Bundesliga Soccer Analysis using Pyspark

Questions to Answer

Who are the winners of the D1 division in the Germany Football Association (Bundesliga) in the last decade?

Which teams have been relegated in the past 10 years?

Does Octoberfest affect the performance of Bundesliga?

In [1]:

```python
#importing required Libraries
import findspark
from pyspark import SparkContext
from pyspark.sql import SparkSession, Window, Row
from pyspark.sql.functions import *
from pyspark.sql.types import *
import matplotlib.pyplot as plt
```

In [2]:

```python
#https://spark.apache.org/docs/latest/sql-getting-started.html
spark = SparkSession \
        .builder \
        .appName("firstSpark") \
        .getOrCreate()
```

In [3]:

```python
# df = spark.read.format('csv').options(header='true').load(".")
def load_dataframe(filename):
    df = spark.read.format('csv').options(header='true').load(filename)
    return df
```

In [4]:

```python
#creating a dataframe
df_matches = load_dataframe('/Users/ruturajharal/Desktop/Bundesliga-analysis-pyspark/Matches.csv')
df_matches.limit(10).show()
```

```
+--------+---+------+----------+--------------+--------------+----+----+---+
|Match_ID|Div|Season|      Date|      HomeTeam|      AwayTeam|FTHG|FTAG|FTR|
+--------+---+------+----------+--------------+--------------+----+----+---+
|       1| D2|  2009|2010-04-04|    Oberhausen|Kaiserslautern|   2|   1|  H|
|       2| D2|  2009|2009-11-01|   Munich 1860|Kaiserslautern|   0|   1|  A|
|       3| D2|  2009|2009-10-04| Frankfurt FSV|Kaiserslautern|   1|   1|  D|
|       4| D2|  2009|2010-02-21| Frankfurt FSV|     Karlsruhe|   2|   1|  H|
|       5| D2|  2009|2009-12-06|         Ahlen|     Karlsruhe|   1|   3|  A|
|       6| D2|  2009|2010-04-03|  Union Berlin|     Karlsruhe|   1|   1|  D|
|       7| D2|  2009|2009-08-14|     Paderborn|     Karlsruhe|   2|   0|  H|
|       8| D2|  2009|2010-03-08|     Bielefeld|     Karlsruhe|   0|   1|  A|
|       9| D2|  2009|2009-09-26|Kaiserslautern|     Karlsruhe|   2|   0|  H|
|      10| D2|  2009|2009-11-21| Hansa Rostock|     Karlsruhe|   2|   1|  H|
+--------+---+------+----------+--------------+--------------+----+----+---+
```

In [5]:

```python
#converting to pandas dataframe
df_matches.limit(10).toPandas()
```

Out[5]:

	Match_ID	Div	Season	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR
0	1	D2	2009	2010-04-04	Oberhausen	Kaiserslautern	2	1	H
1	2	D2	2009	2009-11-01	Munich 1860	Kaiserslautern	0	1	A
2	3	D2	2009	2009-10-04	Frankfurt FSV	Kaiserslautern	1	1	D
3	4	D2	2009	2010-02-21	Frankfurt FSV	Karlsruhe	2	1	H
4	5	D2	2009	2009-12-06	Ahlen	Karlsruhe	1	3	A
5	6	D2	2009	2010-04-03	Union Berlin	Karlsruhe	1	1	D
6	7	D2	2009	2009-08-14	Paderborn	Karlsruhe	2	0	H
7	8	D2	2009	2010-03-08	Bielefeld	Karlsruhe	0	1	A
8	9	D2	2009	2009-09-26	Kaiserslautern	Karlsruhe	2	0	H
9	10	D2	2009	2009-11-21	Hansa Rostock	Karlsruhe	2	1	H

In [6]:

```python
df_matches.printSchema()
```

```
root
 |-- Match_ID: string (nullable = true)
 |-- Div: string (nullable = true)
 |-- Season: string (nullable = true)
 |-- Date: string (nullable = true)
 |-- HomeTeam: string (nullable = true)
 |-- AwayTeam: string (nullable = true)
 |-- FTHG: string (nullable = true)
 |-- FTAG: string (nullable = true)
 |-- FTR: string (nullable = true)
```

In [7]:

```python
#Next step would be renaming some of the columns
old_cols = df_matches.columns[-3:]
new_cols = ["HomeTeamGoals", "AwayTeamGoals", "FinalResult"]

#combine 2 rdd's
old_new_cols = [*zip(old_cols, new_cols)]
for old_col, new_col in old_new_cols:
    df_matches = df_matches.withColumnRenamed(old_col, new_col)
```

In [8]:

```python
#checking the pandas structure

df_matches.limit(5).toPandas()
```

Out[8]:

	Match_ID	Div	Season	Date	HomeTeam	AwayTeam	HomeTeamGoals	AwayTeamGoals	FinalResult
0	1	D2	2009	2010-04-04	Oberhausen	Kaiserslautern	2	1	H
1	2	D2	2009	2009-11-01	Munich 1860	Kaiserslautern	0	1	A
2	3	D2	2009	2009-10-04	Frankfurt FSV	Kaiserslautern	1	1	D
3	4	D2	2009	2010-02-21	Frankfurt FSV	Karlsruhe	2	1	H
4	5	D2	2009	2009-12-06	Ahlen	Karlsruhe	1	3	A

Who are the winners Bundesliga in the last decade?

In [9]:

```python
df_matches.limit(10).toPandas()
```

Out[9]:

	Match_ID	Div	Season	Date	HomeTeam	AwayTeam	HomeTeamGoals	AwayTeamGoals	FinalResult
0	1	D2	2009	2010-04-04	Oberhausen	Kaiserslautern	2	1	H
1	2	D2	2009	2009-11-01	Munich 1860	Kaiserslautern	0	1	A
2	3	D2	2009	2009-10-04	Frankfurt FSV	Kaiserslautern	1	1	D
3	4	D2	2009	2010-02-21	Frankfurt FSV	Karlsruhe	2	1	H
4	5	D2	2009	2009-12-06	Ahlen	Karlsruhe	1	3	A
5	6	D2	2009	2010-04-03	Union Berlin	Karlsruhe	1	1	D
6	7	D2	2009	2009-08-14	Paderborn	Karlsruhe	2	0	H
7	8	D2	2009	2010-03-08	Bielefeld	Karlsruhe	0	1	A
8	9	D2	2009	2009-09-26	Kaiserslautern	Karlsruhe	2	0	H
9	10	D2	2009	2009-11-21	Hansa Rostock	Karlsruhe	2	1	H

The approach is to aggregate the home and away game results separately creating two dataframes: home and away.

In [10]:

```python
#If Home team Win then result = 1, Away team win result = 1, Draw result =1
df_matches = df_matches.withColumn('HomeTeamWin', when(col('FinalResult') == 'H', 1).otherwise(0)) \
                        .withColumn('AwayTeamWin', when(col('FinalResult') == 'A', 1).otherwise(0)) \
                        .withColumn('GameTie', when(col('FinalResult') == 'D', 1).otherwise(0))
```

In [11]:

```python
df_matches.limit(10).toPandas()
```

Out[11]:

	Match_ID	Div	Season	Date	HomeTeam	AwayTeam	HomeTeamGoals	AwayTeamGoals	FinalResult	HomeTeamWin	AwayTea
0	1	D2	2009	2010-04-04	Oberhausen	Kaiserslautern	2	1	H	1	
1	2	D2	2009	2009-11-01	Munich 1860	Kaiserslautern	0	1	A	0	
2	3	D2	2009	2009-10-04	Frankfurt FSV	Kaiserslautern	1	1	D	0	
3	4	D2	2009	2010-02-21	Frankfurt FSV	Karlsruhe	2	1	H	1	
4	5	D2	2009	2009-12-06	Ahlen	Karlsruhe	1	3	A	0	
5	6	D2	2009	2010-04-03	Union Berlin	Karlsruhe	1	1	D	0	
6	7	D2	2009	2009-08-14	Paderborn	Karlsruhe	2	0	H	1	
7	8	D2	2009	2010-03-08	Bielefeld	Karlsruhe	0	1	A	0	
8	9	D2	2009	2009-09-26	Kaiserslautern	Karlsruhe	2	0	H	1	
9	10	D2	2009	2009-11-21	Hansa Rostock	Karlsruhe	2	1	H	1	

In [12]:

```python
#bundesliga is a D1 division and we are interested in season <= 2010 and >= 2000
bundesliga = df_matches \
                    .filter((col('Season') >= 2000) &
                            (col('Season') <= 2010) &
                            (col('Div') == 'D1'))
```

In [13]:

```python
bundesliga.limit(10).toPandas()
```

Out[13]:

	Match_ID	Div	Season	Date	HomeTeam	AwayTeam	HomeTeamGoals	AwayTeamGoals	FinalResult	HomeTeamWin	AwayTeamWi
0	21	D1	2009	2010-02-06	Bochum	Leverkusen	1	1	D	0	
1	22	D1	2009	2009-11-22	Bayern Munich	Leverkusen	1	1	D	0	
2	23	D1	2009	2010-05-08	M'gladbach	Leverkusen	1	1	D	0	
3	24	D1	2009	2009-08-08	Mainz	Leverkusen	2	2	D	0	
4	25	D1	2009	2009-10-17	Hamburg	Leverkusen	0	0	D	0	
5	26	D1	2009	2010-04-17	Stuttgart	Leverkusen	2	1	H	1	
6	27	D1	2009	2010-03-20	Dortmund	Leverkusen	3	0	H	1	
7	28	D1	2009	2009-10-31	Schalke 04	Leverkusen	2	2	D	0	
8	29	D1	2009	2009-08-22	Freiburg	Leverkusen	0	5	A	0	
9	30	D1	2009	2010-02-21	Werder Bremen	Leverkusen	2	2	D	0	

In [14]:

```python
# home team features
home = bundesliga.groupby('Season', 'HomeTeam') \
       .agg(sum('HomeTeamWin').alias('TotalHomeWin'),
            sum('AwayTeamWin').alias('TotalHomeLoss'),
            sum('GameTie').alias('TotalHomeTie'),
            sum('HomeTeamGoals').alias('HomeScoredGoals'),
            sum('AwayTeamGoals').alias('HomeAgainstGoals')) \
       .withColumnRenamed('HomeTeam', 'Team')
```

In [15]:

```
home.limit(10).toPandas()
```

Out[15]:

	Season	Team	TotalHomeWin	TotalHomeLoss	TotalHomeTie	HomeScoredGoals	HomeAgainstGoals
0	2005	Kaiserslautern	5	7	5	26.0	33.0
1	2006	Cottbus	6	6	5	21.0	22.0
2	2001	St Pauli	4	9	4	19.0	28.0
3	2005	Mainz	6	4	7	31.0	23.0
4	2006	Hamburg	4	4	9	22.0	19.0
5	2003	Stuttgart	9	1	7	29.0	13.0
6	2003	Hansa Rostock	10	6	1	34.0	18.0
7	2007	Hansa Rostock	5	8	4	17.0	21.0
8	2001	M'gladbach	6	6	5	21.0	21.0
9	2002	M'gladbach	10	2	5	31.0	11.0

In [16]:

```
#away game features
away =  bundesliga.groupby('Season', 'AwayTeam') \
        .agg(sum('AwayTeamWin').alias('TotalAwayWin'),
             sum('HomeTeamWin').alias('TotalAwayLoss'),
             sum('GameTie').alias('TotalAwayTie'),
             sum('AwayTeamGoals').alias('AwayScoredGoals'),
             sum('HomeTeamGoals').alias('AwayAgainstGoals'))  \
        .withColumnRenamed('AwayTeam', 'Team')
```

In [17]:

```
away.limit(10).toPandas()
```

Out[17]:

	Season	Team	TotalAwayWin	TotalAwayLoss	TotalAwayTie	AwayScoredGoals	AwayAgainstGoals
0	2005	Kaiserslautern	3	10	4	21.0	38.0
1	2006	Cottbus	5	9	3	17.0	27.0
2	2001	St Pauli	0	11	6	18.0	42.0
3	2005	Mainz	3	10	4	15.0	24.0
4	2006	Hamburg	6	5	6	21.0	18.0
5	2003	Stuttgart	9	5	3	23.0	11.0
6	2003	Hansa Rostock	2	8	7	21.0	36.0
7	2007	Hansa Rostock	3	12	2	13.0	31.0
8	2001	M'gladbach	3	7	7	20.0	32.0
9	2002	M'gladbach	1	12	4	12.0	34.0

In [18]:

```
#season features
window = ['Season']
window = Window.partitionBy(window).orderBy(col('WinPct').desc(), col('GoalDifferentials').desc())
table = home.join(away, ['Team', 'Season'],  'inner') \
    .withColumn('GoalsScored', col('HomeScoredGoals') + col('AwayScoredGoals')) \
    .withColumn('GoalsAgainst', col('HomeAgainstGoals') + col('AwayAgainstGoals')) \
    .withColumn('GoalDifferentials', col('GoalsScored') - col('GoalsAgainst')) \
    .withColumn('Win', col('TotalHomeWin') + col('TotalAwayWin')) \
    .withColumn('Loss', col('TotalHomeLoss') + col('TotalAwayLoss')) \
    .withColumn('Tie', col('TotalHomeTie') + col('TotalAwayTie')) \
    .withColumn('WinPct', round((100* col('Win')/(col('Win') + col('Loss') + col('Tie'))), 2)) \
    .drop('HomeScoredGoals', 'AwayScoredGoals', 'HomeAgainstGoals', 'AwayAgainstGoals') \
    .drop('TotalHomeWin', 'TotalAwayWin', 'TotalHomeLoss', 'TotalAwayLoss', 'TotalHomeTie', 'TotalAwayTie') \
    .withColumn('TeamPosition', rank().over(window))
```

In [19]:

```
table_df = table.filter(col('TeamPosition') == 1).orderBy(asc('Season')).toPandas()
table_df
```

Out[19]:

	Team	Season	GoalsScored	GoalsAgainst	GoalDifferentials	Win	Loss	Tie	WinPct	TeamPosition
0	Bayern Munich	2000	62.0	37.0	25.0	19	9	6	55.88	1
1	Leverkusen	2001	77.0	38.0	39.0	21	7	6	61.76	1
2	Bayern Munich	2002	70.0	25.0	45.0	23	5	6	67.65	1
3	Werder Bremen	2003	79.0	38.0	41.0	22	4	8	64.71	1
4	Bayern Munich	2004	75.0	33.0	42.0	24	5	5	70.59	1
5	Bayern Munich	2005	67.0	32.0	35.0	22	3	9	64.71	1
6	Stuttgart	2006	61.0	37.0	24.0	21	6	7	61.76	1
7	Bayern Munich	2007	68.0	21.0	47.0	22	2	10	64.71	1
8	Wolfsburg	2008	80.0	41.0	39.0	21	7	6	61.76	1
9	Bayern Munich	2009	72.0	31.0	41.0	20	4	10	58.82	1
10	Dortmund	2010	67.0	22.0	45.0	23	5	6	67.65	1

From this we can see that Bayern Munich Have won 6 times in the period of 2000-2010 which is more than 60% in 10 years

Q2. Which teams were relegated in th past 2000 - 2010

In [44]:

```
df_teams = load_dataframe('/Users/ruturajharal/Desktop/Bundesliga-analysis-pyspark/Teams.csv')
df_teams.limit(5).toPandas()
```

Out[44]:

	Season	TeamName	KaderHome	AvgAgeHome	ForeignPlayersHome	OverallMarketValueHome	AvgMarketValueHome	StadiumCapac
0	2017	Bayern Munich	27	26	15	597950000	22150000	75
1	2017	Dortmund	33	25	18	416730000	12630000	81
2	2017	Leverkusen	31	24	15	222600000	7180000	30
3	2017	RB Leipzig	30	23	15	180130000	6000000	42
4	2017	Schalke 04	29	24	17	179550000	6190000	62

In [46]:

```
# lets check for 2010s
relegated = table.filter((col('TeamPosition') == 16) |
                (col('TeamPosition') == 17) |
                (col('TeamPosition') == 18)).orderBy(asc('Season'))
relegated.filter(col('Season') == 2010).toPandas()
```

Out[46]:

	Team	Season	GoalsScored	GoalsAgainst	GoalDifferentials	Win	Loss	Tie	WinPct	TeamPosition
0	Wolfsburg	2010	43.0	48.0	-5.0	9	14	11	26.47	16
1	Ein Frankfurt	2010	31.0	49.0	-18.0	9	18	7	26.47	17
2	St Pauli	2010	35.0	68.0	-33.0	8	21	5	23.53	18

We can see from the above table that team Wolfsburg, Ein Frankfuty and St Pauli have been relegated during season 2010

In [ ]:

In [ ]:

In [ ]:

In [ ]: