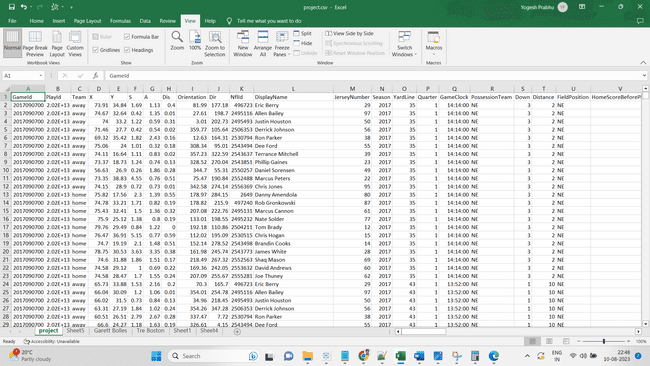
**Problem Statement**: - Design an analytics framework to identify underperforming players in the Arizona Cardinals NFL team for potential trading, while sourcing high-performing players during the trading window.

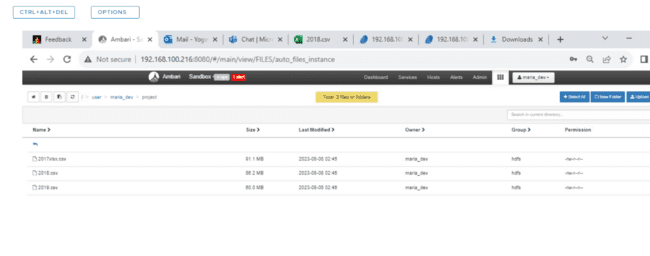
**To understand the performance of an individual player based on the following 3 factors.**

1. Position experience
2. Performance ranking
3. Weather impact
4. Final ranking as per previous 3 factors

**Available data set.**

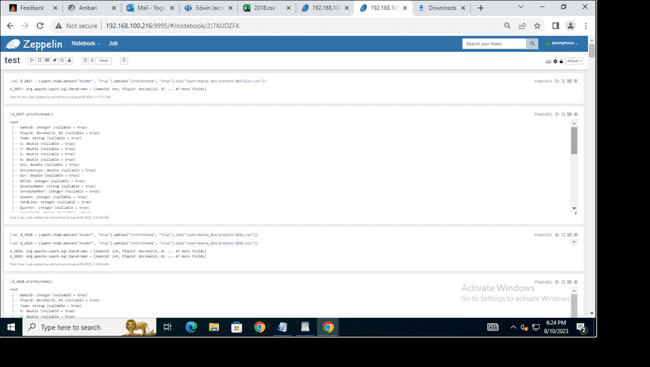


**Uploaded files on HDFS**



**Creating main data frame in zipline to perform the analysis**

* val d\_2017 = (spark.read.option("header", "true").option("inferSchema", "true").csv("/user/maria\_dev/project/2017xlsx.csv"))
* val d\_2018 = (spark.read.option("header", "true").option("inferSchema", "true").csv("/user/maria\_dev/project/2018.csv"))
* val d\_2019 = (spark.read.option("header", "true").option("inferSchema", "true").csv("/user/maria\_dev/project/2019.csv"))



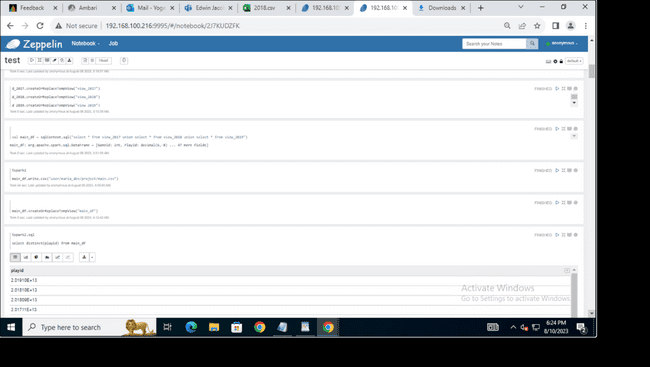
**Created data frames for separate files based on years**

* d\_2017.createOrReplaceTempView("view\_2017")
* d\_2018.createOrReplaceTempView("view\_2018")
* d\_2019.createOrReplaceTempView("view\_2019")
* val main\_df = sqlContext.sql(" select \* from view\_2017 union

select \* from view\_2018 union

select \* from view\_2019")

* main\_df.createOrReplaceTempView("main\_df")
* select \* from Main\_df



**Query to check first factor (Who played most on that particular position and ranked using window function).**

select

displayname,

position,number\_1,

dense\_rank () over ( order by number\_1 desc ) as Rank\_count

from   
(select

displayname,

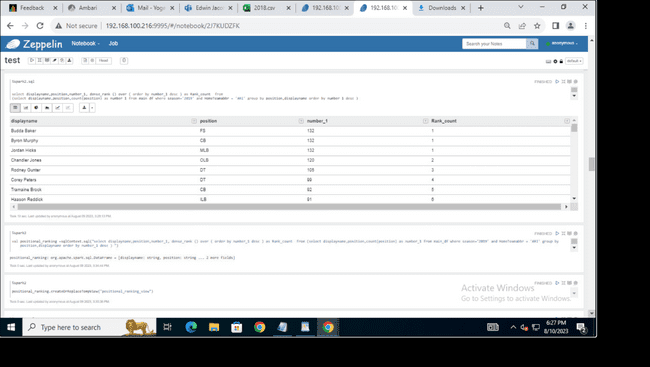
position,

count(position) as number\_1 from main\_df

where season='2019' and HomeTeamabbr = 'ARI'

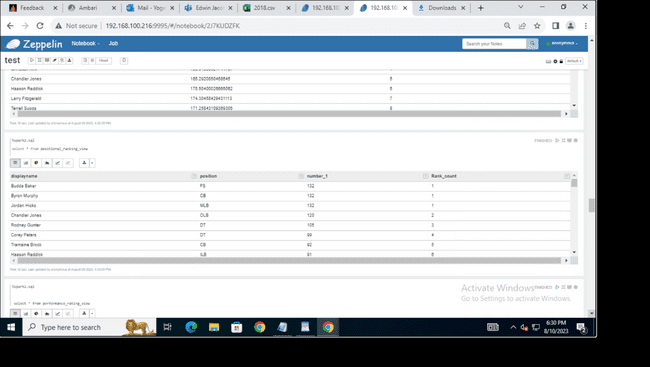
group by position,displayname

order by number\_1 desc )



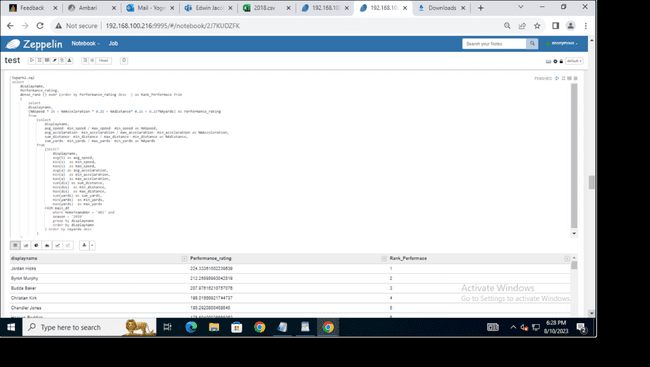
**Created a separate view for Positional ranking view**

* select \* from positional\_ranking\_view



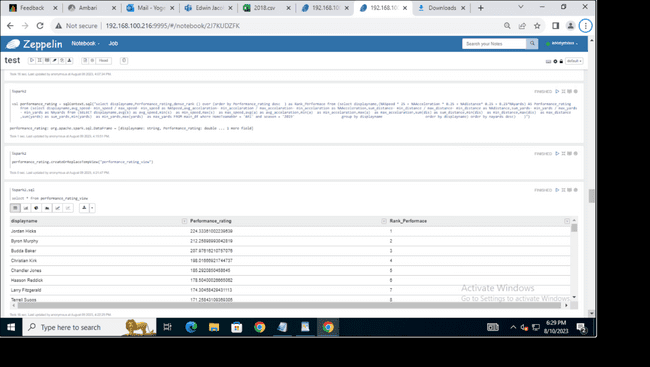
**Query to check second factor. (Performance ranking based on speed, acceleration, yards, and distance. Ranking them based on the performance index)**

select   
    displayname,  
    Performance\_rating,  
    dense\_rank () over (order by Performance\_rating desc  ) as Rank\_Performace from  
    (     
        select  
        displayname,  
        (NASpeed \* 25 + NAAcceleration \* 0.25 + NAdistance\* 0.25 + 0.25\*NAyards) AS Performance\_rating  
        from   
            (select   
                displayname,  
                avg\_speed- min\_speed / max\_speed- min\_speed as NASpeed,  
                avg\_accelaration- min\_accelaration / max\_accelaration- min\_accelaration as NAAcceleration,  
                sum\_distance- min\_distance / max\_distance- min\_distance as NAdistance,  
                sum\_yards- min\_yards / max\_yards- min\_yards as NAyards  
            from   
                (SELECT   
                    displayname,  
                    avg(S) as avg\_speed,  
                    min(s)  as min\_speed,  
                    max(s)  as max\_speed,  
                    avg(a) as avg\_accelaration,  
                    min(a)  as min\_accelaration,  
                    max(a)  as max\_accelaration,  
                    sum(dis) as sum\_distance,  
                    min(dis)  as min\_distance,  
                    max(dis)  as max\_distance,  
                    sum(yards) as sum\_yards,  
                    min(yards)  as min\_yards,  
                    max(yards)  as max\_yards  
                FROM main\_df  
                    where HomeTeamabbr = 'ARI' and   
                    season = '2019'     
                    group by displayname   
                    order by displayname  
                ) order by nayards desc  
            )



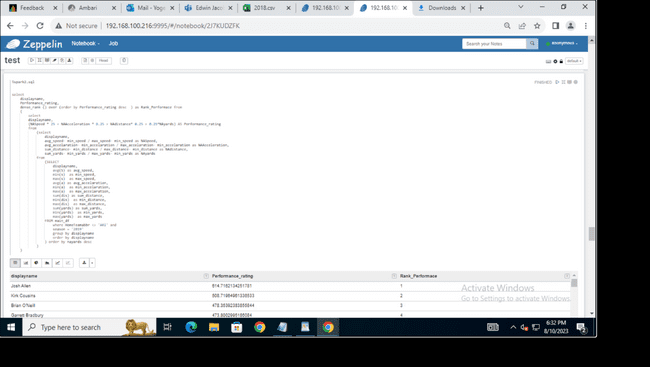
**Created a separate view for performance ranking**

* select \* from performance\_rating\_view



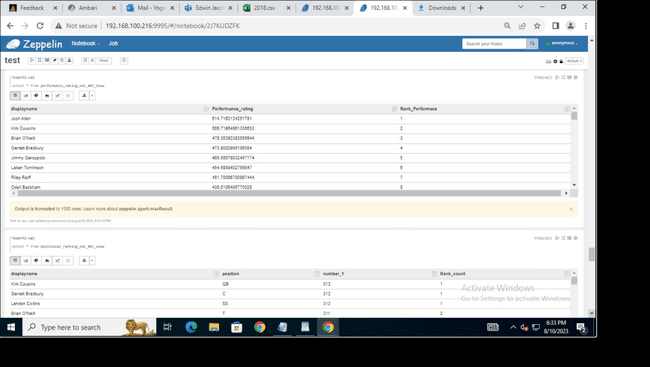
**Query to check second factor for player who are not in playing 11. (Performance ranking based on speed, acceleration, yards, and distance. Ranking them based on the performance index)**

select   
    displayname,  
    Performance\_rating,  
    dense\_rank () over (order by Performance\_rating desc  ) as Rank\_Performace from  
    (     
        select  
        displayname,  
        (NASpeed \* 25 + NAAcceleration \* 0.25 + NAdistance\* 0.25 + 0.25\*NAyards) AS Performance\_rating  
        from   
            (select   
                displayname,  
                avg\_speed- min\_speed / max\_speed- min\_speed as NASpeed,  
                avg\_accelaration- min\_accelaration / max\_accelaration- min\_accelaration as NAAcceleration,  
                sum\_distance- min\_distance / max\_distance- min\_distance as NAdistance,  
                sum\_yards- min\_yards / max\_yards- min\_yards as NAyards  
            from   
                (SELECT   
                    displayname,  
                    avg(S) as avg\_speed,  
                    min(s)  as min\_speed,  
                    max(s)  as max\_speed,  
                    avg(a) as avg\_accelaration,  
                    min(a)  as min\_accelaration,  
                    max(a)  as max\_accelaration,  
                    sum(dis) as sum\_distance,  
                    min(dis)  as min\_distance,  
                    max(dis)  as max\_distance,  
                    sum(yards) as sum\_yards,  
                    min(yards)  as min\_yards,  
                    max(yards)  as max\_yards  
                FROM main\_df  
                    where HomeTeamabbr <> 'ARI' and   
                    season = '2019'     
                    group by displayname   
                    order by displayname  
                ) order by nayards desc  
            )



**Query to check performace ranking for player who are not in playing 11**

select \* from performance\_rating\_not\_ARI\_View



**Created a separate view of weather impact for player who are not in playing11.**

select

displayname,

weather\_impact,

dense\_rank() over(order by weather\_impact desc ) as ranking\_weather\_impact

from

(Select displayname,

((avg\_speed+(0.6\*avg\_temperature) +(0.2\*avg\_humidity)+(0.2\*avg\_windspeed))\*0.5 +(avg\_accelaration+(0.6\*avg\_temperature)+(0.2\*avg\_humidity)+(0.2\*avg\_windspeed))\*0.5) as weather\_impact

from (

SELECT

displayname,

avg(S) as avg\_speed,

min(s) as min\_speed,

max(s) as max\_speed,

avg(a) as avg\_accelaration,

min(a) as min\_accelaration,

max(a) as max\_accelaration,

ifnull(avg(temperature),68) as avg\_temperature,

ifnull(avg(humidity),56) as avg\_humidity,

ifnull(avg(windspeed),8) as avg\_windspeed

FROM main\_df

where HomeTeamabbr <> 'ARI' and season = '2019'

group by displayname

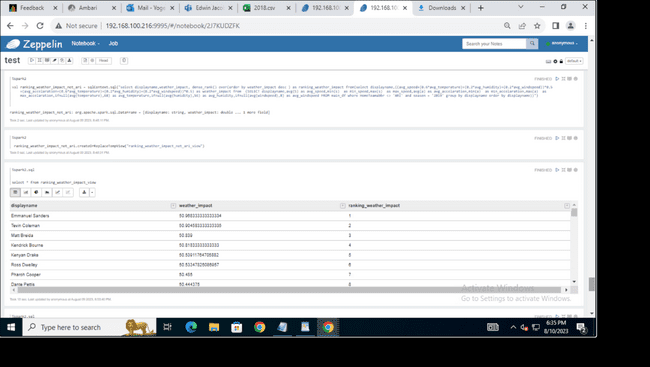
order by displayname

)

)

**Query to check the created view of performance ranking for players who are not in playing 11.**

select \* from ranking\_weather\_impact\_not\_ari\_view



**Query to check overall rank if we consider all three factors Position ranking, performance ranking and weather impact on players who are playing11.**

select name,

final\_rank,

dense\_rank() over(order by final\_rank )

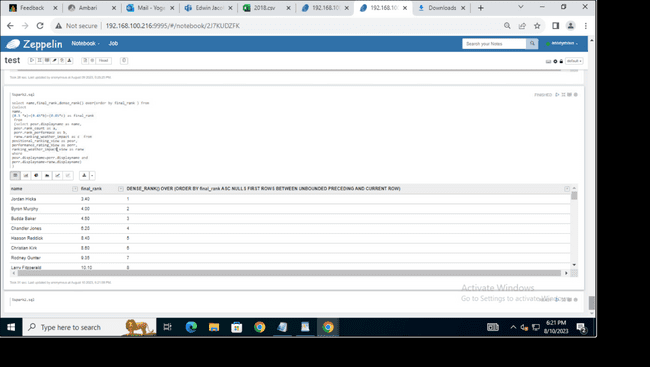
from (select name,  
 (0.5 \*a)+(0.45\*b)+(0.05\*c) as final\_rank  
 from  
 (

select

posr.displayname as name,   
posr.rank\_count as a,  
perr.rank\_performace as b,  
ranw.ranking\_weather\_impact as c

from   
positional\_ranking\_view as posr,  
performance\_rating\_View as perr,   
ranking\_weather\_impact\_view as ranw   
where   
posr.displayname=perr.displayname and   
perr.displayname=ranw.displayname

)  
 )



**Query to check overall rank if we consider all three factors Position ranking, performance ranking and weather impact on players who are not playing11.**

select name,

final\_rank,

dense\_rank() over(order by final\_rank )

from (select name,

(0.5 \*a)+(0.45\*b)+(0.05\*c) as final\_rank

from (

select posr.displayname as name,

posr.rank\_count as a,

perr.rank\_performace as b,

ranw.ranking\_weather\_impact as c

from

positional\_ranking\_not\_ari\_view as posr,

performance\_rating\_not\_ari\_View as perr,

ranking\_weather\_impact\_not\_ari\_view as ranw

where posr.displayname=perr.displayname and perr.displayname=ranw.displayname

) )