DS 6003
Assignment 1, Spark

Andrew Dahbura, amd6ua

Motivation

-Continue using batting data example from previous notebooks.

-Engineer new features

    -%doubles, triples, HR of player's hits

    -RBI per hit and per at bat

    -RBI per games played

-ML pipeline and display evaluation metrics

Code Snippets

In [8]:
```
#Collect first 10 rows
df.take(10)
```

Out[8]:
```
[Row(playerID='colemjo02', yearID=1890, stint=1, teamID='PHI', lgID='NL', G=1, AB=0, R=0, H=0, 2B=0, 3B=0, HR=0, RBI=
0.0, SB=0.0, CS=0.0, BB=0, SO=0.0, IBB=0.0, HBP=0.0, SH=0.0, SF=0.0, GIDP=0.0),
 Row(playerID='brynato01', yearID=1891, stint=1, teamID='BSN', lgID='NL', G=1, AB=0, R=0, H=0, 2B=0, 3B=0, HR=0, RBI=
0.0, SB=0.0, CS=0.0, BB=0, SO=0.0, IBB=0.0, HBP=0.0, SH=0.0, SF=0.0, GIDP=0.0),
 Row(playerID='dunnian01', yearID=1891, stint=1, teamID='NY1', lgID='NL', G=1, AB=0, R=0, H=0, 2B=0, 3B=0, HR=0, RBI=
0.0, SB=0.0, CS=0.0, BB=0, SO=0.0, IBB=0.0, HBP=0.0, SH=0.0, SF=0.0, GIDP=0.0),
 Row(playerID='sulliji01', yearID=1891, stint=1, teamID='BSN', lgID='NL', G=1, AB=0, R=0, H=0, 2B=0, 3B=0, HR=0, RBI=
0.0, SB=0.0, CS=0.0, BB=0, SO=0.0, IBB=0.0, HBP=0.0, SH=0.0, SF=0.0, GIDP=0.0),
 Row(playerID='viaule01', yearID=1892, stint=1, teamID='CL4', lgID='NL', G=1, AB=0, R=0, H=0, 2B=0, 3B=0, HR=0, RBI=
0.0, SB=0.0, CS=0.0, BB=0, SO=0.0, IBB=0.0, HBP=0.0, SH=0.0, SF=0.0, GIDP=0.0),
 Row(playerID='johnsab01', yearID=1893, stint=1, teamID='CHN', lgID='NL', G=1, AB=0, R=0, H=0, 2B=0, 3B=0, HR=0, RBI=
0.0, SB=0.0, CS=0.0, BB=0, SO=0.0, IBB=0.0, HBP=0.0, SH=0.0, SF=0.0, GIDP=0.0),
 Row(playerID='scheija01', yearID=1894, stint=1, teamID='PHI', lgID='NL', G=1, AB=0, R=0, H=0, 2B=0, 3B=0, HR=0, RBI=
0.0, SB=0.0, CS=0.0, BB=0, SO=0.0, IBB=0.0, HBP=0.0, SH=0.0, SF=0.0, GIDP=0.0),
 Row(playerID='sommean01', yearID=1894, stint=1, teamID='BRO', lgID='NL', G=1, AB=0, R=0, H=0, 2B=0, 3B=0, HR=0, RBI=
0.0, SB=0.0, CS=0.0, BB=0, SO=0.0, IBB=0.0, HBP=0.0, SH=0.0, SF=0.0, GIDP=0.0),
 Row(playerID='terryad01', yearID=1894, stint=1, teamID='PIT', lgID='NL', G=1, AB=0, R=0, H=0, 2B=0, 3B=0, HR=0, RBI=
0.0, SB=0.0, CS=0.0, BB=0, SO=0.0, IBB=0.0, HBP=0.0, SH=0.0, SF=0.0, GIDP=0.0),
 Row(playerID='thomato01', yearID=1894, stint=1, teamID='CL4', lgID='NL', G=1, AB=0, R=0, H=0, 2B=0, 3B=0, HR=0, RBI=
0.0, SB=0.0, CS=0.0, BB=0, SO=0.0, IBB=0.0, HBP=0.0, SH=0.0, SF=0.0, GIDP=0.0)]
```

In [9]:
```
#Correlation between number of hits and type of hit -- single, double, triple or HR
print("Pearson's r(H,2B) = {}".format(df.corr("H", "2B")))
print("Pearson's r(H,3B) = {}".format(df.corr("H", "3B")))
print("Pearson's r(H,HR) = {}".format(df.corr("H", "HR")))
```

```
Pearson's r(H,2B) = 0.955159385083123
Pearson's r(H,3B) = 0.7222983866445453
Pearson's r(H,HR) = 0.7989849986141668
```

In [10]:
```
#Additional correlations
print("Pearson's r(G,H) = {}".format(df.corr("G", "H")))
print("Pearson's r(H,RBI) = {}".format(df.corr("H", "RBI")))
print("Pearson's r(AB,RBI) = {}".format(df.corr("AB", "RBI")))
```

```
Pearson's r(G,H) = 0.9179755244596672
Pearson's r(H,RBI) = 0.9405136278412812
Pearson's r(AB,RBI) = 0.931500145781767
```

Code Snippets

In [16]:
```
#Vectorization
from pyspark.ml.linalg import Vectors, VectorUDT

# make a user defined function (udf)
sqlc.registerFunction("oneElementVec", lambda d: Vectors.dense([d]), returnType=VectorUDT())

# vectorize the data frames
train = train.selectExpr("G", "oneElementVec(RBI) as RBI")
test = test.selectExpr("G", "oneElementVec(RBI) as RBI")

print(test.orderBy(test.G.desc()).limit(10))
```

```
DataFrame[G: bigint, RBI: vector]
```

In [17]:
```
# rename to make ML engine happy
train = train.withColumnRenamed("G", "label").withColumnRenamed("RBI", "features")
test = test.withColumnRenamed("G", "label").withColumnRenamed("RBI", "features")
```

Visualization:

RBI’s versus games played