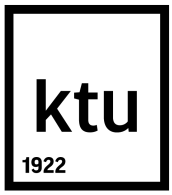



# Kaunas University of Technology

## *Computational Intelligence and Decision Making*

Individual project:

# Legendary Pokémon classification

Applicant:

**Erika Gardini**

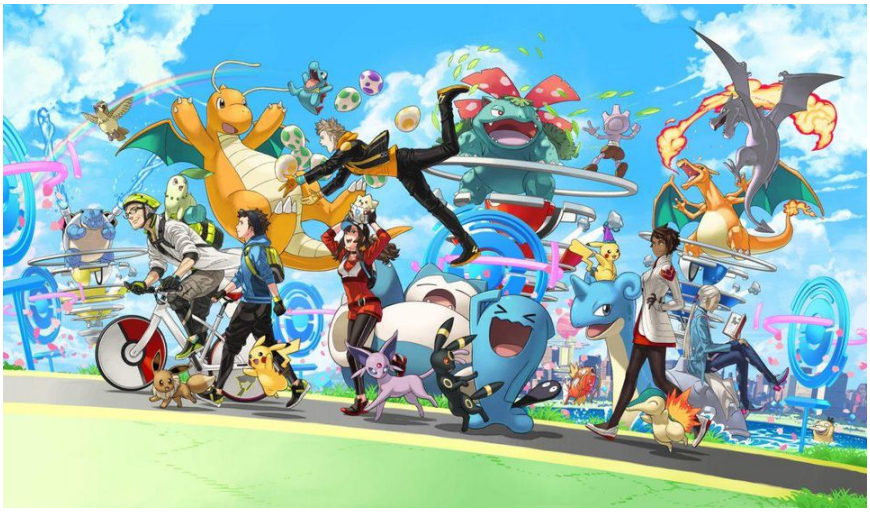
# Table of contents

# 1. Introduction

The aim of this laboratory work is to try different classification techniques in Machine Learning context. To achieve this purpose, it is necessary to select a dataset, to analyze the content, to train a model using part of the information stored and to test it using the remaining part.

The dataset selected is called Pokémon dataset and contains some information about the characteristic of each Pokémon.

Pokémon is a media franchise managed by The Pokémon Company, a Japanese consortium between Nintendo, Game Freak, and Creatures. It is centered on fictional creatures called “Pokémon”, which humans, known as Pokémon Trainers, catch and train to battle each other for sport.



The reason of this choice is to do experiment using amusing data and information without restricting the possibility to obtain significant results in the Machine Learning areas.

## 1.1 Situational Fluency

To achieve the aim of the final task it is important to have a basic gasp of the work and improve the knowledge about the application domain in order to complete the project successfully.

What follow is a summary of the important thing learned about the Pokémon.

The word Pokémon is Macedonian and means “pocket animals”. Each of them has a name (that could be in American or Japanese version) a gender, a type and different shape, dimension, height and weight. It also born, grow, reproduce it selves and die. In the Pokémon is really important the concept of “coach”. In fact, the aim of the game is to catch the Pokémon, train them and use them to challenge other coach and their Pokémon. With respect to this aspect, each Pokémon has a capture rate and also different characteristics: abilities, base happiness, a different amount of damage taken against an attack of a particular type, base HP, base attack, base defense, special attack, special defense and speed. Each of this characteristic can be trained and improved.

Finally, each Pokémon can be identified as common Pokémon or legendary Pokémon. This characteristic depends on how much is simple to catch it.

## 2. Data understanding

After an accurate comprehension of the application domain, it is important to define the prediction subject. In this case, the task is to classify Pokémon in legendary or not, so one Pokémon will be the prediction subject.

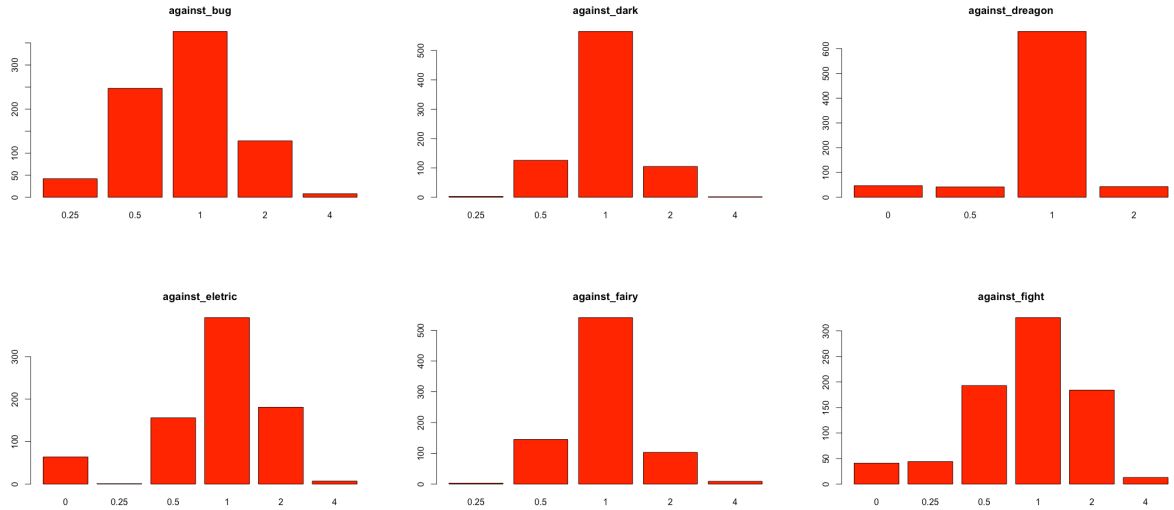
The structure of the dataset required would contain one row per Pokémon, and each row would include a set of descriptive features describing the characteristics of that Pokémon, and a target feature indicating if the Pokémon is legendary of not.

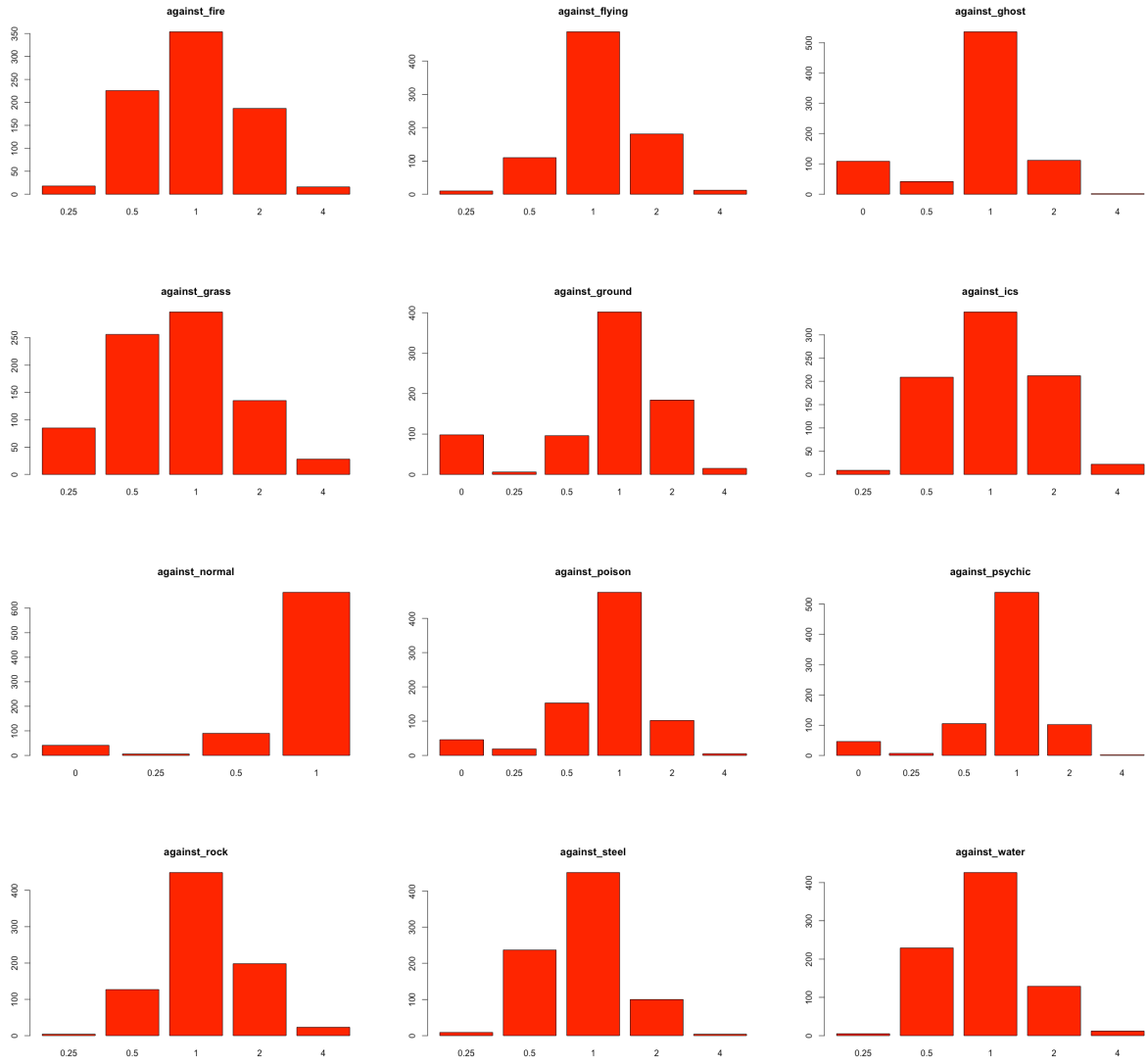
What follow is an accurate description of the dataset available:

| **Number of rows** | 801 |
|---|---|
| **Number of columns** | 41 |
| **Feature** | **Type** |
| abilities | Textual |
| against_# (18 features) | Categorical |
| attack | Numerical |
| base_egg_steps | Numerical |
| base_happiness | Numerical |
| base_total | Numerical |
| capture_rate | Numerical |
| classification | Textual |
| defense | Numerical |
| experience_growth | Numerical |
| height_m | Numerical |

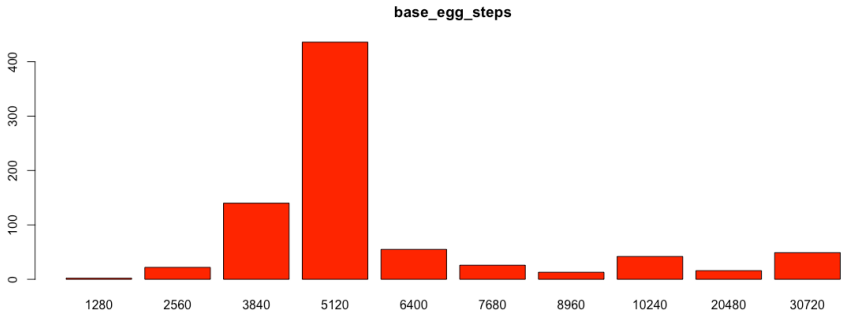
| | |
|---|---|
| hp | Numerical |
| japanese_name | Textual |
| name | Textual |
| percetange_male | Numerical |
| pokedex_number | Numerical |
| sp_attack | Numerical |
| pp_defense | Numerical |
| speed | Numerical |
| type1 | Categorical |
| type2 | Categorical |
| weight_kg | Numerical |
| generation | Numerical |
| is_legendary | Binary |

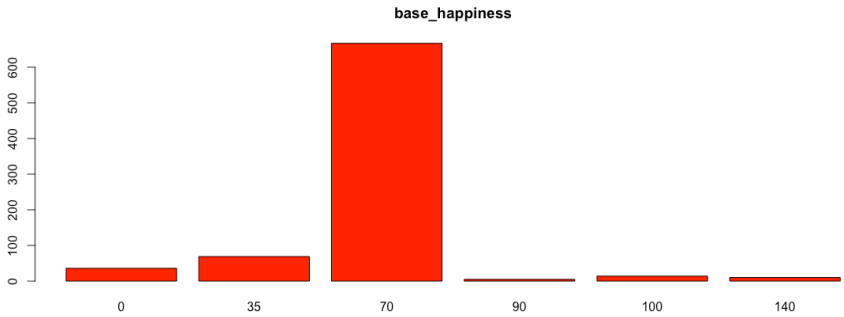
What follows is the graphical representation of the numerical, categorical and binary features:

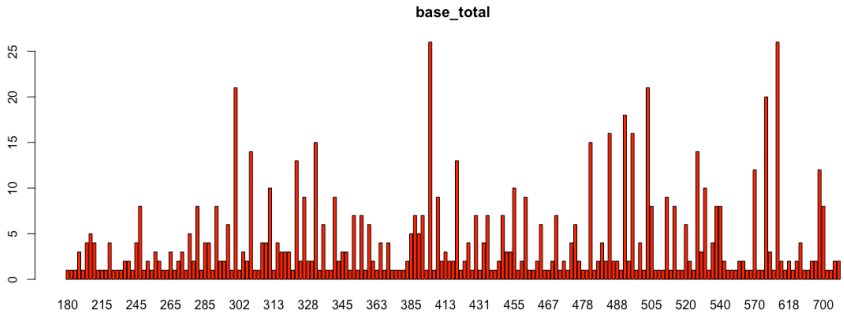


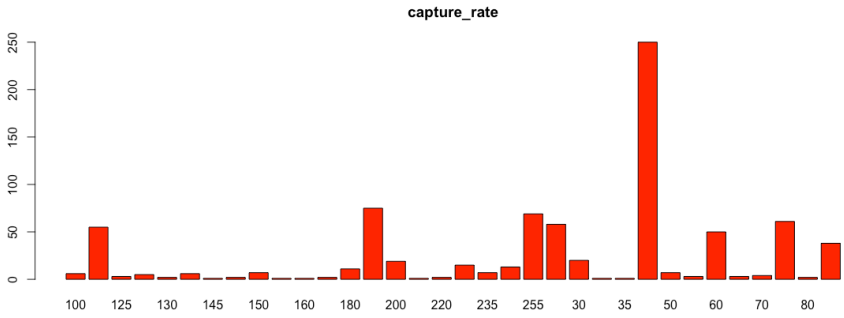


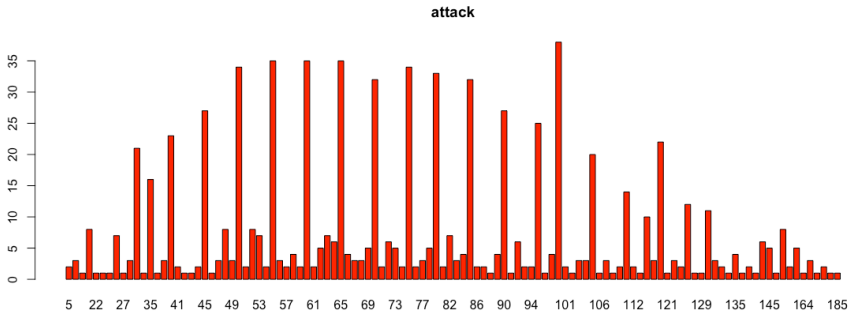

At this step, the main observation is that the different against values are not evenly distributed across the possible values. Instead, the “1” value is much more heavily represented than the other in most of the images.

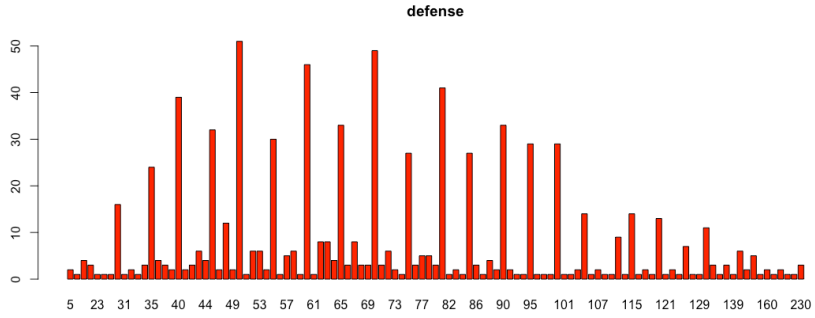

base_egg_steps
0
100
200
300
400
1280
2560
3840
5120
6400
7680
8960
10240
20480
30720

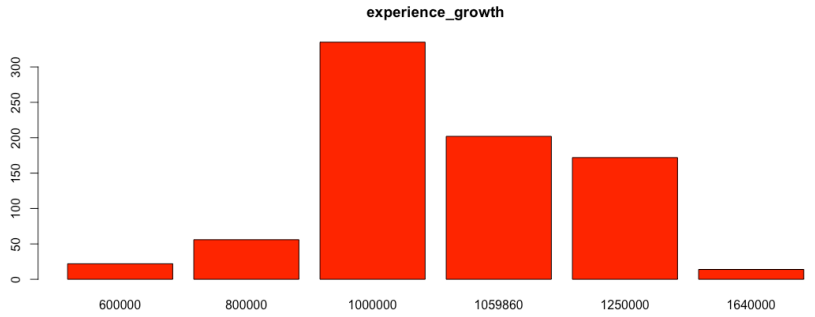

base_happiness
0
100
200
300
400
500
600
0
35
70
90
100
140

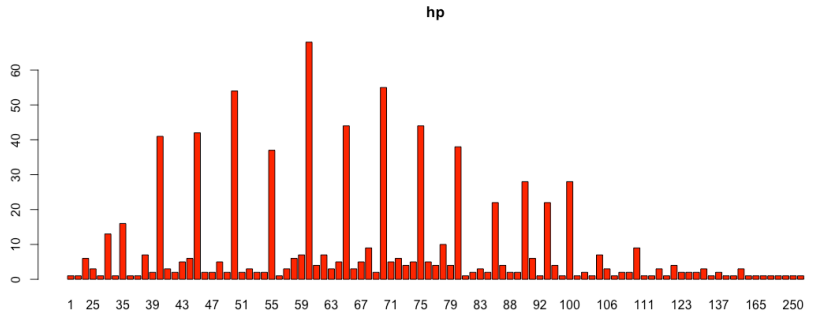

base_total
0
5
10
15
20
25
180 215 245 265 285 302 313 328 345 363 385 413 431 455 467 478 488 505 520 540 570 618 700

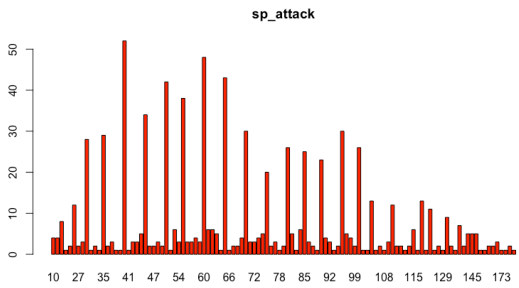

capture_rate
0
50
100
150
200
250
100 125 130 145 150 160 180 200 220 235 255 30 35 50 60 70 80

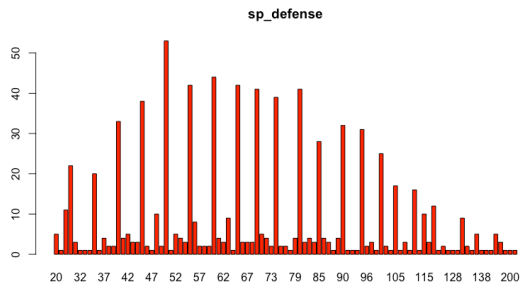

attack
0 5 10 15 20 25 30 35
5 22 27 35 41 45 49 53 57 61 65 69 73 77 82 86 90 94 101 106 112 121 129 135 145 164 185

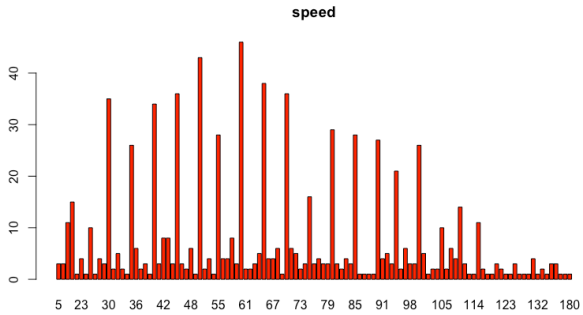

defense
0 10 20 30 40 50
5 23 31 35 40 44 49 53 57 61 65 69 73 77 82 86 90 95 101 107 115 121 129 139 160 230

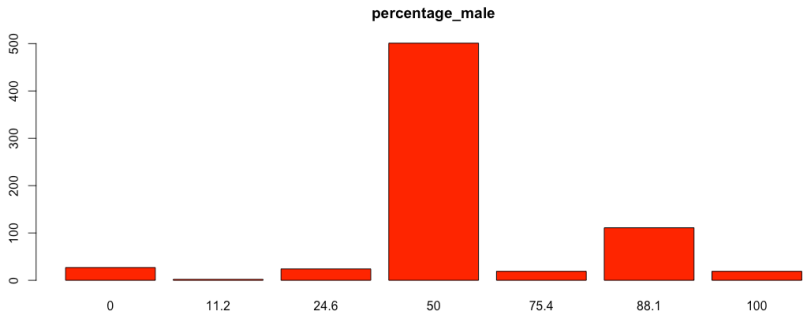

experience_growth
0 50 100 150 200 250 300
600000 800000 1000000 1059860 1250000 1640000

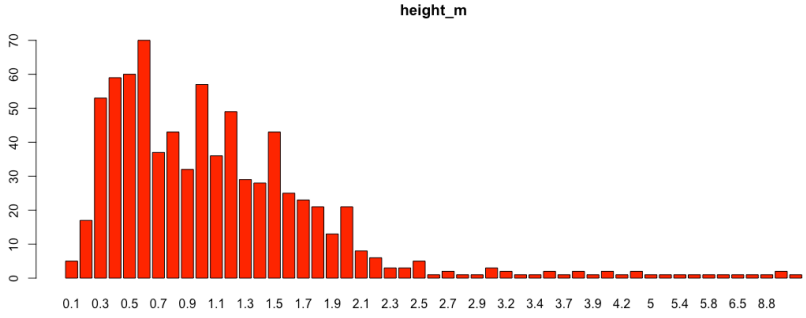

hp
0 10 20 30 40 50 60
1 25 35 39 43 47 51 55 59 63 67 71 75 79 83 88 92 100 106 111 123 137 165 250

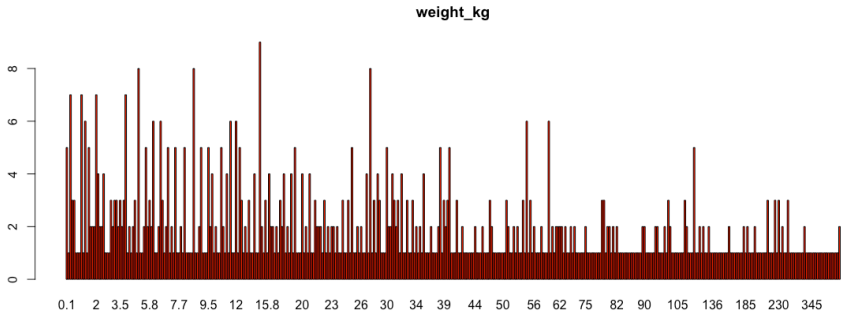

sp_attack
0
10
20
30
40
50
10 27 35 41 47 54 60 66 72 78 85 92 99 108 115 129 145 173

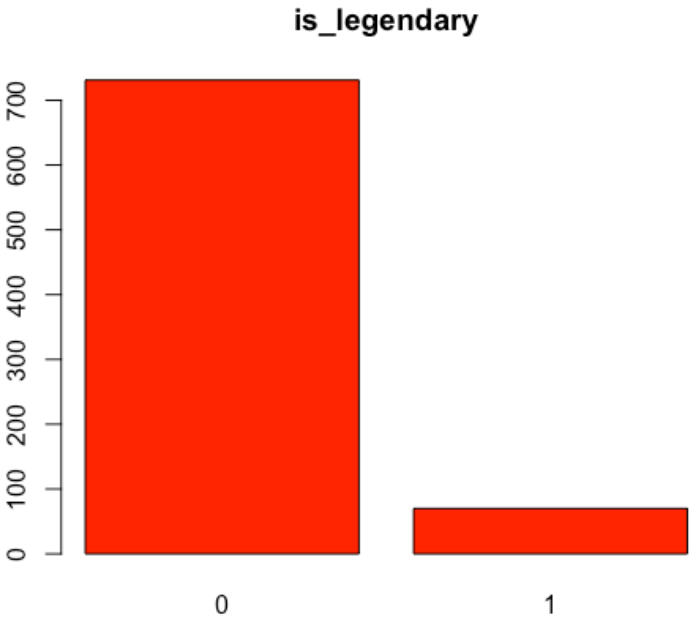

sp_defense
0
10
20
30
40
50
20 32 37 42 47 52 57 62 67 73 79 85 90 96 105 115 128 138 200


speed
0
10
20
30
40
5 23 30 36 42 48 55 61 67 73 79 85 91 98 105 114 123 132 180


percentage_male
0
100
200
300
400
500
0 11.2 24.6 50 75.4 88.1 100


height_m
0
10
20
30
40
50
60
70
0.1 0.3 0.5 0.7 0.9 1.1 1.3 1.5 1.7 1.9 2.1 2.3 2.5 2.7 2.9 3.2 3.4 3.7 3.9 4.2 5 5.4 5.8 6.5 8.8




This last image is the representation of the target feature “is_legendary”. As can be imagined, the quantity of common Pokémon is predominant; on the contrary, there are few legendary Pokémon.

Some features are not reported graphically since the relative images are not significant.

The following tables summarize the quality of the most significant numerical and categorical data contained into the dataset:

| Feature | Count | % Miss. | Card | Mode | Mode freq | Mode % | 2° Mode | 2° Mode freq | 2° Mode % |
|---|---|---|---|---|---|---|---|---|---|
| against_bug | 801 | 0 | 5 | 1 | 376 | 47% | 0.5 | 247 | 31% |
| against_dark | 801 | 0 | 5 | 1 | 565 | 70% | 0.5 | 126 | 16% |
| against_dragon | 801 | 0 | 4 | 1 | 669 | 83% | 0 | 47 | 6% |
| against_electric | 801 | 0 | 6 | 1 | 392 | 49% | 2 | 181 | 23% |
| against_fairy | 801 | 0 | 5 | 1 | 541 | 67% | 0.5 | 145 | 18% |
| against_fight | 801 | 0 | 6 | 1 | 326 | 41% | 0.5 | 193 | 24% |
| against_fire | 801 | 0 | 5 | 1 | 354 | 44% | 0.5 | 226 | 28% |
| against_flying | 801 | 0 | 5 | 1 | 488 | 61% | 2 | 181 | 23% |
| against_ghost | 801 | 0 | 5 | 1 | 536 | 67% | 2 | 112 | 14% |
| against_grass | 801 | 0 | 5 | 1 | 297 | 37% | 0.5 | 256 | 32% |
| against_ground | 801 | 0 | 6 | 1 | 402 | 50% | 2 | 184 | 23% |
| against_ice | 801 | 0 | 5 | 1 | 349 | 44% | 2 | 212 | 26% |
| against_normal | 801 | 0 | 4 | 1 | 664 | 83% | 0.5 | 90 | 11% |
| against_poison | 801 | 0 | 6 | 1 | 476 | 59% | 0.5 | 153 | 19% |
| against_psychic | 801 | 0 | 6 | 1 | 539 | 67% | 0.5 | 105 | 13% |
| against_rock | 801 | 0 | 5 | 1 | 449 | 56% | 2 | 198 | 25% |
| against_steel | 801 | 0 | 5 | 1 | 451 | 56% | 0.5 | 237 | 30% |
| against_water | 801 | 0 | 5 | 1 | 426 | 53% | 0.5 | 229 | 29% |
| type1 | 801 | 0 | 18 | water | 114 | 14% | normal | 105 | 13% |
| type2 | 417 | 384 | 19 | flying | 96 | 23% | ground poison | 34 | 8% |

| Feature | Count | % Miss. | Card | Min | 1st Qrt. | Mean | Median | 3rd Qrt | Max | Std. Dev. |
|---|---|---|---|---|---|---|---|---|---|---|
| attack | 801 | 0 | 114 | 5 | 55 | 77.858 | 75 | 100 | 185 | 32.159 |
| base_egg_steps | 801 | 0 | 10 | 1280 | 5120 | 7191.011 | 5120 | 6400 | 30720 | 6558.22 |
| base_happyness | 801 | 0 | 6 | 0 | 70 | 65.361 | 70 | 70 | 140 | 19.599 |
| base_total | 801 | 0 | 203 | 180 | 320 | 428.377 | 435 | 505 | 780 | 428.377 |
| capture_rate | 801 | 0 | 34 | 3 | 45 | 98.675 | 60 | 170 | 255 | 76.249 |
| defense | 801 | 0 | 109 | 5 | 50 | 73.009 | 70 | 90 | 230 | 30.769 |
| experience_growth | 801 | 0 | 6 | 600000 | 1000000 | 1054996 | 1000000 | 1059860 | 1640000 | 160255.8 |
| height_m | 801 | 20 | 52 | 0.1 | 0.6 | 1.124 | 1 | 1.5 | 14.5 | 1.080 |
| hp | 801 | 0 | 99 | 1 | 50 | 68.959 | 65 | 80 | 155 | 26.576 |
| percentage_male | 801 | 98 | 8 | 0 | 50 | 50 | 50 | 50 | 100 | 20.262 |
| sp_attack | 801 | 0 | 111 | 10 | 45 | 71.306 | 65 | 91 | 194 | 32.354 |
| sp_defense | 801 | 0 | 97 | 20 | 50 | 70.911 | 66 | 90 | 230 | 27.908 |
| speed | 801 | 0 | 113 | 5 | 45 | 66.334 | 65 | 85 | 180 | 28.908 |
| weight_kg | 801 | 20 | 442 | 0.1 | 9 | 61.378 | 27.30 | 64.8 | 999.9 | 109.355 |

Considering the table that has been already illustrated, there are few features with missing values. In the next chapters, it will be necessary to identify a strategy to manage them; one possible strategy could be to overlook the features (only if it doesn't contain a big quantity of information, necessary to improve the classification). Another possibility could be to remove all the rows with two or more missing values. Before to compute this strategy, it is necessary to verify that the percentage of row to delete is low and also is homogeneously distributed; this strategy ensures that there isn't relationship between missing values and common/legendary Pokémon.

Finally, an important aspect is the lack of features with cardinality equal to 1. That means that all the features contain information and could be relevant.

## 3. Data preparation

In this chapter, the data contained into the Pokémon dataset will be processed and transformed to handle data quality issues.

The first amendment of the dataset regards the feature “abilities”. In fact, this feature is textual and it could not be used for the classification. From this feature has been obtained a new derived feature called “num_abilities”; it represents the number of abilities of each Pokémon.

Another derived feature has been obtained calculating the mean of the features against; it is called “against_mean”.

In addition, after the analysis of the features, some of them has been removed since they don’t contain information useful for the identification of the legendary Pokémon; these features are: pokedex_number, Japanese_name, classification, type2, height_m, weight_kg, percentage_male and generation.

Finally, the content of the categorical features “type1” has been converted from string to number (for example: grass = 1, fire = 2, water = 3, …).

These new restricted and derived features form the ABT. What follow are the data quality reports of the ABT:

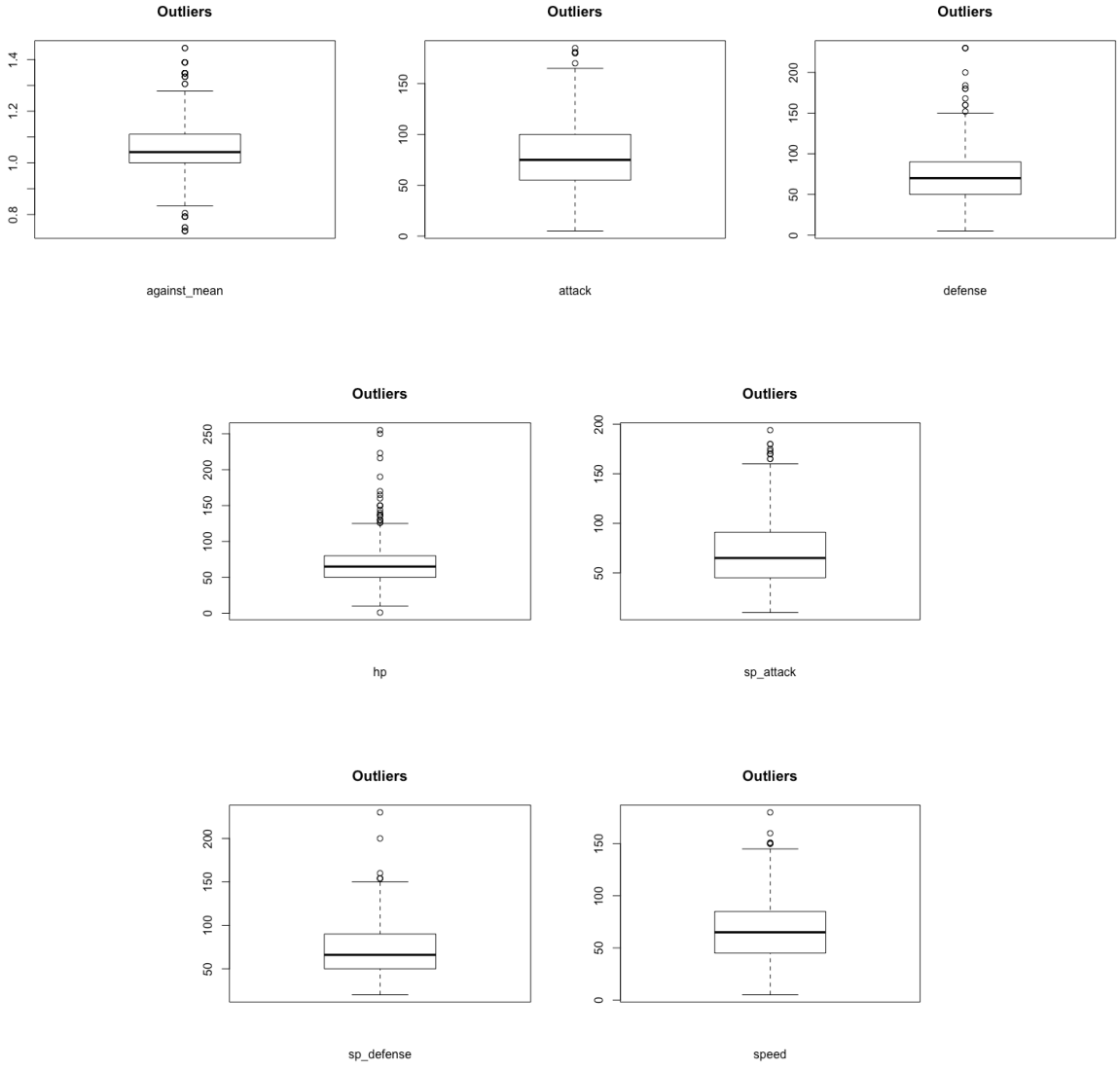
| Feature | Count | % Miss. | Card | Mode | Mode freq | Mode % | 2° Mode | 2° Mode freq | 2° Mode % |
|---|---|---|---|---|---|---|---|---|---|
| num_abilities | 801 | 0 | 5 | 3 | 444 | 55% | 2 | 267 | 33% |
| type1 | 801 | 0 | 18 | 18 | 114 | 14% | 13 | 105 | 13% |

| Feature | Count | % Miss. | Card | Min | 1st Qrt. | Mean | Median | 3rd Qrt | Max | Std. Dev. |
|---|---|---|---|---|---|---|---|---|---|---|
| against_mean | 801 | 0 | 38 | 0.7361 | 1 | 1.058 | 1.0417 | 1.111 | 1.444 | 0.106 |
| attack | 801 | 0 | 114 | 5 | 55 | 77.858 | 75 | 100 | 185 | 32.159 |
| base_egg_steps | 801 | 0 | 10 | 1280 | 5120 | 7191.011 | 5120 | 6400 | 30720 | 6558.22 |
| base_happyness | 801 | 0 | 6 | 0 | 70 | 65.361 | 70 | 70 | 140 | 19.599 |
| base_total | 801 | 0 | 203 | 180 | 320 | 428.377 | 435 | 505 | 780 | 428.377 |
| capture_rate | 801 | 0 | 33 | 3 | 45 | 98.675 | 60 | 170 | 255 | 76.249 |
| defense | 801 | 0 | 109 | 5 | 50 | 73.009 | 70 | 90 | 230 | 30.769 |
| experience_growth | 801 | 0 | 6 | 600000 | 1000000 | 1054996 | 1000000 | 1059860 | 1640000 | 160255.8 |
| hp | 801 | 0 | 99 | 1 | 50 | 68.959 | 65 | 80 | 155 | 26.576 |
| sp_attack | 801 | 0 | 111 | 10 | 45 | 71.306 | 65 | 91 | 194 | 32.354 |
| sp_defense | 801 | 0 | 97 | 20 | 50 | 70.911 | 66 | 90 | 230 | 27.908 |
| speed | 801 | 0 | 113 | 5 | 45 | 66.334 | 65 | 85 | 180 | 28.908 |

Analyzing the new quality report it is possible to note that, for some features, is evident the presence of outliers. In fact, the magnitude of the maximum values in comparison to the median and 3rd quartile is often unusual; this is valid for the features “against_mean”, “attack”, “defense”, “hp”, “sp_attack”, “sp_defense” and “speed”.

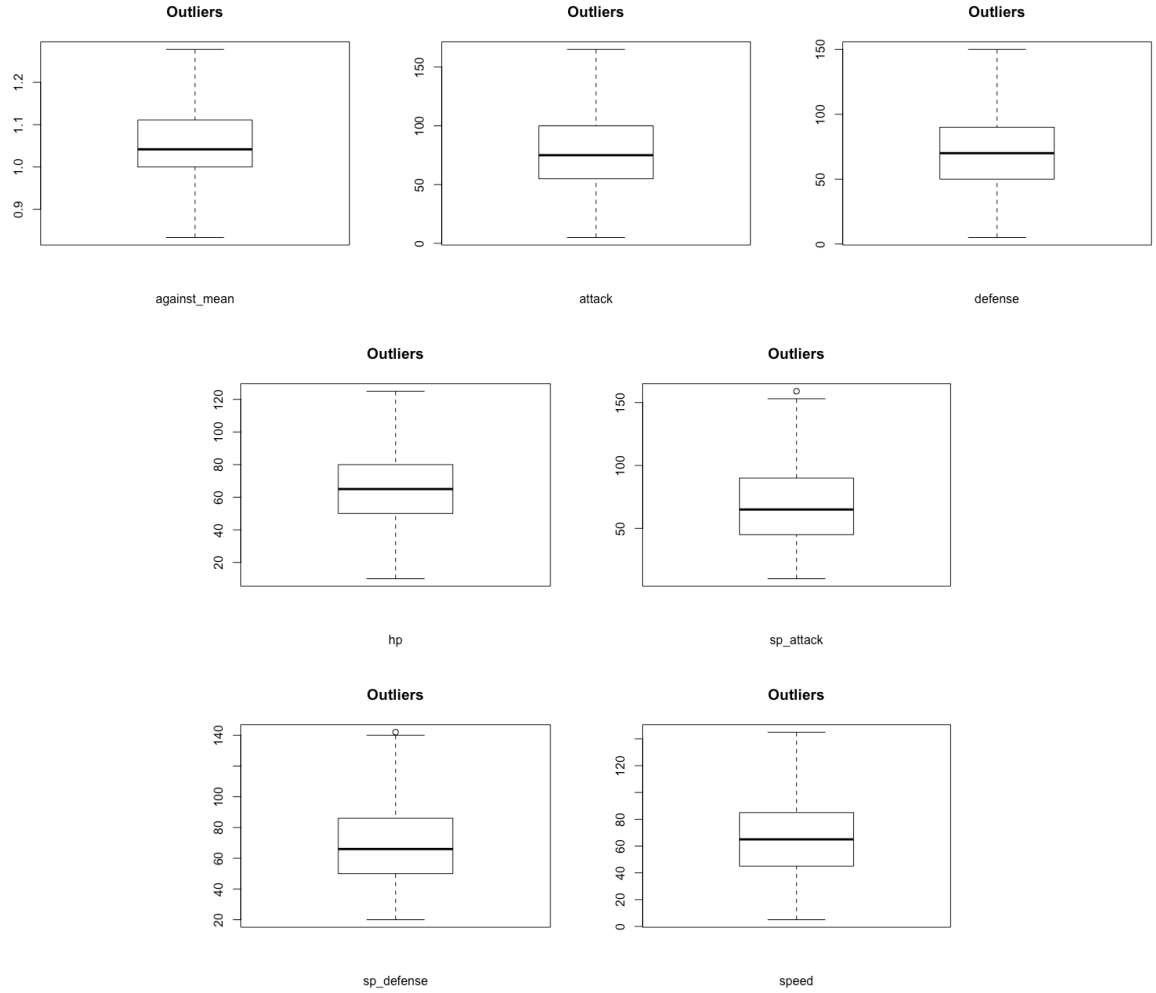
The same thing is true for the magnitude of the minimum values in comparison to the median and the 1st quartile; for example, the feature “against_mean” could contains outliers.

This statement is confirmed also by the boxplot of the mentioned features:




To remove the outliers has been used the clamp transformation; in particular all the values identified as outliers (upper then 3rd quartile value plus 1.5 times the inter-quartile range and lower then 1st quartile value minus 1.5 times the inter-quartile range) are changed with the central value.

The obtained results are:




Finally, normalization has been computed to improve the accuracy of the final predictive model. The only drawback to normalization is that the models become less interpretable; however, in this specific case is not so important to reduce the interpretability.

# 4. Modelling

The descriptive features in the ABT are primarily continuous. For this reason, the k nearest neighbor and the Support Vector Machine has been chosen to create the model. Additionally, the decision tree classification strategy has been considered.

## 4.1 Subdivision of the dataset into train and test set

The ABT has been first of all subdivided into train and test. To achieve this purpose, some row of the dataset has been randomly selected. The training set contains the 90% of the data in the ABT (720 instances) and the test set consisted of the remaining 10% (81 instances).

To verify the quality of this operation, the proportion of the train set and the test set has been verified. What follows are the result obtained:

| Dataset | is_legendary=0 | is_legendary=1 |
|---|---|---|
| Training set | 91.25% | 8.75% |
| Test set | 91.36% | 8.64% |

The table already shows that the information contained into the ABT dataset has been homogeneously distributed into the train set and the test set.

To realize the models, it is not necessary to further modify the available data, since they have been already transformed in numerical and they are all normalized.

## 4.2 Baseline models

The first experiment realized consist of using the full set of descriptive features, create the three different models (trained with the train set) and measure the accuracy using the test set.

### *k* nearest neighbor

To realize this experiment has been used the kNN algorithm with different distances and different values of the parameter k.

The best result obtained is given by k=1 and Euclidean distance, with an overall accuracy of 97.53%.

What follow is the confusion matrix obtained:

| Predicted → Actual ↓ | is_legendary=0 | is_legendary=1 |
|---|---|---|
| **is_legendary = 0** | 74 | 0 |
| **is_legendary = 1** | 2 | 5 |

### Decision trees

To realize this experiment has been used the decision tree classifier. The result obtained has an overall accuracy of 100%.

What follow is the confusion matrix obtained:

| Predicted → Actual ↓ | is_legendary=0 | is_legendary=1 |
|---|---|---|
| **is_legendary = 0** | 74 | 0 |
| **is_legendary = 1** | 0 | 7 |

### Support Vector Machine

To realize this experiment has been used the SVM classifier. The best results obtained is given by kernel = “rbf” (Gaussian distribution), with an overall accuracy of 97.53%.

What follow is the confusion matrix obtained:

| Predicted → Actual ↓ | is_legendary=0 | is_legendary=1 |
|---|---|---|
| **is_legendary = 0** | 73 | 1 |
| **is_legendary = 1** | 1 | 6 |

The initial baseline results are promising; however other actions can be used to improve the accuracy obtained.

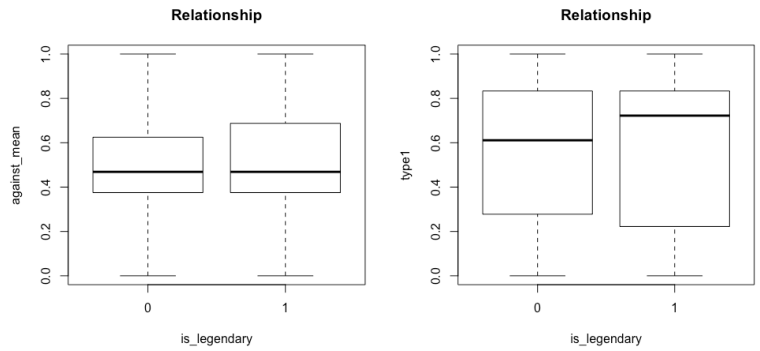
## 4.3 Feature selection

This strategy has been applied to search subsets of features that considered all together improve the performance of the model created with all the features.

To select the best subset of feature, two different approaches has been used:

- calculation of the IG of each feature with respect to the target feature “is_legendary”;
- boxplot analysis, created comparing each feature with the target feature “is_legendary”.

Comparing the results obtained using the two approaches, the features “against_mean” and “type1” are redundant and not useful. In fact, considering the boxplots, is evident that it is impossible deduce information about legendary Pokémon using the feature “against_mean” and “type1”:




This statement is confirmed also by the IG, that is 0.0 for both.

The confusion matrices for the baseline models after feature selection are:

| Predicted → Actual ↓ | is_legendary=0 | is_legendary=1 |
|---|---|---|
| is_legendary = 0 | 74 | 0 |
| is_legendary = 1 | 0 | 7 |

| Predicted → Actual ↓ | is_legendary=0 | is_legendary=1 |
|---|---|---|
| is_legendary = 0 | 74 | 0 |
| is_legendary = 1 | 0 | 7 |

| Predicted → Actual ↓ | is_legendary=0 | is_legendary=1 |
|---|---|---|
| is_legendary = 0 | 73 | 1 |
| is_legendary = 1 | 1 | 6 |

The overall accuracy is respectively, 100 %, 100%, 97,53% for kNN, decision tree and SVM.

As can be seen, the accuracy for the kNN classifier is improved and the accuracy for the SVM and the decision tree classifiers is unchanged. However, the efficiency of the models is improved since the number of the features is reduced.

# 5. Conclusion

At the end of the experiment it is possible to claim that with a restricted knowledge of the Pokémon characteristics it is possible to identify if it is legendary or not.

Considering the obtained results, the kNN and decision tree approaches are the best one, with an overall accuracy of 100%. But the SVM approach gives good results, comparable with the others even if worst.

Surfing on the kaggle web site, has been possible to analyze the actual results and compare it with the results obtained into this experiment.

In this experiment has been used different data process and has been also obtained different derived feature. Other Machine Learning techniques has been used. In fact, in this experiment are used KNN and decision tree models; on the contrary, in the previous experiment has been used the random forest. The SVM approach has been used in booth the experiments (this one and the one cited).

Comparing the results has been possible to say that the actual results are better than the one obtained. The following table summarize the previous and the actual results:

| | **Actual experiment** | **Previous experiment** |
|---|---|---|
| **KNN** | 100% | - |
| **Decision tree** | 100% | - |
| **Random forest** | - | 99.57% |
| **SVM** | 97.53% | 92.73% |

In the end, it is possible to confirm that the strategies used are meaningful and they brought improved results.