mentioned, luck and trainer skill are pretty major limitations of this model since it cannot be quantified. This lines up with the findings such as [1] and [11] Similarly, any strategies that are used outside of team building is not accounted for. This can lead to great teams getting eliminated early or bad teams going farther than anticipated, which can limit how well our model can be evaluated.

There are a lot of more complicated interactions in other regulations that made Regulation H uniquely easier to analyze than any other. For example, in an unrestricted format where legendary Pokémon are permitted, they have a much bigger impact on the meta that makes every team more similar than before. On the other hand, trying to refine and analyze other regulations could be an other project to look at in the future. Another approach that we could take is to try to model how far a team would get in a tournament instead of trying to analyzing the specific placement. This would take a more probabilistic approach than we currently present in this report. You could also build a trainer profile to let past performance at other tournaments inform bracket placements in the future.

CV Scores, Classification Reports, and Confusion Matrices

The CV scores had a low variance, but were consistently higher than the accuracy score in the classification report. This is probably due to data leakage that occurred during classification of the data. We tried to mitigate any leakage we could with a pipeline, but it appears there was still some issues unresolved.

To begin with the classification reports, we will talk about the uniform and quantile categorization models. These models have the same issues in different ways. Both models did not identifying which teams are top-tier, as we can see the class imbalance was too great which let the model guess that each team is in the first two or last two classes and increase those two classes metrics. This is not a good model as it is doing little more than guessing. Luckily, the kmeans strategy showed more promise.

Both kmeans strategies had a wider support of the data. This gave the models the chance to fix the major issues that were present in the first two models. By looking at the random sets, the overall accuracy is the same as the Pikalytics informed sets, however classes 2 and 3 received a 0 in all pertinent metrics and thus is lower in the macro and weighted averages. Finally, the kmeans Pikalytics informed sets performed the best with the highest distribution of support and the scoring the highest in each metric. However, we do admit that these numbers are not great, but with the inherent limits of the model due to properties of the game outside of the control of team building (luck), this is not a terrible result and is far better than randomly guessing the categories of each team.

From the confusion matrices, we see a similar story to each result as the classification report. The first two models picked one half of the classes and guessed everything was there. The second two models we see that there was a lot of misscategorization between the first two classes and barely any correct categorization in the last two classes. However, the Pikalytics informed model did outperform the random guessing overall.

Feature Analysis

From the start, we were initially interested in feature importance. Before starting we assumed that going fast, hitting hard, and using the most popular Pokémon would be the most important features. By looking at the filtered features that were chosen using RFE and their importance, our suspicions were correct as each of these initial guesses were present in the top ten most important features. Additionally we see that it is also important to have high base stats, have a wide range of speeds, use the most helpful items, make sure that your team can handle multiple offensive opposing types, don't pick Pokémon that have many type weaknesses on their own, and that you don't just get every Pokémon put to sleep. These are important findings that could help inform team building in the future even outside of this regulation. I was surprised in particular that weather and terrain were not a chosen features for this model as it was used heavily during this regulation. I assume that these two features compliment other chosen features and thus were discarded in favor of meta usage or move scores.

A Personal Note

Personally I learned a lot about gathering information and object-oriented programming (OOP). Everything that went into this model was scrapped from the internet which had a harsh learning curve in itself, but I also needed to learn about how to efficiently represent the teams and how different Pokémon interact with each other. This is a problem that is perfect for OOP and required multiple type classes. I also learned how important it is to save locally the information that you gather, especially when computational resources are limited. This lets you focus more time on building the model and less time on scraping the data each time you want to run a test.

6 Conclusion

Our model was inherently limited by factors that cannot be quantified such as luck and skill that are not present during team building. These factors can have a big impact on how well a team does at any tournament. That being said, there were models that performed better than others and gave us interesting information. Throughout this project, the most interesting findings were the feature importance results. This information could be used to inform anyone interested in becoming the very best. We found that the most important features are a combination of offense, and defense and are summarized in Figure 3. However, team building is just the first step in winning, and luck or skill are important factors to consider when entering the tournament.

References

[1] Simon Abramov, Alexander Korotin, Andrey Somov, Evgeny Burnaev, Anton Stepanov, Dmitry Nikolaev, and Maria A Titova. Analysis of video game players' emotions and team performance: An esports tournament case study. *IEEE Journal of Biomedical and Health Informatics*, 26(8):3597–3606, 2021.

[2] felixphew. PokePaste, 2025. A dedicated pastebin service for sharing, formatting, and viewing competitive Pokémon teams.

[3] Matt Gifford and Tuncay Bayrak. A predictive analytics model for forecasting outcomes in the national football league games using decision tree and logistic regression. *Decision Analytics Journal*, 8:100296, 2023.

[4] Victoria J Hodge, Sam Devlin, Nick Sephton, Florian Block, Peter I Cowling, and Anders Drachen. Win prediction in multiplayer esports: Live professional match prediction. *IEEE Transactions on Games*, 13(4):368–379, 2019.

[5] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[6] Nimbasa City Post. Pokémon VGC Resources, 12 2019. A curated list of websites, tools, and content creators for competitive Pokémon VGC.

[7] Pikalytics. VGC 2025 Best of 3 Regulation Set H Usage Stats, 2025. Live usage statistics for competitive Pokémon battles in the specified format.

[8] PokeAPI. PokéAPI, 2024. A free, open-source RESTful API for detailed Pokémon game data.

[9] Serebii. Serebii.net Scarlet & Violet Attackdex, 2025. A database detailing every attack move in Pokémon Scarlet and Violet.

[10] Smogon University. Smogon University Stats, 2025. Official monthly usage statistics for competitive Pokémon battling across various formats, hosted by the Smogon community.

[11] Sai Yashwanth Tadishetty and C. Dhatri. A multi-agent pokemon tournament for evaluating strategic reasoning of large language models. *arXiv preprint*, Aug 2025. arXiv:2508.01623v1 [cs.AI].