Statistical analysis of Pokémon

Dzmitry Kurch, DSE

Statistical learning project, June 2023

Abstract

This paper presents an investigation into the application of statistical learning methods for the analysis of the Pokémon dataset. The Pokémon franchise, renowned for its rich and diverse collection of creatures, provides a unique opportunity to explore the capabilities of machine learning techniques. Specifically, this study focuses on employing classification analysis, clustering, and dimensionality reduction methods to gain insights into the underlying patterns and characteristics of the Pokémon universe.

The first phase of the analysis involves classification analysis, where various supervised learning algorithms are trained on a labelled subset of the dataset. By harnessing the power of classification algorithms, we aim to accurately predict the attributes and properties of Pokémon based on their known characteristics. This analysis can be instrumental in assisting trainers and researchers in making informed decisions and developing effective strategies.

Next, dimensionality reduction methods are employed to capture the essential features of the dataset while reducing its complexity. By transforming the original high-dimensional Pokémon dataset into a lower-dimensional space, we aim to extract meaningful representations that facilitate visualization and interpretation.

Furthermore, clustering techniques are applied to group Pokémon based on their similarities. By leveraging unsupervised learning algorithms, we aim to identify inherent clusters or groupings within the Pokémon dataset, which may reveal hidden relationships or evolutionary patterns.

Overall, this research showcases the potential of statistical learning methods in unlocking valuable insights from the Pokémon dataset. The findings from classification analysis, clustering, and dimensionality reduction can contribute to the advancement of Pokémon-related research, gameplay strategies, and educational applications. Moreover, the methodologies presented in this study can serve as a foundation for applying statistical learning techniques to other large and complex datasets in diverse domains.

1. Introduction

Pokémon is media franchise that began as a pair of Role Playing Games (RPG) video games for the original Game. Boy that were developed by Game Freak and published by Nintendo, in 1996. With the passage of time Pokémon increased its popularity and thus its owners ended up producing many animated television shows, films, trading card games, various manga comics, as well as a number of video games of different kinds, like the released augmented reality game Pokémon Go!, that really caught on in 2016.

Pokémon are fictitious animal-like monsters that live in the (of course, also invented) Pokémon world. Pokémon like fighting with each other, and they usually fight according to their (human) trainers' orders. Almost all the Pokémon games include these fights, but in different manners. In some of them the user needs to rely on her or his strategy and in the strength of his or her Pokémon, whereas other video-games are more ability-based. Hence, an interesting fact of this games may well be the way the strength or the ability to fight of a Pokémon is described. This depends, once again, on the type of the video-game. For statistical analysis purposes, the most attractive way of describing the Pokémon is that of the RPGs'. First of all, because a big number of Pokémon have been introduced throughout these years -seven generations of Pokémon with the order of 100 of Pokémon in each of them. Second, in the RPGs each Pokémon is described with a big number of variables. Not only do we have the combat stats (the variables that describe the ability to fight), but also many variables that describe more details of each Pokémon, e.g. the colour or the probability of being female or male.

Thus, we can statically analyse the wide variety of variables used to describe the Pokémon, and there is a chance to find relationships between them, and also to cluster the Pokémon according to some criteria. In the rest of the report we will explore the Pokémon and their corresponding variables that appear in the RPGs.

First we will introduce the variables and instances of the dataset in *Section 2*. We will explore the variables and their potential dependencies in *Section 3*. Once we have studied the variables, we will try to predict type of the Pokémon, in *Section 4*. We are going to use a PCA as a powerful dimensionality reduction method to visualize the data in the *Section 5*. Finally we make some clustering of the Pokémon using the output from the PCA in *Section 6*, and with draw some conclusions in *Section 7*.

2. The dataset and its variables

The dataset is taken from Kaggle, here you may find the original source and the description https://www.kaggle.com/datasets/alopez247/pokemon.

By and large, in total we have 721 samples with 24 variables to analyze. The first two are unique identifiers of the Pokémon, the number in the Pokédex and the name. The Pokédex is encyclopaedia-like tool that can be used in the Pokémon RGBs to get information of the Pokémon. In fact, most of the variables we will use in this work are taken from the Pokédex. From the resting 21 variables, 12 are numerical (10 continuous and 2 discrete), 6 categorical and 3 boolean. Please refer to Appendix section to read the detailed description of those variables.

3. Exploratory Data Analysis

In this section we will try to gain an insight of the distributions of the different variables as well as some relationships between them.

Firstly, we do a quick data quality analysis by exploring the percentage of missing variables in the dataset.

Table 1 - Missing values percentage

Feature name	***Missing percentage***
Egg_Group_2	73.51
Type_2	51.45
Pr_male	10.68

Based on the table above we conclude that situation is not that bad, only 3 out of 24 features has missing values. Moreover, all of those features can be considered as *categories,* which allows us just to add an additional category *Null* and go on with the encoding later.

Now we proceed with a Bivariate variable analysis by combining the Univariate analysis with an attempt to explore the relationship between variables. Let's start with plotting a correlation matrix of the numerical features. As you may see on the picture below there are a few groups of cross-correlated features which were expected based on their nature. For example, Pokémon fight characteristics such as *(speed) attack, (speed) defence and health points* are positively correlated because usually Pokémons are constructed in a "balanced" way. Also an interesting observation is that *catch rate* is negatively correlated with almost all characteristics, which means catching a Pokémon with high fight capability is harder.

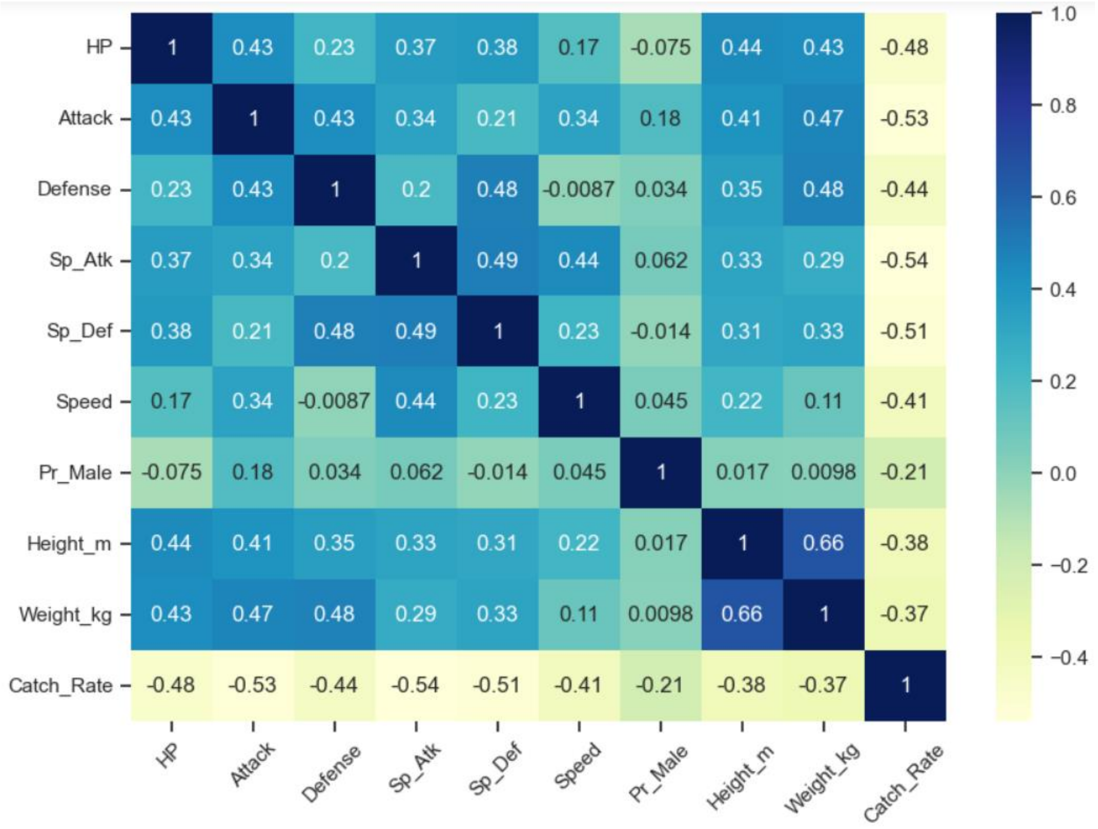


Figure 1 - Correlation matrix of numerical features

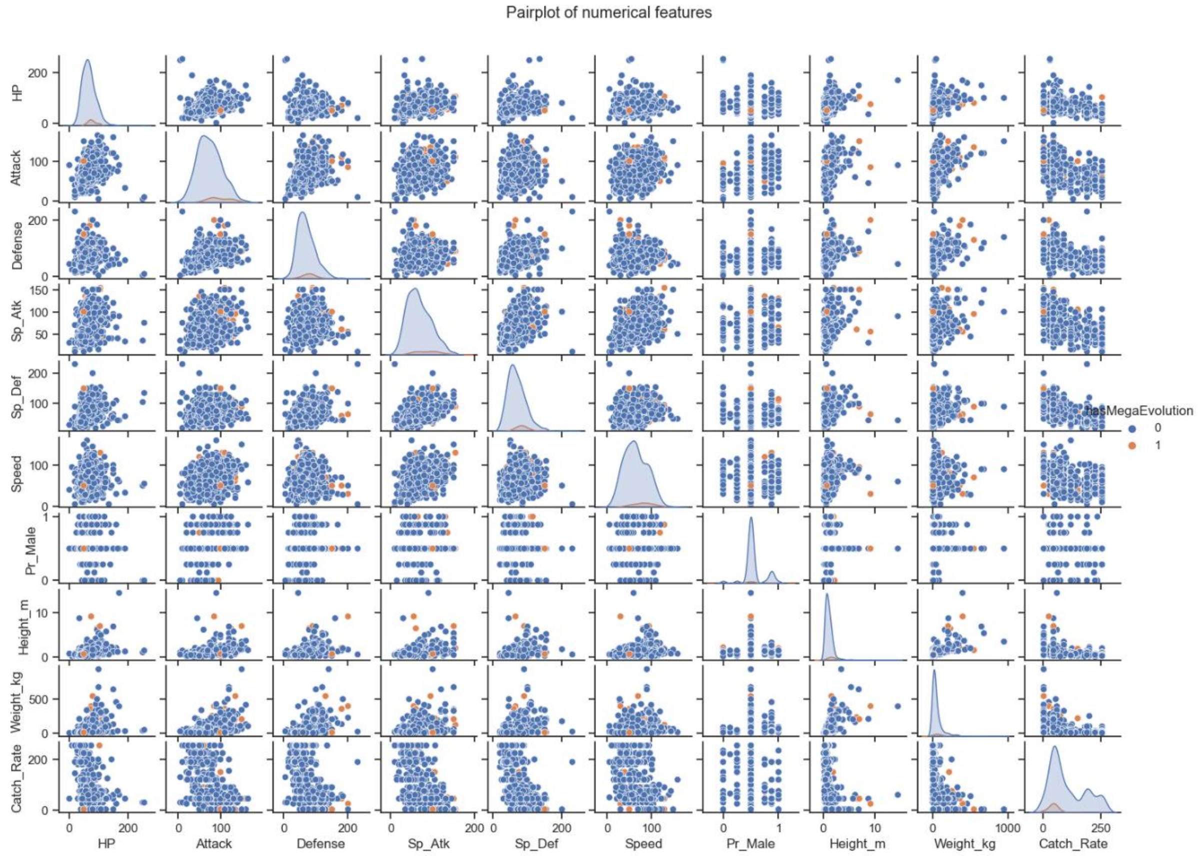


Figure 2 - Pairplot of numerical features

Our main goal for that project will be to predict the *hasMegaEvolution* characteristic which is property that some Pokémon have and allows them to change their appearance, types, and stats during a combat into a much stronger form. Among the Pokémon players they are considered to be very valuable, that is why it is important to be able to know that fact before you put an effort into catching the Pokémon itself. Important note: our dataset will be highly unbalanced containing only ***6.4%*** samples with *hasMegaEvolution* property.

Based on the pair plot above we observe an interesting tendency that most of some Pokémons who hold a mega evolution tend to "cluster" in the right upper corner of the fight characteristics scatter plots, which means that a high value of those characteristics may be a good sign for that type.

After that I analysed the percentage of *hasMegaEvolution* Pokémons across numerical and categorical variables to highlight the potentially most valuable characteristics which can help us to spot that type. Here are only a few examples.

Across Pokémons with the highest attack value (more than 100) there are ***15%*** of the mega-evolutionary, which is almost 3 times higher than the total mean.

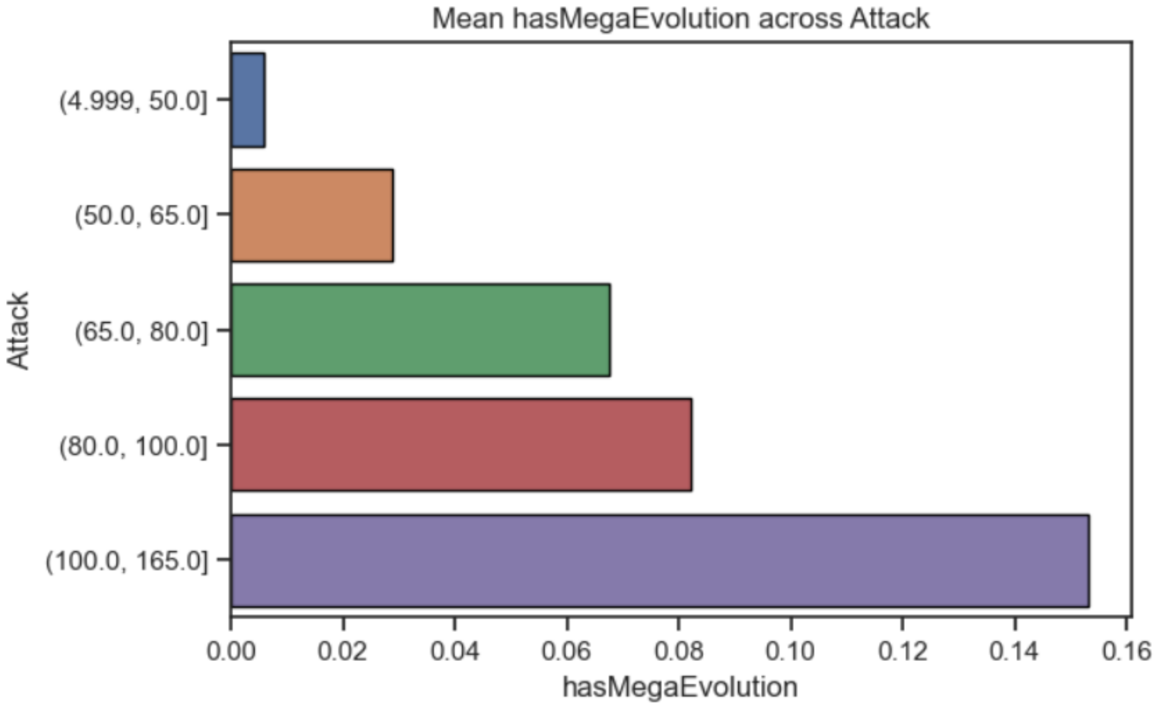


Figure 3 - Distribution of MegaEvolution poke for different attack ranges

Across Pokémons with the defence range [72, 90] there are ***14%*** of the mega-evolutionary ones, which suggests that feature has a good predictive potential.

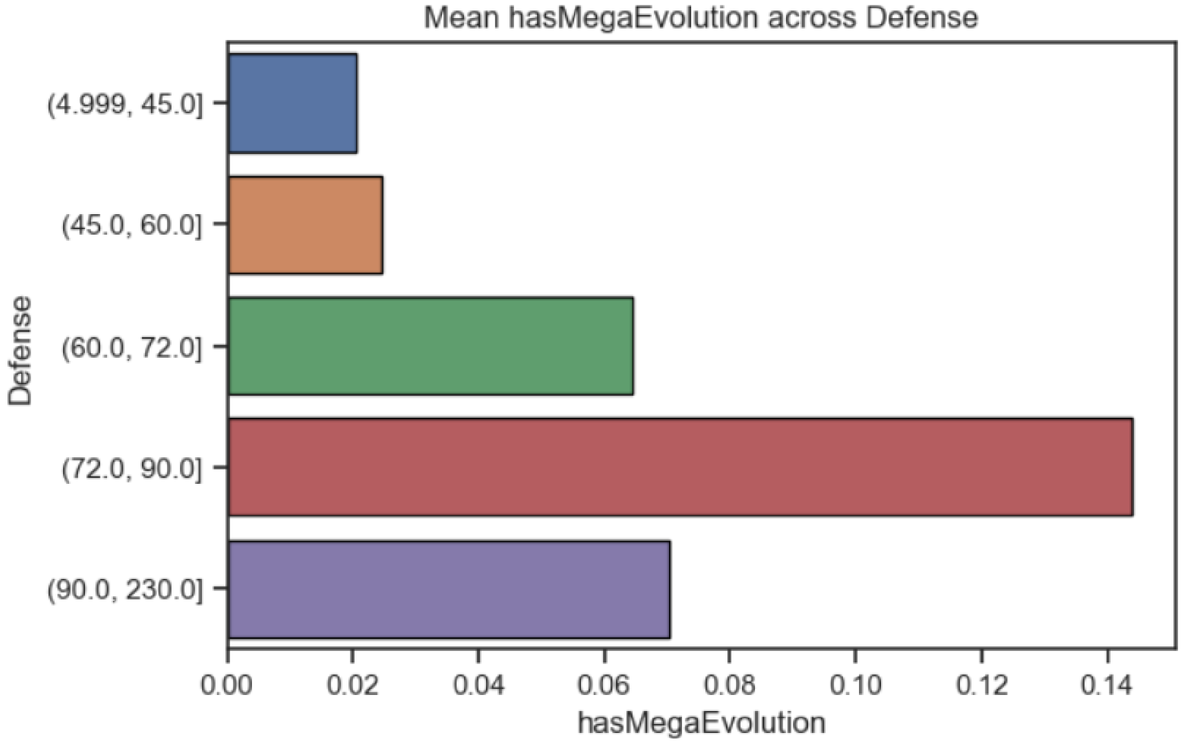


Figure 4 - Distribution of MegaEvolution poke for different defence ranges

A plot above shows us that the *Dragon type* has the highest percentage of mega-evolutionary Pokémons ***25%*** with the 24 samples considered.

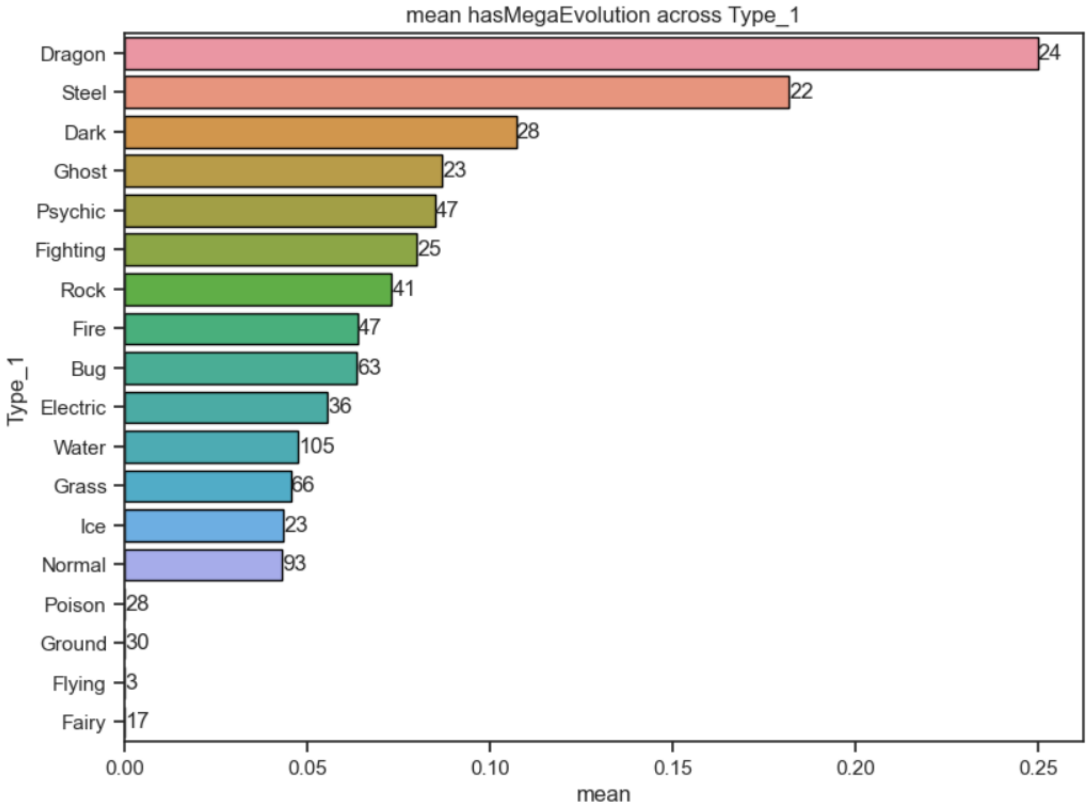


Figure 5 - Distribution of MegaEvolution poke for different Type_1

4. Classification

In this section we introduce an attempt to predict one of the most important characteristics of Pokémon *hasMegaEvolution.* In the previous section we highlighted a few features which can give a good signals for spotting a mega-evolutionary Pokémons. That is a good sign of taking a benefit from combining them together to build the model.

4.1 Data preprocessing

First and foremost, we have to preprocess our data properly. We apply a numerical feature scaling by subtracting a mean and dividing by a standard deviation from each sample. For categorical features we use a One-Hot-Encoding technique by setting to all but the one the dummy variables to avoid a “trap”.

After applying all of those steps we end up with 101 features (dummy + numerical scaled), which will be used to fed up our model.

4.2 Model quality estimates

From the one hand, we are sure we would like to focus on spotting the minor class (mega-evolutionary). But from the other hand, it is hard to guess which type of errors will have less weight – false positives or false negatives? To take care of both precision and recall without prioritizing one over another we choose an F1-score which is simply a harmonic mean of those two as a model quality measure.

To measure the quality of the model in a proper way we split the dataset into two parts – *train* and *test*, where the first one will be used for model hyperparameter tuning and the second for the best model estimation. Moreover, to keep the test dataset “untouched” we extract the necessary parameters for applying a feature scaler and OHE from the train and apply it on the test as if we didn’t see it at all. That is especially important for the OHE, because what happens is that if it encounters an unseen category in the test it handles that as *no category* by setting all dummy variables to zero.

4.3 Logistic regression

The first model we will be using is a simple Logistic regression with L2 regularization where the main hyperparameter will be the strength of the regularization measure by the corresponding coefficient. We apply a hyperparameter tuning using a cross-validation with 5 folds to reduce a bias in our estimation on a train set. Below you may observe a training plot which shows the dependence of the F1-score on the regularization strength:

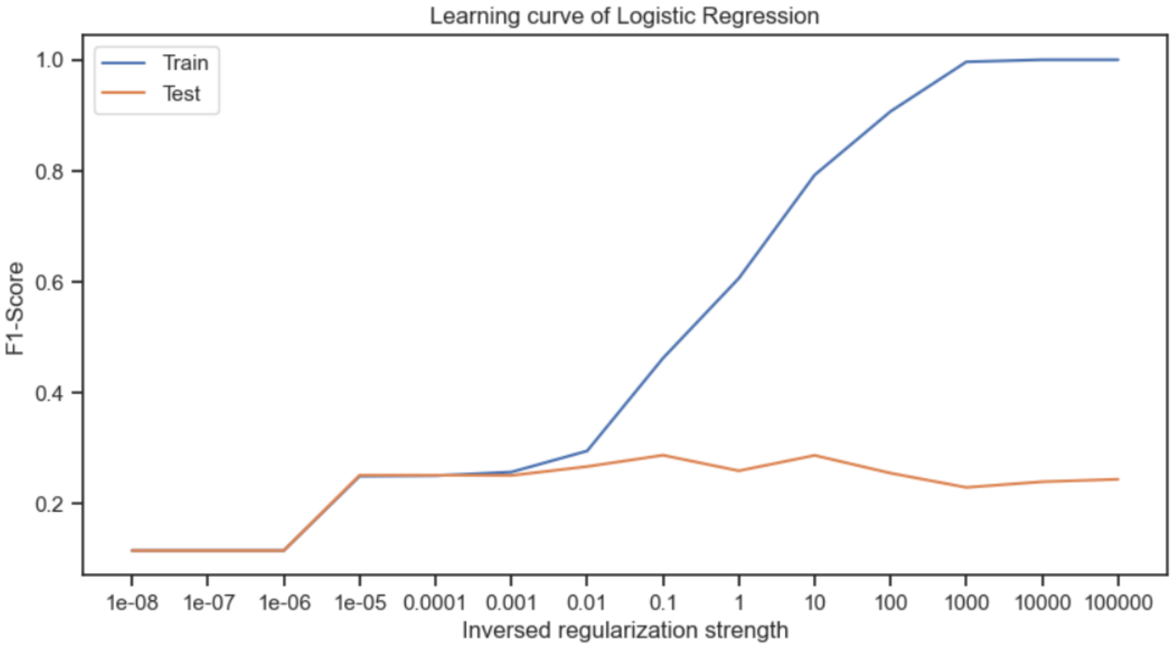


Figure 6 - Learning curve of Logistic Regressio

We can see that models with a strong regularization tend to have almost identical results both on train and test with relatively low scores, which is a sign of underfitting. Then, after some point of relaxing the regularization strength the train and test line start to diverse and clearly end up with an overfitted model. Based on that plot we find out the best value for the *inversed regularization strength* which is 0.1 which provides a descent test result without producing a highly overfitted model. After training the model on the whole train dataset and measuring the quality on the test we end up with the following numbers:

Table 2 - Logistic RegressionTest classification report

class	*0*	*1*
precision	0.95	0.19
recall	0.88	0.38
F1-score	0.91	0.26
support	168	13

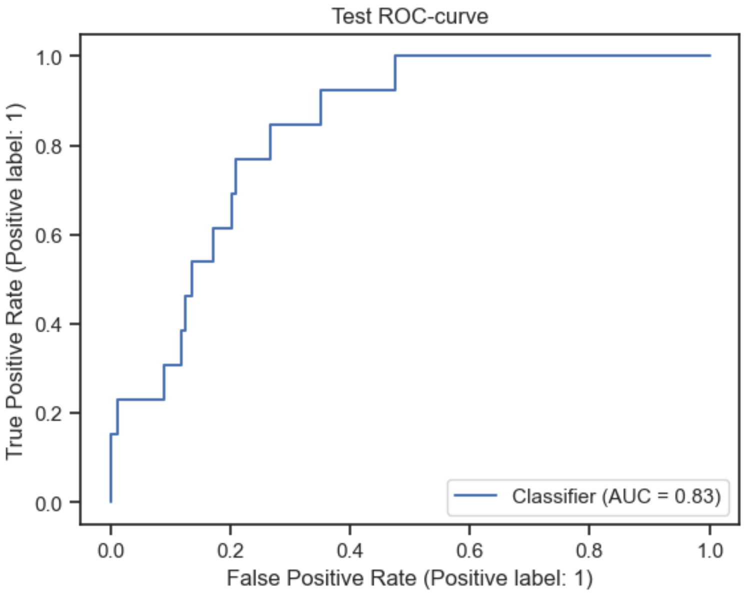


Figure 7 - Logistic Regression Test ROC-curve

By extracting the weights of the Logistic Regression model we can hypothesize (under the certain assumptions) about the contribution of Pokémon characteristics to the probability of mega-evolution.

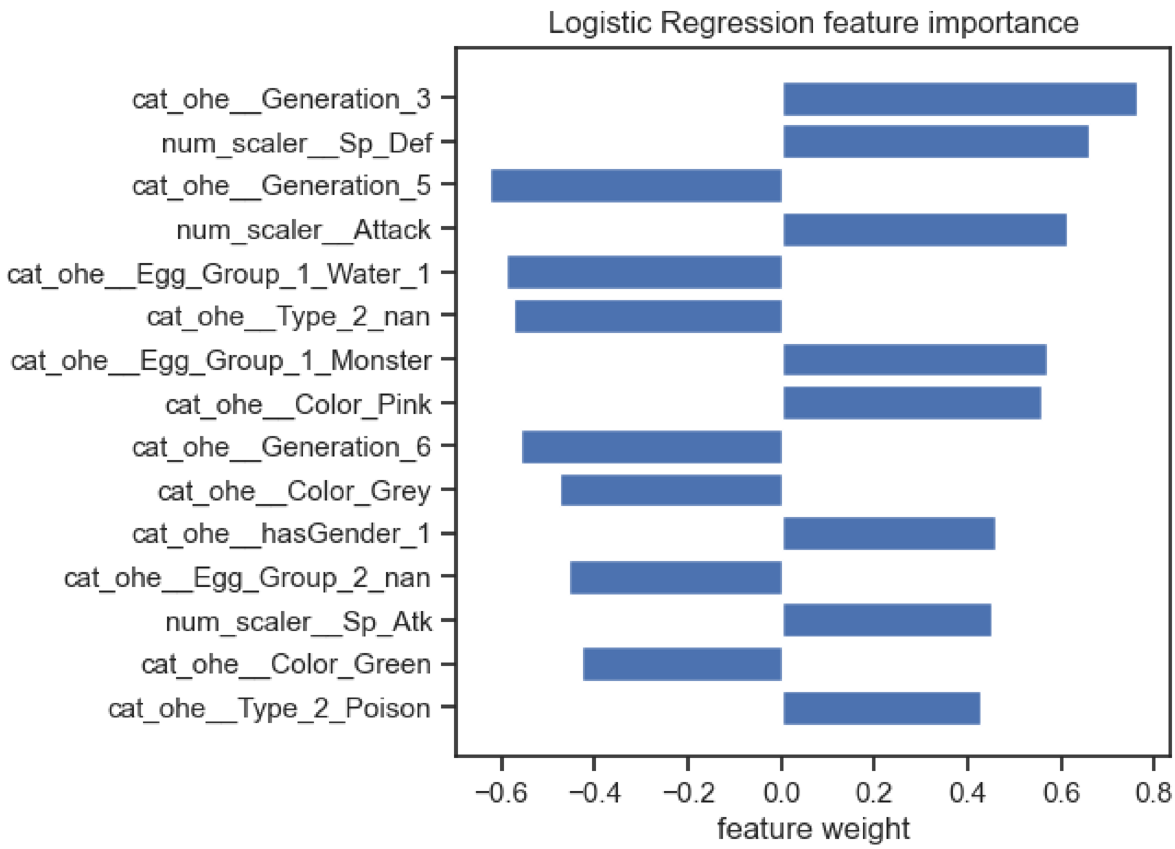


Figure 8 - Logistic Regression feature importance

For example, the Pokémons of the third generation are more likely to be mega-evolutionary, which we have observed during the EDA. Thus we end up with the positive coefficient for the dummy variable “3” of the category *Generation.* All positive characteristics such as *speed attack* or *attack* have the positive impact as expected.

Note that the logistic regression uses the *balanced class weight* that allows us to penalize the model for the errors of the minor class much more.

4.4 Random forest

Next step for us will be training the Random Forest Classifier to identify the mega-evolutionary Pokémons. Note that the same data preprocessing steps were performed and the validation schema remained the same. Unlike the Logistic regression which had only one main hyperparameter Random Forest has many. To tune the multiple hyperparameters we used the Grid Random Search technique which allows us to avoid complexity of brute force consecutive parameter consideration. Instead we fix the parameter distribution and number of iteration which allows us to sample the combinations and select the best one. The hyperparameters and their distribution can be found in the table below:

Table 3 - Random forest classifier hyperparameter grid

Hyperparameter name	*Distribution*
Number of trees	Uniform(10, 100, 10)
Maximum tree depth	Uniform(1, 5)
Minimum sample split	Uniform(2, 20)
Minimum sample leaf	Uniform(1, 10)

As a result the best combination of the hyperparameters based on the Cross-Validation run on train set was *number of trees = 60, maximum tree depth = 3, maximum sample split = 10, minimum sample leaf = 5.* Resulting classification report for the test sample you may find below:

Table 4 - Random Forest Test classification report

class	*0*	*1*
precision	0.97	0.33
recall	0.89	0.69
F1-score	0.93	0.45
support	168	13

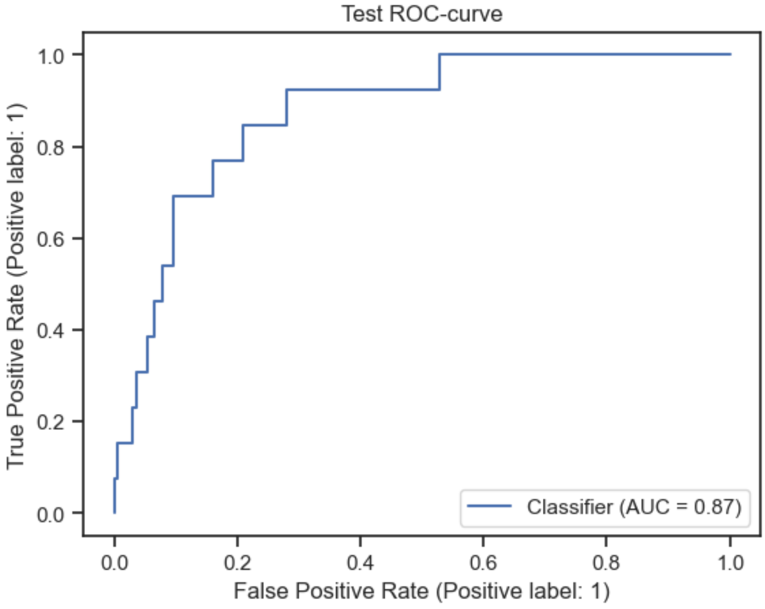


Figure 9 - Random forest Test ROC-curve

As we can see the results have improved significantly due to the Random forest capability of identifying the most important combinations of categorical factors and splitting the sample accordingly.

5. Dimensionality Reduction

In this section we introduce a dimensionality reduction algorithm which will be applied to our data in order to reduce the feature space.

We will be using PCA as a main feature reduction algorithm. Data preprocessing step will be the same as for clustering, please refer to the section 4.1. In order to obtain a reasonable trade-off between the information kept in the components and the resulting feature space we tune the hyperparameter using the "elbow rule".

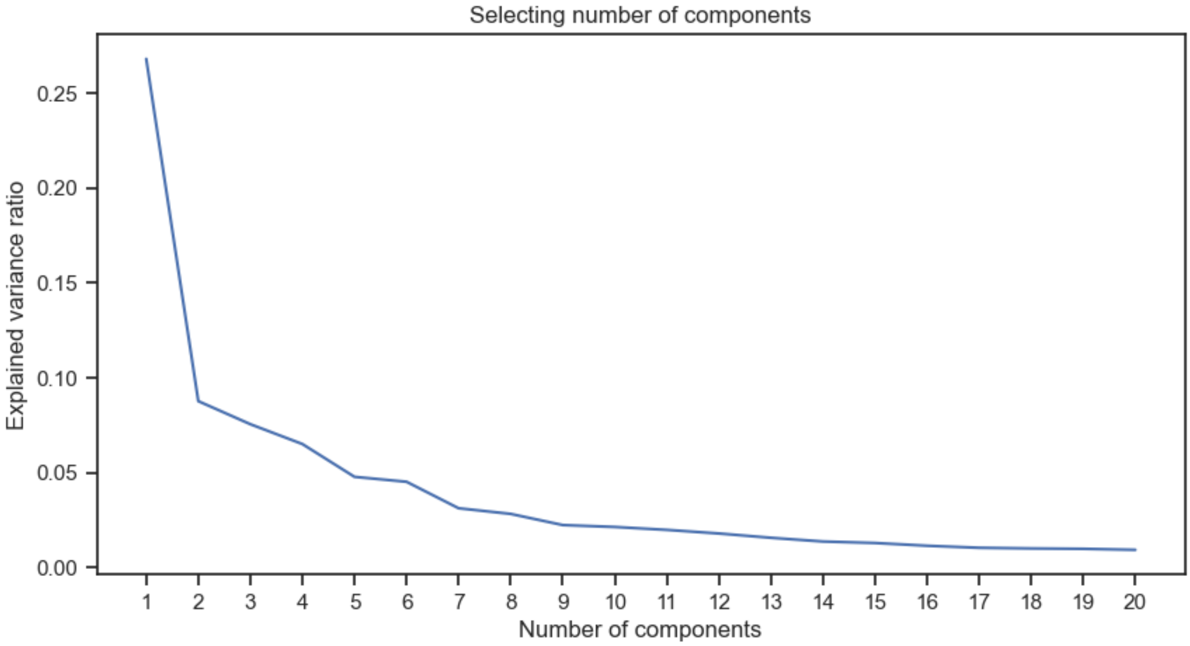


Figure 10 - PCA number of components selection

As you can see on the picture above approximately after adding the 10th component the marginal increase in explained variance is relatively low. By using 10 components in total 70% variance can be explained, which is a sufficient number for our purposes.

6. Clustering

In this section we introduce the Hierarchical clustering algorithm which we will apply to our Pokémon dataset. In order to avoid the *curse of dimensionality* (remember after data preprocessing we end up with 101 features having only 720 samples) we will use the clustering on the top of PCA output from the section 5. In order to achieve a meaningful result we will tune the number of clusters using the *Silhouette score* which balances the inner- and outer-cluster distance.

On the picture below you may see the resulting "training plot" of the Hierarchical clustering algorithm in which we can clearly observe the "optimal" point with k = 3.

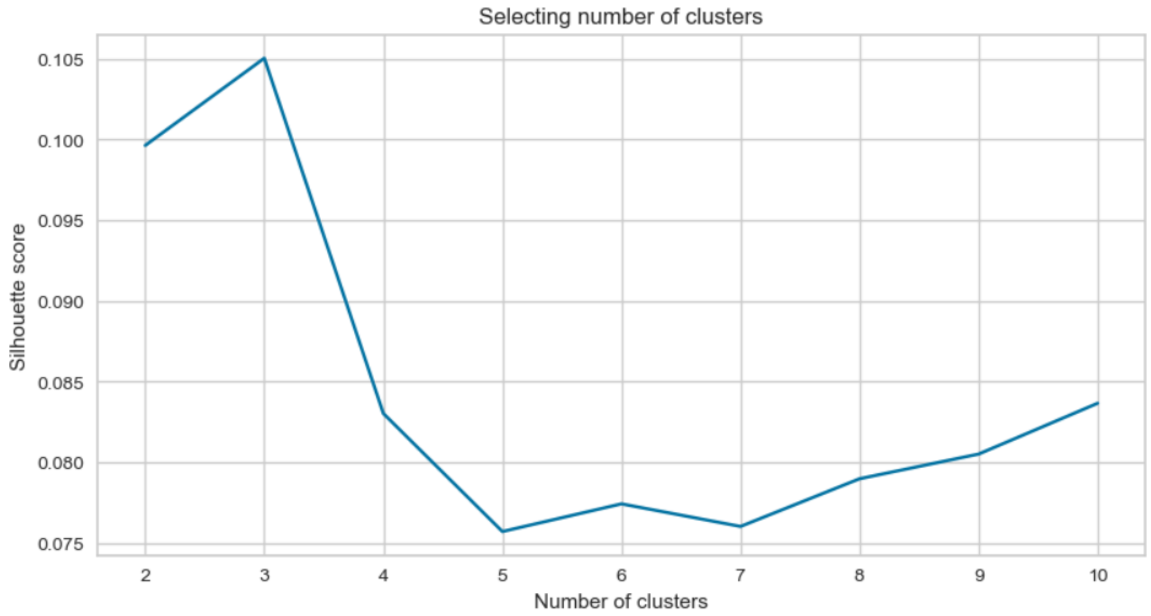


Figure 11 - Tuning the number of clusters

On the picture below you can observe the visualized sample-wise silhouette score for k=3.

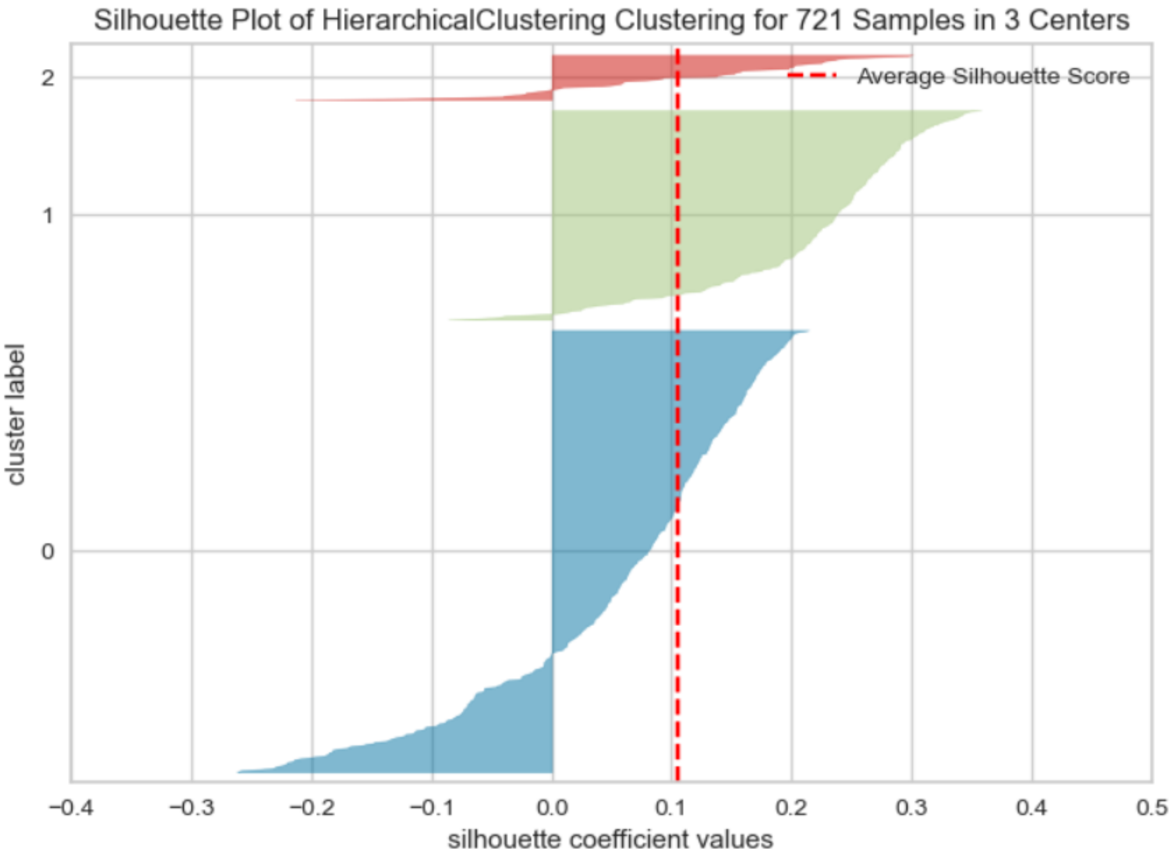
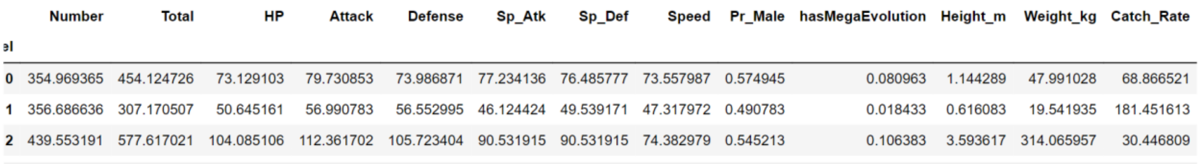


Figure 12 - Hierarchical clustering best silhouette score

As a result we end up with 3 groups. To understand better the similarities we have calculated the mean value of the numerical features for each class.



	Number	Total	HP	Attack	Defense	Sp_Atk	Sp_Def	Speed	Pr_Male	hasMegaEvolution	Height_m	Weight_kg	Catch_Rate
el													
0	354.969365	454.124726	73.129103	79.730853	73.986871	77.234136	76.485777	73.557987	0.574945	0.080963	1.144289	47.991028	68.866521
1	356.686636	307.170507	50.645161	56.990783	56.552995	46.124424	49.539171	47.317972	0.490783	0.018433	0.616083	19.541935	181.451613
2	439.553191	577.617021	104.085106	112.361702	105.723404	90.531915	90.531915	74.382979	0.545213	0.106383	3.593617	314.065957	30.446809

Figure 13 - Cluster similarity analysis

From the picture above we conclude that the *label=1* corresponds to the "weak" Pokémons characterized by small value of all battle characteristics and easy-to-catch rate. It is followed by *label=0* with more advanced creatures, whereas *label=2* collects all the most powerful Pokémons.

By applying PCA to the initial dataset with k=2 we can visualize on the 2d plane the obtained Pokémon groups.

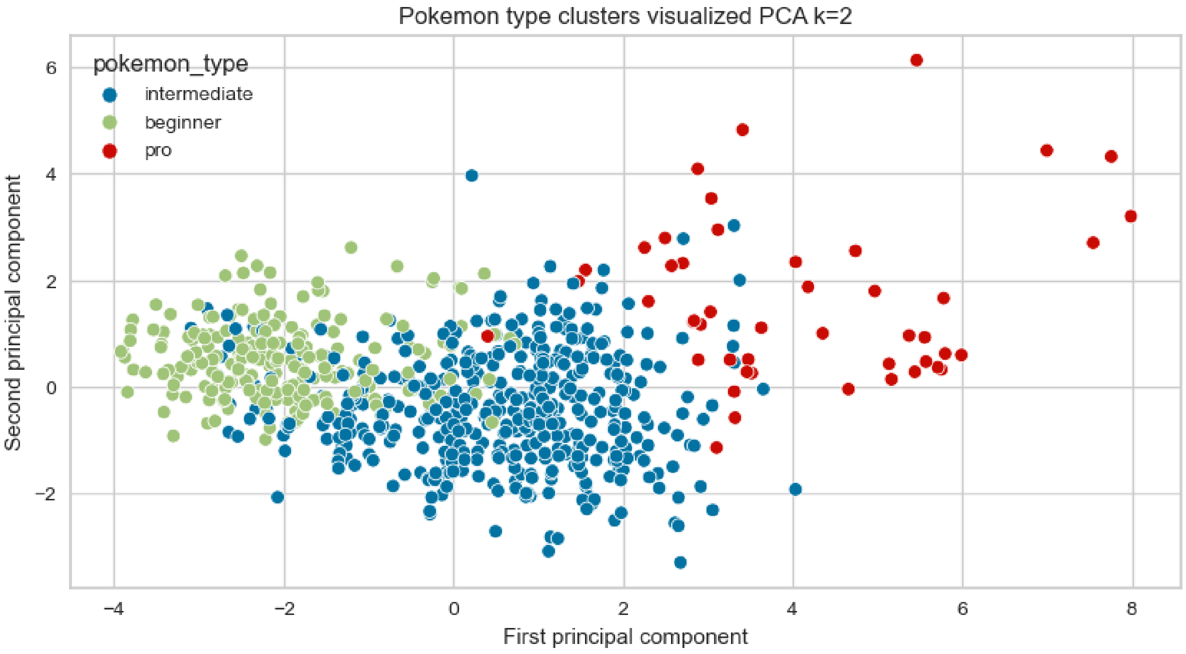


Figure 14 - Pokemon type clusters

Surprisingly the clusters are well-distinguishable even on the 2d plane, we clearly detect all the three resulting groups.

7. Conclusions

In this paper, we explored the application of statistical learning methods to analyse the Pokémon dataset, focusing on classification analysis, clustering, and dimensionality reduction. Through these analyses, we gained valuable insights into the characteristics and patterns within the Pokémon universe.

Overall, this study underscores the significance of statistical learning methods in analysing complex datasets such as Pokémon. The findings presented here contribute to the advancement of Pokémon-related research, gameplay strategies, and educational applications. Furthermore, the methodologies employed in this research can serve as a blueprint for applying statistical learning techniques to other large and intricate datasets in various domains.

As the Pokémon franchise continues to evolve, incorporating statistical learning methods can unlock new avenues for exploration and discovery. By harnessing the power of machine learning and data analysis, we can deepen our understanding of the intricate world of Pokémon, ultimately enhancing gameplay experiences and pushing the boundaries of educational research in the realm of fictional ecosystems.

Appendix

Full list of variables and their description

- ***Type_1***. Primary type of the Pokémon. It is related the nature, with its lifestyle and with the movements it is able to learn for the fighting time. This categorical value can take 18 different values: *Bug, Dark, Dragon, Electric, Fairy, Fighting, Fire, Flying, Ghost, Grass, Ground, Ice, Normal, Poison, Psychic, Rock, Steel*, and *Water*.
- ***Type_2***. Pokémon can have two types, but not all of them do. The possible values this secondary type can take are the same than the variable *Type_1*.
- ***Total***. The sum of all the base battle stats of a Pokémon. It should be a good indicator of the overall strength of a Pokémon. It is the sum of the next six variables. Each of them represents a base battle stat. All the battle stats are continuous yet integer variables, i.e. the number of values they can take is infinite in theory, or just very big in the practice.
- ***HP***. Base health points of the Pokémon. The bigger it is, the longer the Pokémon will be able to stay in a fight before they faint and leave the combat.
- ***Attack***. Base attack of the Pokémon. The bigger it is, the more damage its physical attacks will deal to the enemy Pokémon.
- ***Defense***. Base defense of the Pokémon. The bigger it is, the less damage it will receive when being hit by a physical attack.
- ***Sp_Atk***. Base special attack of the Pokémon. The bigger it is, the more damage its special attacks will deal to the enemy Pokémon.
- ***Sp_Def***. Base special defense of the Pokémon. The bigger it is, the less damage it will receive when being hit by a special attack.
- ***Speed***. Base speed of the Pokémon. The bigger it is, the more times the Pokémon will be able to attack to he enemy.
- ***Generation***. The generation where the Pokémon was released. It is an integer between 1 and 6, so it is a numerical discrete variable. It could let us analyze the development or the growth of the game through the years.
- ***isLegendary***. Boolean indicating whether the Pokémon is legendary or not. Legendary Pokémon tend to be stronger, to have unique abilities, to be really hard to find, and to be even harder to catch.
- ***Color***. Color of the Pokémon according to the Pokédex. The Pokédex distinguishes between ten colors: *Black, Blue, Brown, Green, Grey, Pink, Purple, Red, White*, and *Yellow*.
- ***hasGender***. Boolean indicating the Pokémon can be classified as male or female.

- ***Pr_Male***. In case the Pokémon has Gender, the probability of its being male. The probability of being female is, of course, 1 minus this value. Like Generation, this variable is numerical and discrete, because although it is the probability of the Pokémon to appear as a female or male in the nature, it can only take 7 values: *0, 0.125, 0.25, 0.5, 0.75, 0.875,* and *1*.
- ***Egg_Group_1***. Categorical value indicating the egg group of the Pokémon. It is related with the race of the Pokémon, and it is a determinant factor in the breeding of the Pokémon. Its 15 possible values are: *Amorphous, Bug, Ditto, Dragon, Fairy, Field, Flying, Grass, Human-Like, Mineral, Monster, Undiscovered, Water_1, Water_2,* and *Water_3*.
- ***Egg_Group_2.*** Similarly to the case of the Pokémon types, Pokémon can belong to two egg groups.
- ***hasMegaEvolution***. Boolean indicating whether a Pokémon can mega-evolve or not. Mega-evolving is property that some Pokémon have and allows them to change their appearance, types, and stats during a combat into a much stronger form.
- ***Height_m.*** Height of the Pokémon according to the Pokédex, measured in meters. It is a numerical continuous variable.
- ***Weight_kg.*** Weight of the Pokémon according to the Pokédex, measured kilograms. It is also a numerical continuous variable.
- ***Catch_Rate***. Numerical variable indicating how easy is to catch a Pokémon when trying to capture it to make it part of your team. It is bounded between 3 and 255. The number of different values it takes is not too high notwithstanding, we can consider it is a continuous variable.
- ***Body_Style***. Body style of the Pokémon according to the Pokédex. 14 categories of body style are specified: *bipedal_tailed, bipedal_tailless, four_wings, head_arms, head_base, head_legs, head_only, insectoid, multiple_bodies, quadruped, serpentine_body, several_limbs, two_wings,* and *with_fins*.