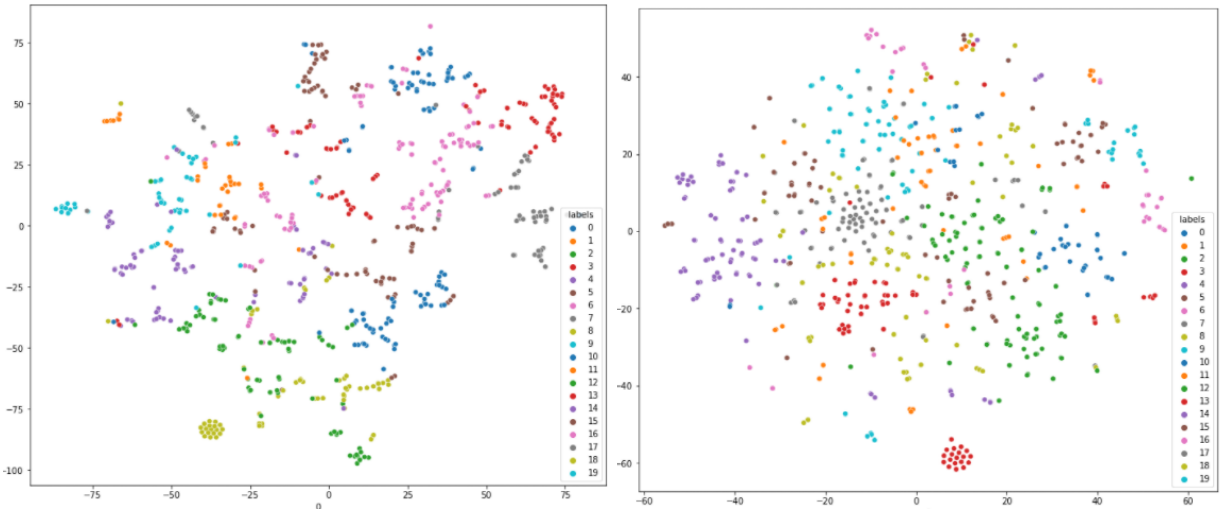
Current Situation: As data science becomes more ubiquitous, it is starting to make its way into the video game industry. Since pokemon is a quantitative turn-based strategy game with lots of freely available data on the web, I chose it as the subject of my project. In particular, I have not seen ANY model which has yet been made to predict the competitive strength of a pokemon from its static features. This limits the ability of pokemon designers (officially at the pokemon company, and unofficially at the smogon create-a-pokemon project) to predict how strong their creations are going to be (whether those creations are single pokemon or entire generations of pokemon). Game quality, sales and development time could be significantly impacted by this lack. There is also an opportunity to use models to understand which pokemon features, or combinations of features, make a pokemon competitive and powerful. This is useful for pokemon designers, but it is also useful for creating tools to assist more technically minded pokemon players.

Project Goal: Given the above situation, the goal was to create a classification model in 2021 which could predict the competitive tier of a pokemon (ZU, PU, NU, RU, UU, OU or Uber) with as high accuracy as possible from the static stats of those pokemon (the 4 static features unique to each pokemon are base stats, typing, abilities and learnset of available moves). The model would not only offer us a prediction tool which is useful in itself, but also would offer insight into the features which were most important to the model (and thus which features are most important to competitive viability of pokemon). For the sake of clarity, we are only making predictions and feature importances for the Smogon 6v6 competitive singles format in generation 8 of pokemon. This gives us a standard ruleset to ensure fairness and ease of comparison, and using the most recent generation makes the model maximally up to date for the sake of making useful predictions about future generations.

The Approach: There are 3 core tasks that we had to perform to complete this project effectively:

1) **Web Scraping and Data Cleaning:** In order to obtain the data on all the pokemon features that we need, in addition to the competitive tiering we are trying to predict, we had to obtain it from somewhere on the web. Fortunately, the strategy pokedex on the smogon website is very comprehensive; it had literally everything we needed in one place, but there were two major problems with it. The first problem is that it required very extensive cleaning to get it into a form in which it could be effectively explored and modeled. This technical and not very interesting process has already been covered in my data wrangling notebook for those who need to know more about it. The second problem is that the cleaned data has only 738 pokemon (which are the rows or observations for the model) but 914 features for each pokemon (which are the columns). Any model we make on such data will have zero degrees of freedom (since there are more features than observations), so it will be a useless overfit model. It's even worse than that, because for classification tasks with only 2 classes, you are generally recommended to have 5 to 10 times more observations than features, and for 7 class models (like what we have) you should have even more than that. So we needed to shrink the number of features to a tiny fraction of its size.

2) **Feature Engineering:** This task also had two main aspects. The first is having an intelligent understanding of the high level roles pokemon play competitively (offense, defense, utility, weather-based) and of the sub-tasks within those roles, so that we could design features that capture a pokemon's ability to play such roles. Using resources such as our web scraped strategydex, smogon viability lists for each competitive tier, and bulbapedia, we were able to create a few dozen such features, but this still didn't shrink the data anywhere near enough, which leads to the second major aspect of our feature engineering. Most of our feature columns are coming from hundreds of one-hot encoded abilities and moves, so we needed a way to aggregate those into a few numbers that could successfully inform a model. Taking inspiration from methods in natural language processing (such as tfidf), we were able to shrink those features into a few numbers (called "cindex") by taking the number of pokemon that actually competitively uses each ability or move divided by the total number of pokemon that have the move or ability available to be used. These ended up being some of our cleverest and most successful features. More details can be found in the EDA notebook.
3) **Clustering:** A look at unsupervised clusters on our dataset in the EDA notebook shows that pokemon weren't being clustered by competitive tier, but nonetheless the pokemon clustered together were remarkably similar. KMeans clustering performed better than other types of clustering, perhaps due to its arbitrariness (since the clusters aren't necessarily natural). When we applied clustering in our models, it turned out to be the most important feature in our best model, so it was possibly essential to the project. Looking at the highly ordered T-SNE visualizations of the data, colored by cluster, makes it easy to see why the clustering is so powerful in making classification decisions easier for models (it has a fibrous structure, in places, and usually sticks together outside of the big yellow cluster in the 2nd visualization):

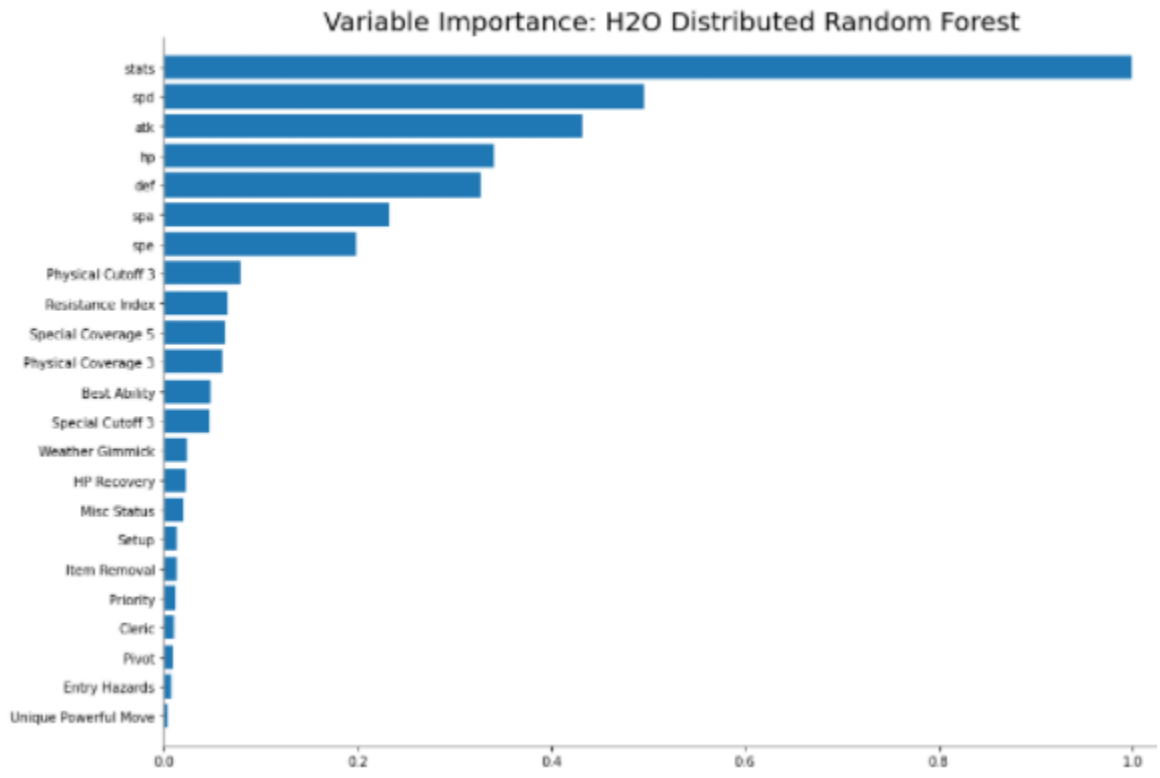


Attempted Models: We tried Logistic Regression, KNN, Decision Tree, and Random Forest for this classification task, but there are a few quirks to our implementation. For each of these models, we tried using clusters and not using clusters to see what gave the best result, and we tried clusters of different sizes (basically, we treated clustering as a complex hyperparameter). Since one of our classes, ZU, is nearly half the length of the data, we also tried using two-stage

versions of each of these modeling types. The first stage would only test for the first large class (ZU), which gave the second stage more discernment in the other 6 smaller classes which are much smaller. Two-stages lead to better performance in most model types. Since one-hot encoded clustering can be such a problem with the sklearn implementation of Random Forest, we used a more specialized library, h2o.ai, for our random forest implementation, which works much better with clustering.

Best Result: Our best model was exactly the one trained using clustering and the h2o.ai library (so it was probably worthwhile). On the training data, the weighted f-score on all classes is 0.887, with the highest class being .956 and the lowest being 0.797. On the testing data, the weighted f-score on all classes is 0.776, with the highest class being .941 and the lowest being 0.556. While this model is far from perfectly accurate, that would be unlikely to expect of a static machine learning model about a subject with changing metagames, changing tiers and unpredictable decisions by humans, such as pokemon. More about the hyperparameters can be learned by looking at the Modeling_RF and Modeling_Final notebooks. This model was a two-stage model, with the first stage being clustered by the 6 base stats and only predicting whether the pokemon was ZU or not, and the second stage being clustered by the entire learnset of moves available to a pokemon and predicting one of the 6 highest tiers (only being used on pokemon that passed the first stage as “not ZU”). The feature importances of both models offer a lot of information about how competitive pokemon could be evaluated:

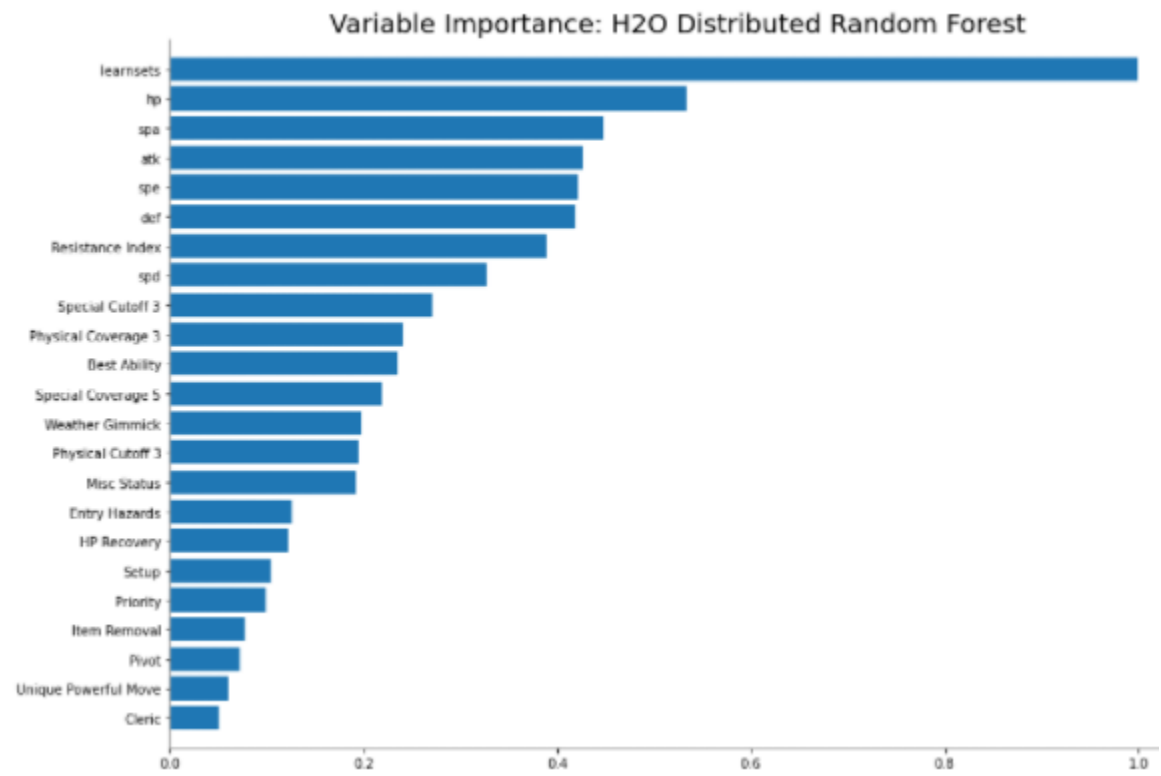
```
first_stage.varimp_plot(23)
```



Here we can see that the stats cluster (at the very top) and the 6 base stats (special defense, attack, hit points, defense, special attack, and speed, in order under stats) are by far the most

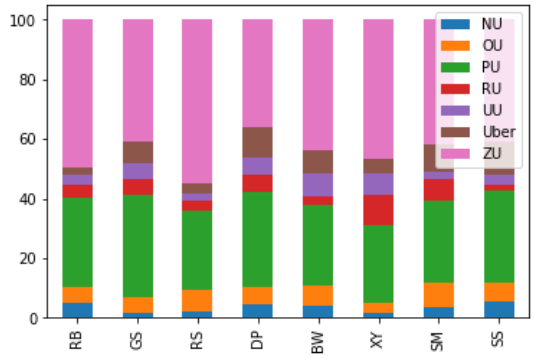
important for determining whether a pokemon is competitive or not. The next most important are the physical and special learnsets (cutoff represents how good the available physical or special moves are, whereas coverage represents how many types the available physical and special moves are good against), the resistance index (which is how good the pokemon type is at resisting attacks), and the Best Ability (how strong the best ability of a pokemon is). The remaining features are insignificant.

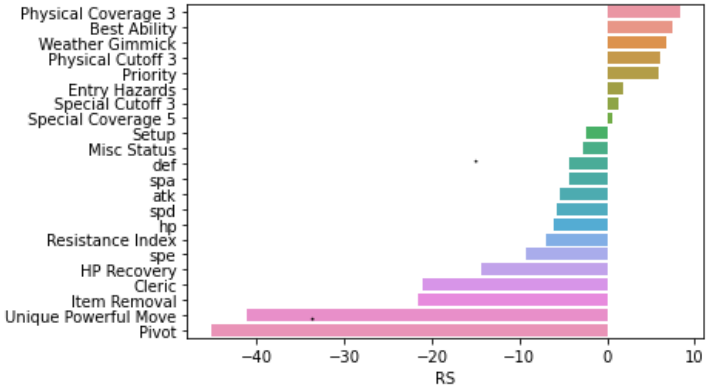
```
second_stage.varimp_plot(23)
```

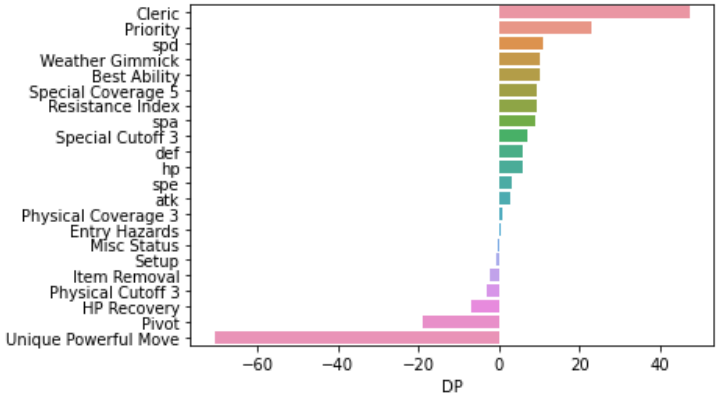


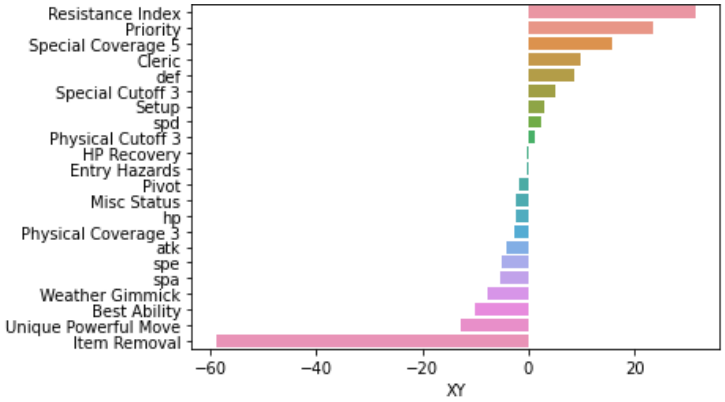
On the other hand, when deciding which of the 6 highest competitive tiers a pokemon belongs to, almost all of the features have a notable influence, though the familiar features from the first model still tend to be the best predictors. The clustering is called learnsets this time, since we clustered based on the moves available to each pokemon. The presence of powerful status moves (Misc Status) and how much a pokemon is able to take advantage of weather (Weather Gimmick) become notably predictive. Access to Entry Hazards also saw a large jump in predictive power and relative ranking to other features (which makes sense since it is a core feature of competitive games). Cleric, Unique Powerful Move and Pivot seem to be consistently some of the least important features, regardless of the stage, so in a future project it might be worth trying to improve the design of such features or considering combining them with better features.

Model Application: Since we intend to apply this model to future generations of pokemon, it only makes sense to use it to evaluate the past generations. So we asked our model what it predicted about the proportions of tiers in each generation:









It is instructive to look at the similarities and differences between RS (Generation 3), DP (Generation 4) and XY (Generation 6). They all have a pretty large ZU (pink) and PU (green) tier, which makes sense since these are the largest classes, especially ZU. Therefore, nothing is terribly skewed in these generations. RS has a large top pink section (ZU, non-competitive pokemon), and this is explained by quite poor stats overall. DP, on the other hand, has quite a small pink ZU section, and a large Uber (reddish-brown) and PU (Green) section. This is because DP is high in most important features, so its Green (PU, the lowest competitive tier) is nearly the same size as its Pink (ZU), since a much larger percentage of its pokemon have some competitive viability (and it's also strong on the highest end, Uber). XY is an interesting middle ground, since it has a large middle red and purple section, which are middling competitive tiers (RU and UU), but it's more average in its pink ZU and small towards the top end (Uber, OU). XY has higher than average on some stats and top features, and low on others, so this intermediate result makes sense.

Concrete Recommendations: As we can see above, it is possible to analyze a new generation of pokemon against the precedent of older generations. This gives a lot more possibility of control to designers of new pokemon generations. It is possible for them to directly compare and make the stacked bar for a new generation as similar, or as different, as desired from the previous generation, by tweaking the presence of various important features of the pokemon in the new generation. It is worth warning that players generally react negatively to large changes in the franchise, but the control is there either way. An even simpler suggestion is to use the

model to test various ideas for the features pokemon creators want to assign to a single pokemon. You can throw many different feature possibilities into the model for a given pokemon and see which tier the model predicts for it, and this directly informs the optimal feature arrangement. Both professional and amateur pokemon creators can use the model this way to make pokemon of a desired or appropriate power level. Finally, the feature importances of both stages of the model can help players organize their search for competitive pokemon to try out on their teams. Players could do this manually using the feature importances and the clusters that it's based on (since similar clusters are such an important feature). It's also possible that more automated tools could be created which would be especially helpful to newer players in suggesting good pokemon for their teams based on exactly such clusters and feature importances. Companies like Mobalytics use data science to build tools that help competitive gamers in a similar manner, giving players recommendations for their skill level and playstyle.

Future Directions: There are a lot of ways to improve the models and methods used in this project, given a lot more time and effort. The most obvious would be using even more advanced AI algorithms we never tried, like deep neural networks, evolutionary agents, reinforcement learning, etc. Though the way we created the features is well-informed to some extent, it contains plenty of arbitrariness, and consulting with top competitive pokemon experts on better features to use could be very fruitful. One example is that WolfeyVGC considered the offensive power of pokemon type in a recent YouTube video released on January 4th 2022 (rather than the defensive power of type that we captured in our Resistance Index feature), and that could make for another good feature to include. We also only built a model for Smogon 6v6 singles, which might not be the only sort of competitive viability designers and players want to control for. Although it would be more difficult due to less organized and less maintained tier lists, it might be possible to create a model for doubles formats like VGC. Finally, it is possible that a model like this (or even better) could be created for each new generation that comes out, giving feedback on previous models, improving from the mistakes of earlier iterations, and staying current so that the predictions made by the model are maximally relevant to the current metagame.