Pokémon App & Machine Learning

Eduardo Martinez

May 7, 2021

Introduction

Pokémon has been popular since the original games – *Pokémon Yellow, Pokémon Red,* and *Pokémon Blue* – were released in 1996 and 1998 (*Pokémon Games*). Modern machine learning research pertaining to *The Pokémon Company* focuses on the relatively new mobile game: *Pokémon Go*. Wang (2017) researched new computer algorithms that would simplify big data collection and analysis related to mobile crowd sensing; this study incorporated Pokémon Go, but the methods and algorithms can be extended to a variety of big data analytics involving crowd sensing and computing. De Roure, Hendler, James, Nurmikko-Fuller, Van Kleek, & Willcox (2019) created a simulator of Pokémon Go that learned from player behaviors in the game in order to study the Social Machines represented in the game. It is challenging to find recent machine learning research on the pokémon games released before 2010 (i.e., the first four generations).

There are sparse articles that research characteristics of actual pokémon and the video games created in the 1990s and 2000s. Liapis (2018) executed AI to map pokémon types to their corresponding traits in the video games; then, he presented algorithms that can be used to invent new pokémon based on the AI learned about various pokémon types. Chen (2014) used game theory and Markov process to compute winning probabilities and expected time needed to complete the entire game. Machine learning research conducted several years ago has depreciated value compared to newer studies because machine learning is evolving at a rapid pace concurrently with technology. Evidently, Liapis' 2018 study involved much more advanced methods than Chen's study, which was conducted in 2014. It would be beneficial to apply modern machine learning and statistical methods on the first four generations of Pokémon.

There are a total of 493 Pokémon in the first four generations. However, players can travel with a maximum of 6 pokémon at a time in the games. (***Note***, these six active pokémon will be referred to as a lineup throughout this paper.) Consequently, players have to make difficult decisions about which pokémon they choose to include in their lineup.

Selecting a lineup is further complicated because there are 18 different Pokémon types (*Pokémon Types & Type Chart*). Evaluating all pokémon types is challenging since the games do not offer methods for comparing multiple pokémon at the same time. Each type has corresponding strengths and weaknesses that generally apply to other pokémon that are the same type. It is advantageous for players to have a primary lineup containing diverse pokémon types to maximize adaptability. Many pokémon are dual types, which may lead a player to believe that a lineup with six dual type pokémon is best. However, dual types are susceptible to more weaknesses since they usually inherit the weaknesses of each of their types.

Once a player identifies a type of pokémon they want to include in their lineup, they still have to decipher which pokémon is ideal from the long list of pokémon for a given type. Players consider other pokémon attributes; for example, base stats, abilities, physical characteristics, etc. are considered. There is no widely used method that players can utilize to quickly and efficiently compare pokémon. However, machine learning and statistical analysis can be created to compare the strengths and weaknesses of multiple pokémon via a user-friendly method.

Although it is intuitive to build the mathematically strongest line up possible, every player has unique preferences. Players might try to build a lineup that maximizes their strengths and minimizes their weaknesses; other players want to include certain types of pokémon in their lineup, whether it is beneficial or not; and some players consider a combination of both of these approaches.

Machine learning can make this decision easier by providing players a filtered list of pokémon based on a specified criteria. Machine learning can learn a player's preferences and provide a corresponding list of pokémon.

Study Background

This project had two primary purposes:

(1) Create a user-interactive Pokémon App to easily compare various characteristics of multiple pokémon simultaneously.
(2) Uncover the key traits that distinguish legendary pokémon from normal pokémon.

Several datasets obtained from Kaggle (Barradas, 2016; Larcher, 2018, Ojeda, 2019; Romero, 2020; Zorrilla, 2017) have been combined to create a new dataset that contains aspects of each dataset used. The variables in the combined dataset are described below:

- **ID**: The Pokémon's unique Pokédex Number, where values range from 1 to 493.
- **Name**: Only the primary pokémon for a given Pokédex Number was included in the data and the others were omitted. Similar to ID, there are 493 unique pokémon names.
- **Generation** (Gen.): The generation that a pokémon made its first appearance in. In this project, Generations 1-4 are included in the dataset used. A generation is a time period in which a set of video games were released around the same time period and contain the same pokémon in each games released during a given generation.
- **Status/Type**: A pokémon's is status either Normal or Legendary. In Generations 1-4 contains 458 Normal and 35 Legendary pokémon.
- **Type 1**: There are a total of 18 pokémon types: Bug, Dark, Dragon, Electric, Fairy, Fighting, Fire, Flying, Ghost, Grass, Ground, Ice, Normal, Poison, Psychic, Rock, Steel, and Water.
- **Type 2**: The secondary type of a dual-type pokémon. NA for this variable indicates that the pokémon does not have a secondary type.
- **Total Stats** (Total S.: The sum of a pokémon's six base stats (Health, Attack, Defense, Special Attack, Special Defense, and Speed).
- **Hit Points** (HP): A pokémon's health (the amount of damage a pokémon can withstand before fainting).
- **Attack**: Potency of melee attacks (e.g., "Tackle").
- **Defense** (Def): How well a pokémon can withstand or shield against melee moves (Attack).

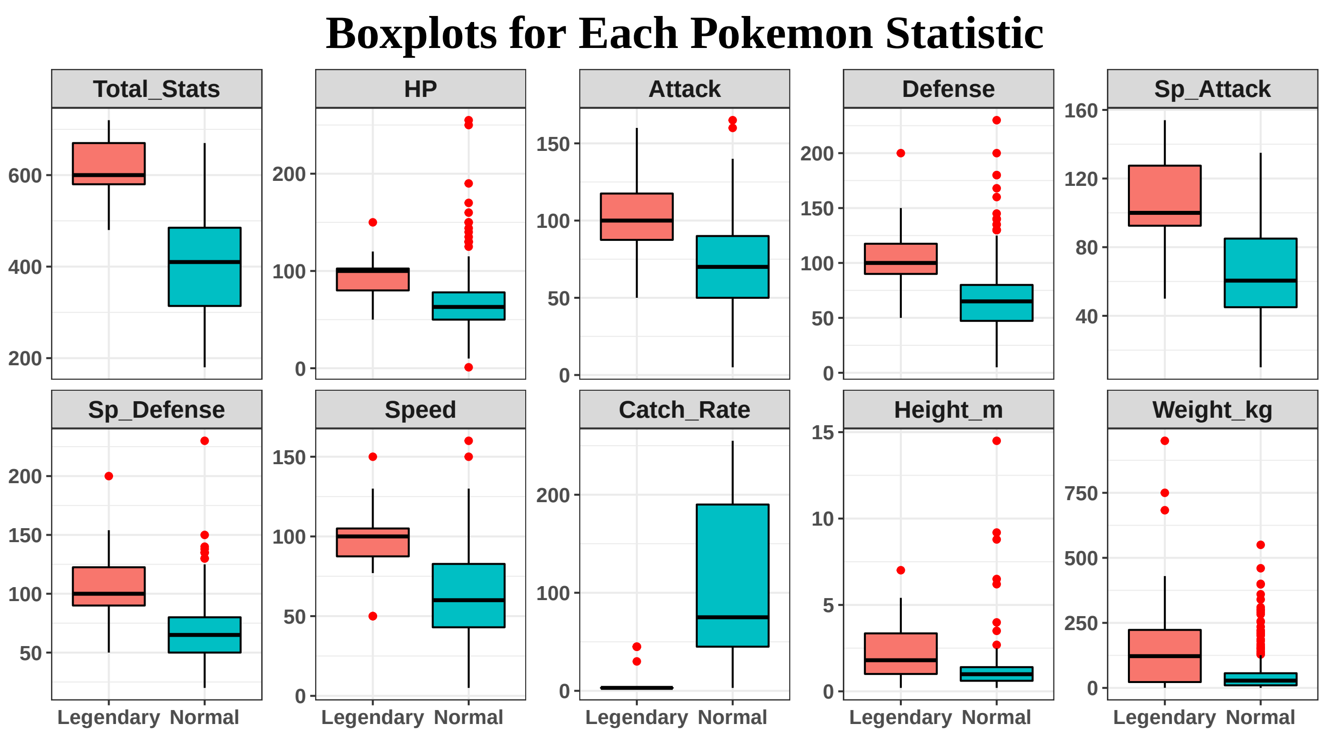
- **Special Attack** (Sp.Att): Compared to regular Attack, Special Attack moves tend to inflict damage via artificial or elemental methods (e.g., "Fire Blast").
- **Special Defense** (Sp.Def): How well a pokémon can withstand special or shield against Special Attacks.
- **Speed**: Quicker pokémon has the advantage of being able to attack first. When an opposing pokémon is slowed down, a pokémon with high Speed has a small chance of being able to attack twice in a row.
- **Defense Against _____** (D_[Type]): General term for 18 variables representing how well a pokémon defends against each of the 18 Types. Inside [Type], input the move type of the attack a pokémon is defending against (e.g. D_Bug).
- **Ability 1**: A pokémon first or only ability. An ability is a special effect activated whenever applicable. Each pokémon has at least 1 special ability.
- **Ability 2**: A pokémon's secondary ability. If the pokémon does not have a secondary ability, the entry will be NA.
- **Catch Rate** (Catch R.): A score representing the probability of catching a pokémon with a PokéBall when the pokémon has full health. Scores range from 3 to 255, calculated using a complex formula; as scores become larger, the probability of catching a pokémon increase, while low scores indicate that it is difficult to catch the associated pokémon.
- **Has Gender** (hasGender): Boolean variable indicating if the pokémon has a gender (TRUE) or not (FALSE).
- **Percent Male**: Proportion of male pokémon for a given species. The female proportion can be calculated using the following formula: 1 - Pr.Male. If a pokémon's gender is not known (hasGender = FALSE), then Pr.Male = NA.
- **Height in Meters** (Height).
- **Weight in Kilograms** (Weight).

Source: The Pokémon Company. *Pokémon video game glossary.*

Descriptions for all abilities are found in pokemondb.net/ability.

Summary statistics for most variables split by pokémon status (Legendary vs. Normal) are displayed in the following table and figure:

Stat	Min	Q1	Median	Mean	Q3	Max	Min	Q1	Median	Mean	Q3	Max
Total Stats	180	314	410	399.43	485	670	480	580	600	615.14	670	720
HP	1	50	63	65.88	78	255	50	80	100	94.54	102.5	150
Attack	5	50	70	71.42	90	165	50	87.5	100	102.29	117.5	160
Defense	5	47.2	65	67.71	80	230	50	90	100	103.46	117.5	200
Special Attack	10	45	60.5	64.97	85	135	50	92.5	100	109.54	127.5	154
Special Defense	20	50	65	66.21	80	230	50	90	100	109.4	122.5	200
Speed	5	43	60	63.24	82.75	160	50	87.5	100	95.91	105	150
Catch Rate	3	45	75	106.26	190	255	3	3	3	8.57	3	45
Height	0.2	0.6	0.99	1.11	1.4	14.5	0.2	1	1.8	2.19	3.36	7.01
Weight	0.1	9.9	28	49.11	56.38	550	0.3	22.5	122	189.71	223	950



Key Hypotheses:

It is hypothesized that two predictors will best distinguish Legendary from Normal Pokémon:

(1) Total Stats because Legendary Pokémon are known for being more powerful than Normal Pokémon.

(2) Catch Rate because Legendary Pokémon are very difficult to catch in the video games.

Methods

Data analysis and Machine Learning was conducted almost entirely using R and RStudio. Microsoft Excel was used quickly make minor adjustments to the data when necessary.

Pokémon App

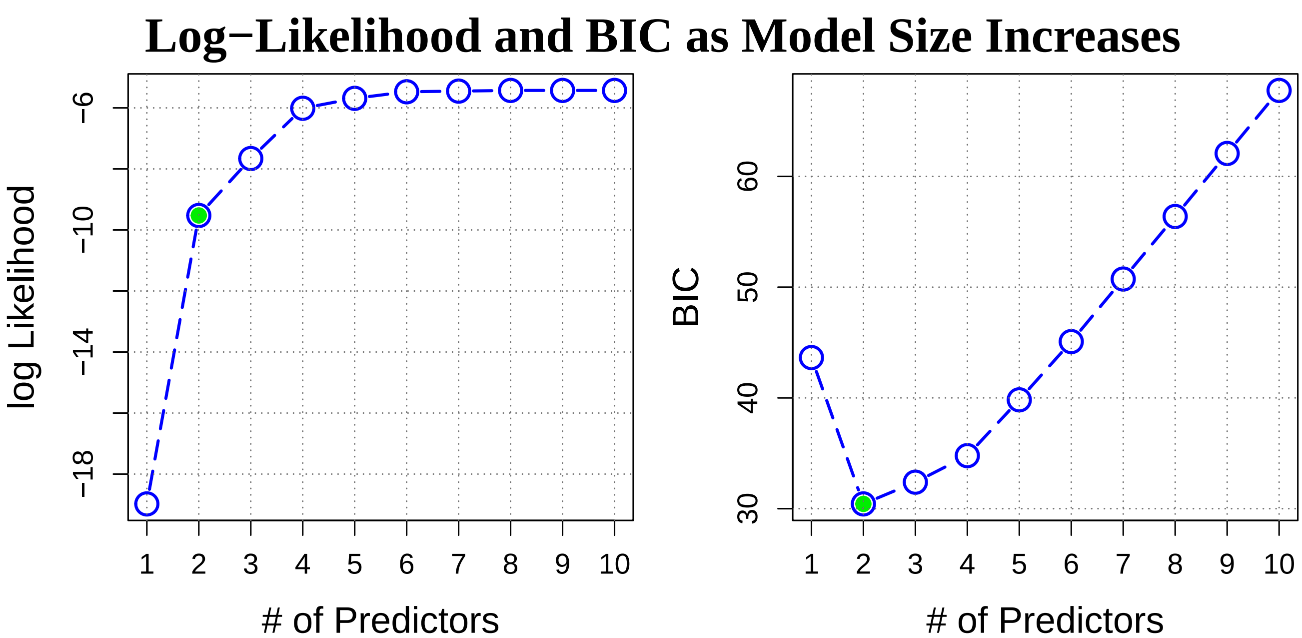
Shiny R was used to create an interactive Pokémon App for searching and comparing multiple pokémon. Compared to current ways of comparing pokémon, this app is user-friendly, and it automatically reacts to any modifications made by the user. The app is publicly published online at https://eduardomartinez9711.shinyapps.io/PokemonShinyApp/.

Classification Methods

Logistic regression and k-Nearest-Neighbours (k-N-N) were the classification methods used to predict whether a pokémon was Legendary or not. The data was randomly split with a train-test-split ration of 60% to 40%. It is crucial to train a regression using data that is large in terms of number of rows; conversely, having a large number of parameters/predictors is problematic as exponentially minimum amount of additional rows would be required (Hastie, Tibshirani, Friedman, 2017; Shar, 2017). Dimension reduction is critical in training a model in order to avoid overfitting the data.

As there were many predictors in the data use, the subset selection was employed via an exhaustive search penalized by the Bayesian Information Criterion (BIC). The goal of BIC and other information criteria is to maximize log-likelihood (Miller, 1984), but this can be achieved by adding more parameters. The BIC punishes models as they add more predictors. An optimal subset is found by minimizing BIC, which results in maximum posterior probability (McLeod &

Xu, 2010). By using an exhaustive search, the best possible model is guaranteed because every possible outcome is checked. This is a very time consuming since there were a total of 10 predicting variables considered. The accompanying R package utilized to efficiently find the best subset was “bestglm” (McLeod, Xu, & Lai, 2020). This process can be visualized as shown in the plot and table below:



BIC	loglik	p	Total S.	HP	Att	Def	Sp.Att	Sp.Def	Speed	Catch R.	Height	Weight
152	-76	0										
44	-19	1								✓		
30	-10	2	✓							✓		
32	-8	3	✓							✓		✓
35	-6	4	✓					✓		✓		✓
40	-6	5	✓	✓				✓		✓		✓
45	-5	6	✓					✓	✓	✓	✓	✓
51	-5	7	✓	✓				✓	✓	✓	✓	✓
56	-5	8	✓	✓			✓	✓	✓	✓	✓	✓
62	-5	9	✓	✓	✓	✓		✓	✓	✓	✓	✓
68	-5	10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

The BIC plot is minimized when two predictors are used ($p = 2$). In comparison, log-likelihood is continuously increases, but the difference in log-likelihood is relatively small for $p > 2$. The corresponding two parameters for the best subset are Total Stats and Catch Rate with an Intercept. All predictors in the model were statistically significant. The predictors included in the best subset model and their key statistics are shown in the table below:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-16.352	5.36	-3.05	0.0023
Total_Stats	0.035	0.01	3.41	0.0006
Catch_Rate	-0.150	0.05	-3.08	0.0021

When predicting using k nearest neighbors, the predictors were first standardized using the following formula $\frac{x_i - \mu}{\sigma}$. It is important to perform some type of scaling for k nearest neighbors as this algorithm classifies a points based on the majority class of its k nearest points measured in Euclidean Distance (Hastie, Tibshirani, Friedman, 2017; Shar, 2017). The same predictors considered in logistic regression were also used in k nearest neighbors. The k nearest neighbors that were tested where $k = 3$, $k = 5$, and $k = 7$. Since there were only 21 and 14 legendaries in the train and Test Set, respectively, accuracy began to decrease when more than 9 nearest neighbors were considered.

Results

Training Set Confusion Matrix

Prediction	True Class	n
Normal	Normal	274
Legendary	Normal	1
Normal	Legendary	2
Legendary	Legendary	19

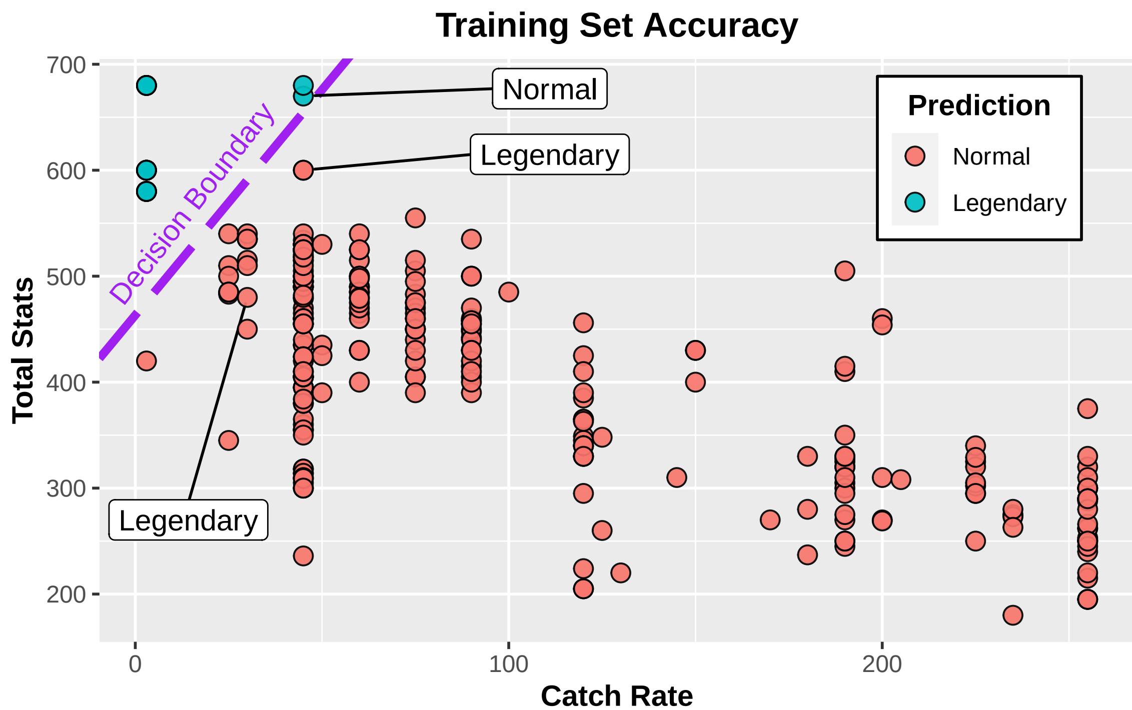
Model Accuracy on Train Set: 98.99%

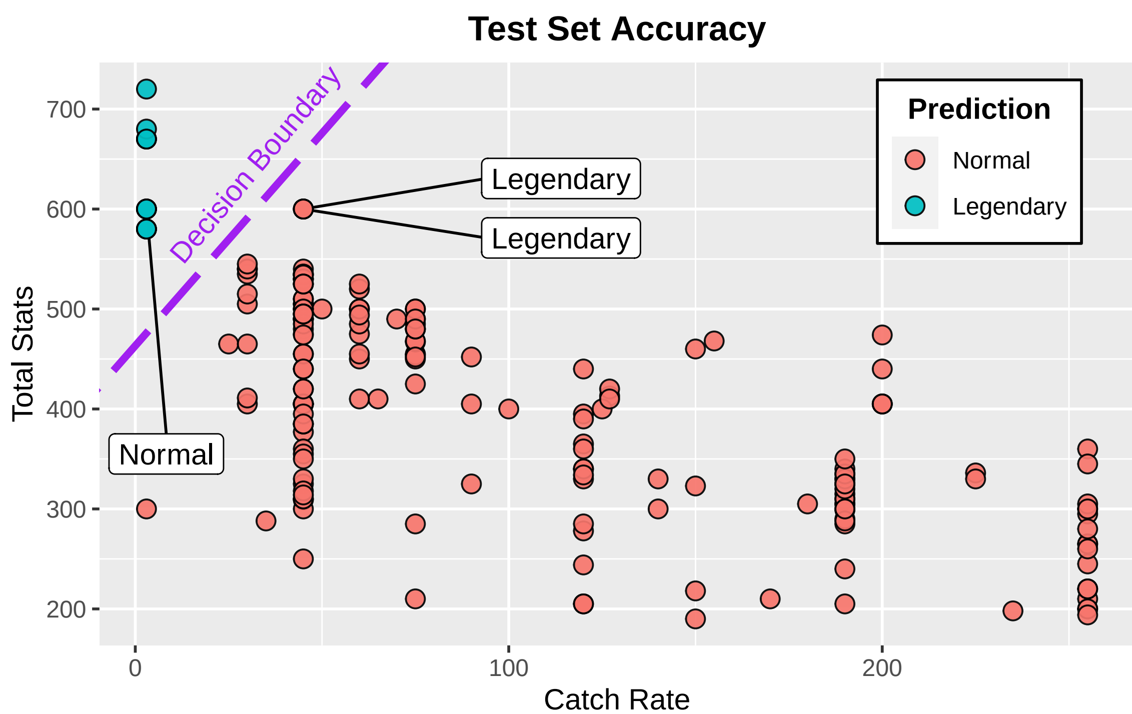
Test Set Confusion Matrix

Prediction	True Class	n
Normal	Normal	182
Legendary	Normal	1
Normal	Legendary	2
Legendary	Legendary	12

Model Accuracy on Test Set: 98.48%

Note: In the next two plots, the purple line at the top left section of each plot represents the decision boundary. All points below and to the right of the line were classified as normal, and points above and to the left were classified as Legendary. If a point is misclassified, its true class is labeled. Three pokemon were misclassified in both the Training and Test Set. In the Test Set, a point seems to labeled as legendary twice, but there are two points with same coordinates.



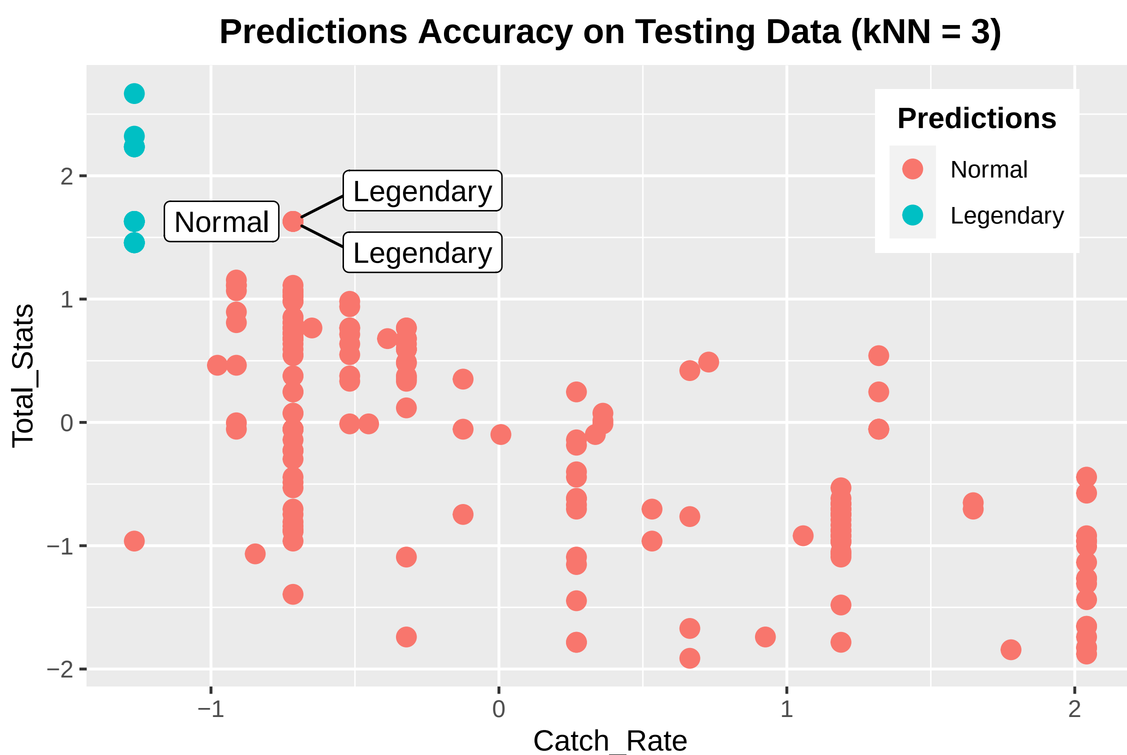


k-Nearest-Neighbords

k Nearest Neighbors: Confusion Matrix

k_3 Class	True Class	n	k_5 Class	True Class	n	k_7 Class	True Class	n
Normal	Normal	182	Normal	Normal	179	Normal	Normal	179
Legendary	Normal	1	Legendary	Normal	4	Legendary	Normal	4
Normal	Legendary	2	Normal	Legendary	0	Normal	Legendary	0
Legendary	Legendary	12	Legendary	Legendary	14	Legendary	Legendary	14

Accuracy on Test Set: $k_3 = 98.48\%$, $k_5 = 97.97\%$, $k_7 = 97.97\%$



The optimal k-Nearest-Neighbors considered when applying this algorithm was the three nearest neighbors ($k = 3$) as it performed the best on the Test Set. Interestingly, when $k = 3$, the algorithm incorrectly classified the same three points as those incorrectly classified by the logistic regression model.

Discussion and Conclusion

The Pokémon App can be very useful for anyone that enjoys playing the older pokémon games. Experienced players will be able to quickly and efficiently compare various pokémon simultaneously. Moreover, the app can be used to determine the best lineup of 6 pokémon possible. Simply, drag the “Min Total Stats” slider to the right. Note, very few pokémon are strong all around. Thus, to see which pokémon have the highest stats overall, users should first lower all other base stats. The app can also be very useful for new players. For instance, if players do not know the type and weaknesses of a wild pokémon or the pokémon used by a gym leader, they can quickly find out by simply typing the pokémon’s name on the search option in the app. The app can be accessed on a mobile device, but it functions best on a computer.

When predicting legendary pokémon, the predictive models developed were very successful. Models consistently achieved over 95% accuracy. Hence, it is not clear if either logistic regression or k-N-N was better. However, a decision boundary can easily be defined using logistic regression. By contrast, there is no decision boundary in k-N-N.

A possible extension of this study can be applied by people that create new pokémon. The findings in this study can be used as a guide on what a legendary pokémon’s Total Stats and Catch Rate should be.

Limitations & Suggestions

The goal in this study was to accurately predict whether a pokémon is legendary or not. Future, studies can attempt to predict different variables. For instance, can weight and height be used to predict other base stats. I anticipate that larger pokémon will have higher defense but slower speed (vise-versa for smaller pokémon).

Further, only two classification methods were tested in this study. Future studies may want to try different methods like a Neural Network, but such studies will not be as meaningful because the accuracy level is already really high with the models utilized. Instead, the other generations of pokémon should be explored to see if they are consistent with the first four generations of pokémon.

The characteristics in the app do not include all qualities about each pokémon. Studies that seek to expand on the app can be useful for players that want to know even more about pokémon.

References

- Barradas, A. (2016) *Pokémon with stats.* [Data set]. Retrieved February 28th, 2020, from www.kaggle.com/abcsds/pokémon.
- Chen, S. D. (2014). A crude analysis of twitch plays pokémon. *ArXiv, 1408.4925*, 1-6.
- De Roure, D., Hendler, J. A., James, D., Nurmikko-Fuller, T., Van Kleek, M., & Willcox, P. (2019). Towards a cyberphysical web science: a social machine's perspective on Pokémon Go! In Proceedings of the 10th ACM Conference on Web Science (WebSci '19). *Association for Computing Machinery*, 65–69. DOI: 10.1145/3292522.3326043.
- Hastie, T., Tibshirani, R., & Friedman, J. H. (2017). *The elements of statistical learning: Data mining, inference, and prediction.* (2nd ed.) New York: Springer.
- Larcher, C. (2018). *Pokémon weakness - Generations 1 to 6.* [Data set]. Retrieved March 29th, 2020, from www.kaggle.com/celiolarcher/pokémonweaknessgen1.
- Liapis, A. (2018). Recomposing the pokémon color palette. In: Sim, K., & Kaufmann, P. (Eds.) Applications of Evolutionary Computation. EvoApplications 2018. *Lecture Notes in Computer Science, 10784*, 308-324. DOI: 10.1007/978-3-319-77538-8_22.
- Mcleod, A.I., Xu, C., & Lai, Y. (2020). Package 'bestglm'. p. 1-48.
- Mcleod, A.I., & Xu, C. (2010). bestglm: Best subset GLM. p. 1–39.
- Miller, A.J. (1984). Selection of subsets of regression models. *J. R. Statist. Soc. A. 147*(3), 389-425.
- Ojeda, A. (2019). *1st to 7th gen pokémon* (Version 3) [Data set]. Retrieved February 28th, 2020, from www.kaggle.com/dizzypanda/1st-to-7th-gen-pokémon.
- Panimate, C., Xiong S., Iida H., & Kondo T. (2015) Evolutionary changes of pokémon game: a case study with focus on catching pokémon. In: Chorianopoulos K., Divitini M., Baalsrud Hauge J., Jaccheri L., & Malaka R. (eds) Entertainment Computing — ICEC 2015. *Lecture Notes in Computer Science, 9353*, 182-194. DOI: 10.1007/978-3-319-24589-8_14.
- *Pokémon data.* (n.d.). Pokémon Database. https://pokemondb.net/.
- *Pokémon games.* (n.d.). Pokémon Database. https://pokemondb.net/.

- *Pokémon types & type chart.* (n.d.). Pokémon Database. https://pokemondb.net/type.
- Romero, M.T. (2020). *Complete Pokémon Dataset.* (Version 5). [Data set]. Retrieved February 28th, 2020, www.kaggle.com/mariotormo/complete-pokémon-dataset-select=pokedex_Updat
- Shar, T. (Dec. 6, 2017). *About train, validation and test sets in machine learning.* Towards data science. URL.
- The Pokémon Company. (n.d.). *Pokémon video game glossary.* Pokémon. www.pokémon.com/us/play-p
- Wang, M. (2017). FollowMe if you can: a study of mobile crowd sensing with Pokémon Go. In Proceedings of the Australasian Computer Science Week Multiconference (ACSW '17). *Association for Computing Machinery*, (Article 39), 1-9. DOI: 10.1145/3014812.3014853.
- Zhang, Z. (2016). Variable selection with stepwise and best subset approaches. *Ann Transl Med., 4*(7), 1-6.
- Zorrilla, A.L. (2017). *Pokémon for data mining and machine learning.* [Data set]. Retrieved February 28th, 2020, from www.kaggle.com/alopez247/pokémon.