ETL con PIG

1 Base de datos

La base de datos que voy a usar la he sacado de: https://www.kaggle.com/datasets , para ser más exactos, he escogido una en la que se tratan eventos que pasan en partidos de futbol: https://www.kaggle.com/secareanualin/football-events

Enlace a mi Dropbox: https://www.dropbox.com/sh/qwmh56qagnjblx1/AABWw4d-xNamPMjJH-_gE2Sta?dl=0

Este dataset se divide en 3 ficheros:

- **Dictionary.txt**: en el que se especifican los detalles de algunos valores numéricos de las variables.
- **Ginf.csv:** Un fichero con algunos detalles sobre los partidos que luego se verán en los eventos.
- **Events.csv:** Dataset principal en el que tenemos 941009 instancias, con 22 atributos y en el que se tienen eventos que suceden durante los partidos. Los atributos y su descripción se pueden ver en la web de descarga del dataset, pero además lo adjunto en el Anexo 1, al final de documento.

Por lo tanto, he subido al servidor este dataset para realizar nuestras pruebas con pig.

2 Pig

2.1 Carga del dataset

Como podemos ver en el script, lo primero que hacemos es, tras abrir pig con "pig –x local", es cargar el fichero csv con el siguiente comando:

```
dt_event = load 'input/events.csv' using PigStorage(',') AS (id_odsp:chararray, id_event:chararray, sort_order:float, time:float, text:chararray, event_type:chararray, event_type2:chararray, side:chararray, event_team:chararray, opponent:chararray, player:chararray, player2:chararray, player_in:chararray, player_out:chararray, shot_place:chararray, shot_outcome:chararray, is_goal:int, location:chararray, bodypart:chararray, assist_method:chararray, situation:chararray, fast_break:int);
```

Donde a cada variable le asignamos el tipo de dato que es.

Si lo que queremos es mostrar esta lectura podemos hacer:

```
dump dt_event;
```

Quedando algo como lo que vemos a continuación:

```
("jTgA8mrd/","jTgA8mrd18",18.0,20.0,"Roberto Trashorras (Rayo Vallecano) wins a free kick in the defensive half.",8,NA,1,"Rayo Vallecano","Real Madrid","roberto trashorras",NA,NA,NA,NA,NA,0,2,NA,0,NA,0)
("jTgA8mrd/","jTgA8mrd19",19.0,21.0,"Corner,  Rayo Vallecano. Conceded by Daniel Carvajal.",2,NA,1,"Rayo Vallecano","Real Madrid","daniel carvajal","daniel carvajal",NA,NA,NA,,0,NA,NA,0,)
("jTgA8mrd/","jTgA8mrd20",20.0,24.0,"Foul by Alejandro GA¡lvez (Rayo Vallecano).",3,NA,1,"Rayo Vallecano","Real Madrid","alejandro galvez",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("jTgA8mrd/","jTgA8mrd21",21.0,24.0,"Karim Benzema (Real Madrid) wins a free kick in the defensive half.",8,NA,2,"Real Madrid","Rayo Vallecano","karim benzema",NA,NA,NA,NA,NA,0,2,NA,0,NA,0)
```

Esta lectura la podríamos almacenar con: (estará al final del script)

```
store dt_event into 'pigResults/EventsProcessed';
```

2.2 Proyección

Si queremos una proyección simple para ver cómo funciona esta, podemos quedarnos con los equipos que realizan cada evento y el jugador principal del evento con:

```
Equipos_y_Jugador = foreach dt_event generate event_team, opponent, player;
```

Si lo mostramos con dump, podemos ver el resultado como el siguiente:

```
("Real Madrid","Rayo Vallecano","karim benzema")
("Rayo Vallecano","Real Madrid","roberto trashorras")
(1,"Rayo Vallecano","Real Madrid")
("Rayo Vallecano","Real Madrid","alejandro galvez")
("Real Madrid","Rayo Vallecano","karim benzema")
grunt>
```

Donde vemos el resultado de las variables que hemos decido obtener de nuestro dataset.

2.3 Selección

He realizado 3 selecciones distintas, para poder hacer 3 agrupamientos y resúmenes distintos para ver distintos resultados. Veamos por lo tanto estas selecciones.

Seleccionaremos los eventos en los que el equipo local ha sido el Real Madrid o la Juventus:

```
RM_J_Local = filter dt_event by (event_team == '"Real Madrid"') OR (event_team == '"Juventus"');
```

```
("jTgA8mrd/","jTgA8mrd6",6.0,6.0,"Daniel Carvajal (Real Madrid) is shown the yellow card for a bad foul.",4,NA,2,"Real Madrid","Rayo Vallecano","daniel carvajal",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("jTgA8mrd/","jTgA8mrd7",7.0,11.0,"Foul by Xabi Alonso (Real Madrid).",3,NA,2,"Real Madrid","Rayo Vallecano","xabi alonso",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("jTgA8mrd/","jTgA8mrd9",9.0,11.0,"Xabi Alonso (Real Madrid) is shown the yellow card for a bad foul.",4,NA,2,"Real Madrid","Rayo Vallecano","xabi alonso",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("jTgA8mrd/","jTgA8mrd14",14.0,17.0,"Daniel Carvajal (Real Madrid) wins a free kick in the defensive half.",8,NA,2,"Real Madrid","Rayo Vallecano","daniel carvajal",NA,NA,NA,NA,NA,0,2,NA,0,NA,0)
("jTgA8mrd/","jTgA8mrd17",17.0,20.0,"Foul by Karim Benzema (Real Madrid).",3,NA,2,"Real Madrid","Rayo Vallecano","karim benzema",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("jTgA8mrd/","jTgA8mrd21",21.0,24.0,"Karim Benzema (Real Madrid) wins a free kick in the defensive half.",8,NA,2,"Real Madrid","Rayo Vallecano","karim benzema",NA,NA,NA,NA,NA,0,2,NA,0,NA,0)
```

Seleccionaremos los eventos en los que el equipo local es el Real Madrid y el evento ha sido una tarjeta roja, con el atributo event_type igual a 6:

```
RM_RedCard = filter dt_event by (event_team == '"Real Madrid"') AND (event_type == '6');
```

```
("vJg7s9Hk/","vJg7s9Hk47",47.0,53.0,"FA¡bio CoentrA£o (Real Madrid) is shown the red card.",6,14,1,"Real Madrid","Espanyol","fabio coentrao",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("EidD9R0r/","EidD9R0r98",98.0,82.0,"Cristiano Ronaldo (Real Madrid) is shown the red card for fighting.",6,14,2,"Real Madrid","Cordoba","cristiano ronaldo",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("d2ZLt3wM/","d2ZLt3wM131",131.0,85.0,"Mesut Ozil (Real Madrid) is shown the red card.",6,14,2,"Real Madrid","Villarreal","mesut ozil",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("8tWd04ym/","8tWd04ym91",91.0,90.0,"FA¡bio CoentrA£o (Real Madrid) is shown the red card.",6,14,2,"Real Madrid","Getafe","fabio coentrao",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("ClVHKXcr/","ClVHKXcr87",87.0,84.0,"Isco (Real Madrid) is shown the red card.",6,14,1,"Real Madrid","Barcelona","isco",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("69LBecmK/","69LBecmK75",75.0,68.0,"Mateo Kovacic (Real Madrid) is shown the red card.",6,14,2,"Real Madrid","Valencia","mateo kovacic",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("fX2hW5hg/","fX2hW5hg71",71.0,75.0,"Cristiano Ronaldo (Real Madrid) is shown the red card for fighting.",6,14,2,"Real Madrid","Athletic Bilbao","cristiano ronaldo",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("StRC9O3T/","StRC9O3T70",70.0,63.0,"Sergio Ramos (Real Madrid) is shown the red card.",6,14,1,"Real Madrid","Barcelona","sergio ramos",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("dEbNSKje/","dEbNSKje6",6.0,6.0,"Antonio AdA¡n (Real Madrid) is shown the red card.",6,14,1,"Real Madrid","Real Sociedad","antonio adan",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
```

Seleccionaremos los eventos en los que el equipo local es el Real Madrid y el evento ha sido un penalti, con el atributo event_type igual a 11:

```
RM_Penalty = filter dt_event by (event_team == '"Real Madrid"') AND (event_type == '11');
```

```
("buWp9Tfb/","buWp9Tfb86",86.0,71.0,"Penalty Real Madrid. Luka Modric draws a foul in the penalty area.",11,NA,1,"Real Madrid","Getafe",NA,NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("WjRDjw1h/","WjRDjw1h81",81.0,71.0,"Penalty Real Madrid. KakA¡ draws a foul in the penalty area.",11,NA,2,"Real Madrid","Celta Vigo",NA,NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("0xxzxg4k/","0xxzxg4k28",28.0,21.0,"Penalty Real Madrid. Cristiano Ronaldo draws a foul in the penalty area.",11,NA,1,"Real Madrid","Malaga",NA,NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("YkU2Uz4M/","YkU2Uz4M109",109.0,90.0,"Penalty Real Madrid. Pepe draws a foul in the penalty area.",11,NA,2,"Real Madrid","Elche",NA,NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("neLVl2o6/","neLVl2o684",84.0,90.0,"Penalty Real Madrid. Gareth Bale draws a foul in the penalty area.",11,NA,1,"Real Madrid","Malaga",NA,NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
("nyyOdjqp/","nyyOdjqp33",33.0,31.0,"Penalty Real Madrid. Isco draws a foul in the penalty area.",11,NA,1,"Real Madrid","Sevilla",NA,NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)
```

2.4 Agrupamientos y resúmenes

En base a nuestras selecciones del apartado anterior vamos a agrupar las 3 selecciones por el equipo rival y vamos a realizar un resumen para cada caso.

Si queremos ver la media del minuto en el que ocurren los eventos, para la primera selección, podemos ver como la mayoría de los resultados es el minuto 45, como era de esperar ya que los eventos ocurren entre el minuto 0 y el 90:

RM_J_Rival = group RM_J_Local by opponent;

minuto_medio = foreach RM_J_Rival generate group, AVG(RM_J_Local.time) as dt_event;

```
("Barcelona",49.87218045112782)
("Frosinone",50.172413793103445)
("Sampdoria",50.05993690851735)
("Celta Vigo",48.54985754985755)
("Fiorentina",48.454918032786885)
("Las Palmas",47.08620689655172)
("Real Betis",50.075520833333336)
("US Pescara",47.55833333333333)
("Villarreal",49.26495726495727)
("Chievo Verona",46.79759519038076)
("Hellas Verona",49.26053639846743)
("Real Sociedad",48.470449172576835)
("Real Zaragoza",47.81460674157304)
("Internazionale",47.746192893401016)
("Rayo Vallecano",47.69827586206897)
("Sporting Gijon",47.833333333333336)
("Athletic Bilbao",47.337868480725625)
("Atletico Madrid",49.47679324894515)
("Real Valladolid",47.42168674698795)
("Racing Santander",48.88172043010753)
("Deportivo La Coruna",48.258620689655174)
```

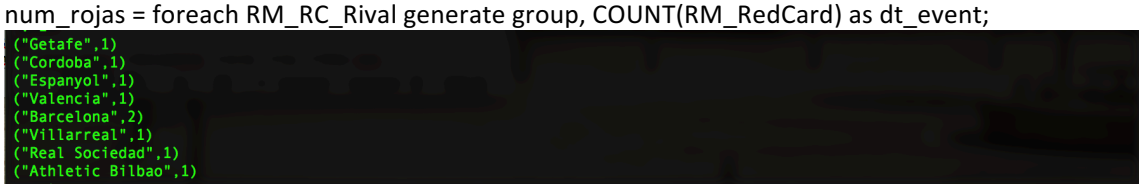
Si nos centramos en la segunda selección, vamos a contar cuantas tarjetas rojas se tienen según el rival al que se enfrenta:

RM_RC_Rival = group RM_RedCard by opponent;

```
("Getafe",{("8tWd04ym/","8tWd04ym91",91.0,90.0,"FA¡bio CoentrA£o (Real Madrid) is shown the red card.",6,14,2,"Real Madrid","Getafe","fabio coentrao",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)})
("Cordoba",{("EidD9R0r/","EidD9R0r98",98.0,82.0,"Cristiano Ronaldo (Real Madrid) is shown the red card for fighting.",6,14,2,"Real Madrid","Cordoba","cristiano ronaldo",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)})
("Espanyol",{("vJg7s9Hk/","vJg7s9Hk47",47.0,53.0,"FA¡bio CoentrA£o (Real Madrid) is shown the red card.",6,14,1,"Real Madrid","Espanyol","fabio coentrao",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)})
("Valencia",{("69LBecmK/","69LBecmK75",75.0,68.0,"Mateo Kovacic (Real Madrid) is shown the red card.",6,14,2,"Real Madrid","Valencia","mateo kovacic",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)})
("Barcelona",{("ClVHKXcr/","ClVHKXcr87",87.0,84.0,"Isco (Real Madrid) is shown the red card.",6,14,1,"Real Madrid","Barcelona","isco",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0),("StRC9O3T/","StRC9O3T70",70.0,63.0,"Sergio Ramos (Real Madrid) is shown the red card.",6,14,1,"Real Madrid","Barcelona","sergio ramos",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)})
("Villarreal",{("d2ZLt3wM/","d2ZLt3wM131",131.0,85.0,"Mesut Ozil (Real Madrid) is shown the red card.",6,14,2,"Real Madrid","Villarreal","mesut ozil",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)})
("Real Sociedad",{("dEbNSKje/","dEbNSKje6",6.0,6.0,"Antonio AdA¡n (Real Madrid) is shown the red card.",6,14,1,"Real Madrid","Real Sociedad","antonio adan",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)})
("Athletic Bilbao",{("fX2hW5hg/","fX2hW5hg71",71.0,75.0,"Cristiano Ronaldo (Real Madrid) is shown the red card for fighting.",6,14,2,"Real Madrid","Athletic Bilbao","cristiano ronaldo",NA,NA,NA,NA,NA,0,NA,NA,0,NA,0)})
```

Ahora veamos la cuenta:

num_rojas = foreach RM_RC_Rival generate group, COUNT(RM_RedCard) as dt_event;



```
("Getafe",1)
("Cordoba",1)
("Espanyol",1)
("Valencia",1)
("Barcelona",2)
("Villarreal",1)
("Real Sociedad",1)
("Athletic Bilbao",1)
```

SI hacemos lo mismo, para la tercera selección, vemos el número de penaltis en función del oponente:

RM_Penal_Rival = group RM_Penalty by opponent;

num_penal = foreach RM_Penal_Rival generate group, COUNT(RM_Penalty) as dt_event;

3 Conclusiones

Hemos realizado pruebas en Pig en modo local. Hemos probado las proyecciones, selecciones, agrupamientos y resúmenes. Se ha visto lo fácil que es realizar este tipo de acciones con esta herramienta y como se puede tratar con muchos tipos de datos.

Anexo 1: 22 características del dataset events.csv

id_odsp -> unique identifier of game (odsp stands from oddsportal.com) -> String
id_event -> unique identifier of event (id_odsp + sort_order) -> String
sort_order -> chronological sequence of events in a game -> Numeric
time -> minute of the game -> Numeric
text -> text commentary -> String
event_type -> primary event. 11 unique events (1-Attempt(shot), 2-Corner, 3-Foul, 4-Yellow Card, 5-Second yellow card, 6-(Straight) red card, 7-Substitution, 8-Free kick won, 9-Offside, 10-Hand Ball, 11-Penalty conceded) -> String
event_type2 -> secondary event. 4 unique events (12 - Key Pass, 13 - Failed through ball, 14-Sending off, 15-Own goal) -> String
side -> 1-Home, 2-Away -> String
event_team -> team that produced the event. In case of Own goals, event team is the team that beneficiated from the own goal -> String
opponent -> Help us describe this column -> String
player -> name of the player involved in main event (converted to lowercase and special chars were removed) -> String
player2 -> name of player involved in secondary event -> String
player_in -> player that came in (only applies to substitutions) -> String
player_out -> player substituted (only applies to substitutions) -> String
shot_place -> placement of the shot (13 possible placement locations, available in the dictionary, only applies to shots) -> String
shot_outcome -> 4 possible outcomes (1-On target, 2-Off target, 3-Blocked, 4-Hit the post) -> String
is_goal -> binary variable if the shot resulted in a goal (own goals included) -> Boolean
location -> location on the pitch where the event happened (19 possible locations, available in the dictionary) -> String
bodypart -> (1- right foot, 2-left foot, 3-head) -> String
assist_method -> in case of an assisted shot, 5 possible assist methods (details in the dictionary) -> String
situation -> 4 types: 1-Open Play, 2-Set piece (excluding Direct Free kicks), 3-Corner, 4-Free kick -> String
fast_break -> binary -> Boolean