proyectoTID

Alejandro Casado Quijada y Gustavo Rivas Gervillas

Introducción

Descripción del dataset

Este dataset contiene datos recogidos de la aplicacion *PokemonGo*, esta aplicación es un juego de realidad aumeta que emplea el GPS del móvil para principalmente localizar y capturar pokemon en el mundo real. El dataset contiene 296021 muestras cada una de las cuales dispone de los siguientes campos:

- **pokemonId**: el identificador del pokemon, denota su clase.
- **latitude**: latitud de la posición donde se ha localizado el pokemon.
- **longitude**: longitud de la posición donde se ha localizado el pokemon.
- **appearedLocalTime**: momento exacto en el que se encontró el pokemon, con el formato yyyy-mm-ddThh-mm-ss.ms.
- **cellId 90-5850m**: la localización goegráfica del pokemon proyectada en una celda S2.
- **appearedTimeOfDay**: momento del día en el que apareció el pokemon (night, evening, afternoon, morning).
- **appearedHour**: hora local de una observación del pokemon.
- **appearedMinute**: minuto local de una observación del pokemon.
- **appearedDayOfWeek**: día de la semana en la que se produjo el avistamiento (Monday, Tuesday, Wednesday, Thursday, Friday, Saturdy, Sunday).
- **appearedDay**: día del avistamiento.
- **appearedMonth**: mes del avistamiento.
- **appearedYear**: año del avistamiento.
- **terrainType**: tipo del terreno donde se avistó el pokemon. Este dato viene dado por un valor número según una tabla de tipos de terreno.
- **closeToWater**: si está el pokemon a 100m del agua o no.
- **city**: ciudad donde se ha visto el pokemon.
- **continent**: continente donde se ha avistado el pokemon.
- **weather**: un string indicando el tiempo que hacía en el momento del avistamiento.
- **temperature**: temperatura en grados Celsius en el momento del avistamiento.
- **windSpeed**: velocidad del viento en el momento del avistamiento km/h.
- **windBearing**: dirección del viento entre 0 y 360 grados.
- **pressure**: presión en el momento del avistamiento en bares.
- **weatherIcon**: el tiempo atmosférico en el momento del avistamiento clasificado según un sistema de categorías más simple que el empleado en *weather* (fog, clear-night, partly-cloudy-night, partly-cloudy-day, cloudy, clear-day, rain, wind).
- **sunriseMinutesMidnight**: tiempo de la aparición relativo al amanecer.
- **sunsetMinutesBefore**: tiempo de la aparición relativo a la puesta de sol.
- **population density**: densidad de población por km^2 en un avistamiento.
- **urbal-rural**: cómo de urbana es la localización donde apareció el pokemon relativa a la *population density* (<200 rural, >= 200 && < 400 midUrban, >= 400 && < 800 subUrban, >800 urban).
- **gymDistanceKm**: distancia al gimnasio más cercano al punto de aparición del pokemon.
- **pokestopDistanceKm**: distancia a la pokestop más cercana al punto de aparicion del pokemon.
- **gymIn100m - pokestopIn5000m**: son atributos booleanos que indican si hay un gimnasio o una pokestop a 100m/200m/.../5000m de la localización donde se avistó el pokemon.
- **cooc1 - cooc151**: booleano que indica si el avistamiento de un pokemon coincidió con el de otro (de una clase entre 1 y 151) en un radio de 100m y en un rango de tiempo de 24 horas.
- **class** dice qué pokemon se trata, y en la página del dataset indica que es el atributo a predecir. ## Preprocesamiento

En primer lugar vamos a ver cuántas muestras y atributos tiene nuestro dataset. Además veremos si las clases están balanceadas, para echo emplearemos el comando `xtab`:

```
#ds <- read.csv("/mnt/Datos/Master/TID/proyectoTID/300k.csv")
ds <- read.csv("300k.csv")
```

```
cat("Hay un total de " , nrow(ds) , " muestras.\n")
```

```
## Hay un total de  296021  muestras.
```

```
cat("Cada muestra tiene " , ncol(ds) , " atributos.\n")
```

```
## Cada muestra tiene  208  atributos.
```

Con lo cual como vemos el número de muestras y de atributos es muy elevado. Además el número de clases a clasificar es muy elevado por lo tanto lo que vamos a hacer es intentar clasificar los pokemon según su tipo, así que vamos a añadir dicha columna al dataset. Los tipos considerados son aquellos que se establecieron en la primera generación de Pokemon y se recogen por medio de las siguientes variables que almacenan una cadena indicando el tipo; estas variables se han añadido por comodidad a la hora de generar el vector de tipos que crearemos a continuación:

```
P = "planta"
A = "agua"
F = "fuego"
B = "bicho"
G = "fantasma"
L = "lucha"
N = "normal"
E = "electrico"
S = "psiquico"
V = "veneno"
H = "hielo"
D = "dragon"
T = "tierra"
R = "roca"
```

A continuación formamos un array considerando el tipo primario de cada pokemon de la primera generación, dando lugar a un array de 151 elementos. Pese que en la app hay algunas especies de pokemon que no aparecen por motivos de indexación, ya que las muestras dan la clase según el número original del pokemon, introducimos el tipo de los 151 pokemon:

```
tipos = c(P,P,P,F,F,F,A,A,A,B,B,B,
  B,B,B,N,N,N,N,N,N,N,V,V,
  E,E,T,T,V,V,V,V,V,V,N,N,
  F,F,N,N,V,V,P,P,P,B,B,B,
  B,T,T,N,N,A,A,L,L,F,F,A,
  A,A,S,S,S,L,L,L,P,P,P,A,
  A,R,R,R,F,F,A,A,E,E,N,N,
  N,A,A,V,V,A,A,G,F,F,R,S,
  S,A,A,E,E,P,P,T,T,L,L,N,
  V,V,T,T,N,P,N,A,A,A,A,A,
  A,S,B,H,E,F,B,N,A,A,A,N,
  N,A,E,F,N,R,R,R,R,R,N,H,
  E,F,D,D,D,S,S)
```

Ahora vamos a proceder a añadir la columna de tipos al dataset que hemos cargado:

```
ds["tipo"] <- tipos[ds$class]
```

Veamos entonces cómo se distribuyen las muestran que tenemos según el tipo de pokemon avistado:

```
T <- xtabs(~ tipo, ds)
print(T)
```

```
## tipo
##      agua     bicho    dragon electrico  fantasma     fuego     hielo
##     37887     57641       664      2903      2419      4755       755
##     lucha    normal    planta  psiquico      roca    tierra    veneno
##      2670    129463     11171     12642      3408      5012     24631
```