**Classification**

Tiati Thelen

Department of Computer Information Sciences, Minnesota State University, Mankato

CIS 677: Individual Studies

Dr. Guarionex Salivia

March 9, 2022

**Contents**

Introduction .….…………………………………..……………………………………………… 3

Decision Trees ……………………….………………………………….………………………. 6

Support Vector Machines ……………………………………………….………………………. 6

K-Nearest Neighbors …………...……………………………………….………………………. 7

Results …………………………….…………….……………………….………………………. 8

References ...……………………….…………………………………….……………………... 11

# Introduction

# Machine Learning problems can be broken down into classification and regression. Classification is defined as categorizing a data into classes. Specifically, the goal of this paper is to classify Pokémon as legendary or base level based on relevant attributes. The data set found was provided by (Barradas, 2016). Three algorithms are implemented and evaluated to determine the best model for the dataset. Decision Trees, Support Vector Machines, and K-Nearest Neighbors will be compared in terms of confusion matrices and accuracy on the test data. All data cleaning, algorithm information, and algorithm implementation was provided through the textbook *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (Géron, 2019).

The data is prepared by removing unnecessary variables. Specifically, the column ‘Name’ and ‘Number’ will be removed since it contains irrelevant data. Another data cleaning was to consider null values. There were columns that contained null values, and of those null values around 40% of the column was null. Therefore, following the textbook those columns were also removed.

Graphical user interface, text, application, email

Description automatically generated

Figure 1: Null Values

The next data cleaning steps were changing all variables that held true and false values into zeros and ones. I also used ordinal encoding for the variable “Color”. This variable needed to be encoded since it held around ten categorical value types. I chose ordinal by the recommendation the book provided. The two figures below show the code and steps taken to complete this.

Table

Description automatically generated

Figure 2: Binary Values

Table

Description automatically generated

Figure 3: Encoding

Lastly, before starting the model training, it is important to create training and testing data frames. Therefore, the y training and testing variables will hold if the Pokémon is legendary or not. I chose to have 33% of the data hidden to be held as testing data. This hidden data will be the consistently the same set across all three models. The figure below shows the implementation of this data split. I did implement this through a module in the sklearn library.

Graphical user interface, text, application, email

Description automatically generated

Figure 4: Training and Testing Data Sets

**Decision Trees**

The first algorithm implemented was Decision Trees. This algorithm can perform both regression and classification tasks. The Scikit-Learn library uses Classification and Regression Tree (CART). CART is preferred for smaller datasets and is becoming more common (Priyam, Abhijeeta & Guptaa, 2013). Overall, this algorithm wants to reduce its cost function. It starts with splitting the training set into two subsets by a single feature, k, and by a threshold tk. These two parameters are chosen by searching for the park (k, tk) that provides the purest subsets. The determination of level of pureness is also weighed on the subset’s size. It will continually split these subsets further with the same logic. It will stop recursing once it reached max depth or if it cannot find a split to reduce impurity. It will then execute tree pruning to produce a simpler tree. Finally, the algorithm will produce an optimal tree solution. (Lewis, 2000).

Decision Trees is a greedy algorithm, thus is will greedily search for a split. It will produce a reasonably good solution. In general, finding an optimal tree is a NP-Complete problem. The prediction complexity is O(log2(m)) and the training complexity is O(nxm(log(m)) (Géron, 2019).

**Support-Vector Machines**

The second algorithm is Support-Vector Machines (SVMs). This algorithm is capable of linear and nonlinear classification, regression, and outlier detection. It is the most popular. It is well suited for classification of complex yet small-medium datasets. It uses statistical learning theory to search for a hypothesis that fits the data but without overfitting (Al-Jarrah, Yoo & Muhaidat, 2015).

It works by fitting the widest possible street between classes. It predicts the class of a new instance x. First, the computation of the decision function is calculated and if the result is positive then it belongs to the positive class (1). If the result is negative, it will be classified in the negative class (0). During training, the value of w and b are found in a way to make the margin as wide as possible while avoiding margin violations (hard margins) or limiting violations (soft margins). The goal is to minimize ||w|| to result in a large margin. This results in a constrained optimization problem (Géron, 2019).

It is preferred to use soft margin classification. This is because it is a balance between keeping the street as wide as possible and limiting margin violations. It is controlled by the hyperparameter c. Thus, if the model is overfitting the programmer should try and reduce c.

**K-Nearest Neighbors**

The last classification algorithm is K-Nearest Neighbors (KNNs). It assumes similar things exist in proximity. It initializes k to the set chosen number of neighbors. Then it calculates the distance between the query example and the current example from the data. It adds the distance and the index of the example to an ordered collection. Then, the collection is sorted from smallest to largest indices. It picks the first k entries and assigns labels accordingly. After storing the training, the set of all parameters must be normalized for faster calculations. A “good” k value can come from cross-validation (Thakallapelli, Ghosh & Kamalasadan, 2016).

As k decreases to one, the predictions become less stable. But as k increases, prediction become more stable and accurate. There is a balance because once k becomes too big more errors will appear. The advantage for this algorithm is that it is easy to use and implement. There is no need to build a model, tune several parameters, or make additional assumptions, lastly the algorithm is versatile is that it can do classification, regression, and search. On the other hand, this algorithm gets significantly slower as the number of examples/predictors increase (Harrison, 2019). This will lead to low efficiency, which is resulting from KNNs being a lazy algorithm, being dependent on a “good” k (Guo, et al., 2003). Lastly, it is expensive to calculate a k if the dataset is large.

**Results**

The following figures show the code and implementation of the previously stated algorithms. These algorithms were compared in terms of accuracy of the hidden test data set. An interesting point is that all these results were above 90%. The best algorithm is K-Nearest Neighbors, then Decision Trees, and lastly SVMs. In conclusion, I would say all these models performed exceptionally well and are acceptable to be used. We can see one is better but not only by a margin.

Table

Description automatically generated

Figure 5: Decision Tree Results

Table

Description automatically generated

Figure 6: KNN Results

Table

Description automatically generated

Figure 7: SVM Results

**References**

Al-Jarrah, O., Yoo, P., & Muhaidat, S. (2015, September). *Efficient Machine Learning for Big Data: A Review*. Science Direct. Retrieved from https://doi.org/10.1016/j.bdr.2015.04.001

Barradas, A. (2016, August 29). *Pokemon with stats*. Kaggle. Retrieved March 9, 2022, from https://www.kaggle.com/abcsds/pokemon

Géron A. (2019). *Hands-on machine learning with scikit-learn and tensorflow concepts, tools, and techniques to build Intelligent Systems*. O'Reilly.

Guo, G., Wang, H., Bell, D., & Bi, Y. (2003, November 3). *KNN model-based approach in classification*. SpringerLink. Retrieved from https://link.springer.com/chapter/10.1007/978-3-540-39964-3\_62

Harrison, O. (2019, July 14). *Machine learning basics with the K-nearest neighbors algorithm*. Medium. Retrieved March 8, 2022, from https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

Lewis, R. (2000). An Introduction to Classification and Regression Tree (CART) Analysis. *Annual Meeting of the Society for Academic Emergency Medicine*.

Priyam, A., Abhijeeta, & Guptaa, R. (2013). Comparative Analysis of Decision Tree Classification Algorithms. *International Journal of Current Engineering and Technology*, *3*(2).

Thakallapelli, A., Ghosh, S., & Kamalasadan, S. (2016, November). *Real-time frequency based reduced order modeling of large power grid*. IEEE Xplore. Retrieved from https://ieeexplore.ieee.org/document/7741877