PREDICTING POKÉMON BATTLES

Exam Submission Spring 2018

(Nintendo, 2017)

STUDENTS: ANDERS FLERON HOLST, LINE BAY KREIBERG & SABINA HULT
SUPERVISOR: SEBASTIAN RISI

Lecturers: Paolo Burelli, Sebastian Risi
Course: KGDAMIN1KU-Spring 2018

Predicting the Outcome of a Pokémon Battle

Sabina Hult, Anders Fleron Holst and Line Bay Kreiberg

***Abstract*—Pokémon is one of the major grossing media franchises in the world. In this project we seek to classify the outcome of a Pokémon battle by using the ID3 algorithm to predict the winner of a head-to-head combat between two Pokémons based on several attributes of the participating Pokémons. k-Medoids clustering is used as well as both a preprocessing tool and in the analysis of the classification results. The main data sets contain 800 individual Pokémons and 50.000 combats. Results show that with the given data available, Speed is the main attribute involved in predicting the winner and with this attribute alone a classification accuracy of appx. 95% was achieved by training the ID3 algorithm on 10% of the original data set and testing on the remaining 90%. Considering "*real world*" Pokémon battles, this seems unlikely, indicating that our data sets lack some complexity and information about game mechanics, leading to the conclusion that the data sets consist of bot fights with no human involvement or applied strategies.**

I. Introduction

Pokémon (Pocket-Monster) is a universe created by Satoshi Tajiri in collaboration with game developer Nintendo. In the video games you play a human trainer that catches Pokémons and battles other trainers and Pokémons to win leagues and be the best Pokémon trainer in the world. A Pokémon battle is turn-based, where each Pokémon does one move/action in each turn. Since its first emergence in 1996 (Pokémon Red, Pokémon Green), the game has become an international phenomenon, lately seen with the release of the augmented reality game for the smartphone platform, Pokémon Go, in 2016 (Bulbapedia, 2018).

The goal of this project is to predict the winner of a Pokémon battle using the ID3 classification algorithm and the k-Medoids clustering algorithm and explore what attributes might be related to winning Pokémon battles.

II. Data Set

The data used in this project originates from a Pokémon Challenge uploaded on Kaggle, which consists of two data sets; combats.csv and pokemon.csv (T7-Pokémon Challenge, 2017). The combat data file consists of 50.000 combats (tuples) and is made up of three columns; the id of the first Pokémon, id of the second Pokémon and id of the winner of the battle.

The second data set, Pokémon.csv, contains 12 columns; Id, Name, Type 1, Type 2, HP, Attack, Defense, Special Attack, Special Defense, Speed, Generation and whether the Pokémon is Legendary. This data set contains 800 tuples and all Pokémons are in level 50.

To extract more information about the Pokémons we found two more data sets on Kaggle. From the first set we extracted type matching information, which shows how different types of Pokémons fare against each other, represented by an attack multiplication factor (Banik, 2017). From the second, we obtained data on each Pokémons' evolutionary chain. This allowed us to create a stage attribute to account for the number (if any) of evolutions a Pokémon has undergone (Parr, 2016).

Since the obtained data sets derived from Kaggle were already clean, data cleaning was done in conjunction with merging data sets from the different Kaggle challenges. This proved to be a bit of a challenge since ID's and names of Pokémons didn't always matchup between the data sets.

A more detailed explanation of the attributes and their connection to Pokémon battles are provided in Appendix A.

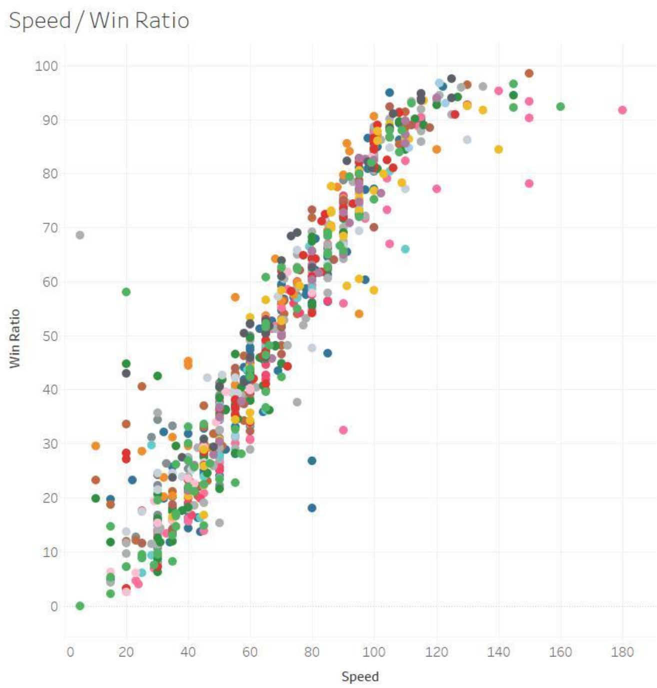
The data mining algorithms used in the project were mostly programmed in Java by the team members. This choice was purely motivated by the increased control over the project. Collaboration and sharing code happened through the version control tool Git/GitHub. The data tuples were stored as Pokémon objects and Combat objects by reading in and parsing the data csv-files.

Since nearly all information about the combats are contained in the attributes of the Pokémons, our object-oriented approach allowed us to minimize the information contained in the Combat objects to Pokémon1, being the Pokémon that attacks first, Pokémon2 and Winner. In practice, we have expanded on this structure in several ways according to the needs of our analysis, e.g. to make discrete attributes for classification and normalized ones for clustering.

To do visualizations with Excel, Tableau and R, we exported data from our program by writing csv-files of the cleaned and preprocessed data, as well as exporting the collections of Pokémons that resulted from clustering.

III. Exploring the Data

Basic exploration of the data was performed in to generate a preliminary understanding of the distributions of the values in the data. We specifically modelled the relationships between a Pokémon's attribute values and their Win Ratio as well as the median attribute value per type of Pokémon. The results can be seen in Appendix B. An important discovery is the correlation between the attribute Speed and the Win Ratio of a Pokémon giving an early indication that Speed seems to be an important attribute for a Pokémon to win a battle.



Sum of Speed vs. sum of Win Ratio. Color shows details about Type 1. Details are shown for Name.

The dots in the graphs represent Pokémons, where the color of the dot correspond to the type of the Pokémon and the trend between Speed and Win Ratio quickly becomes obvious.

IV. Algorithms

Since classification analysis can extracts models (classifiers) to predict categorical class labels (Han, Kamber & Pei, 2012: 327), we chose a classification algorithm as the focus point of our experiments. Specifically, we choose a supervised learning algorithm because the data set has provided us with the class labels we are trying to predict. This makes it possible for the algorithm to train on tuples where it knows which class the tuple belongs to (Han, Kamber & Pei, 2012: 330). In this case whether the first or second Pokémon is the winner of a combat.

A. ID3- Iterative Dichotomiser

The choice fell on the ID3, due to among other things, its property of being an eager learner, meaning most of the work takes place during the training. The prediction phase of the test data therefore requires less work than a lazy learner approach (Han, Kamber & Pei, 2012: 423). This seems to be a more sustainable solution if the result of this project were ever to be used by players to predict the outcome of a battle in the future.

In comparison to a lazy learner approach that might create hyperpolygonal shapes, a decision tree is generally also more describable (Han, Kamber & Pei, 2012: 423), which we found to be a compelling quality.

Since our data set used for the ID3 does not contain missing values, and no known issues related to overfitting has been found, we did not find any convincing arguments to use the C45 over the ID3 (EliteDataScience, 2018; Han, Kamber & Pei, 2012: 340).

The CART algorithm would have been useful for the numeric attributes connected to a Pokémons stats (Brownlee, 2016), but since the data set also contains categorical values and we are interested in the difference in stats between two Pokémons in a battle, we found it easier to discretize the data and use the ID3, rather than CART.

The ID3 was trained on a set of 5.000 combat tuples, and we tested the decision tree on the remaining 45.000 combats from the total set. We experimented with training sets of varied sizes, see Appendix D for more information.

We used sensitivity (true positive rate) and specificity (true negative rate) as measures for the ID3. Accuracy was found by adding these two measures to get the recognition rate (Han, Kamber & Pei, 2012: 365).

To get the central value of the accuracy, we ran the ID3 algorithm a hundred times, each time generating a training set of 5.000 random tuples and adding the remaining tuples to the test set. The accuracy measure for each run was stored and the average was calculated in the end. Every time an accuracy measure for the ID3 is mentioned in this report, we refer to the average of a hundred runs.

As a data preparation step we decided to eliminate continuous attribute values to simplify the resulting decision tree. By utilizing our domain knowledge of Pokémons we arrived at the conclusion that five attribute values should be incorporated; extremely low, low, medium, high and extremely high. In discretizing we decided to split the values by percentiles. Since the extreme categories were included to capture any outliers there might be present in the data, we set the percentiles at the 5 % mark at the top and bottom. The remaining three labels were divided evenly so they each contained 30 % of the data.

B. k-Medoids

The clustering algorithm k-Medoids was used in the project under several stages to visualize and analyze the result of the decision tree, as well as creating labels for attributes and creating a cluster-attribute for the ID3, based on the clustering of Pokémons.

By clustering Pokémons, a form of implicit class was created, that could group similar Pokémon together (Han, Kamber & Pei, 2012: 444), and thereby also make a distinction between Pokémons from different clusters in a battle. Since both the Pokémon data objects and the Combat data objects contained relatively high dimensional data (9 attributes) we used clustering to try to reduce some of this complexity, both in the preprocessing stage with discretization (Han, Kamber, Pei, 2012: 116), and later in the analysis by creating a cluster-attribute (see section Analysis).

The choice of k-Medoids instead of k-Means was mainly motivated by k-Medoids ability to deal with outliers and categorical values (Han, Kamber & Pei, 2012: 454).

Because of the different attribute types used in the analysis, we used the method of bringing all the attributes onto the same scale and determining the similarity between objects based on the attribute type (Han, Kamper & Pei, 2012: 75). The clustering had termination conditions varying on the data set to be clustered, but all were a measure of how many unsuccessful improvements were tried. For example, when clustering the Pokémons (n=800), the termination criteria were when 200 swaps were tried but none executed. In contrast, when clustering the Combats (n=50.000) the termination criteria were set to 30.000. A common challenge when performing cluster analysis is specifying the number of clusters in advance. When we performed clustering with the help of platforms such as Tableau, we allowed it to make the decision and ensure the quality, otherwise we opted for the Elbow method (Kassambara: 2017).

In the clustering process we took steps to keep down the dimensionality of the data, to avoid the curse of dimensionality (Han, Kamper & Pei, 2012: 447). This was especially true when clustering the combats. Since each combat consists of two Pokémons and their attribute values, the combats could potentially have 2^n number of attributes, with n being the number of attributes of the Pokémons.

To shy away from this, we assumed that it is the difference between the attribute values of the two Pokémons that matter. Therefore, in all numerical data referencing the cluster analysis of the combats, positive numbers indicate that Pokémon1 has a higher attribute value than Pokémon2 and negative numbers mean the opposite.

V. Analysis

In our process of running our classification algorithm we did a lot of experiments and updated the nature of our combat objects several times. We refer to these different versions as Three Labels, Clustered Labels, Percentiles and Speed-Tweak.

Considering our choice of classification method, there was a need to generate discrete values for the Combat objects' attributes to reduce the complexity of the classification tree. The initial solution was to assign the Pokémons' discretized attributes (see section Exploring The Data, Algorithm ID3) to the Combats' attributes based on three criteria; HIGHER_LOWER, LOWER_HIGHER and EVEN, e.g. if the first Pokémon's Attack value is lower than the second's, the value of the Combat's Attack attribute is LOWER_HIGHER. We refer to this as the Three Label version.

This approach aims at capturing the essential relationship between the two Pokémons' attribute and utilize this in the classification. The assumption being that any difference in any attribute value will influence the outcome of the battle.

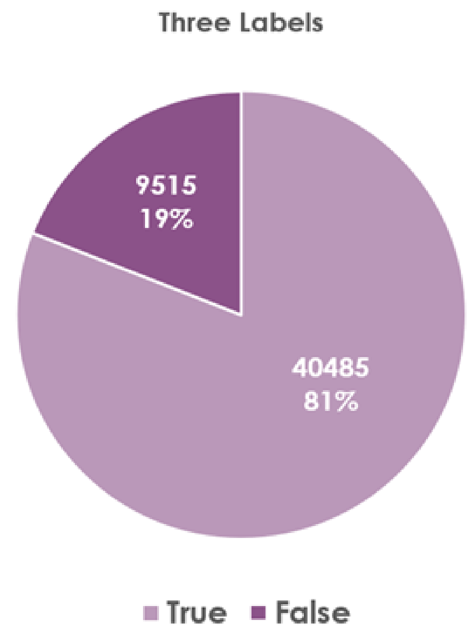


Figure 1: One run of the ID3 with three values per attribute in combat objects.

The classification results can be seen in figure 1, with an accuracy of 81.22 %, which compares well to the results from the baseline performance calculation of 52.79 % (see Appendix C: ZeroR Calculation).

To further improve on the accuracy of the classification, we reexamined our assumptions regarding the data. We decided on focusing on issues regarding our handling of the attribute values of the Pokémons and the aggregation of these into the combats. Since this involved a two-step process; 1) discretizing the Pokémons' numerical values and 2) aggregating these into the combats, our approach reflected this.

The first step involved the assumption, that the attributes could be grouped statically simply by dividing the Pokémons attribute values according to some, at least partially arbitrary, percentiles. This leaves a great deal of influence over the labelling of the attributes in the hands of the analyzer. We decided to try and improve on this by clustering the Pokémons in respects to each numerical attribute, so that we could both obtain a more ideal number of labels and their splitting points for each attribute.

The second step involved the rather simple assumption that the outcome of the classification merely was influenced by whether there existed a difference in the attribute in favor of one of the Pokémons in the Three Label version. This completely disregarded the magnitude of the difference. To counteract this negligence, we decided to create a combat attribute value for each combination of Pokémon attribute value pair, so that instead of LOWER_HIGHER a combat for example could be labelled EXTREMEHIGH_EXTREMELOW. This resulted in a substantial increase in the number of values per attribute for each of the six original numerical attributes in hopes of capturing more information contained in the attributes and thereby achieving a higher accuracy. This version we refer to as Clustered Labels.

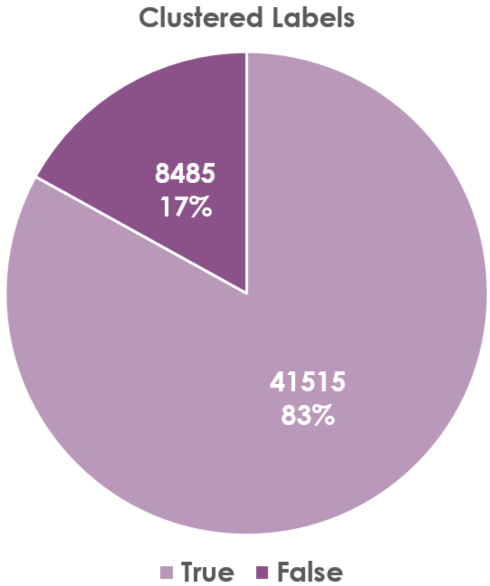


Figure 2: Based on export of one run of the ID3 with clustered values per attribute in combat objects.

Although this did increase the accuracy slightly to 82.2 %, the increase did not indicate that the origins of the false classification had been found. At this point we reconsidered our two-step approach of increasing the complexity of the classification parameters. By regressing to our previous approach of using percentiles to indicate the Pokémons attribute values, while keeping the complexity of the combat values, we maintained a certain level of granularity (25 values per numerical attribute) of the parameters without overcomplicating the process. This we refer to as the Percentiles version.

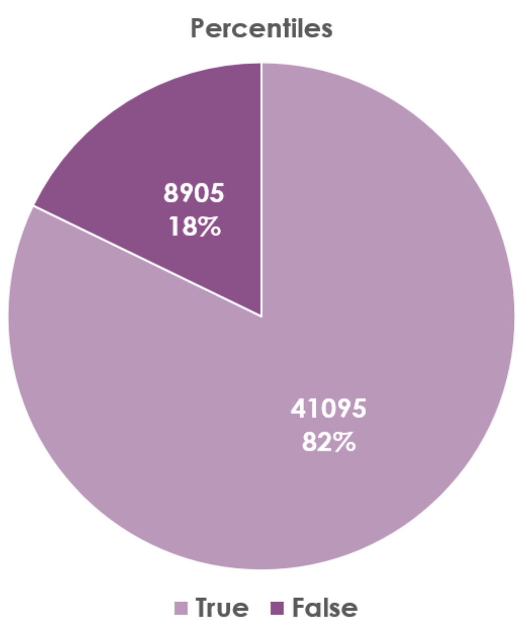


Figure 3: Based on export of one run of the ID3 with percentile values per attribute in combat objects

As can be seen from figure 3, this reduction in complexity did not significantly improve on the precision of our classification. This led us to begin to evaluate the relationships between the attributes more closely. To utilize whatever these relationships may be, we implemented a clustering of the combats. When examining the results of the six clusters of combats, it becomes clear that there indeed seems to be a somewhat clear distribution of the attribute values across the different clusters. An example of this can be seen in figure 4 where the special attack values are mapped by clusters.

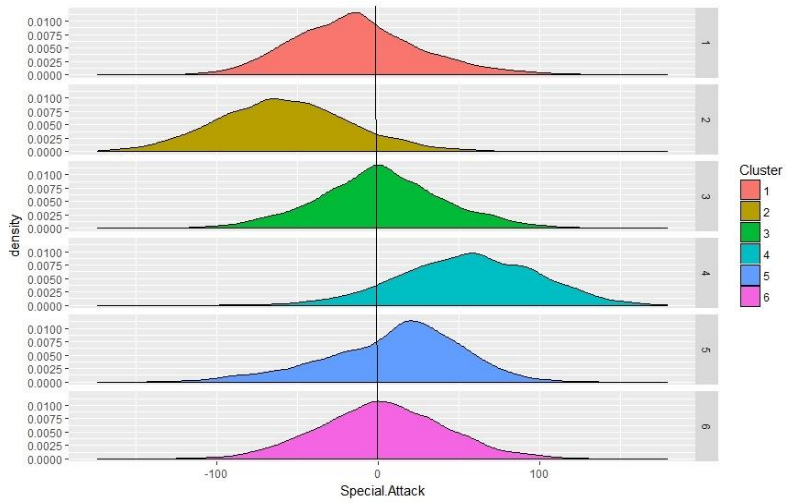


Figure 4: Special Attack mapped by clusters

In this case especially cluster 2 and 4 stands out, where Pokémon2 in general has the highest value in cluster 2 and Pokémon1 has the highest value in cluster 4. On closer scrutiny of all the attributes as distributions over clusters, some degree of inter attribute information does seem to have been captured. If the reason for the inability to classify combats lay in these relationships between the attributes, one could expect the falsely classified combats to localize themselves in one or more clusters, which could guide the investigation further.

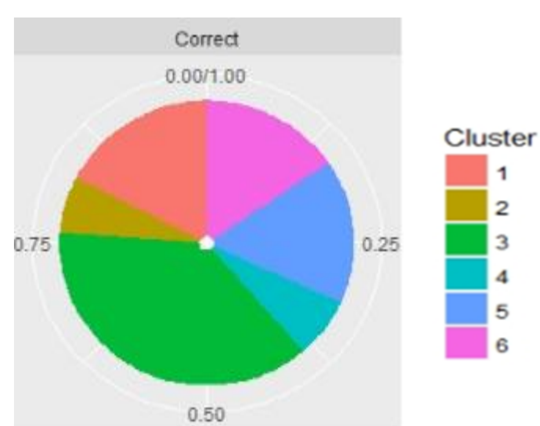


Figure 5: Distribution of correct classified combats over clusters

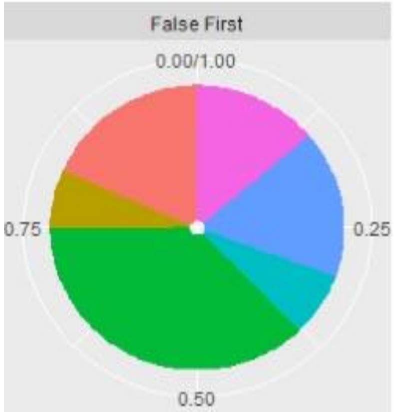


Figure 6: Distribution of falsely classified first win

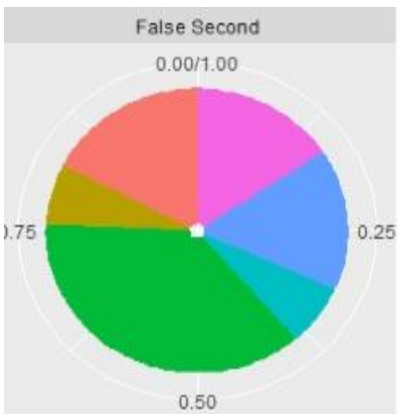


Figure 7: Distribution of falsely classified second win

As can be seen from figure 5, 6 & 7, the correctly and falsely classified combats distribute themselves evenly across the clusters. This indicates that the falsely classified combats are not so because of relationships between attribute values. Another validation of this fact came from the results of adding different cluster attributes to the data set. These attributes came from a clustering of different attribute values of the Pokémons and then adding an attribute carrying the information of which cluster the participants of the combats belonged to. The results did not yield any improvement in accuracy and merits a change of focus in the attempts to heighten the accuracy (see Appendix E: Attribute Accuracy Experiments).

We moved our attention to the attribute Speed, which had a correlation to the Win Ratio of a Pokémon as mentioned earlier. We carried out several experiments by adding and removing attributes in the attribute list for the ID3 (see Appendix E: ID3 Attribute Accuracy Experiments). From these experiments we could conclude that the accuracy of the decision tree depends almost entirely on Speed. Even though the decision tree uses other attributes, if you remove these attributes from the attribute list, and are left with Speed, the decision tree keeps its high accuracy. Furthermore, all the experiments where Speed was in the attribute list, the ID3 would always choose this attribute to split by at the root node, again indicating the importance of Speed to predict the winner of a battle. A deeper investigating of the falsely classified combats revealed that most of the combat tuples had the value EVEN for Speed. We therefore tweaked the attribute Speed and its values when discretizing the Combat objects, so that it was set to according to each Pokémons exact numeric value for Speed, and then setting the combat Speed value according to this, e.g. HIGHER_LOWER, LOWER_HIGHER and EVEN. EVEN in this case would hardly ever be chosen due to the nature of the Pokémon´s disperse numeric values.

This had a major impact on the accuracy of the decision tree, improving the average accuracy to 94.05%, with Speed as the only attribute in the decision tree´s attribute list. Adding the attribute Type to the attribute list improved the accuracy on average with 1%, to 94.96% (see Appendix F: Visual Cluster Trace of Decision Tree Improvement). Experimenting with adding and removing other attributes had no significant impact on the decision tree´s accuracy (see Appendix E: Attribute Selection Experiments).

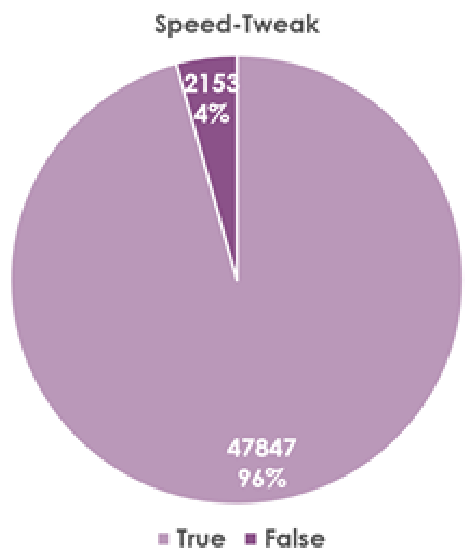
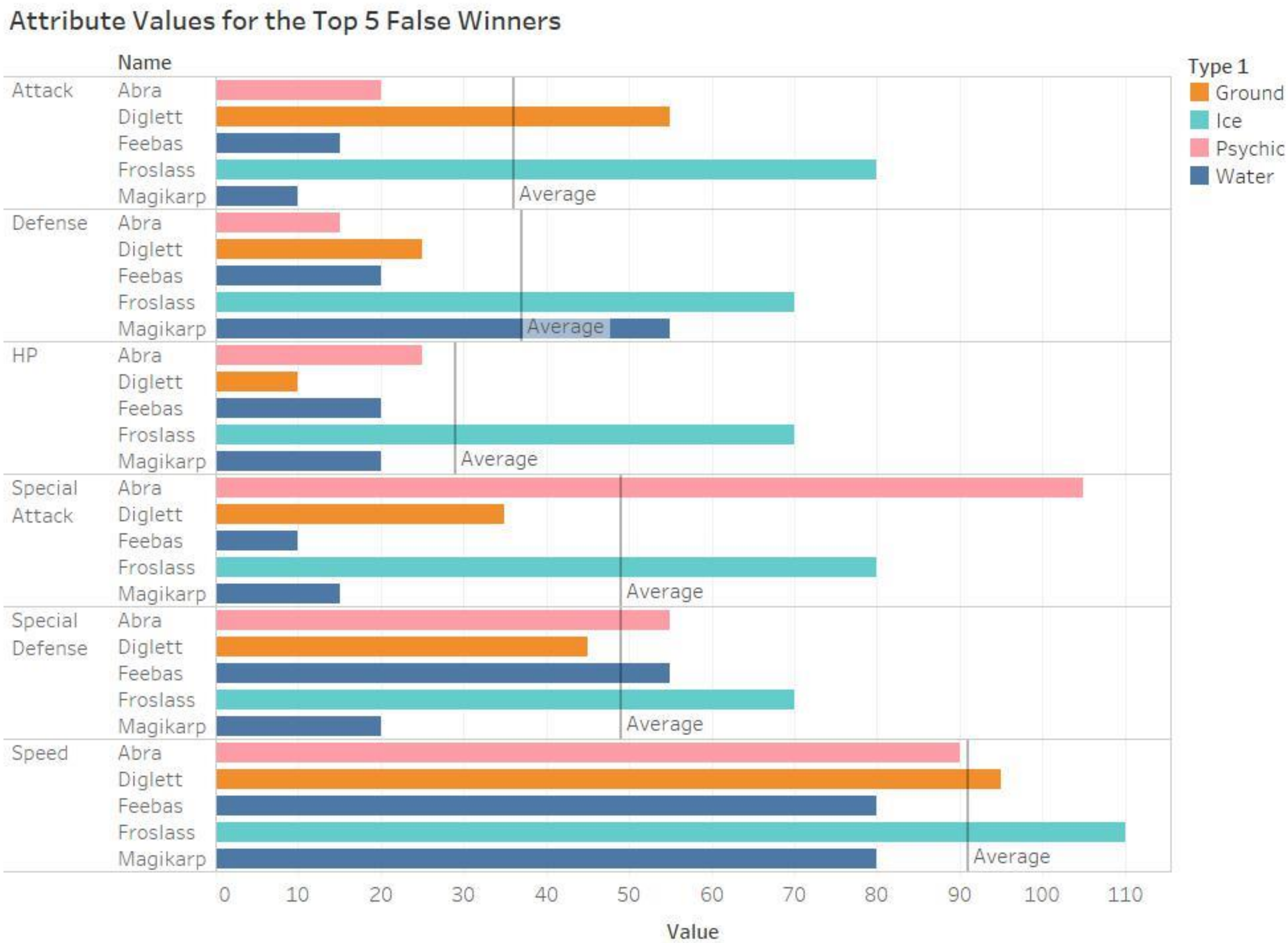


Figure 8: Based on export of one run of the ID3 with the Speed values tweaked attribute in combat objects

VI. Reflection

Our ultimate results of 94.96 % correctly predicted combat outcomes is a very high result. This has prompted us to consider the origins of the data. Any human involvement in the combat process seems improbable considering our classification accuracy. In addition, the overwhelming significance of the Speed attribute further validates this assumption. If Speed indeed were as important as our results suggests, little would be gained from valuing any other attributes which both in theory and practice seems unlikely (see Appendix A: Attribute in the Context of a Battle). A more probable assumption would be that the data set consists of bot generated combat results in standardized settings, where a move/attack is randomly chosen in each turn for both Pokémons. Furthermore, the use of items and being able to switch out a Pokémon, seems to also have been stripped from the generated data. In such a context, with strategy, tactics and skill eliminated from the equation, our results suggest that Speed is the only truly significant attribute.

We took a closer look at the falsely classified combats and the Pokémons involved in those combats by exporting the falsely classified combats and calculated a Falsely Classified Ratio for the Pokémons involved. From this we found the Top 5 False Winners and Top 5 False Losers and investigated their stats.



Attack, Defense, HP, Special Attack, Special Defense and Speed for each Name. Color shows details about Type 1. The view is filtered on Name, which keeps Abra, Diglett, Feebas, Froslass and Magikarp.

For the Top 5 False Winners they all have high Speed values leading to them being classified as winners. Abra, Diglett, Feebas and Magikarp are all Stage 1 Pokémons, meaning that they are on their lowest stage in their evolution and therefore at their weakest point.

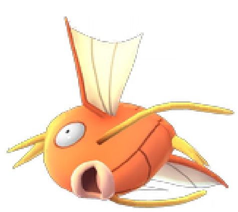


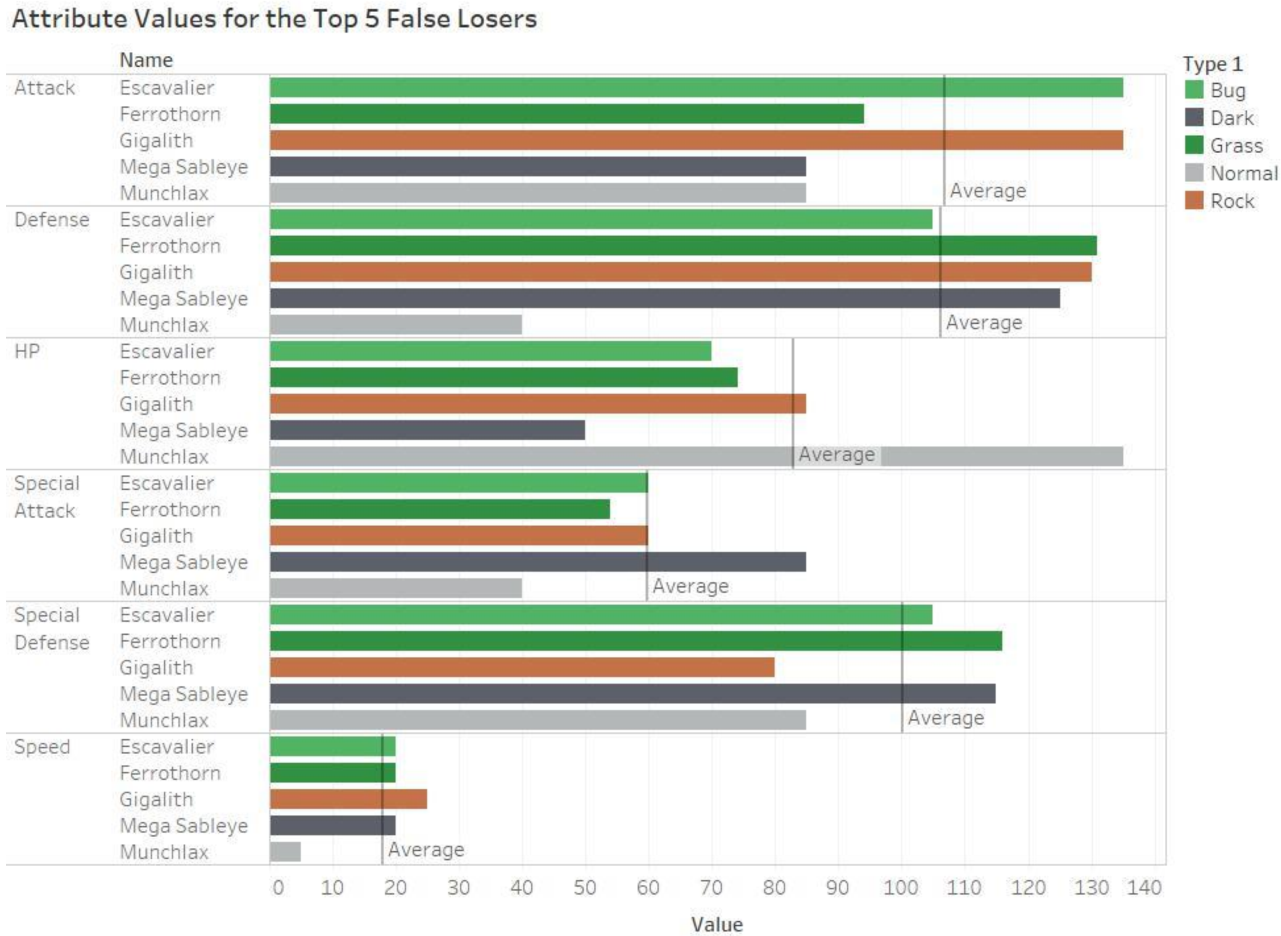
Figure 9: Magikarp (Boost, 2018)

In the case of Magikarp it might be fast and therefore get to attack first, but its attack value is very low and so it won't do much damage. Magikarp also has a move "Splash" that does absolutely nothing (Boost, 2018).

If the moves are randomly chosen and are not made strategically, "Splash" would have a higher probability of getting chosen in this setting than in cases of involving humans or reasoning of any sorts and making Magikarp more likely to lose the fight.

Froslass seems to be an outlier in this little set with relatively high values for each attribute, so we consulted our resident Pokémon expert, Jens Struckmann Hansen, who informed us that this Pokémon has a lot of moves aimed at buffing down the opponent's stats (Attack, Defense, Special Defense, Special Attack, Speed) (Bulbapedia (5), 2018), meaning that it doesn't do direct damage, but rather weakens the opponent so that the next chosen Pokémon can defeat it.

In this setting where we are only looking at one combat in isolation, Froslass is not a great Pokémon, however it might be in "*real world*" combats.



Attack, Defense, HP, Special Attack, Special Defense and Speed for each Name. Color shows details about Type 1. The view is filtered on Name, which keeps Escavalier, Ferrothorn, Gigalith, Mega Sableye and Munchlax.

For the Top 5 False Losers, again we see a close correlation to the Speed attribute, here it being very low for every Pokémon in this set. In comparison the median Speed for a Pokémon in the entire data set is 65.

Figure 10: Munchlax (Pokémon Database (1), 2018)

Munchlax is the weakest Pokémon of the five losers who are really winners and by far the slowest Pokémon of the five with a Speed value of 5.

However, even though Munchlax stats are lower than the others, it has an unusual high HP which means that it can take a lot of damage without fainting. Munchlax has the highest Falsely Classified Ratio of all Pokémons, being falsely classified in 60,17 % of the combats its involved in.

This information might indicate that we are lacking some valuable information regarding Pokémons, for example what moves they apply in a given combat, since this probably highly influences the outcome of a battle, especially in "*real world*" settings.

VII. Conclusion

Our investigation of the Pokémon combat data set, we have succeeded in predicting the outcomes of the battles with 94.96% accuracy. This was achieved, almost exclusively, with the use of the Speed attribute, which indicates, that the data stems from a lab like setting. Extrapolation of the results to "*real world*" Pokémon combats might prove difficult since the data does not take strategy, tactics or skills into account. What can be said is that in the absence of these factors, Speed seems to be the only truly relevant attribute in determining the outcome of a Pokémon head-to-head combat.

References

Banik, R. (2017). The Complete Pokémon data set - Data on more than 800 Pokémons from all 7 generations. Retrieved May 10, 2018 from: https://www.kaggle.com/rounakbanik/Pokémon/data

Boost, R. (2018). Pokémon GO Magikarp Raid Boss | Max CP | Evolution | Moves | Spawns. Retrieved May 12, 2018, from https://rankedboost.com/pokemon-go/magikarp/

Brownlee, J. (2016, April 18). Classification And Regression Trees for Machine Learning. Retrieved May 8, 2018, from: https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/

Bulbapedia. (2018). Pokémon. Retrieved May 10, 2018 from: https://bulbapedia.bulbagarden.net/wiki/Pok%C3%A9mon

Bulbapedia (1). (2018, May 9). Statistic. Retrieved May 10, 2018, from https://bulbapedia.bulbagarden.net/wiki/Statistic

Bulbapedia (2). (2018, May 8). Type. Retrieved May 10, 2018, from https://bulbapedia.bulbagarden.net/wiki/Type

Bulbapedia (3). (2018, May 8). Legendary Pokémon. Retrieved May 9, 2018, from https://bulbapedia.bulbagarden.net/wiki/Legendary_Pok%C3%A9mon

Bulbapedia (4), (2018, March 25). Party. Retrieved May 10, 2018, from https://bulbapedia.bulbagarden.net/wiki/Party

Bulbapedia (5). (2018, May 8). Froslass (Pokémon). Retrieved May 12, 2018, from https://bulbapedia.bulbagarden.net/wiki/Froslass_(Pokémon)

EliteDataScience. (2018, February 08). Overfitting in Machine Learning: What It Is and How to Prevent It. Retrieved May 10, 2018, from https://elitedatascience.com/overfitting-in-machine-learning

Han, J., Kamber, M., Pei, J. (2012). Data Mining: Concepts and Techniques. (3rd ed.). Morgan Kaufmann

Kassambara, A. (2017). Determining the Optimal Number of Clusters: 3 Must Know Methods. Retrieved May 10, 2018 from http://www.sthda.com/english/articles/29-cluster-validation-essentials/96-determining-the-optimal-number-of-clusters-3-must-know-methods/

Nintendo. (2017, June 06). New Pokémon Fighting Game Announced, Pokken Tournament DX. Retrieved May 13, 2018, from http://www.mtv.co.uk/pokken-tournament/news/pokken-tournament-dx

Parr, D. (2016), Pokedex - a richer Pokémon data set. Retrieved May 10, 2018 from: https://www.kaggle.com/davidrgp/Pokedex/data

Pokémon Database. (2018). Pokémon item list. Retrieved May 10, 2018, from https://Pokémondb.net/item/all
Pokémon Database (1). (2018). Munchlax. Retrieved May 12, 2018, from https://pokemondb.net/pokedex/munchlax

T7-Pokémon Challenge. (2017). Pokémon- Weedle´s Cave. Retrieved May 10, 2018 from: https://www.kaggle.com/terminus7/Pokémon-challenge/data

ZeroR. (n.d.). Retrieved March 21, 2018, from http://chemeng.utoronto.ca/~datamining/dmc/zeror.htm

Appendix A: Attribute in the Context of a Battle

Attribute	Description
Attack	The amount of damage a Pokémon deals to another Pokémon, when using a physical move/attack (Bulbapedia (1), 2018).
Defense	The amount of damage “taken in” to a Pokémon’s HP, when it is hit with a physical attack. A high defense would mean a physical attack will do less damage (Bulbapedia (1), 2018).
HP	“Hit points”, is the amount of damage a Pokémon can take before fainting and losing the battle (Bulbapedia (1), 2018).
Speed	Who goes first in the next turn? If a Pokémon has a higher speed than its opponent, it will generally make a move in the next turn before the opponent (Bulbapedia (1), 2018).
Special Defense	Determines the amount of damage a Pokémon takes when hit with a special move (attack) (Bulbapedia (1), 2018).
Special Attack	Determines how much damage a Pokémon does when using a special move (Bulbapedia (1), 2018).
Stage	Pokémons have different evolutionary stages. Certain Pokémons can evolve as their level grows and they become stronger with better stats. Often there is an evolutionary chain of 3 or 2 evolutions (Bulbapedia (1), 2018). 
Legendary	A legendary Pokémon is very rare and very powerful, you often meet a them late in the game, where you can try and catch it (Bulbapedia (3), 2018). 

Type

Type describes properties of a Pokémon and its moves (attacks). Up until now there exist 18 different types.

In a battle these properties play a part in the amount of damage a Pokémon can deal to another Pokémon, and how weak they are against those attacks (Bulbapedia (2), 2018). For instance, a water Pokémon that attacks a fire Pokémon with a water move, will be able to deal 2 times the amount of damage it would normally deal, while a fire Pokémon's move on a water Pokémon will only be able to deal half the amount of damage it normally deals. Due to the many types is quite a complicated mechanic.

Below is a table of the matrix:

×	Defending type																	
Attacking type	NORMAL	FIGHT	FLYING	POISON	GROUND	ROCK	BUG	GHOST	STEEL	FIRE	WATER	GRASS	ELECTR	PSYCHC	ICE	DRAGON	DARK	FAIRY
NORMAL	1×	1×	1×	1×	1×	½×	1×	0×	½×	1×	1×	1×	1×	1×	1×	1×	1×	1×
FIGHT	2×	1×	½×	½×	1×	2×	½×	0×	2×	1×	1×	1×	1×	½×	2×	1×	2×	½×
FLYING	1×	2×	1×	1×	1×	½×	2×	1×	½×	1×	1×	2×	½×	1×	1×	1×	1×	1×
POISON	1×	1×	1×	½×	½×	½×	1×	½×	0×	1×	1×	2×	1×	1×	1×	1×	1×	2×
GROUND	1×	1×	0×	2×	1×	2×	½×	1×	2×	2×	1×	½×	2×	1×	1×	1×	1×	1×
ROCK	1×	½×	2×	1×	½×	1×	2×	1×	½×	2×	1×	1×	1×	1×	2×	1×	1×	1×
BUG	1×	½×	½×	½×	1×	1×	1×	½×	½×	½×	1×	2×	1×	2×	1×	1×	2×	½×
GHOST	0×	1×	1×	1×	1×	1×	1×	2×	1×	1×	1×	1×	1×	2×	1×	1×	½×	1×
STEEL	1×	1×	1×	1×	1×	2×	1×	1×	½×	½×	½×	1×	½×	1×	2×	1×	1×	2×
FIRE	1×	1×	1×	1×	1×	½×	2×	1×	2×	½×	½×	2×	1×	1×	2×	½×	1×	1×
WATER	1×	1×	1×	1×	2×	2×	1×	1×	1×	2×	½×	½×	1×	1×	1×	½×	1×	1×
GRASS	1×	1×	½×	½×	2×	2×	½×	1×	½×	½×	2×	½×	1×	1×	1×	½×	1×	1×
ELECTR	1×	1×	2×	1×	0×	1×	1×	1×	1×	1×	2×	½×	½×	1×	1×	½×	1×	1×
PSYCHC	1×	2×	1×	2×	1×	1×	1×	1×	½×	1×	1×	1×	1×	½×	1×	1×	0×	1×
ICE	1×	1×	2×	1×	2×	1×	1×	1×	½×	½×	½×	2×	1×	1×	½×	2×	1×	1×
DRAGON	1×	1×	1×	1×	1×	1×	1×	1×	½×	1×	1×	1×	1×	1×	1×	2×	1×	0×
DARK	1×	½×	1×	1×	1×	1×	1×	2×	1×	1×	1×	1×	1×	2×	1×	1×	½×	½×
FAIRY	1×	2×	1×	½×	1×	1×	1×	1×	½×	½×	1×	1×	1×	1×	1×	2×	2×	1×

These matchups are suitable for Generation VI onward.

(Bulbapedia (2), 2018)

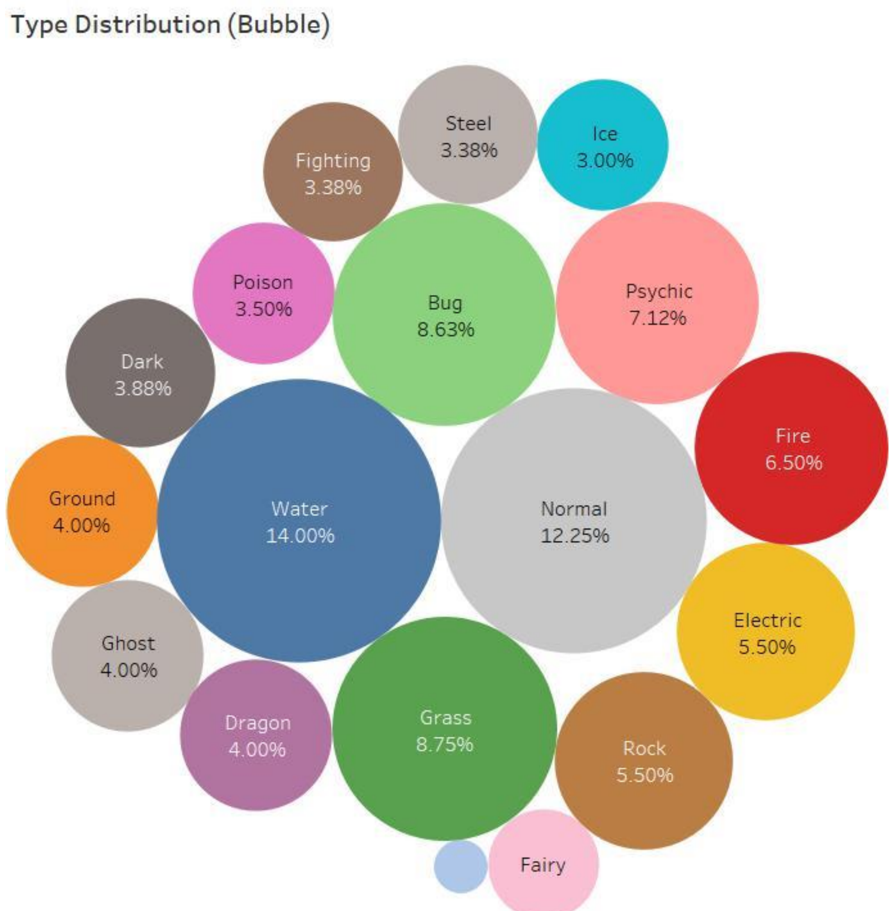
Some Pokémons can be two types. So, if a Pokémon has type 1: fire and type 2: rock, it would deal 4 times the amount of damage to a Pokémon of type bug, because bug Pokémons are weak against both fire and rock.

Level (not applicable in this project)	When a Pokémon gains more experience as it wins Pokémon battles, it's level grows, thereby making the Pokémons stats grow. This project is working with a data set where all Pokémon are in level 50, making them easier to compare to one another (Bulbapedia (1), 2018).
Items (not applicable in this project)	During a battle a Pokémon trainer can use one move/action on using an item on a Pokémon, instead of attacking. There exist several different categories of items, some of the most useful in a battle are the following: • *Berries*: You can feed your Pokémon a berry in a battle, some berries will restore a portion of the Pokémon´s HP, some will make it stronger against certain types, some will improve its attack or defense for a certain amount of time etc. • *Medicine*: Potions are often used to heal a Pokémon's HP, some are used to "awaken" a Pokémon if it fell asleep during a fight, or heal it from paralysis, confusion or burn which are status effects inflicted by certain attacks and special attacks. • *Other items*: other items can help a Pokémon evolve to its mega-evolution (very strong) during the battle (Pokémon Database, 2018).
Withdraw a Pokémon from a battle and send in another from the trainer's team (not applicable in this project).	A Pokémon trainer can have a team of up to 6 Pokémons. The trainer can use an action in a battle, to switch out his battling Pokémon. Often this is done to keep the Pokémons from fainting, and other tactical consideration such as weakening the opponents Pokémon with passive moves, and then switch to a Pokémon with high physical attacks, to bring the opponents HP down (Bulbapedia (4), 2018).

Appendix B: Basic Descriptive Visualization of the Data

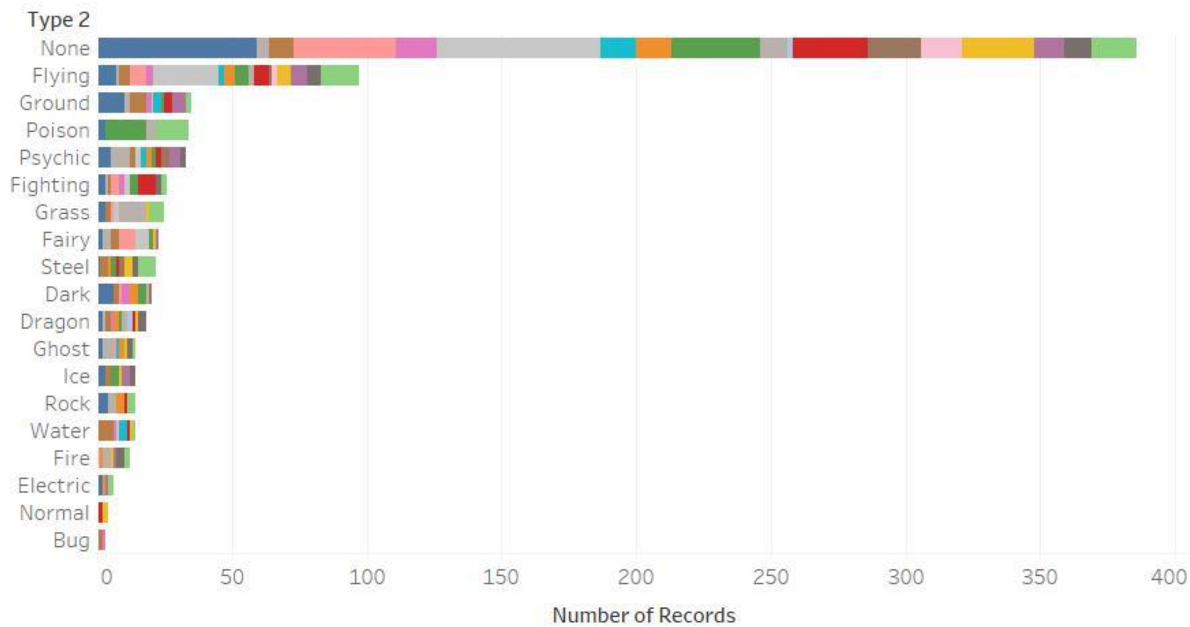
Color Legend for visualizations

Distribution of Types



Type 1 and % of Total Number of Records. Color shows details about Type 1. Size shows % of Total Number of Records. The marks are labeled by Type 1 and % of Total Number of Records. Percents are based on each column of the table.

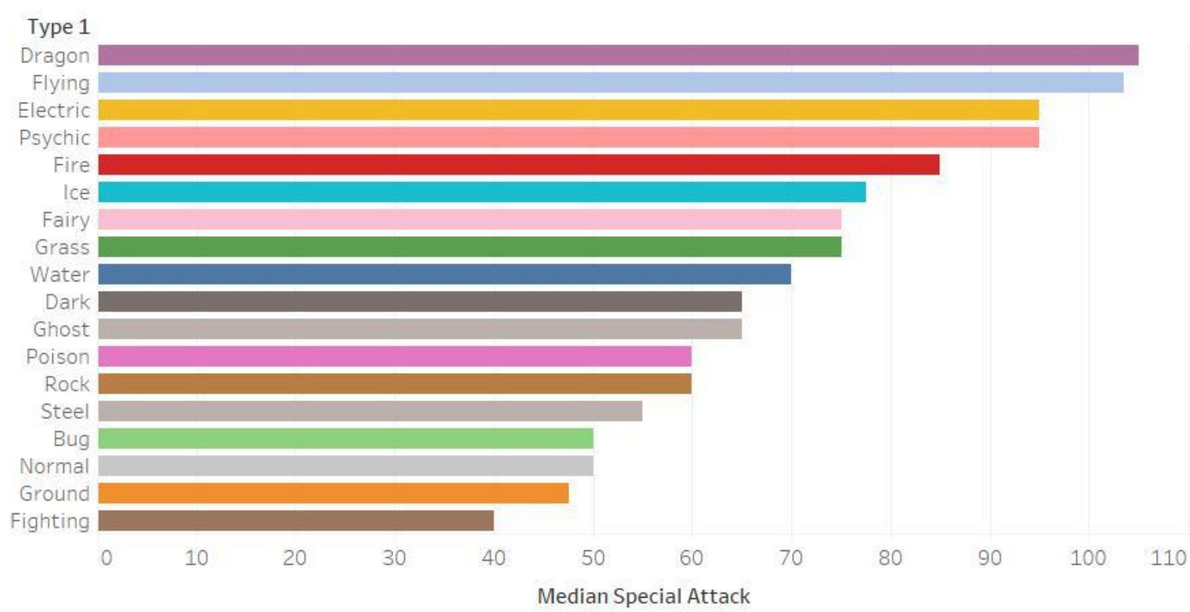
Type 2



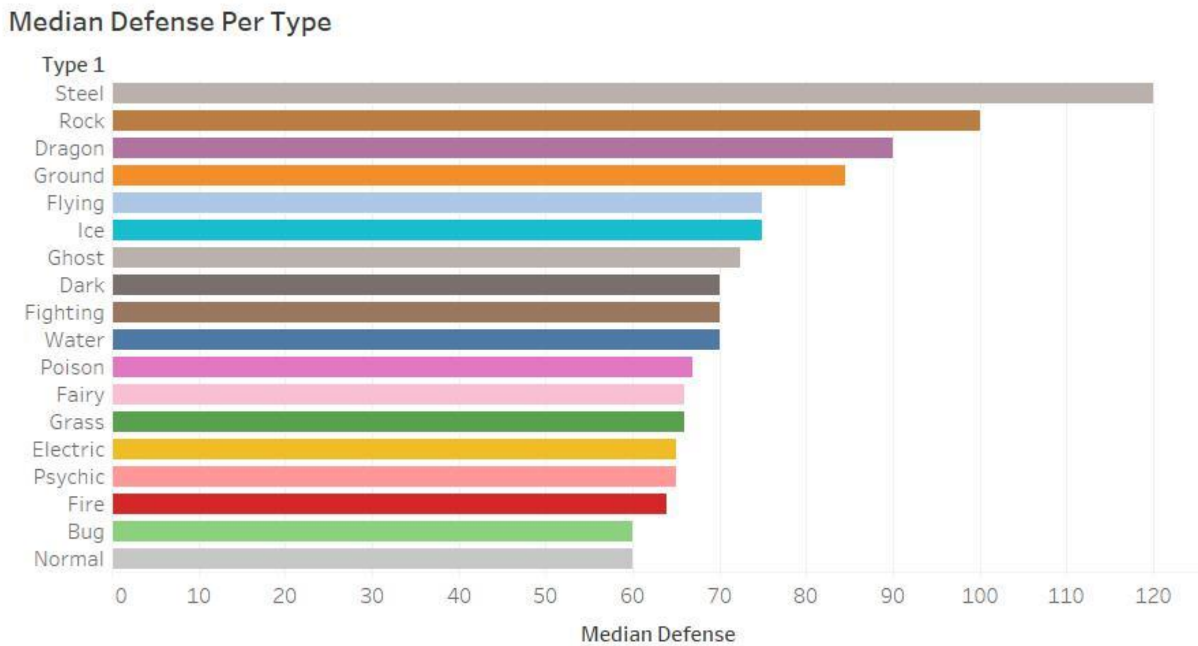
Sum of Number of Records for each Type 2. Color shows details about Type 1.

Median Attribute Values Per Type

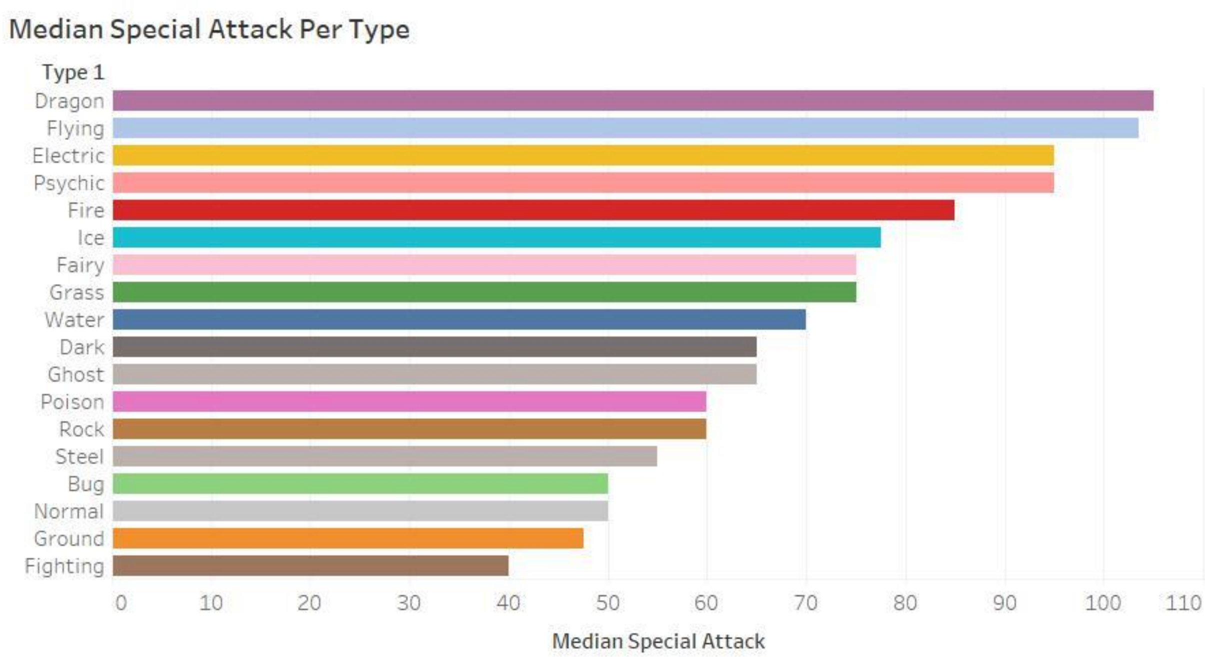
Median Special Attack Per Type



Median of Special Attack for each Type 1. Color shows details about Type 1.

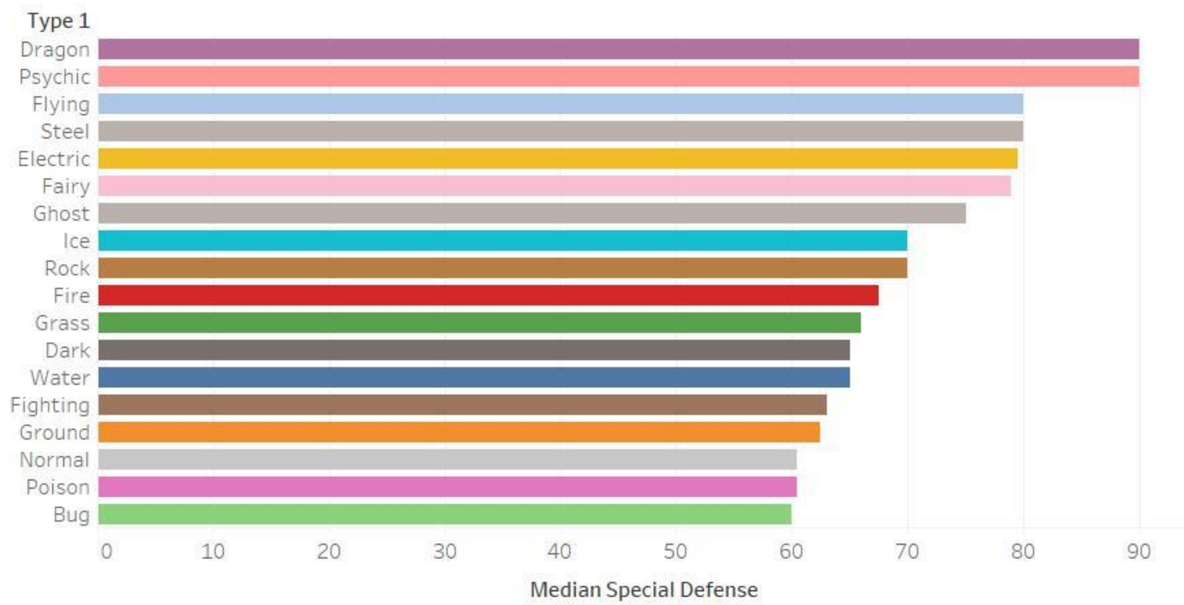


Median of Defense for each Type 1. Color shows details about Type 1.



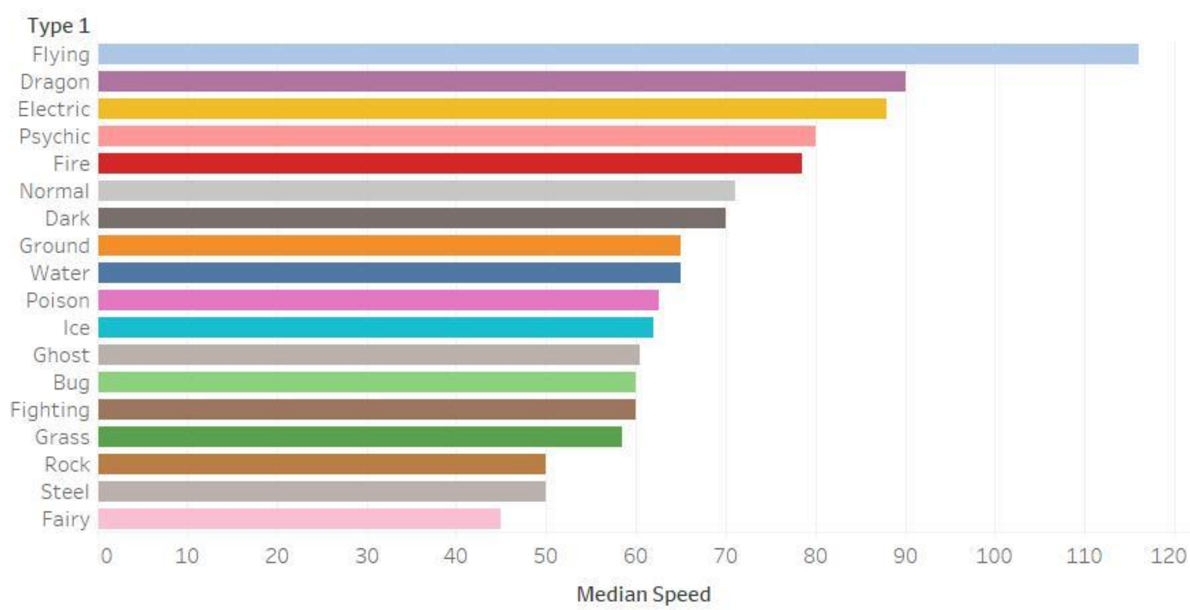
Median of Special Attack for each Type 1. Color shows details about Type 1.

Median Special Defense Per Type

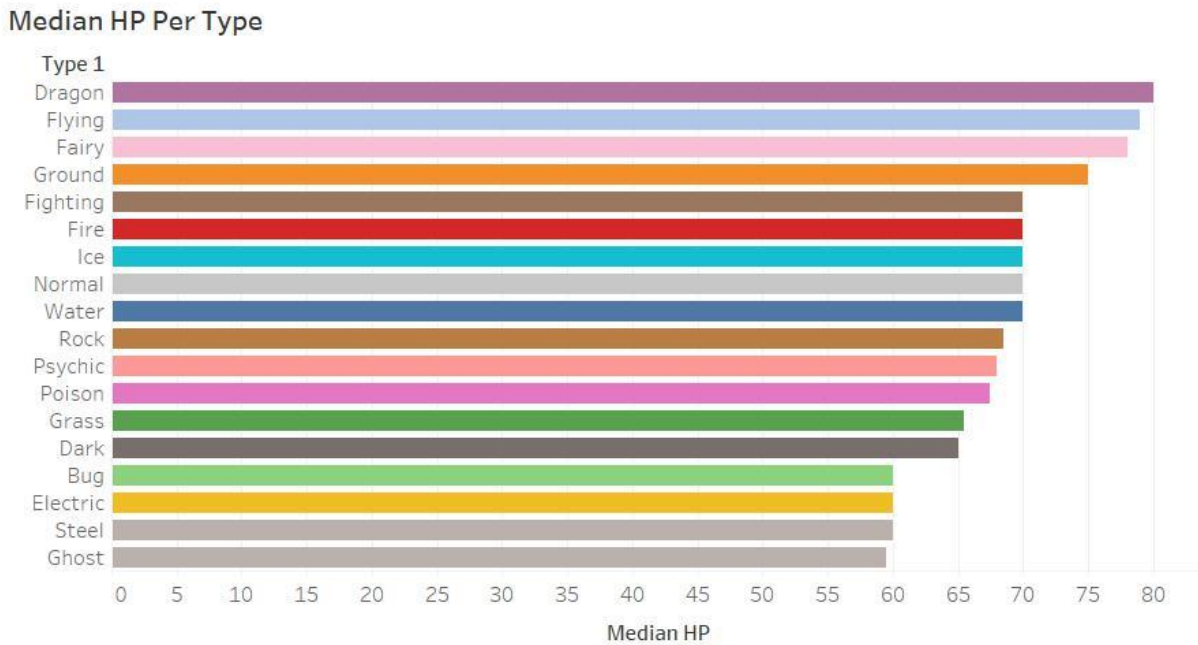


Median of Special Defense for each Type 1. Color shows details about Type 1.

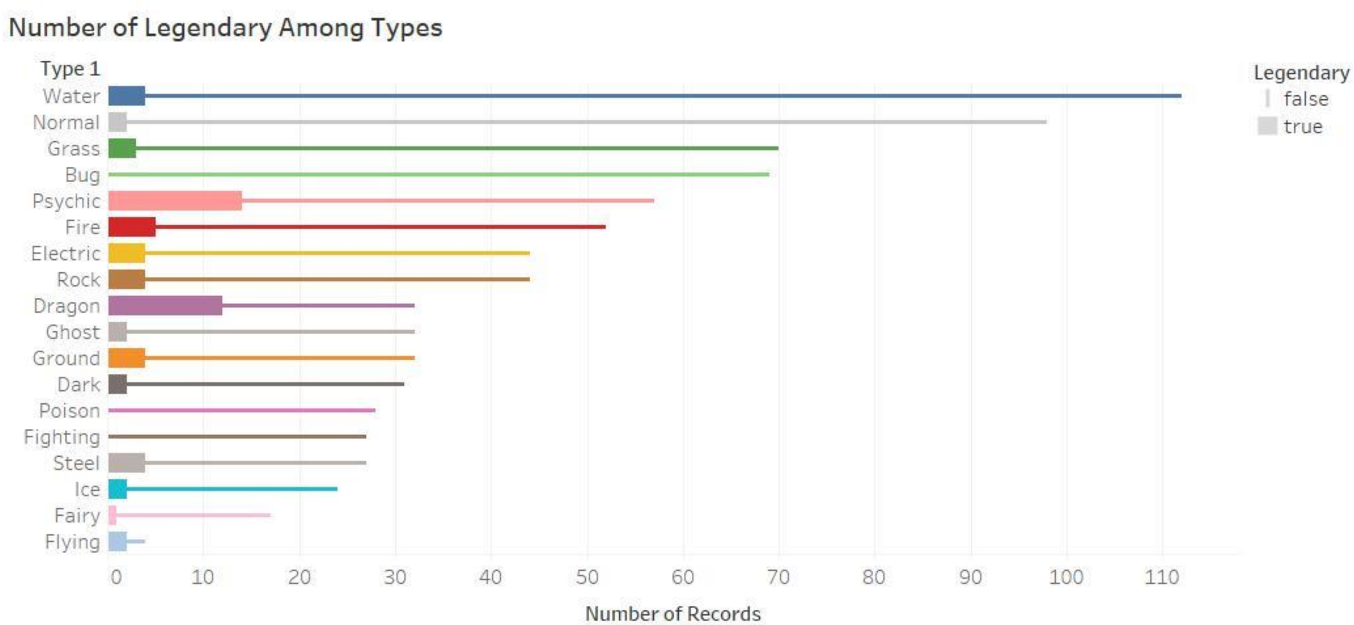
Median Speed Per Type



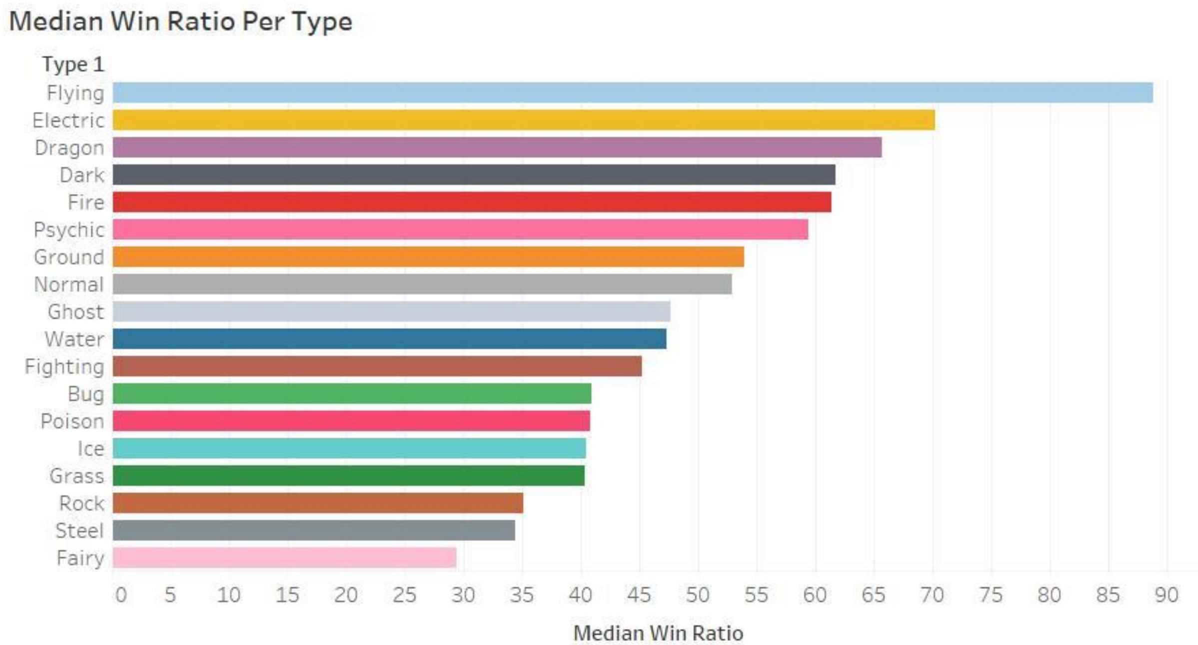
Median of Speed for each Type 1. Color shows details about Type 1.



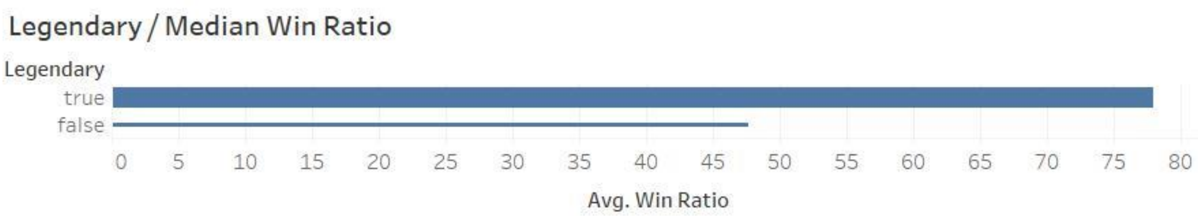
Median of HP for each Type 1. Color shows details about Type 1.



Sum of Number of Records for each Type 1. Color shows details about Type 1. Size shows details about Legendary. The view is filtered on Legendary, which keeps false and true.

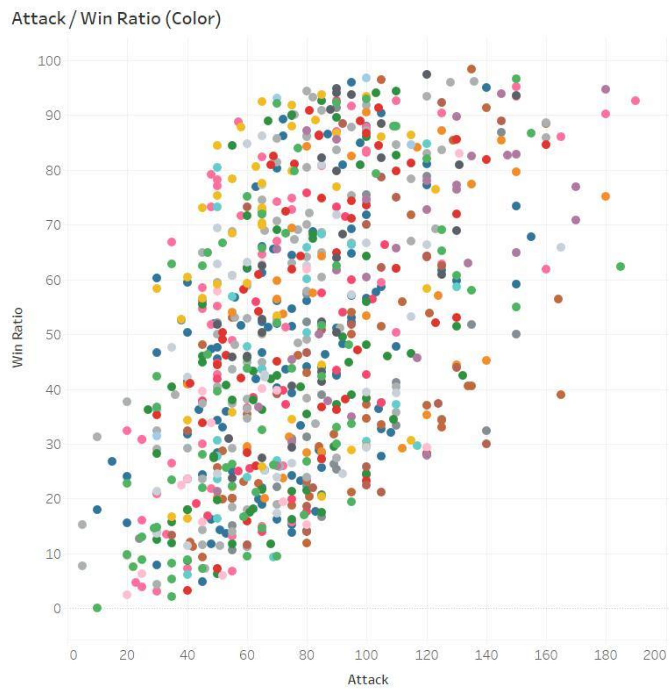


Median of Win Ratio for each Type 1. Color shows details about Type 1.

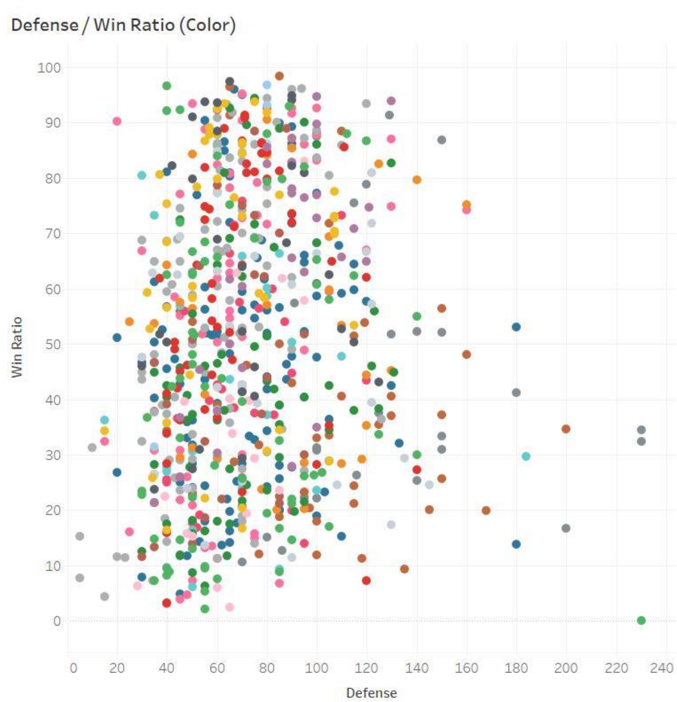


Average of Win Ratio for each Legendary. Size shows details about Legendary.

Win Ratio and Attribute Scatterplots

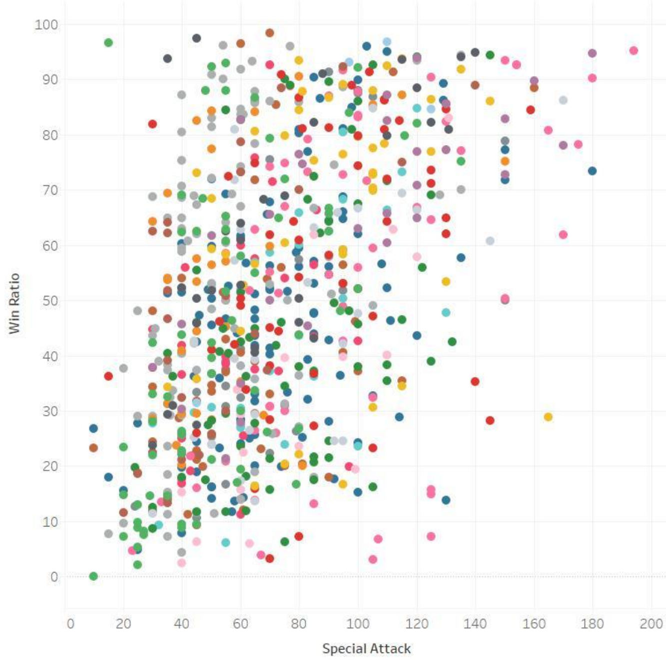


Sum of Attack vs. sum of Win Ratio. Color shows details about Type 1. Details are shown for Name.



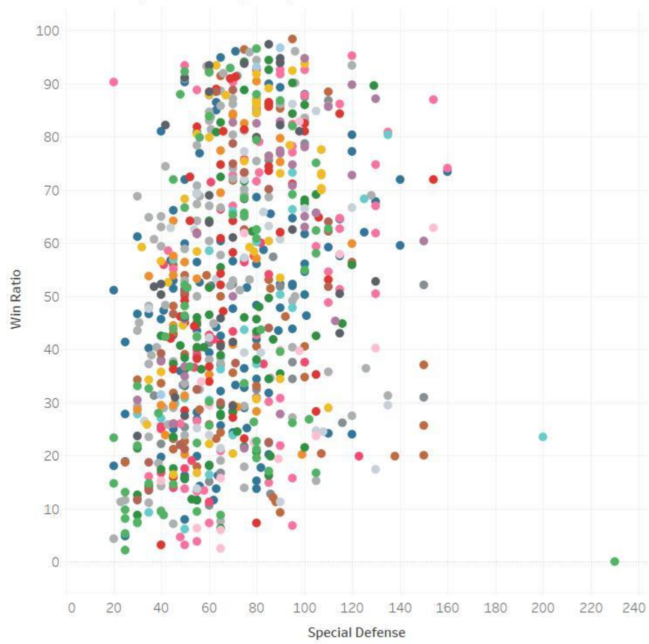
Sum of Defense vs. sum of Win Ratio. Color shows details about Type 1. Details are shown for Name.

Special Attack / Win Ratio (Color)



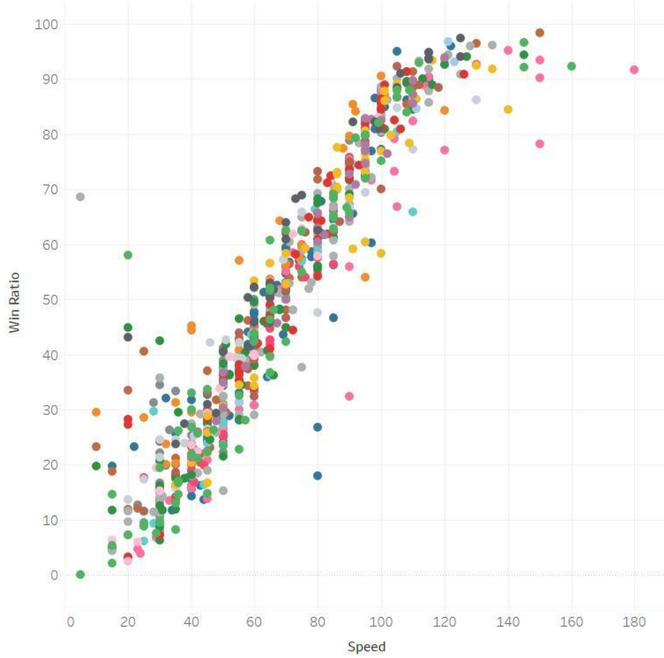
Sum of Special Attack vs. sum of Win Ratio. Color shows details about Type 1. Details are shown for Name.

Special Defense / Win Ratio (Color)



Sum of Special Defense vs. sum of Win Ratio. Color shows details about Type 1. Details are shown for Name.

Speed / Win Ratio (Color)



Sum of Speed vs. sum of Win Ratio. Color shows details about Type 1. Details are shown for Name.

HP / Win Ratio (Color)

Sum of HP vs. sum of Win Ratio. Color shows details about Type 1. Details are shown for Name.

Appendix C: ZeroR Calculation

Baseline performance (ZeroR. (n.d.).

ZeroR on Combats set.

Majority voting: Pokémon 2 winner

$$Acuracy = \frac{26399(correct\ Pokemon2\ guess)+0(correct\ Pokemon1\ guess)}{(26399+0)+(0(false\ Pokemon1\ guess)+23601(false\ Pokemon2\ guess))} \times 100 = 52.798\%$$

Appendix D: Experimenting with Size of Testing Set

Combat objects with three labels, 100 runs

Size of Testing Set (tuples)	Accuracy (%)
1000	78,61
2500	79,32
5000	80,12
10000	80,97
20000	81,70

Combat objects with percentiles, 100 runs

Size of Testing Set (tuples)	Accuracy (%)
1000	82,92
2500	82,97
5000	83,08
10000	82,55
20000	82,62

Appendix E: ID3 Attribute Selection Experiments

The average of 100 runs on random training set of size 5000 tuples

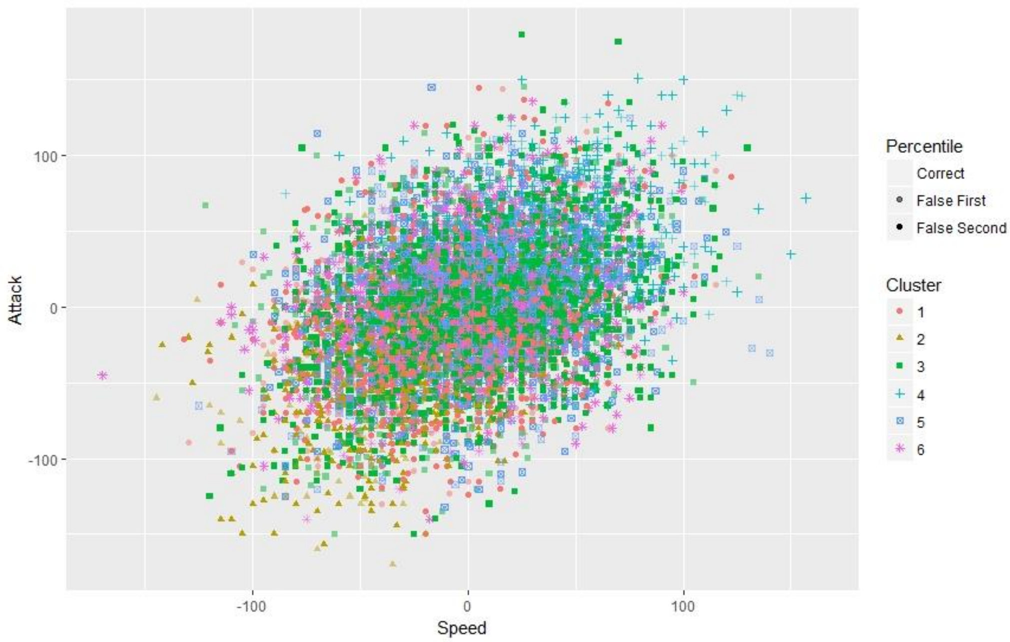
ID3 (with percentiles) trained on:	**Accuracy**
All attributes	81.8 %
All attribute except Speed	63,07%
Speed	83.01%
Attack	64,17%
Defense	58.02%
HP	60.884%
Special Defense	60.65%
Special Attack	63.4%
Legendary	57.01%
Stage	61.3%
Type	53.09%
Speed, Type	82.8%
Speed, Legendary	83.07%
Speed, Attack	82.84%
Speed, Special Attack	82.8%
Speed, HP	82.77%
Speed, Defense	82.82%
Speed, Special Defense	82.8%

Speed, Stage	82.8%
Speed, Legendary, Attack	82,839%
Speed, Legendary, Defense	82,29%
Speed, Legendary, HP	82.4%
Speed, Legendary, Special Attack	82.02%
Speed, Legendary, Special Defense	82.28%
Speed, Legendary, Stage	82.7%
Speed, Legendary, Type	82.7%
Legendary, Attack	65.4%
Legendary, Attack, Stage	66.15%
Legendary, Attack, Stage, Type	63.38%
Attack, Special Attack, Stage	64.3%
Attack, Special Attack, Stage, Legendary	64.3%
Speed, ClusterAttribute(Legendary, Attack, Stage)	82.8%
Speed, Legendary, ClusterAttribute(Speed, Special Attack, Attack, Stage)	82.79%
Speed, ClusterAttribute(Everything except speed)	82.77%
Speed, ClusterAttribute(HP, Defense)	82.82%
Speed, ClusterAttribute(Defense, Special Defense)	82.81%
Speed, ClusterAttribute(HP, Defense, Special Defense)	82.7%
Speed, ClusterAttribute(Attack, HP)	82.83%
Speed, ClusterAttribute(Attack, Type)	82.78%

ID3 (with speed tweaking) trained on:	**Accuracy**
All attributes (not clustering)	92.62%
All attribute except Speed and ClusterAttribute	62,99%
Speed	94.05%
Speed and Legendary	94.0509%
Speed, Stage	94.06%
Speed, Attack	93.99%
Speed, Type	94.95%
Speed, Defense	94.03%
Speed, HP	94.01%
Speed, Special Attack	94.04%
Speed, Special Defense	94.02%
Speed, Type, Legendary	94.87%
Speed, Type, Stage	94.46%
Speed, ClusterAttribute(Type, Stage)	94.05%
Speed, ClusterAttribute(Type, Generation)	94.05%
Speed, ClusterAttribute(Type, Legendary)	94.05%
Speed, ClusterAttribute(Everything except Speed)	94.054%
Speed, ClusterAttribute(Attack, Type)	94.057%

Appendix F: Visual Cluster Trace of Decision Tree Improvement

Before Speed tweaking:



After Speed tweaking: