Class: CS5137 – Machine Learning
From: Tyler Rose and Seth Dippold
Date: 12/08/2017
Subject: Gotta Type 'em All – An Analysis of Classification Algorithms

INTRODUCTION

This project faced the challenge of classifying a given Pokémon based on its different statistics: Hit Points, Attack, Defense, Special Attack, Special Defense, and Speed. Pokémon are based off the original Pocket Monsters in Japan; for more information, go to https://en.wikipedia.org/wiki/Pokémon. Given that new Pokémon are released roughly every two years, this would allow anyone to determine the type of the Pokémon, the available types are Bug, Dark, Dragon, Electric, Fairy, Fighting, Fire, Flying, Ghost, Grass, Ground, Ice, Normal, Poison, Psychic, Rock, Steel, and Water. This problem is challenging because of the small amount of statistics in comparison with the large number of types. Through experience with the many games of Pokémon, many players consider certain types of Pokémon to be weaker. This project aims to make these considerations fact through classification by Decision Tree, SVM, Naïve Bayes, and Nearest Neighbors algorithms. The python library, Scikit-Learn [2], implementations of these algorithms will be used for our classification.

BASIC APPROACH

This project will be approached by splitting the dataset of necessary information into a training and validation set. The training set will contain 4/5 of the data while the validation set will contain the remaining 1/5 of the data. Each of the classification algorithms will then be used with these sets to find the accuracy based on the training data and the accuracy based on the validation data. This ran 100 times for each algorithm to find the average and maximum accuracies.

The SVM approach was split up into two of the different kernels that Scikit-Learn offers: Linear and Radial Basis Function (RBF). Techniques using RBF utilize some method to determine a group of centers. Usually, clustering is used to select the first subset of centers. In Gaussian RBF, each of these centers is used as the center for a local Gaussian function. [1] RBF is a non-linear classification kernel so its decision boundaries will most likely fit better with our data.

EXPERIMENTAL SETUP

The dataset used was the Pokémon with stats dataset from Kaggle.com found at https://www.kaggle.com/abcsds/pokemon. Before the data was used all the Mega evolutions in the dataset were omitted as they could be considered outliers. From here the code did all other processing to separate the data into the statistics and the types used for the machine learning algorithms.

EXPERIMENTAL RESULTS

The results from this experiment were quite definitive that it is not possible to classify Pokémon into their types by their given statistics. Each of the five algorithms used showed a validation accuracy of under 30% on average with Nearest Neighbors providing the highest accuracy as can be seen in Table 1. This result was surprising as we were expecting the decision tree to have the highest result based off the findings in An Empirical Comparison of Supervised Learning Algorithms [3]. Given that a decision tree can create minute discrepancies based on each statistic as shown in Figures 1 and 2, it was originally thought to be able to fit the data well. Although the decision tree had very good results on the training data, it showed that this was highly over fit when it did not perform nearly as well on the validation set. The other surprising part to this was how well the RBF SVM fit the data. After such a poor fit with the Linear SVM, an accuracy increase of almost half was shocking.

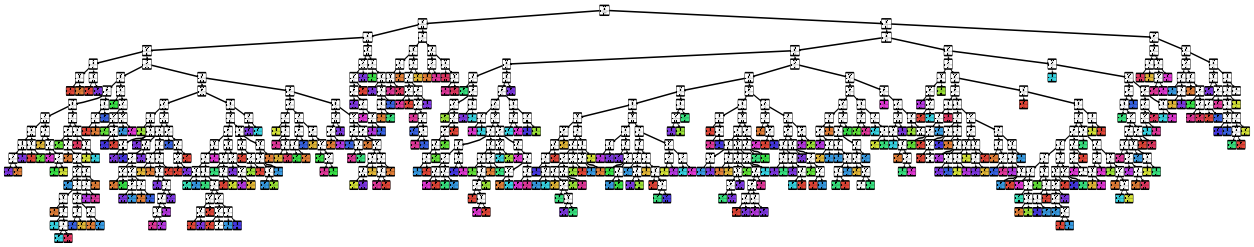

Figure 1: Decision Tree on Statistics and Typing

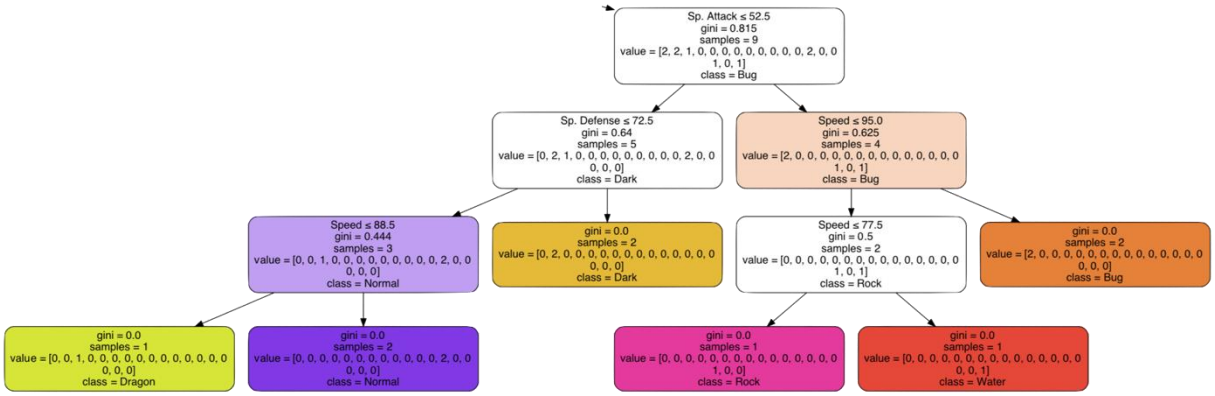

Figure 2: Part of the Decision tree in Figure 1

Algorithm Name	Average Training Accuracy	Average Validation Accuracy	Maximum Validation Accuracy
Decision Tree	98.80%	16.63%	23.68%
Linear SVM	8.82%	8.55%	24.34%
RBF SVM	98.84%	15.01%	21.71%
Nearest Neighbors	44.61%	21.55%	28.95%
Naïve Bayes	23.10%	18.01%	25.66%

Table 1: Accuracies of Machine Learning Algorithms on fitting Pokémon statistics to types

CONCLUSION

As hypothesized from the beginning, our accuracies reported in **Table 1** show that the "Pokemon with stats" dataset is not classifiable. Further work could be done by adding each Pokémon's abilities to the dataset and use that to classify type. Also, it could be possible to classify or further prove the unclassifiable nature of this dataset by using other algorithms such as Neural Networks.

Works Cited

[1] S. R. Gunn, "Support Vector Machines for Classification and Regression," University of Southampton, Southampton, 1998.

[2] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. Vanderplas, A. Joly, B. Holt and G. Varoquaux, API design for machine learning software: experiences from the scikit-learn project, Ithaca: Cornell University, 2013.

[3] R. Caruana and A. Niculescu-Mizil, "An Empirical Comparison of Supervised Learning Algorithms," Cornell University, Ithaca, 2006.