Classify Them All – Predicting Pokémon Types with Machine Learning (2025)

Scott A. Ratchford

Abstract – **Pokémon are fictional creatures from a video game series of the same name. There are over 1,000 Pokémon species at the time of publication, and each species is labeled with exactly 1 or 2 "types" from a list of 18 types (see Appendix A).**

Although the appearance of a Pokémon may provide some indication of its type, it is sometimes difficult for a human to identify the type(s) of a Pokémon species from appearance alone. Machine learning methods may provide deeper insight into the connections between the appearance of Pokémon, their in-game statistics, and their types.

We seek to answer three questions. How do Pokémon statistics relate to Pokémon types? How do Pokémon colors relate to Pokémon types? Can the combination of Pokémon statistics and colors be used to accurately predict Pokémon types?

We found that the in-game statistics of Pokémon have an insignificant correlation with Pokémon types. The colors of each Pokémon have significant correlations with Pokémon types. The combination of these two data sets provides improved prediction accuracy, but the color data is far more significant that other data.

I. INTRODUCTION

A Pokémon may have exactly 1 or 2 unique types. These types are not evenly distributed across Pokémon. The order of Pokémon species' types has no meaning. Some types are far more common than others, and some combinations of types (such as Ground and Fairy) do not exist. Because of the sparse and irregular distribution of types, Pokémon type prediction requires multi-label classification. We will attempt to predict both types for each Pokémon, scoring our methods on "either-type accuracy" (correctly predicting at least 1 of the up to 2 types) and "both-type accuracy" (correctly predicting both of the 2 types, or 1 type and a value of "None" for the other type).

We used a combination of 2 data sets to perform our analysis. The first data set is known as the in-game statistics data set. Pokémon originate in video games, and these games assign various statistics to the Pokémon. These statistics include health points, attack, defense, special attack, and special defense. Our data set also includes a total of the given statistics. This total is used as an approximate measure of the overall strength of a Pokémon, which may also prove relevant to type classification.

The second data set is known as the image data set. This data includes exactly 1 image of every Pokémon, taken from their appearance in video games. We will extract data from these images, but we will not use the images themselves for our classification.

These two data sets will be used to answer our research questions: How do Pokémon statistics relate to Pokémon types? How do Pokémon colors relate to Pokémon types? Can the combination of Pokémon statistics and colors be used to accurately predict Pokémon types?

II. METHODS

To answer the questions we raised, we performed several types of analysis. First, we performed preprocessing on our two datasets. The Pokémon Statistics dataset includes information about 1025 unique Pokémon, but the "Pokémon with Stats and Images" dataset has information about 1215 Pokémon, including alternate forms. These alternate forms are quite complex to separate and would require extensive explanation if they were included in our analysis. For simplicity, we have removed alternate forms of Pokémon, leaving us with 959 Pokémon to analyze.

Next, we calculated the colors present in each Pokémon image. The image files store the color of each pixel as 4 numbers, each in the range [0, 255]. These numbers represent the red, green, blue, and alpha (transparency) of each pixel, otherwise known as RGBA. We ignored all fully transparent (alpha 255) pixels. To translate the color of each pixel into names that humans are more familiar with, we used a list of 9 colors (see Appendix B). For each pixel, we calculated the closest defined color by Euclidean distance, treating the red, green, and blue values as axes in a 3-dimensional space (see Appendix C). Next, these color values were transformed into percentage values based on the percentage of the non-transparent pixels in the image. For example, the image of the Pokémon Rayquaza contains 3531 non-transparent pixels. Approximately 34.834325% of these pixels are closest to black, so the black value was stored as 34.834325. We assigned values for each of the 9 colors in this way. As the last step of color analysis, we scaled the color data (see Appendix D).

After color data, we removed the Pokémon species which have peculiarities which prevent us from performing proper type predictions. Some Pokémon species possess alternate forms, such as Mega, Galarian, and Huisian forms. Because the image data set did not possess images for all the unique Pokémon forms, we chose to exclude all Pokémon with alternate forms from our analysis. This step removed 64 Pokémon (256 rows of the data set, due to the number of alternate forms) which were originally present in the in-game statistics dataset.

For the last step of data preprocessing, we randomly split the data into training and testing sets. 80% of Pokémon were used for training data, and the remaining 20% were used for testing data. If a Pokémon species' in-game statistics were used for training, it was also used for color data training. If a Pokémon species' in-game statistics were used for testing, it was also used for color data testing. This way, the same Pokémon were used for training or testing across both data sets.

2 machine learning models were trained on the in-game statistics dataset. The data used for predictions were the health points, attack, defense, special attack, special defense, speed, and statistics total of each Pokémon species. The target values were the types of each species. If a species only possesses a single type, the second type value was treated as a value called "None". Thus, the models used predicted the first type from the list of 18 Pokémon types, and they predicted the second type from a list of 19 possible values (the 18 types and a "None" value).

The 2 machine learning models trained exclusively on in-game statistics were a decision tree classifier and a k-nearest neighbors classifier. Both models were wrapped in a multioutput wrapper known as a multi output classifier. This wrapper automatically extended the functionality of the models to predict 2 types for each Pokémon, not just 1. The wrapped models were optimized using a grid search cross-validation on a set of hyperparameters designed to avoid overfitting to the training data. These models were used determine if Pokémon of the same type often had similar statistics, such as Rock and Ground types having high defense values. The decision tree and k-nearest neighbors models were chosen for this task due to their non-linearity. With these models, outlier values have little impact on the classification process.

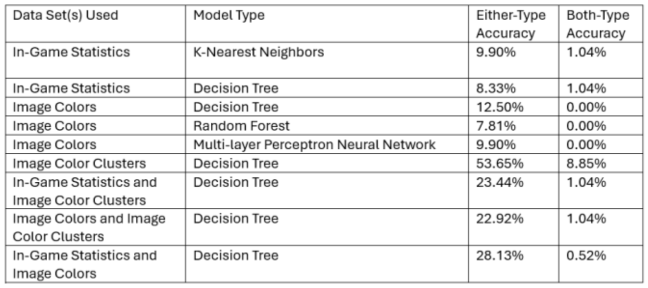
3 machine learning models were trained exclusively on the image color data. These models were also wrapped in a multioutput classifier and used grid search cross-validation optimization. The models selected for image color analysis were a decision tree, a random forest, and a multi-layer perceptron neural network. These decision tree and random forest were chosen for their ability to split Pokémon into smaller and smaller groups based on specific criteria, such as a high percentage of black pixels potentially indicating a Pokémon possessed the Dark type.

A new data set, the image color clustering dataset, was created using k-means clustering. Each of the training Pokémon species were assigned to 1 of 8 clusters. This assignment is a type of dimensionality reduction used to reduce the number of features in a future combination of the previous data sets. A decision tree was trained on the image color clusters using the same process as the previously discussed models. Because only 9 colors were used for image analysis, we anticipated that these would fall into clusters approximating the distribution of colors, such as a group of Pokémon that are mostly black and gray, a group of Pokémon that are comprised of warm colors (red, orange, and yellow), and so on.

Finally, 3 decision trees were trained on combinations of the other data sets. One model was trained on both the in-game statistics and the image colors, another model was trained on both the in-game statistics and the image color clusters, and another model was trained on both the image color and image color clusters. The intention in training these models was to find larger trends in the combination of the data sets. The increased amount of data in these combined data sets drastically increased training time. Because of the limited computing resources available to the author, a lightweight model type like the decision tree was required.

III. Results

Fig. 1 presents the accuracy of each model. "Either-Type Accuracy" is the percentage of instances in which the model correctly predicted at least 1 of a Pokémon species' types (ignoring the second type when it is not present). "Both-Type Accuracy" is the percentage of instances in which the model correctly predicted both of a Pokémon species' types (or 1 if the species has only a single type).



Data Set(s) Used	Model Type	Either-Type Accuracy	Both-Type Accuracy
In-Game Statistics	K-Nearest Neighbors	9.90%	1.04%
In-Game Statistics	Decision Tree	8.33%	1.04%
Image Colors	Decision Tree	12.50%	0.00%
Image Colors	Random Forest	7.81%	0.00%
Image Colors	Multi-layer Perceptron Neural Network	9.90%	0.00%
Image Color Clusters	Decision Tree	53.65%	8.85%
In-Game Statistics and Image Color Clusters	Decision Tree	23.44%	1.04%
Image Colors and Image Color Clusters	Decision Tree	22.92%	1.04%
In-Game Statistics and Image Colors	Decision Tree	28.13%	0.52%

Fig. 1. A table of model accuracy.

Clearly, the in-game statistics of a Pokémon has very little correlation with its type(s). The performance of the models using only this data set was only slightly different from a purely random prediction of the type(s). The image colors have a slightly higher correlation with Pokémon type(s) than in-game statistics do. Surprisingly, the model with the highest performance in either-type and both-type accuracies is the decision tree trained solely on image color clusters. This suggests that Pokémon can be accurately clustered into several color groups, like Pokémon with mostly warm colors, as discussed previously. The combinations of the in-game statistics with image colors greatly outperformed their individual models. Thus, the author's hypothesis that the combination of these data sets would yield higher results is supported. The decision trees trained on the combinations of color clusters with other data sets underperformed the decision tree trained solely on color clusters. This indicates that the color clusters are the only highly significant data in predicting Pokémon type(s).

Before performing the analysis, the author anticipated a much higher correlation between color and type. It is likely that the accuracy of these models is far lower than expected because of the difficulty of performing multi-label classification. If each Pokémon possessed only a single type, it is likely that the accuracy of each model's predictions would be much greater. Additionally, the usage of the multi-output classifier wrapper greatly lowered the potential performance of the models.

IV. Conclusion

The in-game statistics of Pokémon has no significant correlation with type. This is proven by the fact that type prediction accuracy based on in-game statistics was very low. These predictions performed approximately the same as random guessing.

While Pokémon color is moderately correlated with Pokémon type, the data that is extracted from the color data is much more significant than the color data itself. It is possible that by using a different set of colors, such as additional color definitions or updating the specific RGB values of each color, the utility of the color data would be significantly improved.

Color data is only a single aspect of Pokémon images. Further research should be conducted using additional data from the images. This could include the overall shapes (such as angularity or roundness) of the images or physical features (such as wings, paws, or tails) present in the images. Due to the high time commitment in labeling the images in these ways, these suggestions are outside the scope of this paper.

Overall, type predictions were complicated by certain characteristics of Pokémon typing itself. Firstly, Pokémon may have 1 or 2 types. If all Pokémon possessed exactly 1 type or exactly 2 types, type classification could be performed separately for both types. Because Pokémon may or may not have 2 types, we used multi-label type classification, which significantly reduces the likelihood of correct predictions.

V. Appendix

A. Pokémon Types

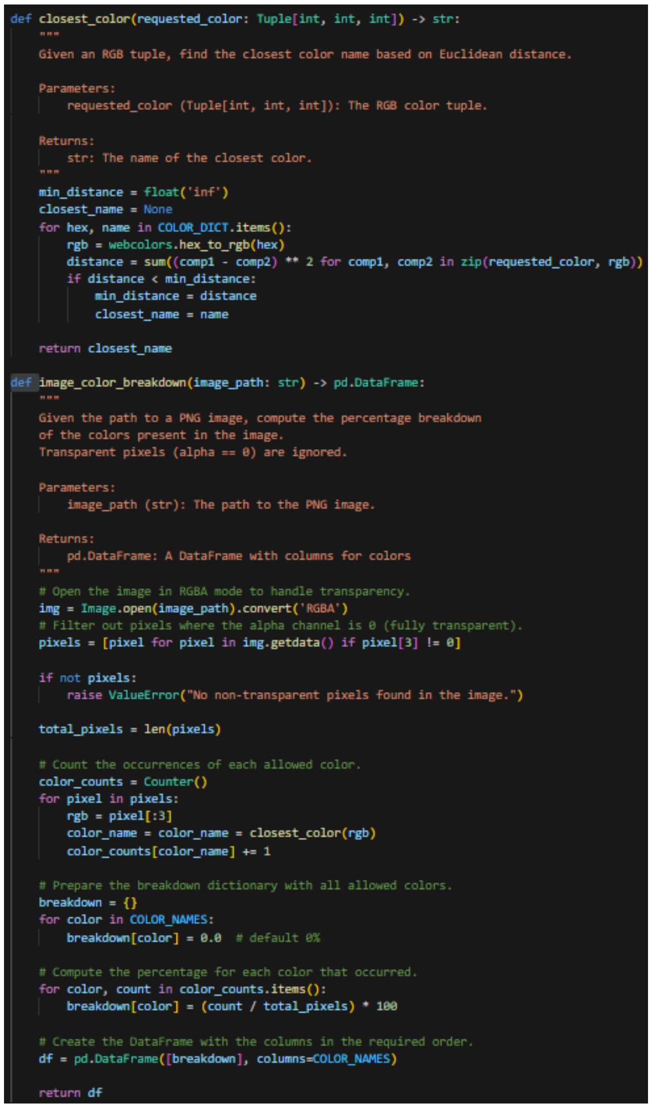
The 18 types of Pokémon are Normal, Electric, Rock, Grass, Dragon, Water, Ice, Steel, Psychic, Fairy, Dark, Ground, Fighting, Bug, Flying, Fire, Poison, and Ghost.

B. Color Definitions

Color Name	Red	Green	Blue
black	0	0	0
blue	0	0	255
pink	255	0	255
green	0	128	0
purple	128	0	128
red	255	0	0
white	255	255	255
yellow	255	255	0
orange	210	146	20

Fig. 2. The table of color definitions.

C. Color Analysis

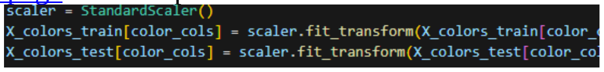


```python
def closest_color(requested_color: Tuple[int, int, int]) -> str:
    """
    Given an RGB tuple, find the closest color name based on Euclidean distance.

    Parameters:
        requested_color (Tuple[int, int, int]): The RGB color tuple.

    Returns:
        str: The name of the closest color.
    """
    min_distance = float('inf')
    closest_name = None
    for hex, name in COLOR_DICT.items():
        rgb = webcolors.hex_to_rgb(hex)
        distance = sum((comp1 - comp2) ** 2 for comp1, comp2 in zip(requested_color, rgb))
        if distance < min_distance:
            min_distance = distance
            closest_name = name

    return closest_name

def image_color_breakdown(image_path: str) -> pd.DataFrame:
    """
    Given the path to a PNG image, compute the percentage breakdown
    of the colors present in the image.
    Transparent pixels (alpha == 0) are ignored.

    Parameters:
        image_path (str): The path to the PNG image.

    Returns:
        pd.DataFrame: A DataFrame with columns for colors
    """
    # Open the image in RGBA mode to handle transparency.
    img = Image.open(image_path).convert('RGBA')
    # Filter out pixels where the alpha channel is 0 (fully transparent).
    pixels = [pixel for pixel in img.getdata() if pixel[3] != 0]

    if not pixels:
        raise ValueError("No non-transparent pixels found in the image.")

    total_pixels = len(pixels)

    # Count the occurrences of each allowed color.
    color_counts = Counter()
    for pixel in pixels:
        rgb = pixel[:3]
        color_name = color_name = closest_color(rgb)
        color_counts[color_name] += 1

    # Prepare the breakdown dictionary with all allowed colors.
    breakdown = {}
    for color in COLOR_NAMES:
        breakdown[color] = 0.0  # default 0%

    # Compute the percentage for each color that occurred.
    for color, count in color_counts.items():
        breakdown[color] = (count / total_pixels) * 100

    # Create the DataFrame with the columns in the required order.
    df = pd.DataFrame([breakdown], columns=COLOR_NAMES)

    return df
```

Fig. 3. The Python code used to analyze image colors.

D. Color Scaling

Color scaling was performed using the StandardScaler from Python module scikit-learn. See this webpage for an explanation.



```python
scaler = StandardScaler()
X_colors_train[color_cols] = scaler.fit_transform(X_colors_train[color_
X_colors_test[color_cols] = scaler.fit_transform(X_colors_test[color_co
```

Fig. 4. Usage of the StandardScaler.

E. Color Clustering Visualization

Due to the high dimensionality (9 dimensions) of the color clusters, we used an unconventional approach to represent these colors. The solid lines on the graph represent each Pokémon in the training data, and the dotted lines represent the clusters.

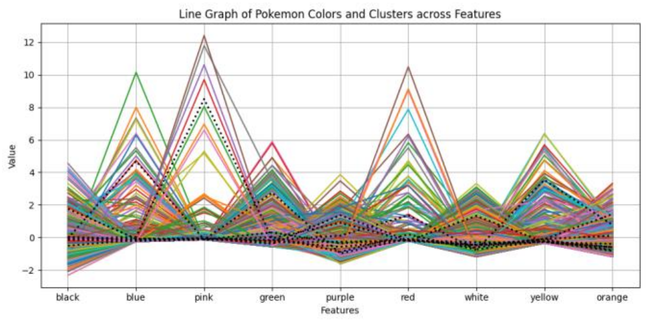


Fig. 5. A graph of the color clusters.

VI. Sources

[1] Christoffer MS, "Pokemon with Stats and Images" *Kaggle*, Sep. 2024.
[2] K. Arnav, "Pokemon Pokedex with Stats" *Kaggle*, Mar. 2024.
[3] Scikit-learn, "StandardScalar - scikit-learn 1.6.1 documentation", 2025.