Predicting Pokemon Primary Type

Victory (Victor) Ma

Features

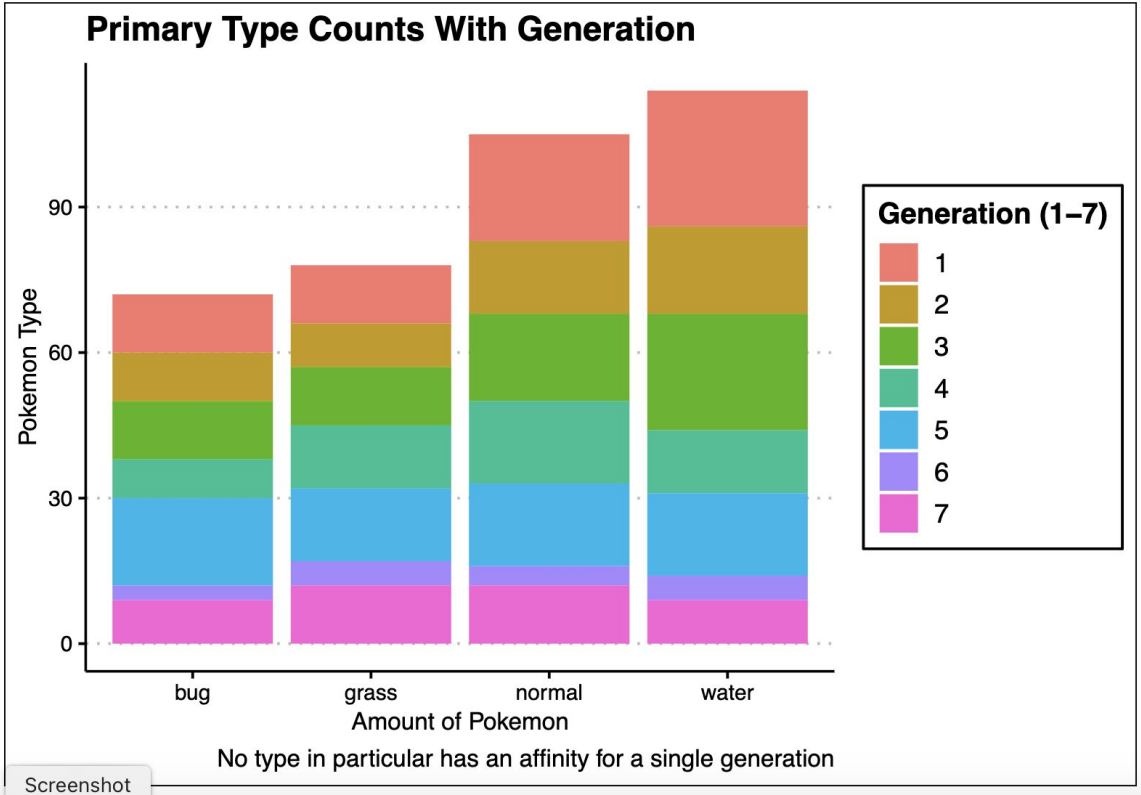
This project aims to predict the primary type (type_1) of a Pokemon based on its various characteristics.
Accurately predicting a Pokemon's primary type can offer insights into game design principles and the underlying patterns that go into a successful video game franchise.

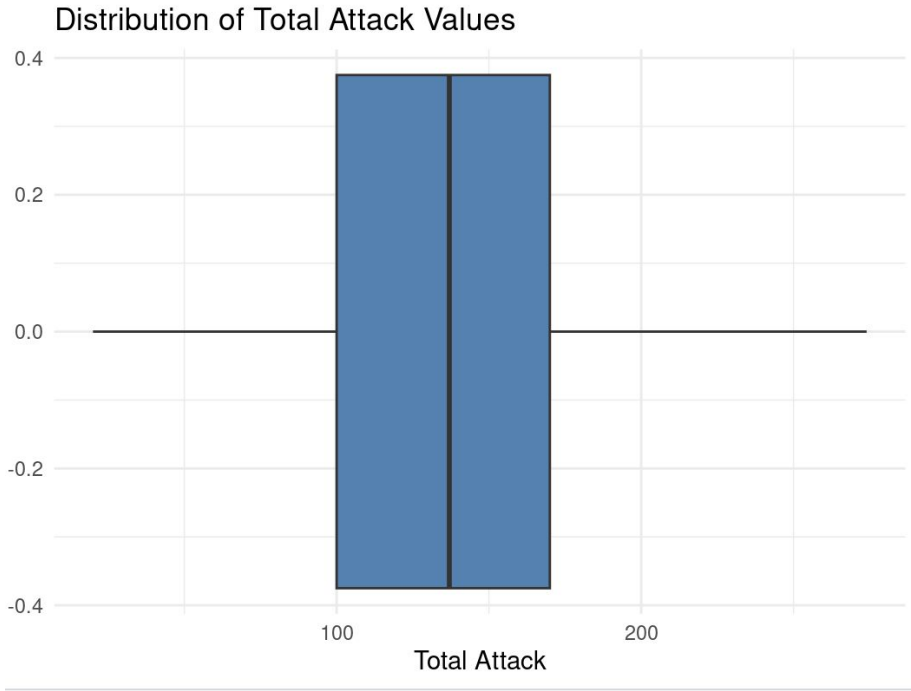
Table 1: Definitions of Variables

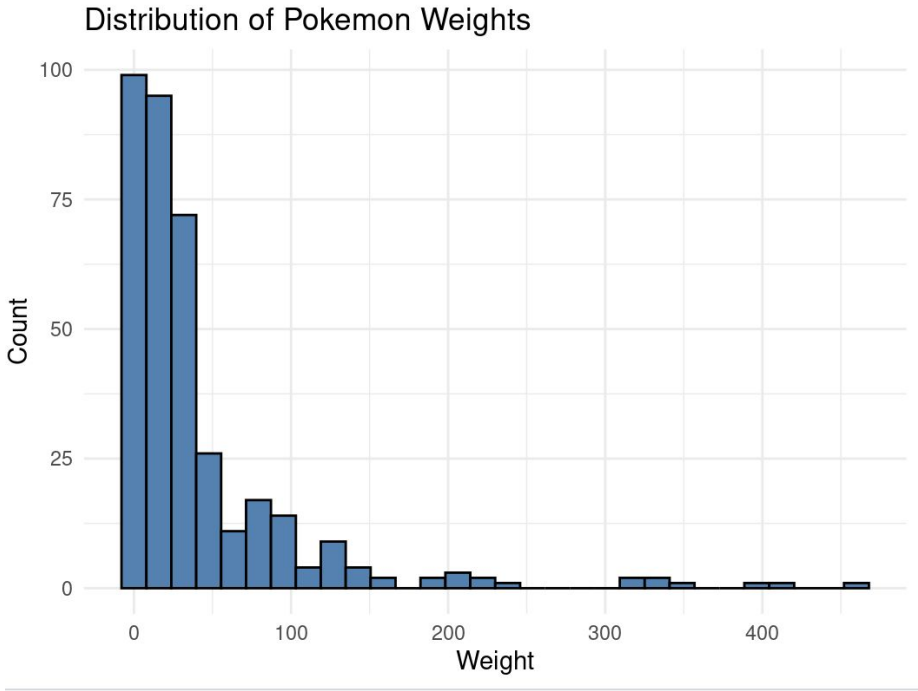
Name	Variable	Role	Definition	Values
height	Explanatory	Numerical	The height of each pokemon.	Number Value
weight	Explanatory	Numerical	The weight of each pokemon.	Number Value
base_experience	Explanatory	Numerical	The base experience of each Pokemon.	Number Value
type_1	Response	Categorical	The primary type.	1 out of 4 different types
has_type_2	Explanatory	Categorical	Whether it has a secondary type.	Yes/No (0 or 1)
hp	Explanatory	Numerical	The HP (hit points).	Number Value
total_attack	Explanatory	Numerical	The total attack points.	Number Value
total_defense	Explanatory	Numerical	The total defense points.	Number Value
speed	Explanatory	Numerical	The speed.	Number Value
has_color_2	Explanatory	Categorical	Whether it has a secondary color.	Yes/No (0 or 1)
has_color_f	Explanatory	Categorical	Whether it has a final color.	Yes/No (0 or 1)
egg_group_1	Explanatory	Categorical	The primary egg group.	1 out of 15 egg groups
has_egg_group_2	Explanatory	Categorical	Whether it has a secondary egg group.	Yes/No (0 or 1)
generation_id	Explanatory	Categorical	The generation ID of each Pokemon.	Value 1-7 Indicating Generation

Response Distribution

There are eighteen possible types that a Pokemon could be. We were recommended by Professor Davila to reduce this down to four, for simplicity. We noticed that generation appear to be distributed quite evenly between the types.

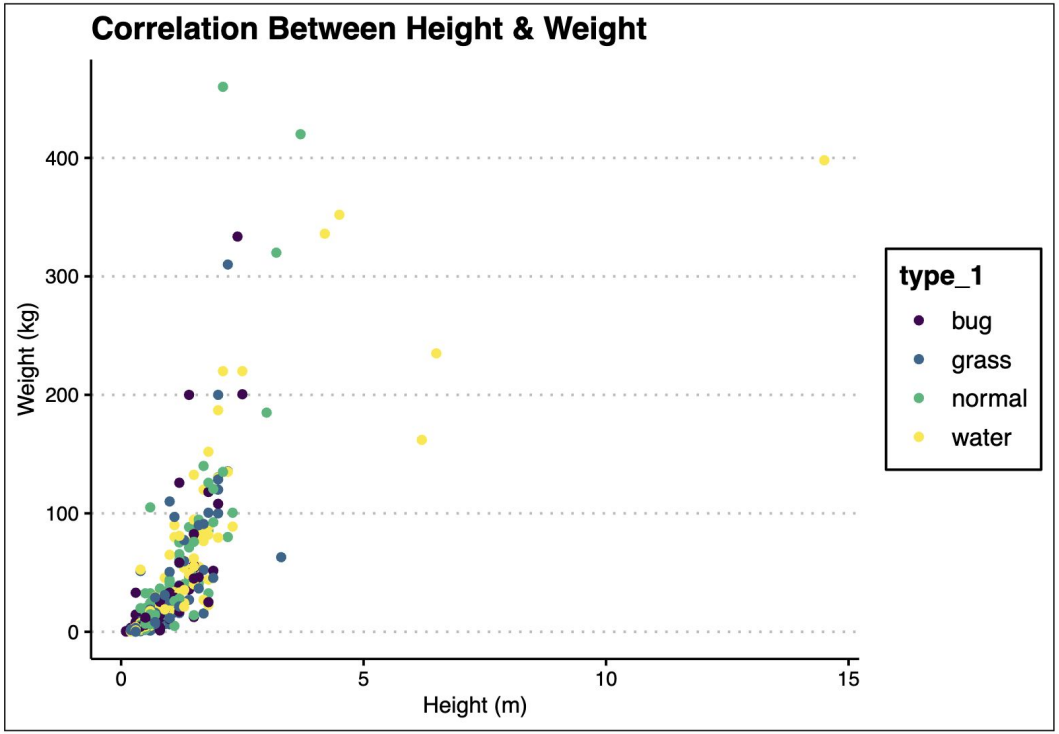



Distribution of Total Attack Values
0.4
0.2
0.0
-0.2
-0.4
100
200
Total Attack


Distribution of Pokemon Weights
100
75
50
25
0
Count
0
100
200
300
400
Weight

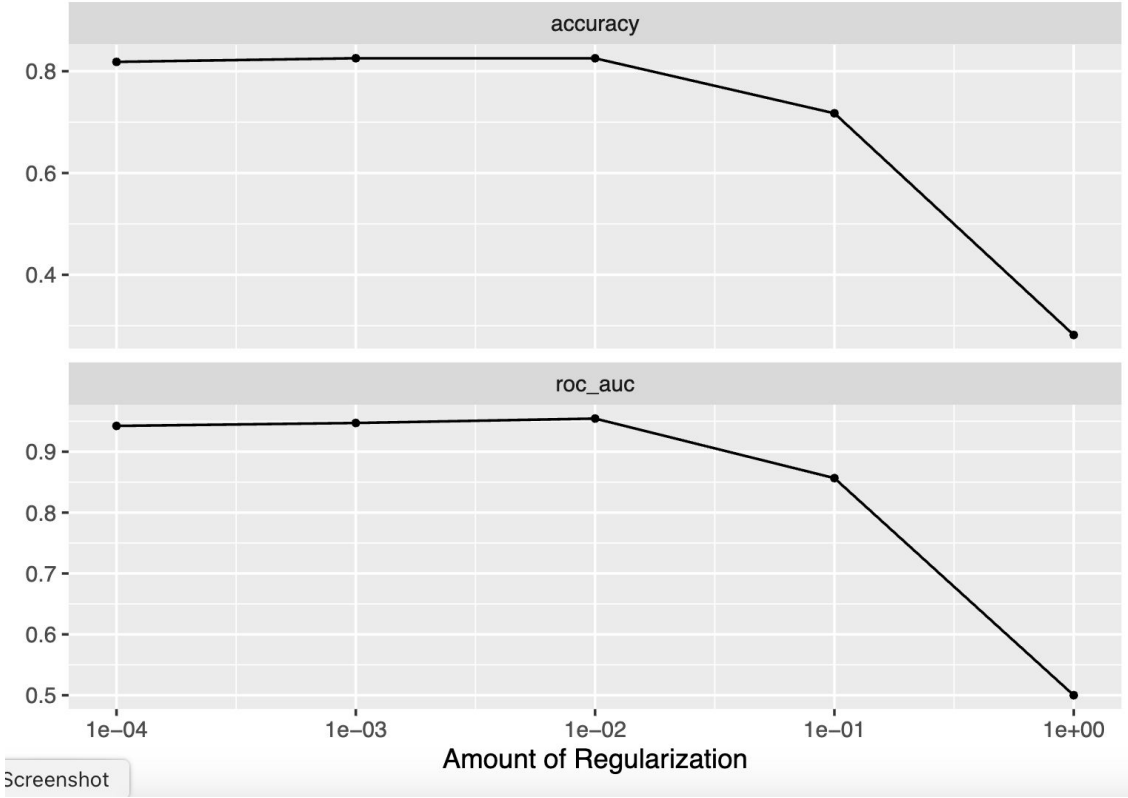
Notable Feature Distribution

There seems to be a moderately positive correlation between height and weight. This scatterplot gives us an idea on some key physical attributes that we can find in some types. Here, we can see that many outliers are water types (in yellow).



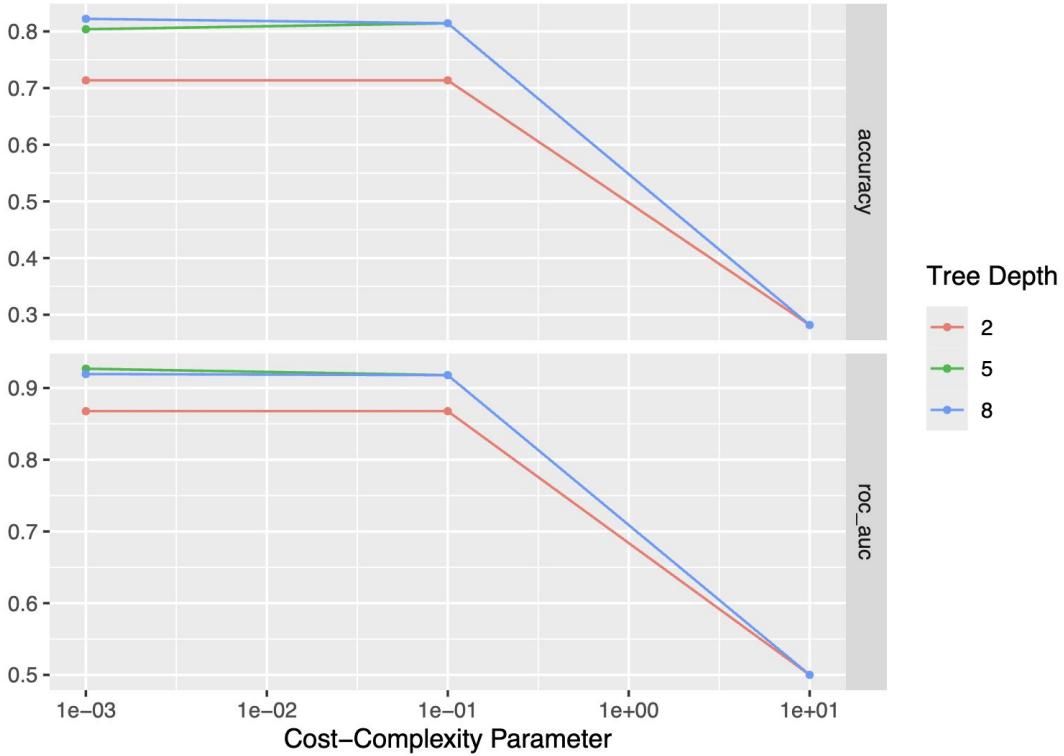
First Model: LASSO

Our initial testing found that ‘color_1’ appears to be making our models overfit, perhaps due to the fact that many if not all Pokemon have a certain color that they make them distinct from other Pokemon. Here, we optimize for ROC AUC and receive a high accuracy at 81.5% at penalty 0.01.



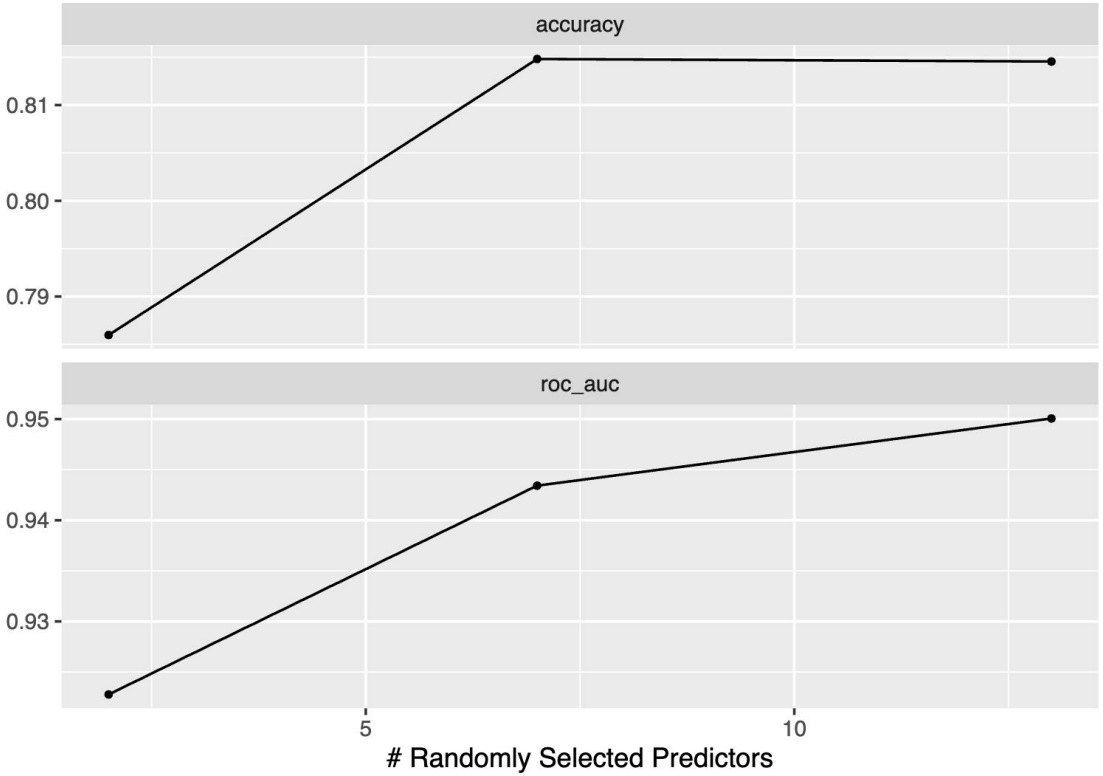
Second Model: Decision Tree

For our second model, we used a Decision Tree. Our ROC AUC value is optimized at around a tree_depth of 5 and a cost complexity of 0.1.



Random Forest

This graph shows how the number of randomly selected predictors (mtry) influences the accuracy and ROC AUC of our Random Forest model. Optimal performance appears to be achieved at around the value of 13.



Conclusion

Based on the test set accuracy, the Multinomial LASSO Regression model performed the best, with an accuracy of 81.5%, precision of 83.7%, and a sensitivity of 80.9%. According to the confusion matrix, our model had the least amount of trouble predicting 'bug' types, while having a bit of trouble with 'grass' and 'normal' types. Due to the nature of the video game, it is possible that 'bug' Pokemon tend to share more similar traits in comparison to other Pokemon types among each other (small, green, weak, etc.).

Table 2: Model Comparison Summary

Model	Accuracy	Most.Important.Variables
Multinomial LASSO Regression	0.815	egg_group_1, generation_id, hp, has_egg_group2, total_attack
Classification Decision Tree	0.753	egg_group1, total_attack, has_egg_group_2, height, speed
Random Forest Classification	0.774	egg_group1, has_egg_group_2, hp, generation_id, weight

Conclusion

This project successfully explored the predictability of a Pokemon's primary type (type_1) using various attributes such as combat statistics, physical characteristics, and categorical identifiers. Through comprehensive data cleaning, wrangling, and application of the machine learning models we learned from class, we were able to uncover patterns beneficial for successful game development.

Table 2: Model Comparison Summary

Model	Accuracy	Most.Important.Variables
Multinomial LASSO Regression	0.815	egg_group_1, generation_id, hp, has_egg_group2, total_attack
Classification Decision Tree	0.753	egg_group1, total_attack, has_egg_group_2, height, speed
Random Forest Classification	0.774	egg_group1, has_egg_group_2, hp, generation_id, weight