**Confusion matrix multiclass 3 class**

%pyspark

from pyspark.sql import SQLContext

sc.\_jsc.hadoopConfiguration().set("fs.s3a.endpoint", "es-si-s3-z2.eecloud.nsn-net.net")

sc.\_jsc.hadoopConfiguration().set("fs.s3a.access.key", "9LHONDBX17WLEA8SCG1S")

sc.\_jsc.hadoopConfiguration().set("fs.s3a.secret.key", "mE39w2rXZkfNCMDDYHqslRQlBDZM9lSNp82HMvZ5")

sqlCtx = SQLContext(sc)

path="s3a://csd-cell-site-degradation/development/csd-telefonica/data-lake/data\_science/temp\_model/highly\_degraded/"

kpi\_name="3g\_dpcr\_voice\_15days\_gteq2\_1000\_Features"

df\_predictions = sqlCtx.read.load("{}{}/batch1/cassandra\_based/predictions/\*.csv".format(path, kpi\_name), format="com.databricks.spark.csv", header="true", inferschema="true")

predicted\_cell\_as\_highly\_degraded= df\_predictions.select("predicted\_class").where(df\_predictions.predicted\_class=='Highly-Degraded').count()

non\_degraded\_cell = df\_predictions.select("actual\_class").where(df\_predictions.actual\_class=='Non-Degraded').count()

highly\_degraded\_cell = df\_predictions.select("actual\_class").where(df\_predictions.actual\_class=='Highly-Degraded').count()

relevant\_prediction = df\_predictions.where((df\_predictions.actual\_class=='Highly-Degraded') & (df\_predictions.predicted\_class=='Highly-Degraded')).count()

precision= float(relevant\_prediction)/float(predicted\_cell\_as\_highly\_degraded)

recall= float(relevant\_prediction)/float(highly\_degraded\_cell)

print(precision)

print(recall)

print(non\_degraded\_cell)

print(highly\_degraded\_cell)

print(predicted\_cell\_as\_highly\_degraded)

print(relevant\_prediction)