P8106 Midterm Project

Phoebe Mo(km3624)

Mar 26, 2021

1. INTRODUCTION

1.1 Motivation and Objective

In this project, I use a dataset that contains different attributes and catch rates for various pokemons to try to understand the relationship between these attributes and catch rates. Here are some questions I want to anwser: Which predictor(s) play important roles in predicting catch rates? Which type of model (linear or non-linear) serves as a better method to predict the catch rate?

1.2 Data Preparation and Cleaning

The original dataset has 20 predictors such as HP, and the outcome 'catch_rate'. After cleaning the names of these variables, I did the following steps to clean the data:

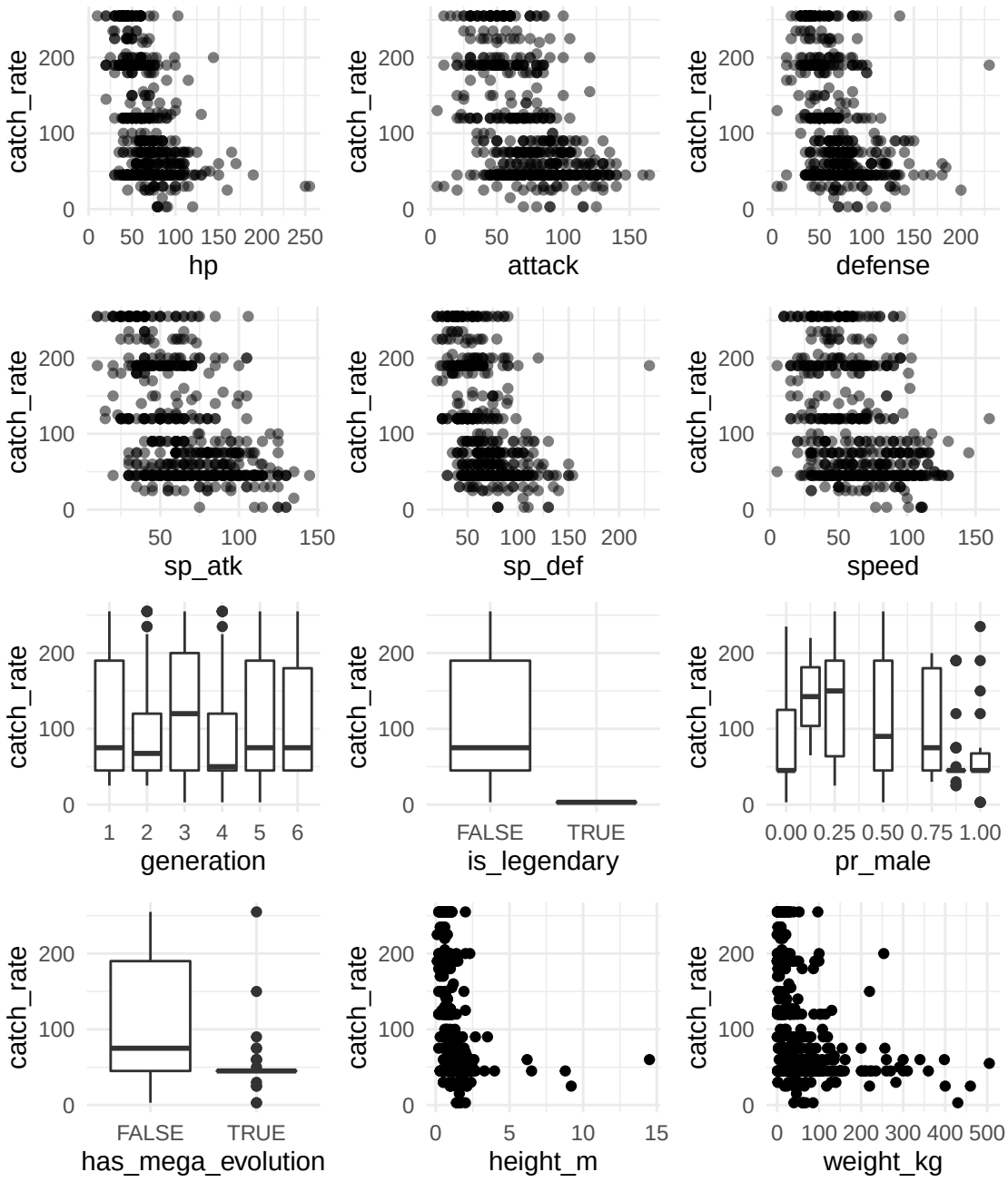
1.2.1 Notice that 'type_2' and 'egg_group_2' are indicators of if a pokemon has a second type or belongs to a second egg group. The 'Null' values in the data means the pokemon does not has the second type/group, so I changed them into "none" to make them as a category to be meaningful;

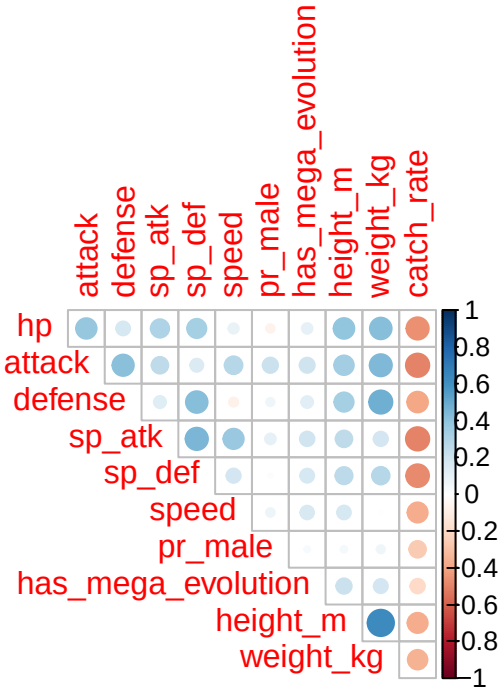
1.2.2 The 'generation' predictor is originally a numeric type, but it has only 6 integer values, so I decided to mutate it to be categorical;

1.2.3. Irrelavant variables 'number' and 'name' are dropped. 'total', which is the total base battle statistic for each pokemon, is calculated and reflected in other battle attributes such as 'hp' and 'attack'. So I dropped it to avoid intercollinearity. Later, after plotting the correlation map, I found 'weight_kg' and 'height_m' has relatively high correlation(correlation plot is shown in section 2). After consideration, I chose to drop 'height_m' since it may has less effect on catch rate compared to 'weight_kg'. 'has_gender' is also dropped because all values are "TRUE".

2. Exploratory Analysis / Visualization

After plotting scatter plots for continuous predictors and boxplots for categorical predictors, significant trends were not observed in different types, egg groups, body_styles or colors. However, the battle attributes: 'hp', 'attack', 'defense', 'sp_atk', 'sp_def', and 'speed' seem to have a negative association with catch rate. The 'weight_kg' also seens to have negative association with catch rate. And the pokemons that are legendary have significantly lower catch rates. Below are some selected visualizations:


catch_rate
hp
attack
defense
sp_atk
sp_def
speed
generation
is_legendary
FALSE
TRUE
pr_male
has_mega_evolution
height_m
weight_kg



3. Models

After the data cleaning procedure, there will be totally 17 predictors included in the models. Since the final dataset contains about 644 observations, I decided to separate it into 75% training data and 25% test data. Repeated 10-fold cross-validation is selected to do model training and select the best tuning parameter (which has the lowest RMSE given by repeated cv) for the following models in caret function.

3.1 Multiple Linear Regression

First, I fit a multiple linear regression to see if any interesting significant associations can be observed. Using this model, we assume the outcome is following normal distribution, which is not perfect for our data. However, since our main objective is to do predictions, we can ignore this temporarily. It turns out that the fitted model has Adjusted R-squared = 0.5767 which is not too bad. The model shows that 'hp', 'attack', 'defense', 'sp_atk', 'sp_def', 'speed', 'pr_male' play important roles in predicting catch rate. Especially, 'hp', 'defense', 'sp_atk', 'pr_male' and 'speed' have very small p-value.

3.2 Lasso, Ridge, and Elastic Net

Then, I use shrinkage methods including lasso, ridge, and elastic net to make penalization to shrink the coefficients as lambda becomes larger. They mainly focus on fitting a linear model and reducing the RSS. Although ridge regression shrinks the coefficients, it will still remain all the predictors in the result. Conversely, lasso regression can help us do the variable selection. In fact, although the dummy variable 'type_2Water' is labeled as significant in the MLR, it is eliminated by the lasso model. Elastic Net regression combines the penalties from both lasso and ridge methods. In all these three models, data is scaled and centered to ensure fairness of shrinking.

The tuning parameters lambda is chosen by the caret function after we specify a range. The one that has the minimum RMSE is selected.

Overall, the shrinkage methods give us similar direction of association between the predictors and catch rate. And they seem to have a larger coefficients for the battle attributes(e.g. hp) and smaller coefficients for the remaining predictors.

3.3 GAM model

Based on the visualization plots, we can see that some continuous variables have curve-like trend in the right tail. In this case, I consider using GAM model in order to include this trend into the analysis. For such predictors, I allow GAM to make it non-parametric smoothing term. GCV is used to choose the degree of freedom for the model.

After fitting, it is observed that the smooth terms of 'hp', 'attack', 'defense', 'sp_atk', 'speed', and the coefficients of 'type_1Poison', 'type_2Water', 'generation3', 'pr_male', 'egg_group_1Undiscovered', 'body_stylemultiple_bodies', are significant. These result greatly coincides with the ones given by previous model.

3.4 Model Comparisons

As can see from the following first table, the predictors that are significant in the MLR model are also found to have a relatively large associations in all other models. Results given by GAM are not included, but GAM shows similar pattern, except that it also include some other significant dummy variables.

From the second RMSE table, I select the mean RMSE as a standard to select the model. GAM has the smallest training RMSE = 41.510, which means it did the best fitting to the training data. This is reasonable since we have observed some nonlinear trends in some of the continuous variables, so fitting GAM may be a good choice. Therefore, GAM will be chosen to predict the data. For the test RMSE, lasso has the best performance of 47.772 while GAM is a little bit behind. For both train RMSE and test RMSE, the MLR model has the worst performances.

Table 1: Coefficients of Predictors for each Model

	lm_estimate	lm_pvalue	ridge_coef	lasso_coef	enet_coef
(Intercept)	222.544	0.001	107.064	107.064	107.064
type_1Poison	73.035	0.036	5.480	4.313	4.484
type_2Water	77.767	0.048	1.992	0.000	0.000
hp	-0.829	0.000	-14.489	-16.157	-16.268
attack	-0.402	0.002	-11.851	-12.347	-12.342
defense	-0.550	0.000	-9.728	-12.047	-11.986
sp_atk	-0.592	0.000	-12.214	-12.971	-12.911
sp_def	-0.321	0.021	-9.698	-9.346	-9.320
speed	-0.724	0.000	-14.149	-16.248	-16.456
generation3	18.496	0.032	4.391	1.325	1.445
pr_male	-46.909	0.001	-7.418	-8.563	-8.615
body_styleinsectoid	35.563	0.037	5.396	2.912	3.124

Table 2: Train and Test RMSE for each Model

	lm	ridge	lasso	enet	GAM
train RMSE	54.543	51.500	50.419	50.422	41.510
test RMSE	49.260	48.011	47.772	47.783	48.414

4. Conclusions

For the model selection, since GAM has the smallest training RMSE, GAM has been selected to do future catch rate prediction.

The most important predictors given by MLR and shrinkage methods model are: 'hp', 'attack', 'defense',

'sp_atk', 'sp_def', 'speed', 'pr_male', and they all have a negative association with catch_rate. There are also several significant dummy variables from some categorical predictors, but since they may be just one type of, for example, body style or pokemon type, the whole categorical predictor itself may not be seemed as important predictor for catch rate.

This result is greatly coincide with my expectation. Before doing the training, I expect the battle attributes may be important predictors for catch rate, because it is reasonable that the pokemon which has better battle attributes should be harder to catch. From the result, it can be seen that this expectation is confirmed. However, I did not expect the percent of male('pr_male') will also be a significant predictor of catch rate, which is surprising.

5. Limitations

5.1 Although lasso, ridge, and elastic net method still provide us a prospective of how each important predictor associates with catch rate, it is relatively hard to interpret the coefficients given by them.

5.2 Since our number of observations is not small compared to the number of predictors and the correlation between each pair of predictors is not that large, the use of shrinkage methods may not be a very good fit;

5.3 The GAM model does not truly select the tuning parameter and do the model training, therefore this may be the limitation in the GAM model. However, since it has the smallest training RMSE, it is still chosen as our best method to do the prediction in this project.