p8106_midterm_wk2343

Gavin Ko

4/4/2020

1. Introduction

Brief about Pokemon and Interested Question

Pokemon is a well desinged video game characters that have detailed numeric settings about their characteristics. Among all pokemons, some of them are classfied as "legendary". In this project, we wish to predict whether a pokemon is legendary by the properties of that specific pokemon.

Data cleaning process

As characters in well designed video game, we assume that there's already tidy and organized statistics for each pokemon. Thankfully, we do find a dataset recording detailed information of all pokemons from generation 1 to 6 on kaggle (source: https://www.kaggle.com/alopez247/pokemon).

After downloading and reading in the dataset, Our dataset have 721 pokemons' data listed with 23 variables. These variables include their name, basic information like body weight, height and battle related information like attack, defense.

After a quick look on the dataset, we find that `total`, which indicates *species strength*, is simply the sum of `hp`, `attack`, `defense`, `sp_atk`, `sp_def` and `speed`. Also, to aviod collinearity issue, we would forfeit these six variables and keep `total` for further analysis. On the other hand, `type`, `color`, `egg_group` and `body_style` are classified as characters but should be factors instead. Therefore, we need to factorize them.

2. EDA

Table 1 for grouped data

Since we're mainly interested in whether a pokemon is legendary or not, we can summarize the dataset grouped by legendary status.

	Normal (N = 675)	Legendary (N = 46)
type_1		
- Bug	63 (9.3%)	0 (0.0%)
- Dark	26 (3.9%)	2 (4.3%)
- Dragon	17 (2.5%)	7 (15.2%)
- Electric	33 (4.9%)	3 (6.5%)
- Fairy	16 (2.4%)	1 (2.2%)
- Fighting	25 (3.7%)	0 (0.0%)
- Fire	42 (6.2%)	5 (10.9%)
- Flying	2 (0.3%)	1 (2.2%)
- Ghost	22 (3.3%)	1 (2.2%)
- Grass	64 (9.5%)	2 (4.3%)
- Ground	28 (4.1%)	2 (4.3%)

	Normal (N = 675)	Legendary (N = 46)
- Ice	21 (3.1%)	2 (4.3%)
- Normal	91 (13.5%)	2 (4.3%)
- Poison	28 (4.1%)	0 (0.0%)
- Psychic	39 (5.8%)	8 (17.4%)
- Rock	38 (5.6%)	3 (6.5%)
- Steel	18 (2.7%)	4 (8.7%)
- Water	102 (15.1%)	3 (6.5%)
type_2		
-	352 (52.1%)	19 (41.3%)
- Bug	3 (0.4%)	0 (0.0%)
- Dark	16 (2.4%)	0 (0.0%)
- Dragon	11 (1.6%)	3 (6.5%)
- Electric	5 (0.7%)	1 (2.2%)
- Fairy	17 (2.5%)	1 (2.2%)
- Fighting	16 (2.4%)	3 (6.5%)
- Fire	7 (1.0%)	2 (4.3%)
- Flying	78 (11.6%)	9 (19.6%)
- Ghost	11 (1.6%)	1 (2.2%)
- Grass	18 (2.7%)	0 (0.0%)
- Ground	29 (4.3%)	1 (2.2%)
- Ice	9 (1.3%)	1 (2.2%)
- Normal	4 (0.6%)	0 (0.0%)
- Poison	31 (4.6%)	0 (0.0%)
- Psychic	24 (3.6%)	3 (6.5%)
- Rock	14 (2.1%)	0 (0.0%)
- Steel	18 (2.7%)	1 (2.2%)
- Water	12 (1.8%)	1 (2.2%)
total		
- Mean (SD)	404.16 (98.64)	620.22 (44.99)
- Median (IQR)	410.00 (316.00, 490.00)	600.00 (580.00, 677.50)
generation		
- 1	147 (21.8%)	4 (8.7%)
- 2	95 (14.1%)	5 (10.9%)
- 3	125 (18.5%)	10 (21.7%)
- 4	96 (14.2%)	11 (23.9%)
- 5	146 (21.6%)	10 (21.7%)
- 6	66 (9.8%)	6 (13.0%)
color		
- Black	29 (4.3%)	3 (6.5%)
- Blue	125 (18.5%)	9 (19.6%)
- Brown	105 (15.6%)	5 (10.9%)
- Green	74 (11.0%)	5 (10.9%)
- Grey	65 (9.6%)	4 (8.7%)
- Pink	39 (5.8%)	2 (4.3%)
- Purple	62 (9.2%)	3 (6.5%)
- Red	70 (10.4%)	5 (10.9%)
- White	48 (7.1%)	4 (8.7%)
- Yellow	58 (8.6%)	6 (13.0%)
has_gender		
- False	37 (5.5%)	40 (87.0%)
- True	638 (94.5%)	6 (13.0%)
pr_male		

	Normal (N = 675)	Legendary (N = 46)
- Mean (SD)	0.55 (0.20)	0.75 (0.42)
- Median (IQR)	0.50 (0.50, 0.50)	1.00 (0.62, 1.00)
egg_group_1		
- Amorphous	41 (6.1%)	0 (0.0%)
- Bug	66 (9.8%)	0 (0.0%)
- Ditto	1 (0.1%)	0 (0.0%)
- Dragon	10 (1.5%)	0 (0.0%)
- Fairy	30 (4.4%)	0 (0.0%)
- Field	169 (25.0%)	0 (0.0%)
- Flying	44 (6.5%)	0 (0.0%)
- Grass	27 (4.0%)	0 (0.0%)
- Human-Like	37 (5.5%)	0 (0.0%)
- Mineral	46 (6.8%)	0 (0.0%)
- Monster	74 (11.0%)	0 (0.0%)
- Undiscovered	27 (4.0%)	46 (100.0%)
- Water_1	74 (11.0%)	0 (0.0%)
- Water_2	15 (2.2%)	0 (0.0%)
- Water_3	14 (2.1%)	0 (0.0%)
egg_group_2		
-	484 (71.7%)	46 (100.0%)
- Amorphous	8 (1.2%)	0 (0.0%)
- Bug	2 (0.3%)	0 (0.0%)
- Dragon	35 (5.2%)	0 (0.0%)
- Fairy	17 (2.5%)	0 (0.0%)
- Field	31 (4.6%)	0 (0.0%)
- Flying	6 (0.9%)	0 (0.0%)
- Grass	32 (4.7%)	0 (0.0%)
- Human-Like	15 (2.2%)	0 (0.0%)
- Mineral	8 (1.2%)	0 (0.0%)
- Monster	1 (0.1%)	0 (0.0%)
- Water_1	13 (1.9%)	0 (0.0%)
- Water_2	8 (1.2%)	0 (0.0%)
- Water_3	15 (2.2%)	0 (0.0%)
has_mega_evolution		
- False	634 (93.9%)	41 (89.1%)
- True	41 (6.1%)	5 (10.9%)
height_m		
- Mean (SD)	1.06 (0.92)	2.45 (1.72)
- Median (IQR)	0.89 (0.61, 1.40)	1.96 (1.50, 3.20)
weight_kg		
- Mean (SD)	46.89 (65.96)	201.80 (197.17)
- Median (IQR)	25.50 (9.00, 55.50)	196.50 (56.55, 293.75)
catch_rate		
- Mean (SD)	106.63 (74.94)	6.65 (11.97)
- Median (IQR)	75.00 (45.00, 190.00)	3.00 (3.00, 3.00)
body_style		
- bipedal_tailed	150 (22.2%)	8 (17.4%)
- bipedal_tailless	101 (15.0%)	8 (17.4%)
- four_wings	18 (2.7%)	0 (0.0%)
- head_arms	35 (5.2%)	4 (8.7%)
- head_base	30 (4.4%)	0 (0.0%)
- head_legs	17 (2.5%)	0 (0.0%)

	Normal (N = 675)	Legendary (N = 46)
- head_only	33 (4.9%)	1 (2.2%)
- insectoid	30 (4.4%)	0 (0.0%)
- multiple_bodies	15 (2.2%)	0 (0.0%)
- quadruped	123 (18.2%)	12 (26.1%)
- serpentine_body	26 (3.9%)	3 (6.5%)
- several_limbs	13 (1.9%)	0 (0.0%)
- two_wings	54 (8.0%)	9 (19.6%)
- with_fins	30 (4.4%)	1 (2.2%)

We can find some interesting triats from this grouping summary:

1) There's a total of 46 legendary pokemon among 721 pokemons, which is around 6%.
2) **type**: The most popular type for legendary pokemons are Flying (19.6%), Psychic(17.4%) and Dragon(15.2%).
3) **total(species strength)**: while the mean of normal pokemons are around 400, legendary pokemon seems to have much higher average at 620.
4) **has_gender**: While most normal pokemons do have gender(94.5%), legendary pokemons are the opposite(13.0%). This make discussing it's male proportion(`pr_male`) not proper since the sample size is too small.
5) **egg_group**: For legendary pokemons, we have **ALL** of them with Undiscoverd egg group. Therefore, once we know that a specific pokemon has this kind of egg, they're highly possible to be legendary.
6) **height and weight**: The average height and weight of legendary pokemons (2.45m, 201kg) seem to be much larger than those of normal pokemons' (1.06m, 47kg).
7) **catch rate**: legendary pokemons owns much lower average catch rate (6.65%) compared to those of normal pokemons(> 100%).

For further analysis, we would focus on these variables to build the prediction model.

3. Prediction Model Building

Apparently, `legendary` is a binary status, so we need to build a non-linear classification model.

Logistic Regression

Choosing Predictors

It doesn't seems working with such a large set of categorical varibles. Due to limited knowledge I have, I can only limit the discussion to continuous and binary predictors and try again. As a result, I kept `total`, `has_gender`, `height`, `weight` and `catch_rate` as predictors.

```
set.seed(88)
model.glm <- train(x = pokemon_data_final[rowTrain,c(3,5,8:10)],
                   y = pokemon_data_final$is_legendary[rowTrain],
method = "glm", metric = "ROC", trControl = ctrl)
```

LDA Method

Since our response variable `is_legendary` is a binary outcome, it can be treated as categorical and we can make this porblem a classification question. Under this scenario, we can apply linear discriminant analysis.

```
# Model building
lda.fit <- lda(is_legendary ~ total + has_gender + height_m + weight_kg + catch_rate,
               data = pokemon_data_final, subset = rowTrain)
```

QDA Method

Another approach to classification problems is quadratic discriminant analysis.

```
# Model building
qda.fit <- qda(is_legendary ~ total + has_gender + height_m + weight_kg + catch_rate,
               data = pokemon_data_final, subset = rowTrain)
```

NB Method

```
# Model building
nbGrid <- expand.grid(usekernel = c(FALSE,TRUE),
                      fL = 2,
                      adjust = seq(0, 1.5, by = .1))

model.nb <- train(x = pokemon_data_final[rowTrain, c(3,5,8,9,10)],
                  y = pokemon_data_final$is_legendary[rowTrain],
                  method = "nb",
                  tuneGrid = nbGrid,
                  metric = "ROC",
                  trControl = ctrl)
```

Comparison of Training/ Testing Performance

```
# prediction performance
glm.pred <- predict(model.glm, newdata = pokemon_data_final[-rowTrain,], type = "prob")
lda.pred <- predict(lda.fit, newdata = pokemon_data_final[-rowTrain,], type = "prob")
qda.pred <- predict(qda.fit, newdata = pokemon_data_final[-rowTrain,], type = "prob")
nb.pred  <- predict(model.nb, newdata = pokemon_data_final[-rowTrain,], type = "prob")

# roc curve building
roc.glm <- roc(pokemon_data_final$is_legendary[-rowTrain], glm.pred[, 2],
               levels = c("False", "True"))
```

```
## Setting direction: controls < cases
```

```
roc.lda <- roc(pokemon_data_final$is_legendary[-rowTrain], lda.pred$posterior[ ,2],
               levels = c("False", "True"))
```

```
## Setting direction: controls < cases
```

```
roc.qda <- roc(pokemon_data_final$is_legendary[-rowTrain], qda.pred$posterior[ ,2],
               levels = c("False", "True"))
```

```
## Setting direction: controls < cases
```

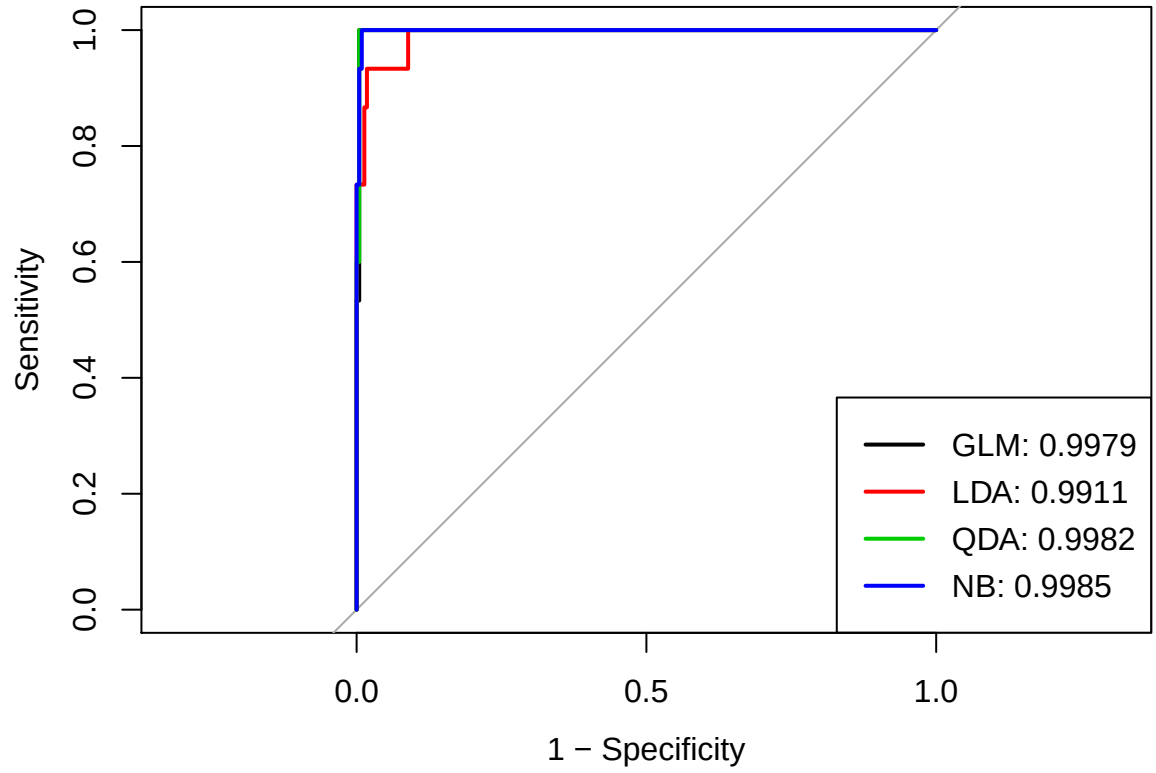
```
roc.nb <- roc(pokemon_data_final$is_legendary[-rowTrain], nb.pred[, 2], levels = c("False", "True"))
```

```
## Setting direction: controls < cases
```

```
# auc
auc <- c(roc.glm$auc[1], roc.lda$auc[1], roc.qda$auc[1], roc.nb$auc[1])

# roc curve comparison
plot(roc.glm, legacy.axes = TRUE)
plot(roc.lda, col = 2, add = TRUE)
plot(roc.qda, col = 3, add = TRUE)
```

```r
plot(roc.nb, col = 4, add = TRUE)
modelNames <- c("GLM","LDA","QDA","NB")
legend("bottomright", legend = paste0(modelNames, ": ", round(auc, 4)), col = 1:4, lwd = 2)
```



4. Conclusion

Important Predictors

It's not easy to tell which predictor is more important under the complicated mathematical structure of discrimination analysis. However, we can use logistic regression model as a reference of the importance of each predictors.

```r
summary(model.glm$finalModel)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.19048  -0.00410  -0.00016   0.00000   1.06732
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -26.564727  12.263639  -2.166  0.03030 *
## total          0.047479   0.020525   2.313  0.02071 *
```

```
## has_genderTrue  -0.801533   1.035287  -0.774  0.43880
## height_m         0.772251   0.427424   1.807  0.07080 .
## weight_kg       -0.003798   0.003453  -1.100  0.27134
## catch_rate      -0.154592   0.054792  -2.821  0.00478 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 229.954  on 480  degrees of freedom
## Residual deviance:  30.403  on 475  degrees of freedom
## AIC: 42.403
##
## Number of Fisher Scoring iterations: 12
```

Accordingly, the two major components in predicting whether a pokemon is legendary are having gender or not, height and catch rate. This is consistent to what we've observed in exploratory data analysis. On the other hand, species strength(`total`) doesn't seem to have a huge effect on determining a pokemon to be legendary. This might be a result of the large scale of species strength.

Model Comparison

All of the models in use have extremely high accuracy with AUC > 0.99. Among them, Naive Bayes Model stands out as the best model in AUC aspect. This is kind of contradictory to intution cause NB approach are suppose to be more suitable for larger p. Therefore, dispite the high AUC value, I would choose QDA as the final model.