# Predicting Combat Results from Pokémon Characteristics

STOR 565 Final Report
Dr. Zhengwu Zhang
Team Anonymous
5/3/2022

## Abstract

Pokémon is a series of video games with continuous popularity around the globe. This paper aims to study Pokémon combats and explore the relationship between the characteristics of Pokémons and their combat results. Obtained from Kaggle (Pokémon- Weedle's Cave | Kaggle), the datasets contained descriptive data of 800 Pokémons and 50,000 combat results. We used classification methods including logistic regression, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis(QDA), K-Nearest Neighbors(KNN), Support Vector Machines(SVM), and tree models. We concluded that Random Forest was the most effective model in predicting winners of Pokémon combats with an accuracy of 96.68%, with speed difference between Pokémons as the most important predictor.

## 1. Introduction

Pokémon was a game series developed by Nintendo and was later adapted into animations and movies. Pokémons are creatures of all shapes and sizes in the wild or raised by trainers. In many Pokémon games, the player is a trainer who catches Pokémons, trains them, and battles against other trainers' Pokémons.

Pokémon battles are turn-based. Each battle consists of repeated turns, and each turn requires all participants to choose an action to take. The outcomes of those actions are revealed immediately through changes in Pokémon status, such as a decrease in hitpoint, which directly influence the rest of the combat. Eventually, the Pokémon reaches zero hit points first loss, and the one that lasts longer wins (The Cave of Dragonflies, 2019). Inspired by the game process, we were curious about how well the combat results can be predicted given the characteristics of two Pokémons. We also wanted to explore the effectiveness of various classification methods in this scenario. With this research, we hoped to explore the underlying game mechanism of Pokémon and inspire players to prepare better combat strategies.

## 2. Research Methodology

This research methodology details the empirical approach we will use to predict combat results using Pokémon characteristics provided in the game setting. I

### 2.1 Description of Data

We obtained files containing Pokémon characteristics and combat results from Kaggle. The "Pokémon" dataset contains 800 distinct Pokémons and 10 of their attribute values. A brief description of Pokémon's attribute values is provided below.

- 1st type = Primary type
- 2nd type = Secondary type

- Hit Points = Maximum damage a Pokémon can take
- Attack = Value of attack damage that a Pokémon can cause
- Defense = The extent a Pokémon can mitigate damage from attacks
- Special Attack= Value of special attack damage that a Pokémon can cause
- Special Defense = The extent a Pokémon can mitigate damage from special attacks
- Speed = Value that determines the place a Pokémon will strike in Battle

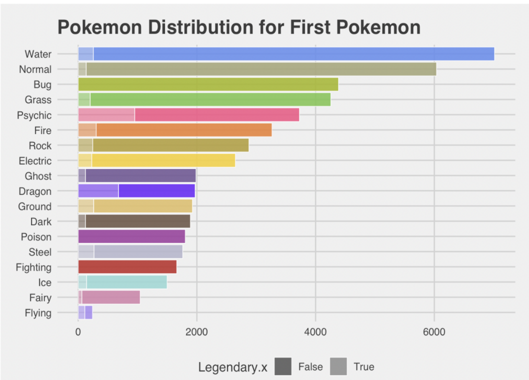
The “Combat” dataset records 50,000 combats between different Pokémons and the corresponding results.

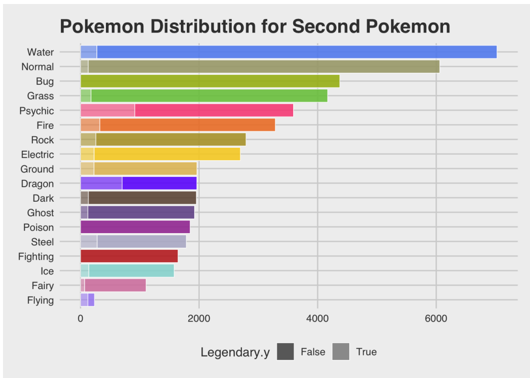
- First_Pokémon: ID for Pokémon that attacks first
- Second_Pokémon: ID for Pokémon that moves later
- Winner: ID for the winning Pokémon

We cleaned the data by eliminating redundant variables such as the names of Pokémon and Pokémon types. We integrated the two datasets using the “dplyr” package in R for effective modeling. The new dataset includes the characteristics of both the first and second Pokémon and their combat results. To make the combat results easier to use in classification, we used 1 to denote combat in which the first Pokémon wins, and 0 otherwise.

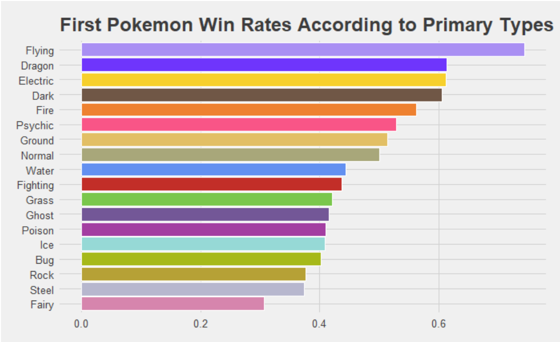
### 2.2 Exploratory Analysis

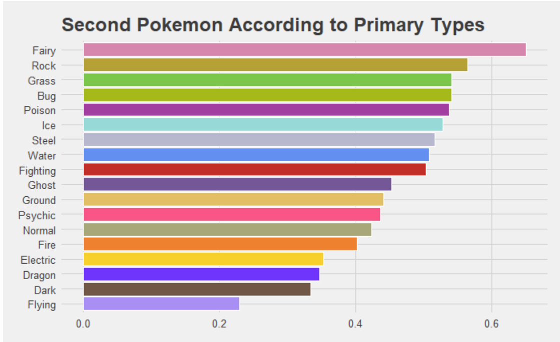
We were curious about how types and legendary status impact the combat results. Therefore, we examined the distribution of primary types of both the first and the second Pokémon with the proportion of legendary Pokémons.



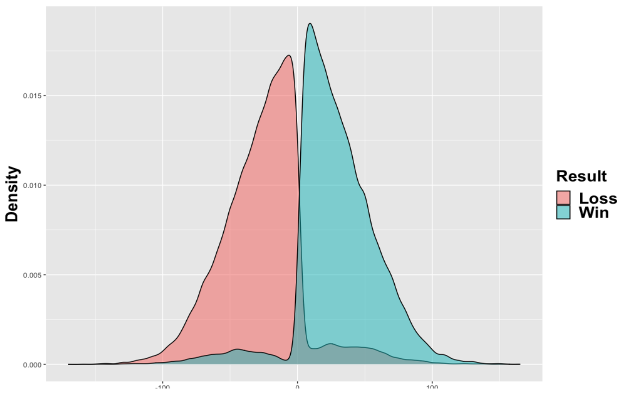




The win rate distributions concerning the first and second Pokémons’ types are shown below. Surprisingly, the Pokémons with dominant winning rates as first attackers, such as flying, dragon, and electric Pokémons, experienced a plummet in winning rates when they moved as the latter attackers. This shift implied that there might be more complex factors influencing the match results, which demanded further studying.




We then drew several density plots to observe the distribution of our variables. The plot below showed a clear boundary around 0, meaning that the speed differences are mostly positive when the first Pokémon wins. Additionally, we added 6 variables to the original dataset to form the second dataset highlighting the differences between rivals in terms of attack, defense, hitpoints, special attack, special defense, and speed.



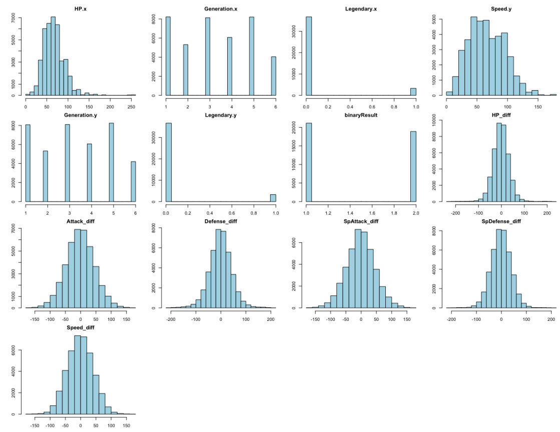
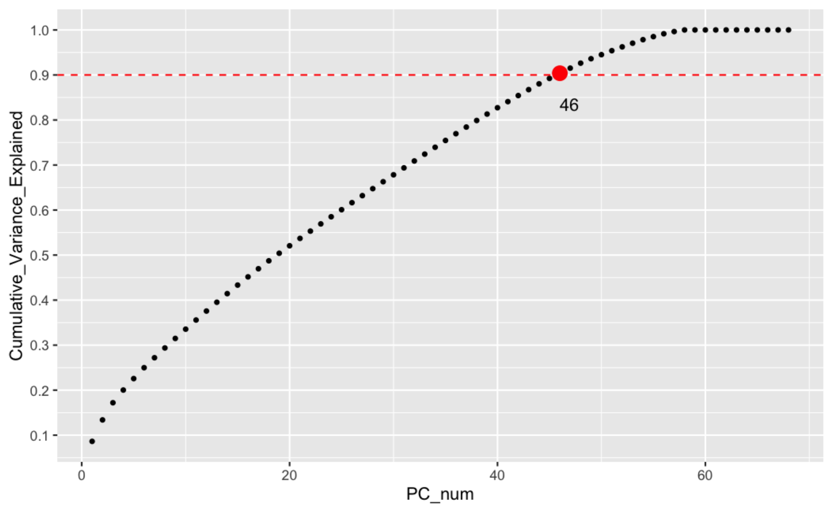



**Figure: Difference in Speed of Two Pokémons** **Figure: Histograms of Variables After Subset Selection**

Next, we performed subset selection, including best subset selection, forward stepwise selection, and backward stepwise selection. The results of different subset selection methods were similar and led us to our third dataset. The remaining numeric variables were mostly normal, and the six newly added variables remained, suggesting that they were important predictors worth considering. We also conducted principal component analysis (PCA) with the second dataset (including the difference variables) to reduce dimensionality. We discovered the first 46 components that cumulatively account for 90% variance of the original dataset, as is shown in the graph.



Our fourth dataset was constructed to highlight the impact of the Speed_diff variables. To sum up, there were 4 datasets used in our project. For each dataset, 80% of the data were split into the training segmentation and 20% into the testing segmentation.

### 2.3 Method Overview

In this study, we used R to perform multiple commonly used binary classification algorithms, including logistic regression, LDA, QDA, SVM, KNN, decision tree, and random forest to form prediction models for Pokémon combat results. With these models, we would be able to identify the most effective predictors from all potential predictors based on processed Pokémon features. Cross-validation, AUC (area under ROC curve), and accuracy were used to evaluate models and compare their performances.

## 3. Results and Analysis

### 3.1 Logistic Regression

Logistic regression was first conducted on the dataset with additional difference variables (the second dataset). The model output indicated that Speed_diff is the most significant variable in terms of p-value, agreeing with our previous observation. Therefore, we fed the dataset without Speed_diff to the Logistic Regression model to derive the influence of the Speed_diff variable.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4740  567
         1  541 4152

               Accuracy : 0.8892
                 95% CI : (0.8829, 0.8953)
    No Information Rate : 0.5281
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7776

 Mcnemar's Test P-Value : 0.4526

            Sensitivity : 0.8976
            Specificity : 0.8798
         Pos Pred Value : 0.8932
         Neg Pred Value : 0.8847
             Prevalence : 0.5281
         Detection Rate : 0.4740
   Detection Prevalence : 0.5307
      Balanced Accuracy : 0.8887
```

```
Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    -0.2914602  0.0757060  -3.850 0.000118 ***
HP.x            0.0011396  0.0008276   1.377 0.168490
Generation.x    0.0357609  0.0088261   4.052 5.08e-05 ***
Speed.y         0.0008721  0.0006875   1.269 0.204574
Generation.y   -0.0585585  0.0088409  -6.624 3.51e-11 ***
HP_diff         0.0011523  0.0006344   1.816 0.069329 .
Attack_diff     0.0094992  0.0004432  21.434  < 2e-16 ***
Defense_diff    0.0017266  0.0004264   4.049 5.14e-05 ***
SpAttack_diff  -0.0012511  0.0004413  -2.835 0.004584 **
SpDefense_diff  0.0012948  0.0005119   2.529 0.011425 *
Speed_diff      0.0650111  0.0007803  83.319  < 2e-16 ***
```

The second logistic regression model has worse performance. Specifically, the accuracy decreased by around 15% from 0.8892 to 0.7499; AUC decreased approximately 10% from 0.9265 to 0.8276. Sensitivity decreased from 0.8976 to 0.7599, and specificity decreased from 0.8798 to 0.7387. It proved Speed_diff's significance as a predictor. Hitpoints, speed, generation, and all the difference variables were influential in terms of p-value as well.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4013 1233
         1 1268 3486

               Accuracy : 0.7499
                 95% CI : (0.7413, 0.7584)
    No Information Rate : 0.5281
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.4984

 Mcnemar's Test P-Value : 0.4966

            Sensitivity : 0.7599
            Specificity : 0.7387
         Pos Pred Value : 0.7650
         Neg Pred Value : 0.7333
             Prevalence : 0.5281
         Detection Rate : 0.4013
   Detection Prevalence : 0.5246
      Balanced Accuracy : 0.7493
```

```
Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.6421494  0.0629531  26.085  < 2e-16 ***
HP.x           0.0102438  0.0007043  14.544  < 2e-16 ***
Generation.x  -0.0065291  0.0073596  -0.887    0.375
Speed.y       -0.0355406  0.0005228 -67.977  < 2e-16 ***
Generation.y  -0.0314775  0.0073404  -4.288 1.8e-05 ***
HP_diff       -0.0067319  0.0005528 -12.178  < 2e-16 ***
Attack_diff    0.0159735  0.0003717  42.979  < 2e-16 ***
Defense_diff  -0.0061212  0.0003564 -17.174  < 2e-16 ***
SpAttack_diff  0.0080798  0.0003544  22.797  < 2e-16 ***
SpDefense_diff 0.0053822  0.0004271  12.601  < 2e-16 ***
```

## 3.2 LDA & QDA

For both linear and quadratic discriminatory analysis (LDA & QDA), we used the dataset with principal components (the fourth dataset) in order to avoid multicollinearity issues. From the confusion matrix attached below, we obtained an LDA model with an accuracy of 0.8555, a sensitivity of 0.8669, and a specificity of 0.8428. The AUC value was 0.9149. On the other hand, the QDA model reported accuracy of 0.7337, a sensitivity of 0.7434, and a specificity of 0.7228; the AUC value was 0.7961. We concluded that LDA performed better than QDA for our dataset, with an approximate difference of 0.12 for each value, indicating the combat results were more likely to be linearly divided than quadratically divided.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 4578  742
         1  703 3977

               Accuracy : 0.8555
                 95% CI : (0.8485, 0.8623)
    No Information Rate : 0.5281
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.71

 Mcnemar's Test P-Value : 0.3175

            Sensitivity : 0.8669
            Specificity : 0.8428
         Pos Pred Value : 0.8605
         Neg Pred Value : 0.8498
             Prevalence : 0.5281
         Detection Rate : 0.4578
   Detection Prevalence : 0.5320
      Balanced Accuracy : 0.8548
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 3926 1308
         1 1355 3411

               Accuracy : 0.7337
                 95% CI : (0.7249, 0.7423)
    No Information Rate : 0.5281
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.466

 Mcnemar's Test P-Value : 0.3727

            Sensitivity : 0.7434
            Specificity : 0.7228
         Pos Pred Value : 0.7501
         Neg Pred Value : 0.7157
             Prevalence : 0.5281
         Detection Rate : 0.3926
   Detection Prevalence : 0.5234
      Balanced Accuracy : 0.7331
```

**Figure: The Confusion Matrix for LDA (left) and QDA (right) Models**

### 3.3 KNN & SVM with three kernels

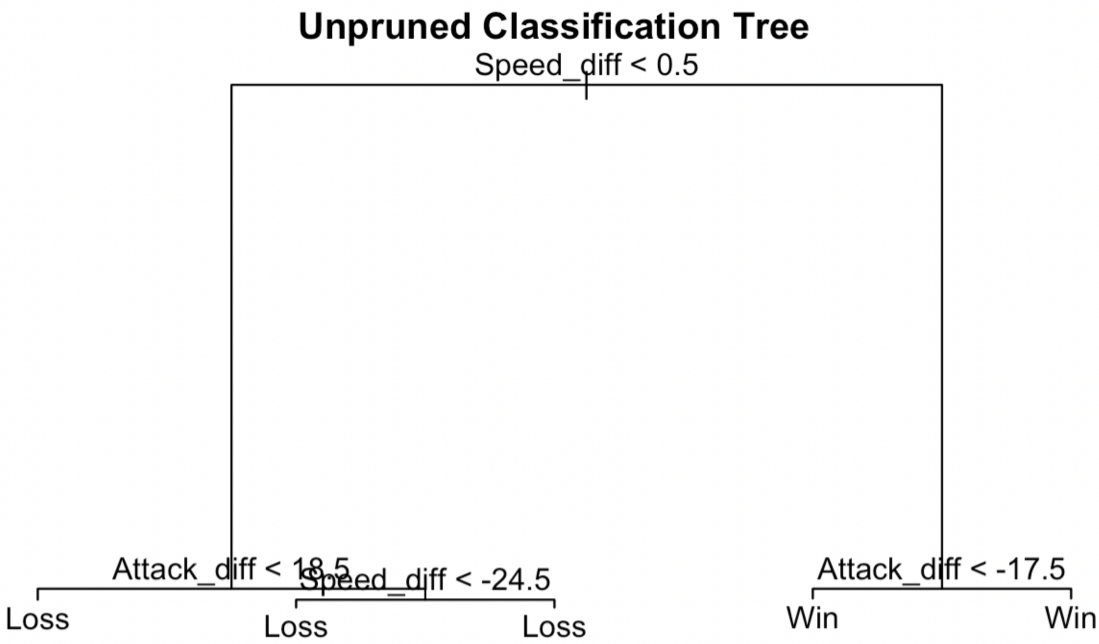
The KNN method considering the Euclidean distance worked well with numerical attributes, which justified our earlier decision to transform categorical variables into binary variables. After experimenting, we discovered that the model performs best with a k-value of 49, yielding an accuracy of 0.8449 and an AUC of 0.8433.

To fully understand the SVM model, we experimented with three different kernels: linear, polynomial, and radial. We expected higher accuracy on the linear kernel based on the results of LDA. Due to the large size of the dataset, we only chose 1,000 rows of data randomly to feed the three models.

The linear kernel with the best cost of 0.5 achieved an accuracy of 0.8185 and an AUC of 0.8881. The polynomial kernel with the best cost of 1 and a degree of 3 had 0.7308 accuracy and 0.8124 AUC. In addition, with the best cost of 1 and a gamma of 0.01, the radial kernel produced an accuracy of 0.6813 and an AUC of 0.732. Overall, it was reasonable to conclude that an SVM with a linear kernel performs best with the dataset of this study.

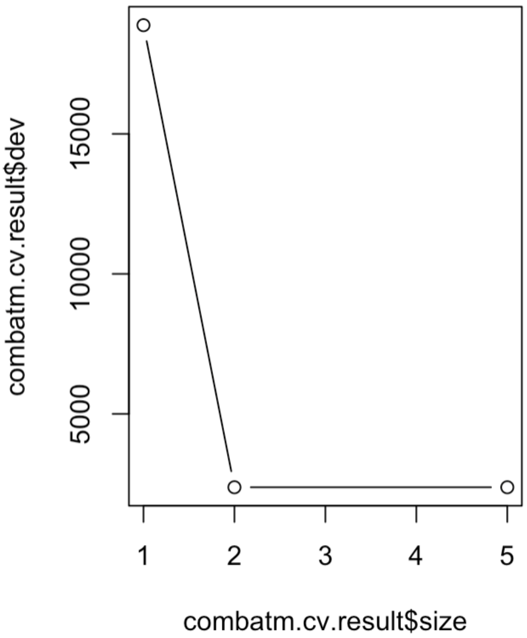
### 3.4 Decision Tree & Random Forest

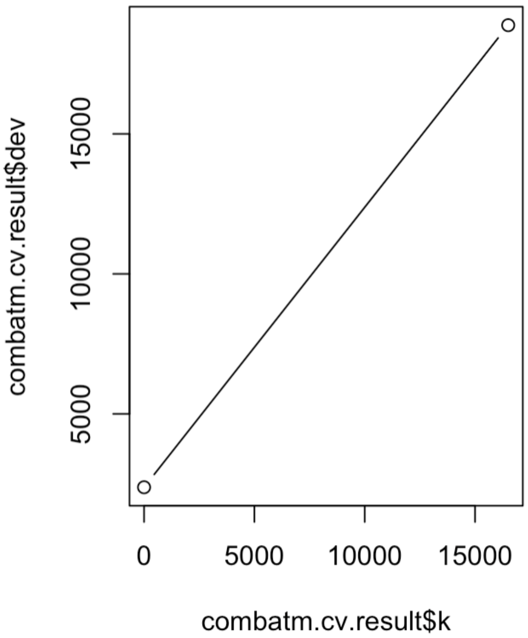
For the decision tree models, we first experimented with a classification tree. We converted the binary win results to "Win" and "Loss" to realize a more straightforward expression. The unpruned classification tree had 5 terminal nodes and a misclassification rate of 0.05948. Since post pruning usually results in a better tree than pre-pruning, weakest-link pruning was employed. The improved classification tree gave us a 94.06% accuracy. The nodes showed that a fight would be predicted as a loss if the speed difference between a Pokémon and its rival is less than 0.5, regardless of the attack difference. Although this is not 100% true for every single game, the model does allow us to generalize this finding. The unpruned classification tree can be found below:

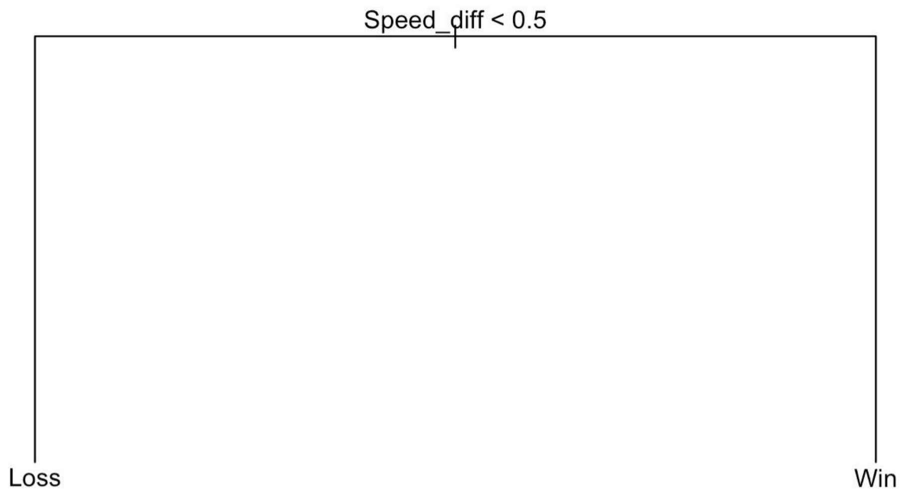


```
Classification tree:
tree(formula = binaryResult ~ ., data = combatm.train)
Variables actually used in tree construction:
[1] "Speed_diff"  "Attack_diff"
Number of terminal nodes:  5
Residual mean deviance:  0.3935 = 15740 / 40000
Misclassification error rate: 0.05948 = 2379 / 40000

combatm.tree_pred    0    1
             Loss 4902  215
             Win   379 4504
```



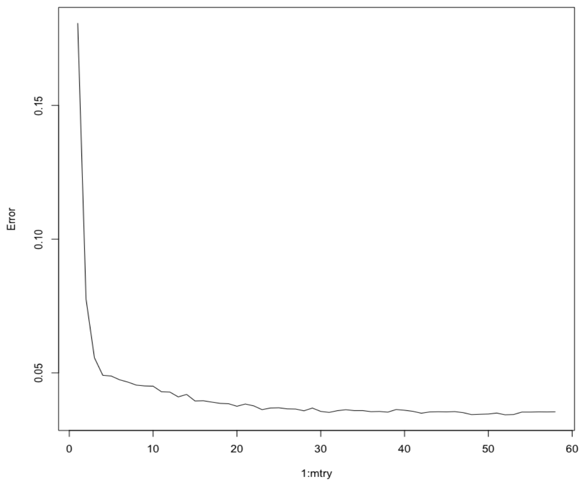



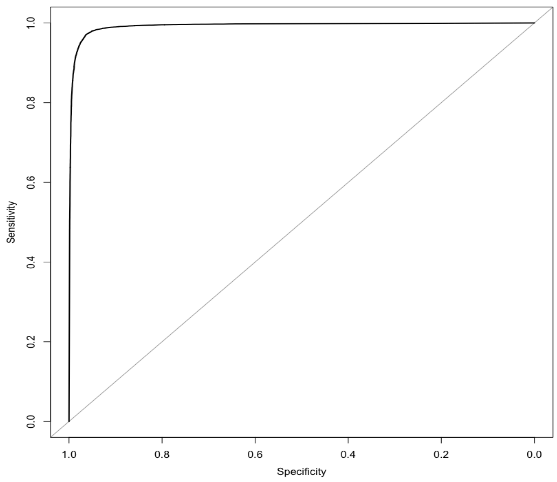



```
combatm.prune.pred    0    1
                Loss 4902  215
                Win   379 4504

Classification tree:
snip.tree(tree = combatm.result_tree, nodes = 2:3)
Variables actually used in tree construction:
[1] "Speed_diff"
Number of terminal nodes:  2
Residual mean deviance:  0.4465 = 17860 / 40000
Misclassification error rate: 0.05948 = 2379 / 40000
```

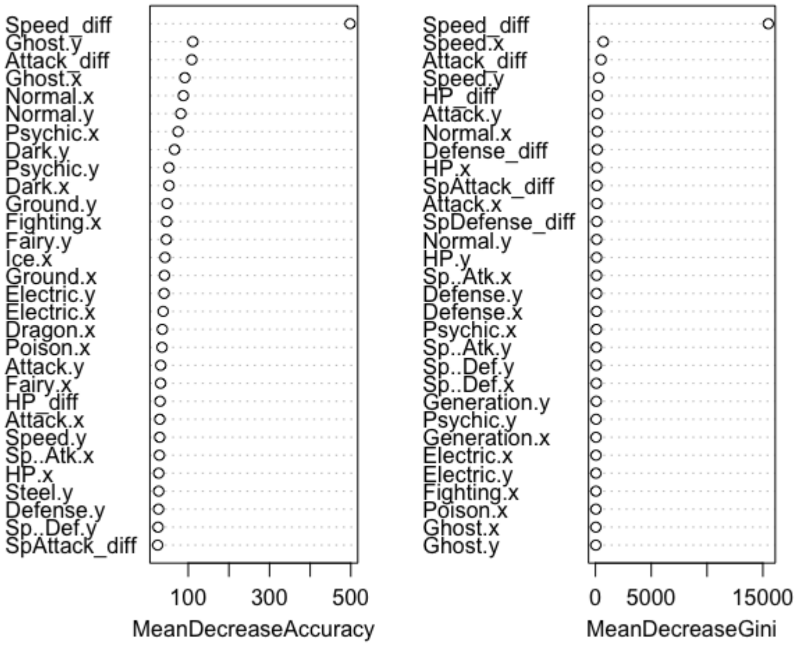
We then move on to the random forest model consisting of multiple tree models. We loop through the mtry and obtain an optimal value of 52 while getting the smallest error. And using that optimal mtry in the random forest model. The model yielded an accuracy of 0.9668 and an AUC of 0.9923, which rank random forest as the top-performing algorithm. The results also agree with previous findings that the variable Speed-diff is way more important than the others.




The figures above demonstrated that Speed.diff is by far the most important variable in our model, which follows the previous result we obtained from the decision tree model. Overall, the tree models provide additional evidence of the first-mover advantage.

```
combatm.best.pred    0    1
               Loss 5109  160
               Win   172 4559
Setting levels: control = Loss, case = Win
Setting direction: controls < cases
Area under the curve: 0.9923
```




## 4. Limitations

Our current methodology has several weaknesses that future studies can improve on. Firstly, although the first type of Pokémon is considered in our model, we are not able to clearly identify the differences in characteristic value between the various types. Specifically, it is difficult to quantify how each type of Pokémon would perform differently when facing every other type of Pokémon. Secondly, the variable secondary type is not included in this study as some Pokémons did not have a secondary type. Thirdly, our study did not provide an answer for how first-mover advantage changes along with Pokémon types, which would be an interesting question to look into. Furthermore, some of the models, such as KNN, used part of the dataset instead of all due to computational constraints. Lastly, other advanced techniques, such as neural networks, are not incorporated in this study but have the potential to yield more effective and accurate predictions.

## 5. Conclusion

In this study, six machine-learning-based classification algorithms were used to predict the winning rate of single combat with Pokémon characteristics. The prediction models yield an accuracy of around 88-95%, and the random forest model produces the best overall performance both in terms of accuracy and AUC. The p-values provided by the logistic regression model and the importance scores suggested in the tree models can be used in subset selection and influential variable identification. By applying the same model to modified datasets, we can also explore the impact of each variable on the prediction. The results also suggest that the speed difference between Pokémons has overwhelming predictive power compared with other variables, and the first mover in combat does enjoy an advantage.

Future studies in Pokémon data can take additional variables such as the secondary types into consideration. More advanced techniques such as neural networks can also be used to explore the inner connection between the variables and potentially boost prediction outcomes.

## References


1. https://en.wikipedia.org/wiki/Pok%C3%A9mon_(video_game_series)
2. https://www.dragonflycave.com/mechanics/battling-basics#whatis