

Práctica PIG

Big Data 2

Alberto Armijo Ruiz

| | |
|-----------------------------------|----------|
| Base de datos usada | 3 |
| Sentencias PIG | 3 |
| Proyección | 3 |
| Selección | 4 |
| Agrupación y cálculo sobre grupos | 5 |

Base de datos usada

Como base de datos se ha utilizado un conjunto de test de un conjunto de datos sacado de la página web de la Universidad de Irvine, el conjunto de datos se puede encontrar en el siguiente enlace: [DotaGames](#)

Este conjunto de datos contiene 102944 instancias en total; de estos el 10% pertenecen al conjunto de test. Cada una de las instancias de este conjunto de datos contienen los datos de una partida del videojuego *Dota2*. Por cada partida hay 116 columnas, las 4 primeras representan el equipo ganador de la partida, el ID del clúster, el modo de juego de la partida, el tipo de partida (competitivo, público o privada); el resto de columnas representan los diferentes personajes del juego, cada una de estas columnas contienen qué equipo lo ha elegido, si es que ha sido elegido.

El conjunto de datos que se ha utilizado es una modificación del conjunto de datos de test; ya que dicho conjunto utiliza valores enteros para representar las diferentes opciones. En el conjunto de datos utilizado estos valores enteros se han cambiado por valores categóricos para darle más sentido a las sentencias dentro de impala. El conjunto de datos se puede encontrar en el siguiente enlace:

<https://drive.google.com/file/d/1c2Pr1-CgL7FTEdc4Z9p56m8vWe9zUxOT/view>

Una vez dentro de la máquina virtual de Cloudera, se debe importar el archivo .csv a la sistema de archivos hdfs. Tras esto, se debe entrar a la shell de Impala y se crea una base de datos, en este caso llamada *practica*. Una vez creada la base de datos se crea una tabla para contener el archivo .csv.

Sentencias PIG

Proyección

Como operación de proyección se han seleccionado las columnas *WinnerTeam*, *ClusterID*, *GameMode* y *GameType*. Para ello se ha utilizado la siguiente sentencia:

```
projection = foreach dotagames generate WinnerTeam, ClusterID, GameMode,  
GameType;  
store projection into 'pigResults/projection' using PigStorage(',');
```

El resultado obtenido es el siguiente.



```
Red Team,223,Reverse Captain's Mode ,Unranked Game
Blue Team,227,Reverse Captain's Mode ,Unranked Game
Red Team,136,Captain's Mode,Unranked Game
Blue Team,227,Captain's Mode,Unranked Game
Blue Team,184,Captain's Mode,Ranked Game
Blue Team,231,Captain's Mode,Unranked Game
Blue Team,152,Captain's Mode,Unranked Game
Red Team,153,Captain's Mode,Unranked Game
Red Team,223,Reverse Captain's Mode ,Unranked Game
Red Team,153,Captain's Mode,Unranked Game
Red Team,133,The Greeviling,Unranked Game
Red Team,124,Captain's Mode,Ranked Game
Blue Team,154,Captain's Mode,Ranked Game
Blue Team,151,Captain's Mode,Unranked Game
Red Team,223,The Greeviling,Unranked Game
Blue Team,154,Captain's Mode,Ranked Game
Red Team,227,Captain's Mode,Ranked Game
Red Team,224,Captain's Mode,Ranked Game
Blue Team,225,Captain's Mode,Unranked Game
Red Team,187,Captain's Mode,Unranked Game
Red Team,225,Reverse Captain's Mode ,Ranked Game
Red Team,152,Captain's Mode,Ranked Game
Red Team,231,Captain's Mode,Unranked Game
:
```

Selección

Como operación de selección se han obtenido todas las entradas en las que gana el equipo azul. Para ello se ha utilizado la siguiente sentencia:

```
selection = filter dotagames by WinnerTeam == 'Blue Team';  
store selection into 'pigResults/selection' using PigStorage(',');
```

El resultado de esta operación es la siguiente.

```

Blue Team,227,Reverse Captain's Mode ,Unranked Game,None,None,None,None,None,Non
e,None,None,Blue Team,None,None,None,None,Blue Team,None,None,None,Blue Team,Non
e,None,Red Team,None,None,Red Team,None,None,Red Team,None,None,None,None,Non
one,None,None,None,None,None,None,None,Red Team,None,None,None,None,None,Non
ne,None,None,None,None,None,None,Red Team,None,None,None,None,None,None,Non
e,None,None,None,None,None,None,None,None,None,None,None,None,None,None,Non
e,None,None,None,None,Blue Team,None,None,None,None,None,None,Blue Team,None,Non
e,None,None,None,None,None,None,None,None,None,Red Team,None,None,None,Red
Team,None,None,None,None,None
Blue Team,227,Captain's Mode,Unranked Game,Red Team,None,None,None,None,None,Non
e,None,Blue Team,None,None,None,None,None,None,None,None,None,None,None,Non
e,Red Team,Red Team,None,None,None,None,None,None,None,None,None,None,None,Non
None,None,Red Team,None,None,None,None,None,None,None,None,None,None,None,Bl
ue Team,None,Red Team,None,Blue Team,None,None,None,None,None,None,None,Non
ne,None,None,None,None,None,None,None,None,None,None,None,None,None,Non
ne,None,None,Blue Team,None,Red Team,None,None,None,None,None,None,None,Non
e,None,None,Blue Team,None,None,None,None,None,None,None,None,Red Team,None
, None, None, None, None
Blue Team,184,Captain's Mode,Ranked Game,None,None,None,Red Team,None,None,None,
Red Team,None,None,None,None,None,None,None,None,None,None,Blue Team,N
one,None,Red Team,None,Red Team,None,None,None,None,None,None,None,Non
e,None,None,Blue Team,None,None,Red Team,None,Blue Team,None,None,None,Non
e,None,None,None,None,None,None,None,None,Red Team,None,None,Blue Team,
:

```

Agrupación y cálculo sobre grupos

En este apartado se han realizado dos operaciones, la primera cuenta el número de partidas ganadas por el equipo azul en cada clúster; la siguiente muestra el número de partidas ganadas por el equipo azul y el total de partidas jugadas en ese servidor. Para la primera operación se debe usar las siguientes sentencias:


```

measure_by_cluster = group selection by ClusterID;
num_wins_by_team = foreach measure_by_cluster generate group,
    COUNT(selection.WinnerTeam) as wins;

store num_wins_by_team into 'pigResults/group_op' using PigStorage(',');

```

El resultado de esta operación es el siguiente.

A screenshot of a text editor window with a light blue border and a vertical scrollbar on the right. The text inside the editor is a list of pairs of numbers, separated by a comma, representing cluster IDs and win counts. The list starts with '111,51' and ends with '161,18', followed by a colon on a new line. The text is left-aligned and uses a monospaced font.

```
111,51
112,47
121,43
122,36
123,46
124,27
131,29
132,43
133,51
134,51
135,47
136,35
137,39
138,43
144,77
145,71
151,417
152,402
153,401
154,387
155,384
156,392
161,18
:
```

La primera columna representa el ID del clúster, la segunda el número de partidas ganadas por el equipo azul.

Para la segunda operación se debe añadir lo siguiente a las sentencias anteriores; además se puede eliminar la sentencia que guarda los datos (store ...).

```
total_measure_by_cluster = group dotagames by ClusterID;
num_wins_total = foreach total_measure_by_cluster generate group,
  COUNT(dotagames.WinnerTeam) as win_total;
totalvsbluewins = join num_wins_total by group, num_wins_by_team by group ;
totalvsbluewins = foreach totalvsbluewins generate $0,$1,$3;
store totalvsbluewins into 'pigResults/wins' using PigStorage(',');
```

El resultado de esta sentencia es la siguiente.

| |
|-------------|
| 111,80,51 |
| 112,97,47 |
| 121,74,43 |
| 122,74,36 |
| 123,85,46 |
| 124,59,27 |
| 131,64,29 |
| 132,96,43 |
| 133,81,51 |
| 134,92,51 |
| 135,83,47 |
| 136,75,35 |
| 137,70,39 |
| 138,80,43 |
| 144,146,77 |
| 145,132,71 |
| 151,764,417 |
| 152,745,402 |
| 153,756,401 |
| 154,738,387 |
| 155,714,384 |
| 156,769,392 |
| 161,27,18 |

En la primera columna se puede ver el ID del cluster, en la segunda se puede ver el número de partidas jugadas en el clúster y en la tercera el número de partidas ganadas por el equipo azul.