Detecting Pokémon Types Using a Variety of Data Mining and Machine Learning Techniques

Ryan Hildebrant[1], Garrison Price[1], Jonathan D'Anneo[1]
[1]Department of Computer Science, San Diego State University, San Diego, CA 92182, USA
rhildebrant@sdsu.edu, gprice2922@sdsu.edu, jdanneo1078@sdsu.edu

***Abstract*—Pokémon is a role-playing game based around building a small team of monsters to battle other monsters in a quest to become the best trainer [1]. Over the years, Pokémon has expanded to many other mediums including the following: trading card game, clothing line, television series, and a theme park. However, the main idea of collecting, fighting, and trading the monsters has remained as the cornerstone to the brand. In this paper, we analyze a set of 898 unique Pokémon to try to predict their type based on various attributes. These attributes include their combat stats, weaknesses to other monsters, height, and weight. To achieve this, we explore a variety of popular classification and clustering machine learning techniques. Specifically we compare the results of a neural network [10], support vector machine [9], KNN-model [4] with a set of K values, Naive Bayes Methods [11], and Decision Trees [12]. For all of these models, we provide clear performance metrics and show that the Poly Kernel SVM [9] method is best. Additionally, we use a data mining technique known as association rule mining [5] to show patterns in the types of weaknesses.**

***Index Terms*—Data Mining, Machine Learning, Pokémon, Prediction**

I. Introduction

In this paper, we focus on classifying the type of a Pokémon, regardless of their legendary or mythical status. We do this for two reasons: (1) previous work found in [6] uses a random forest model to identify only legendary Pokémon, (2) we believe that identifying the type of each Pokémon is the best first step for future work. As Pokémon is an active franchise, new species are continuously being added to the existing Pokémon roster. The newest iterations, *Pokémon Sword* and *Pokémon Shield*, introduced 30 new Pokémon to the franchise. Finding a model that accurately identifies a Pokémon's type allows for predictions on what future Pokémon types might be added to the game, as well as allow for detection of anomalies within each Pokémon type.

II. Task Description

This paper has two major goals. First, we would like to use the five popular machine learning classification methods to try to predict the types of Pokémon. To do our analysis, we split the data set into two parts for training and testing each model. Each model can require different hyper-parameters, or a slightly different form of the data. For example, in the SVM models, we differentiate the hyper-parameters by the type of kernel used to train the model. Although, for every model, we specify that the $type_1$ feature, which specifies a Pokémon's primary type, is the class variable we are trying to identify. There are 18 different types in which a Pokémon can be classified; these are as follows: *'Grass' 'Fire' 'Water' 'Bug' 'Normal' 'Poison' 'Electric' 'Ground' 'Fairy' 'Fighting' 'Psychic' 'Rock' 'Ghost' 'Ice' 'Dragon' 'Dark' 'Steel' and 'Flying'*. Every Pokémon in this data set has a primary type, with some also having a secondary type. Additionally, they have specified values for *attack, special attack, defense, and special defense*, as well as, a distinct base ability set. In section III, we will discuss how challenging this amount of class variables can be for the simpler classification models. Once we have used the models to classify all of the Pokémon in the test set, we will provide detailed results of the accuracy of each model. From there, we will expand the results and provide an in-depth analysis using a confusion matrix and a scatter matrix.

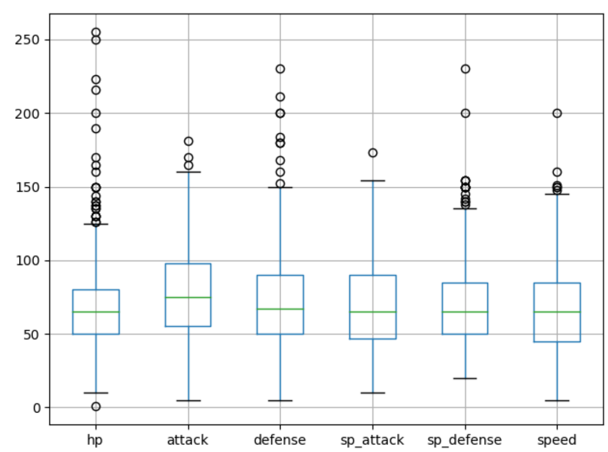

Fig. 1. Box Plot of Pokémon Stats

The second goal is to use a data mining technique to compare the weaknesses of monsters and see if there are patterns with high support, confidence, and lift. To achieve this, we classify a bound for weaknesses and classify all of the monsters within this range. This allows us to see if there are any frequently occurring types of weaknesses and helps improve our overall analysis. For example, in *Table V*, we show that every Pokémon that is weak against Ice and Fire will also be weak against Flying. This is a useful starting point for determining the best Pokémon team composition, which will discuss in *Section V*.

III. Major Challenges and Solutions

We encountered three major challenges when performing our experiments on this data set. The first challenge that we encountered was that there are some Pokémon that have ”variants.” These variants are categorized as the same Pokémon with the same Pokédex ID, but they might have vastly different stats and/or even types. We solved this by removing the duplicate entries for each Pokémon from the data set. In general we tried to save the base variants for the duplicate entries, but when there were no base variants, we opted to keep the first entry.

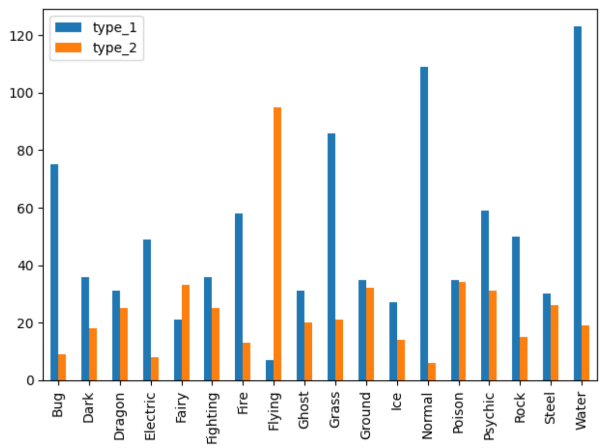

Fig. 2. Histogram of Pokémon Types

The second challenge that we encountered was that there are many combination of types that a Pokémon can belong to. This caused some of the simpler classification methods to struggle to discern types effectively, given that we are working with a relatively small data set. We solved this by maximizing the amount of data used to train the models while still maintaining a large enough sample for testing. We opted to forgo a validation set and stick with the 80 percent/20 percent split for the training and testing sets.

The third challenge that we encountered was that there are many Pokémon with two types, but not all Pokémon have two types. If we were to attempt to classify Pokémon for each possible pair of types, the problem from the second challenge above becomes even worse. There are 160 unique pairs of types in the data set of only 898 Pokémon. We solved this by simply dropping the second type column and only using the first type as our labels. This reduced the number of labels down to just 18, which was still high for some of the classifiers but much more effective. Ideally, both types would be used as labels and we see this as a good opportunity for future work.

IV. Experiments

A. Dataset Description

The data set we use is from the popular online data science website, Kaggle [2]. There were a number of data sets with Pokémon statistics, but we elected to choose a particular data set that is frequently updated by Mario Tormo Romero [3]. This data set includes 1044 Pokémon and is the cleanest data set we could locate, as well as contained the most information about each Pokémon. Additionally, we could validate the usefulness of the data set, as many projects were listed that involved it. Along with the data set we downloaded, there are two additional data sets that are a bit smaller with slightly less Pokémon recorded. We noticed a significant overlap in all of the data sets, so we chose the largest of the three. Once we had the data, we began preprocessing the data to remove unwanted features and rows. Some columns include names for the monsters in other languages, while others included the base ability sets, birth rates, capture rates, etc. While this information was interesting, we decided it was not useful for classification purposes. Given that the data set already had high dimensinoality, we removed these columns in order to reduce the data set down to about 30 columns.

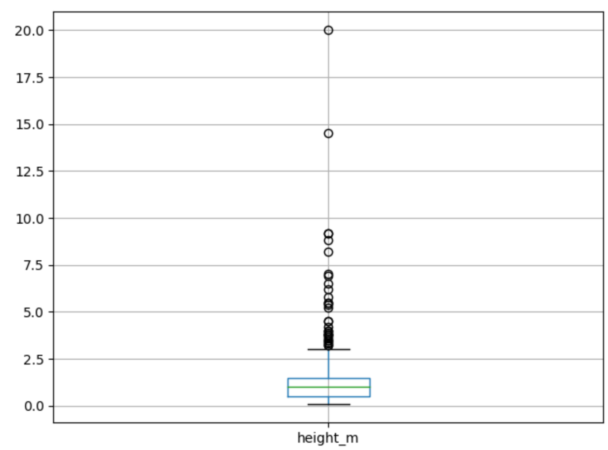

Fig. 3. Box Plot of Pokémon Heights

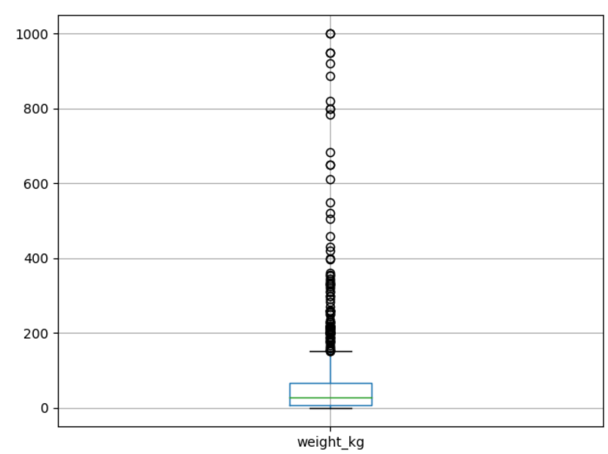

Fig. 4. Box Plot of Pokémon Weights

While still preprocessing the data, we also noticed that many of the Pokémon were repeated multiple times in the data set; this is because some unique monsters can have many forms

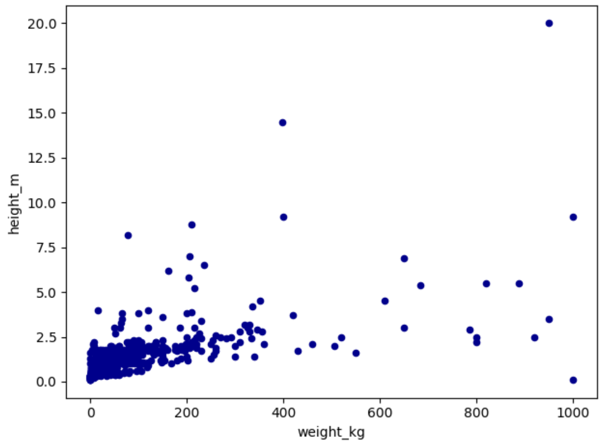

Fig. 5. Scatter Plot of Pokémon Heights and Weights

and are classified with different names based on the form they are currently in. After carefully reviewing these various forms, we noticed that the majority of their feature values did not change. Because of this, we choose to drop the variations and only keep a Pokémon's base form. Additionally, some Pokémon have different stats based on their gender. After reviewing the statistic values, we elected to keep the female variants of those Pokémon.

B. Evaluation Metrics

In order to get a more accurate representation of the performance of each model, ten separate evaluations were performed, each with randomization in the splits of training and testing sets.

For evaluating the performance of the various models, we consider the accuracy of each model with respect to [8]. For each model, we calculate the accuracy based on a confusion matrix. Additionally, we provide the confusion matrices for the best performing models to show the complexity of the classification class. In addition to this, we provide an overview of certain models given different hyper-parameters where appropriate, i.e. different k-values for the KNN models or the classifier type for the Naive Bayes.

Other metrics that are used for evaluation include precision, recall, F1-score, and support. For these metrics, we provide an overview table of the minimum, maximum, and average value scores for each classification model. We choose to provide a less comprehensive overview since this is a classification problem.

Lastly, we provide distance matrices based on the different measurement mediums used within our neural networks. The distance measurements we considered are the Euclidean Distance, the Mahalanobis Distance, and the Minkowski distance with an r value of 1. We choose these distances because we felt that they provided the greatest visual scope of patterns within the data.

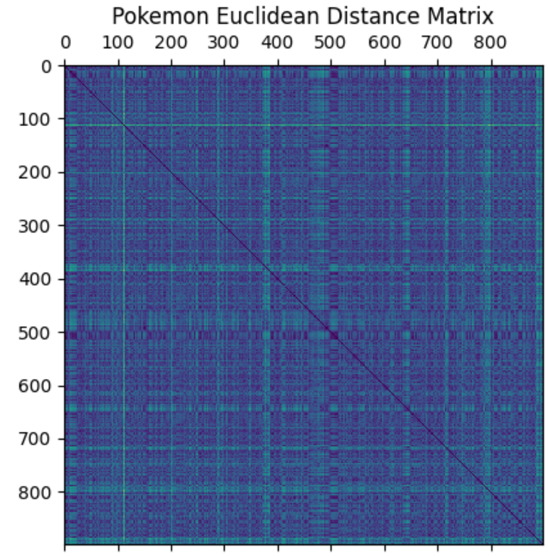

Fig. 6. Pokémon Euclidean Distance Matrix

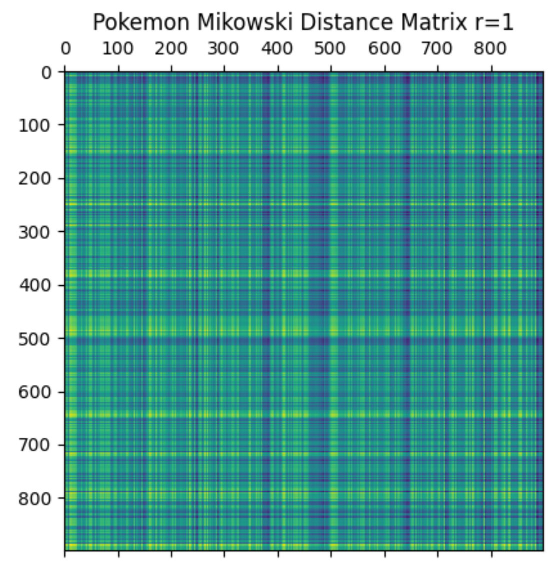

Fig. 7. Pokémon Minkowski Distance Matrix (r=1)

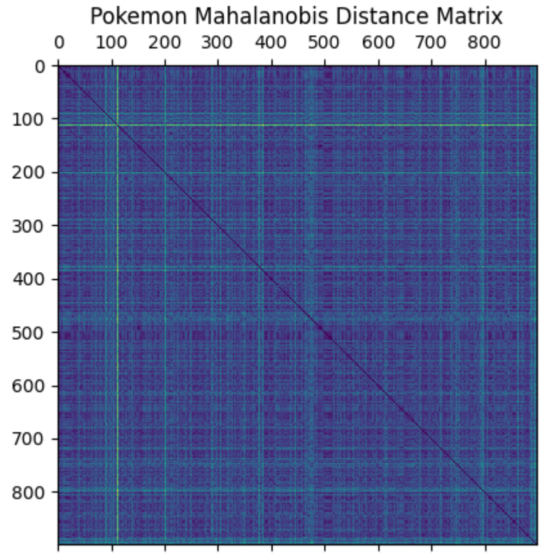

Fig. 8. Pokémon Mahalanobis Distance Matrix

C. Major Results

In this sub-section, we provide a graphical overview of the evaluation metrics outlined in the previous section across all of the classification models.

We start by providing confusion matrices for the best-performing SVM, Decision Tree, Naive Bayes, and Neural Network models.

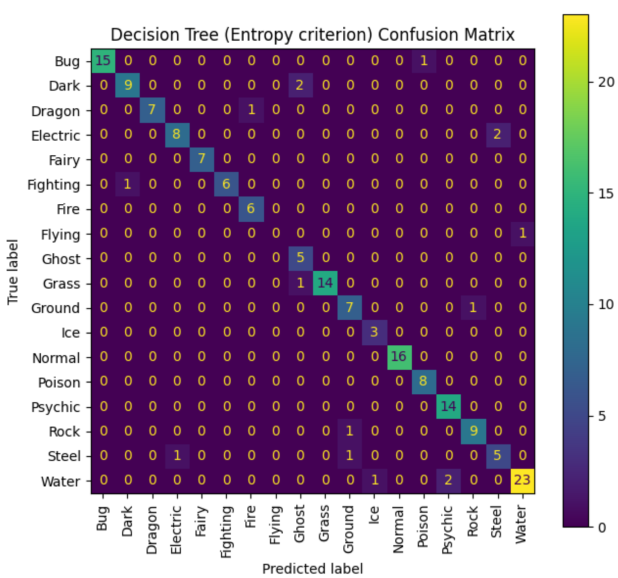

Fig. 9. Best-performing decision tree classifier's confusion matrix

As you can see, almost all of the models were able to correctly identify bug, grass, normal, psychic, and water types. In total, the Support Vector Machine Poly-Kernel method only makes 10 incorrect classifications. By looking at the

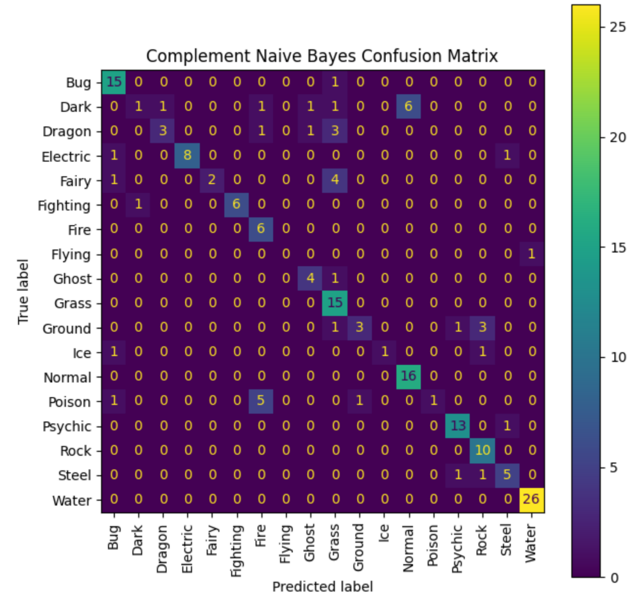

Fig. 10. Best-performing naive bayes classifier's confusion matrix

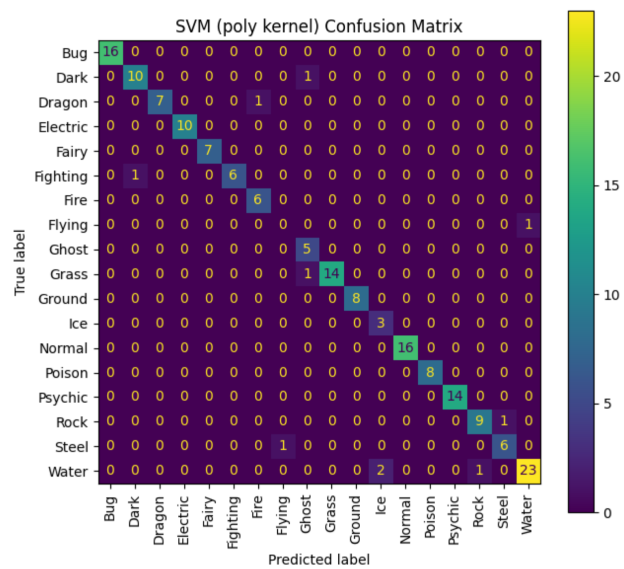

Fig. 11. Best-performing SVM classifier's confusion matrix

confusion matrices, there is clearly an uneven distribution in types of Pokémon. Even with the randomization of the splits in training and testing sets, there is consistently a large number of Pokémon noted above, while there was rarely more than a few of type flying, ghost, and ice. Given this, it should be noted that while accuracy is one of the metrics we use for evaluation of the models, it may not necessarily be the best depiction of a model's performance, as it is important for the models to accurately predict the lesser-populated Pokémon types.

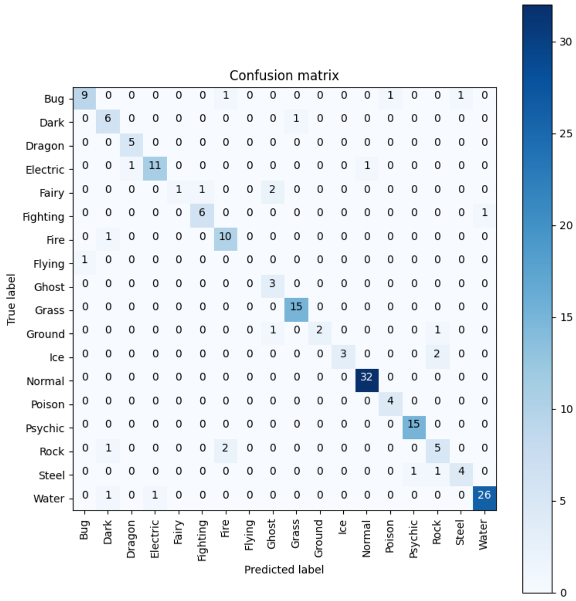

Fig. 12. Best-performing NN classifier's confusion matrix

For our graphs, we choose to show the accuracy of the different models because our focus is primarily on classifying the type. For these figures, we take the average across all 10 experiments, each time testing a number of hyperparameters or variants of a particular model.

In some cases, other models preformed very well for a particular shuffle, but this was not a clear representation of the model's performance. This was rarely the case though, as the difference between the best and worst performance for each model was typically within 5-6% percent. Some models, particularly the Decision Tree and Naive Bayes models, had a difference of 10+% between their best and worst performances. A graph depicting the best and worst performance of each model is also included. We do not consider the run-time performance as a factor in this paper because most of the implementations are designed by using optimized libraries and the total running time is very short.

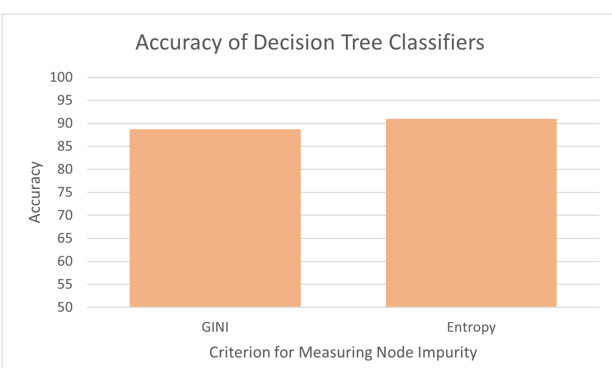

Fig. 13. Decision tree model accuracies - best results from 10 experiments

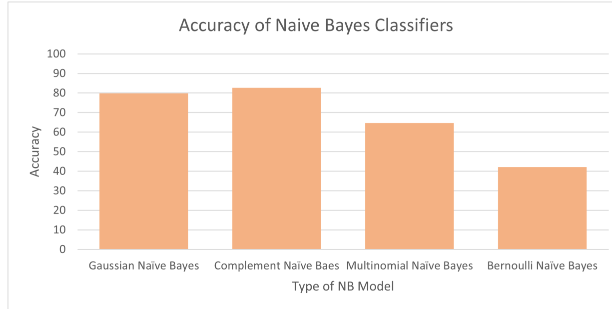

Fig. 14. Naive Bayes model accuracies - best results from 10 experiments

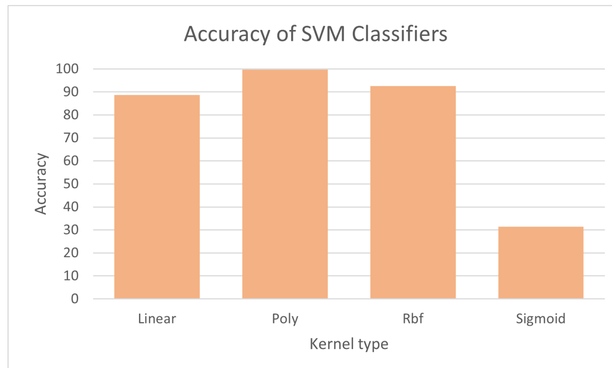

Fig. 15. SVM Model accuracies - best results from 10 experiments

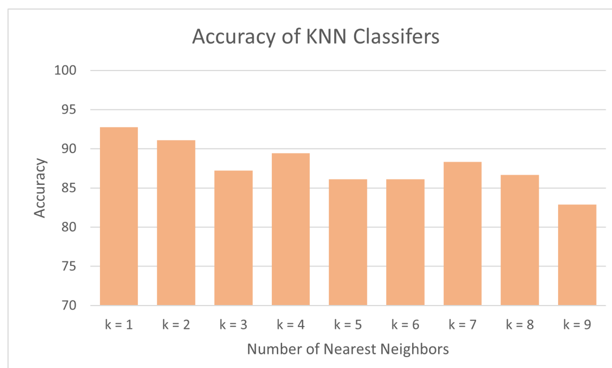

Fig. 16. KNN Model - Best Results From 10 Experiments

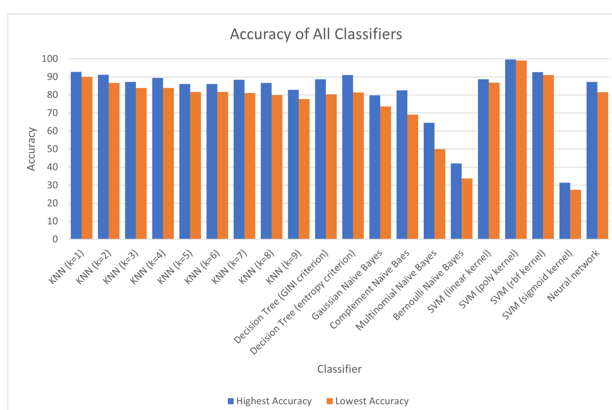

Fig. 17. Highest and lowest accuracies of all classifiers - from 10 experiments

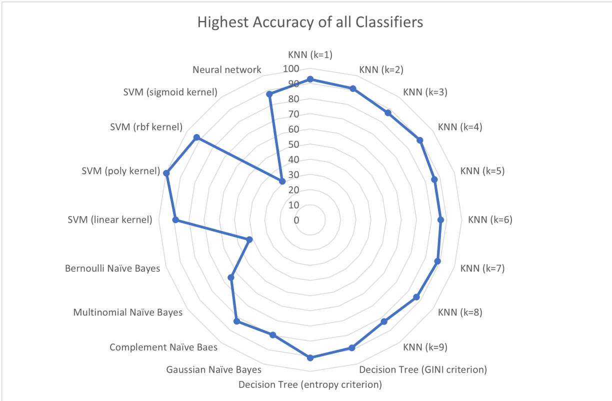

Fig. 18. Highest accuracies of all classifiers - from 10 experiments

As noted above, while accuracy is important, we also provide the minimum, maximum, and average precision, recall, and F1-score for the best performing variants of each model to get an even broader interpretation of each model's performance. Here we choose to also include the minimum and maximum values to provide a more robust view of each classifier. One additional note here is that we did not consider any models beyond these because the results were so consistently strong.

TABLE I
PRECISION FOR BEST-PERFORMING CLASSIFIERS OF EACH TYPE

Model	Min	Max	Average
KNN (K=1)	0.000	1.000	0.858
Decision Tree (Entropy)	0.500	1.000	0.875
Complement Naive Bayes	0.200	1.000	0.781
SVM (poly kernel)	0.714	1.000	0.939
Neural Network	0.000	1.000	0.849

TABLE II
RECALL FOR BEST-PERFORMING CLASSIFIERS OF EACH TYPE

Model	Min	Max	Average
KNN (K=1)	0.000	1.000	0.871
Decision Tree (Entropy)	0.571	1.000	0.880
Complement Naive Bayes	0.500	1.000	0.838
SVM (poly kernel)	0.714	1.000	0.943
Neural Network	0.000	1.000	0.831

TABLE III
F1 SCORES FOR BEST-PERFORMING CLASSIFIERS OF EACH TYPE

Model	Min	Max	Average
KNN (K=1)	0.000	1.000	0.855
Decision Tree (Entropy)	0.615	1.000	0.867
Complement Naive Bayes	0.308	1.000	0.782
SVM (poly kernel)	0.769	1.000	0.938
Neural Network	0.000	1.000	0.823

Lastly, the following tables provide an overview of the Association Rule Mining results with support, confidence and lift values.

TABLE IV
SUPPORT VALUES GIVEN A MINIMUM SUPPORT THRESHOLD OF 0.11 FOR WEAKNESSES AGAINST CERTAIN TYPES

Type Weak Against	Support
Weak Against Ice	0.285078
Weak Against Rock	0.269488
Weak Against Fire	0.259465
Weak Against Fighting	0.248330
Weak Against Ground	0.242762
Weak Against Flying	0.241648
Weak Against Electric	0.223831
Weak Against Grass	0.197105
Weak Against Bug	0.173719
Weak Against Water	0.172606
Weak Against Fire	0.171492
Weak Against Fairy	0.151448
Weak Against Ghost	0.145880
Weak Against Dark	0.136971
Weak Against Poison	0.136971
Weak Against Dark and Ghost	0.132517
Weak Against Steel	0.130290
Weak Against Psychic	0.129176
Weak Against Flying and Ice	0.129176
Weak Against Ice and Fire	0.120267
Weak Against Flying, Ice, and Fire	0.120267
Weak Against Grass and Electric	0.115813
Weak Against Ground and Water	0.110245

TABLE V
CONFIDENCE AND LIFT VALUES GIVEN A MINIMUM CONFIDENCE THRESHOLD OF 0.4 FOR WEAKNESSES AGAINST CERTAIN TYPES

Antecedent	Consequent	Confidence	Lift
Against Dark	Against Ghost	0.967480	6.632036
Against Ghost	Against Dark	0.908397	6.632036
Against Ice and Fire	Against Flying	1.000000	4.138249
Against Flying and Ice	Against Fire	0.947368	3.323191
Against Fire	Against Flying	0.660944	2.735152
Against Flying	Against Fire	0.660944	2.735152
Against Water	Against Ground	0.638710	2.631015
Against Ground	Against Water	0.454128	2.631015
Against Electric	Against Grass	0.517413	2.625067
Against Grass	Against Electric	0.587571	2.625067
Against Flying and Fire	Against Ice	0.701299	2.460024
Against Ice	Against Flying	0.421875	1.745824
Against Flying	Against Ice	0.525346	1.745824
Against Ice	Against Fire	0.421875	1.625939
Against Fire	Against Ice	0.463519	1.625939

D. Analysis

Based on the results outlined in the previous sub-section, we can clearly see that the SVM poly-kernel method is the best at predicting Pokémon types, as it has both the highest accuracy and F1-score. In certain metrics such as F1-score, the SVM model clearly outperforms the other models with a 7.1% better average score. Many of the models achieve a strong accuracy overall, however, there are a few that performed very poorly, particularly the Bernoulli Naive Bayes model and an SVM model with a sigmoid kernel. Another thing to note is that the K values for the KNN models did not change the accuracy very much. Overall the variance from 1 to 9 stayed within a range of 10%. Interestingly, as a general trend, the lower the k value, the better the model performed. This is likely due to the fact that the data set we have is small (less than 1000 entries) with an uneven distribution between classes, so higher K values can lead to misclassification of Pokémon types with a smaller population.

When we began our experiments, we hypothesized that the neural network would produce the best results. Yet, after much fine tuning, we could get the accuracy results of the neural network to achieve a score above 90%. The best performing neural network model was a sequential model with three dense layers with the two intermediate activation functions being rectified linear unit (ReLU) functions and the output activation function being a softmax function. Dropout and batch normalization techniques were also tested but did not increase the accuracy of the model. Intuitively, the dimensionality reduction approach to Support Vector Machines worked very well for this classification task. This is because SVM models attempt to maximum the boundaries between between grouped data points. Since we only considered attributes that did not have high unpredictability, the model was robust enough to pick up the relationships between the features and each type.

In addition to the classification model results, we provide two tables for the Association Rule Mining results showing the support, confidence and lift. From these patterns, we learned some interesting trends about Pokémon. More specifically, we learned that around 1/4 of Pokémon are weak to Ice, Rock, Fire, Fighting, Ground, or Flying. Being that the maximum team size is 6, it seems like it would be very useful to have a Pokemon, or move, of each type within any given roster, as this would provide for a type-advantage against the largest number of Pokémon/trainers one may encounter. Additionally, we noticed some groupings with very high confidence, such as weaknesses of Ice and Fire, and Flying and Ice. There is also a direct correlation between the confidence rating and the lift. As you can see, all of these values have a very positive lift rating. Expanding these results would be very helpful for picking the secondary types of Pokémon on a particular team.

V. Conclusion and Future Works

In this paper, we set out to accomplish two goals for the Pokémon data set. First, we implemented five different machine learning models to predict a given Pokémon's type based on a large set of features. Second, we used association pattern mining to find common patterns in Pokémon weaknesses and determine if certain patterns were correlated. Overall, we believe that the majority of these models could be used for successful classification of Pokémon type. We also believe that the pattern mining results are useful for further work that would involve creating the best Pokémon team composition.

For our future work, we would like to consider expanding these knowledge discovery techniques to the data set to identify the best Pokémon team given an enemy composition or a constraint, such as non-legendary types only. This particular problem introduces many additional challenge, as our models would require data and knowledge beyond the scope of this data set. For example, this data set has each Pokémon's base move set; however, Pokémon can learn additional moves and their current move set could be a large combination of abilities. Some abilities may be clearly better than the base set and thus need to be considered. Additionally, the data regarding what move(s) each Pokémon can learn by means of a TM [13]. Using our the results from our association pattern mining experiments, we have a good idea of the balance of ability types that we could consider. Using this as a starting point may help reduce the complexity of this particular example. Another example to consider is type pairings and what secondary type skills are beneficial to an entire team composition. Lastly, we would like to test and verify these future works by performing ”Pokémon Battles”. To achieve this, we plan to use the popular Pokémon simulator, Pokémon Showdown! [7]. Because of the complexities of these problems, we have decided to use this paper as a starting point to expand on future work.

References

[1] The Editors of Encylopaedia Britannica, “ Pokémon” Britannica.

[2] www.kaggle.com

[3] https://www.kaggle.com/mariotormo/complete-pokemon-dataset-updated-090420

[4] Guo, Gongde and Wang, Hui and Bell, David and Bi, Yaxin. (2004). KNN Model-Based Approach in Classification.

[5] Slimani, Thabet & Lazzez, Amor. (2014). Efficient Analysis of Pattern and Association Rule Mining Approaches. International Journal of Information Technology and Computer Science. 6. 10.5815/ijitcs.2014.03.09.

[6] Identifying Legendary Pokémon using The Random Forest Algorithm: https://bit.ly/3npWC1L

[7] https://pokemonshowdown.com/

[8] Ting K.M. (2011) Confusion Matrix. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA.

[9] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[10] https://www.tensorflow.org/datasets/keras_example

[11] https://scikit-learn.org/stable/modules/naive_bayes.html

[12] https://sklearn.org/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[13] https://bulbapedia.bulbagarden.net/wiki/TM