Predicting Pokémon Types Using Clustering and Classification

Code Submission

Justin Zhang, Isaac Baguisa, Alex Faassen

2025-04-04

Note: Commented out unnecessary "test" outputs.

Data

Pre-processing

```
library(png)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
setwd("Data")

## Load Stats
stats = read.csv("pokemon_stats.csv", header = TRUE)
# Check
# str(stats)
# Image path
stats$image_path = paste0("images/", tolower(stats$name.simple), ".png")
# Check
# head(stats$image_path)

## Load Image Registry
img_reg = read.csv("pokemon_img.csv", header = TRUE)
# Has Image
stats$has_img = tolower(stats$name.simple) %in% img_reg$Name
# No Img
stats[!stats$has_img, "name"]
```

```
## character(0)
```

```
## Load Images
# Test: Abomasnow
abomasnow = readPNG("images/abomasnow.png")
ggplot()+
  annotation_raster(abomasnow,xmin=-Inf,xmax=+Inf,ymin=-Inf,ymax= +Inf)
```

```
# Pixels
# str(abomasnow)
pix = prod(dim(abomasnow)[1:2])

# Image List
image_list = list.files("images", pattern = "*.png") %>% paste("images/", ., sep = "")

# Helper fn
flatten_img = function(img_path) {
  img = readPNG(img_path) # Read image as raster array (dim: [120, 120, 4])
  return(as.vector(img[,,-4])) # Flatten to vector + Remove Alpha
}

# Image Dataset
images = sapply(image_list, flatten_img) %>% t() %>% as.data.frame() %>%
  mutate(image_path = image_list) %>% select(image_path, everything())
# colnames(images) = c("image_path", rep())
# Check
images[1:5, 1:5]
```

```
##                                          image_path V1 V2 V3 V4
## images/abomasnow.png             images/abomasnow.png  0  0  0  0
## images/abra.png                       images/abra.png  0  0  0  0
## images/absol.png                     images/absol.png  0  0  0  0
## images/accelgor.png                images/accelgor.png  0  0  0  0
## images/aegislash-blade.png images/aegislash-blade.png  0  0  0  0
```

```
# str(images)

## Match Datasets
# Match the rows to represent the same pokemon, in the same order.
images = images %>%
  semi_join(stats, by = "image_path") %>%
  arrange(match(image_path, stats$image_path))
# Check
mean(stats$image_path == images$image_path)
```

```
## [1] 1
```

```
## Save Datasets
# save(stats, images, file = "pokemon.RData")
```

EDA Visuals

```
#
```

Image Dimension Reduction

```
#
```

Clustering

```
#
```

Classification

```
#
```

Results

```
#
```