%pyspark

from pyspark.sql.types import StructType, StructField, IntegerType, StringType

from pyspark.sql.functions import from\_unixtime, regexp\_replace

from pyspark.ml.feature import Tokenizer, StopWordsRemover, NGram

from pyspark.sql.functions import explode, udf, avg, sum

from pyspark.sql.functions import lower

from pyspark.sql.functions import lit

data = spark.read.csv("s3://msbabana2952020/data.csv", sep = "|", header = True)

#clean data types

data = data.withColumn("ups\_int", data["ups"].cast(IntegerType()))

data = data.withColumn("downs\_int", data["downs"].cast(IntegerType()))

data = data.withColumn("score\_int", data["score"].cast(IntegerType()))

data = data.withColumn("created\_utc", data["created\_utc"].cast(IntegerType()))

#extract dates

data = data.withColumn("created\_date", from\_unixtime(data["created\_utc"], "yyyy-MM-dd HH:mm:ss"))

data = data.withColumn("created\_month", from\_unixtime(data["created\_utc"], "MM"))

data = data.withColumn("created\_day", from\_unixtime(data["created\_utc"], "dd"))

data = data.withColumn("created\_hour", from\_unixtime(data["created\_utc"], "HH"))

data = data.withColumn("created\_dayOfWeek", from\_unixtime(data["created\_utc"], "E"))

data.createOrReplaceTempView("reddit")

data.columns

%pyspark

mov\_data = spark.read.csv("s3://msbabana2952020/movies\_may\_2015\_gbo.csv", sep = ",", header = True)

mov\_data.columns

%pyspark

movies\_df = spark.sql("select \* from reddit where subreddit = 'movies'")

movies\_df.createOrReplaceTempView("movies")

sqlContext.cacheTable("movies")

#376601 rows

%pyspark

mov\_names = mov\_data.select('movie\_name').distinct()

mov\_names = mov\_names.withColumn("movie\_name\_lower", lower(mov\_names["movie\_name"]))

mov\_names = mov\_names.withColumn("movie\_name\_lower\_esc", regexp\_replace("movie\_name\_lower", "'", "''"))

mov\_names = mov\_names.collect()

%pyspark

field = [StructField("created\_date",StringType(), True), StructField("ups\_int", IntegerType(), True), StructField("downs\_int", IntegerType(), True), StructField("score\_int", IntegerType(), True), StructField("body", StringType(), True), StructField("movie\_name",StringType(), True), StructField("bod\_esc", StringType(), True) ]

schema = StructType(field)

all\_posts = spark.createDataFrame(sc.emptyRDD(), schema)

for row in mov\_names:

mov\_posts = spark.sql("select created\_date, ups\_int, downs\_int, score\_int, body from movies where lower(body) like '%{0}%'".format(row["movie\_name\_lower\_esc"]))

mov\_posts = mov\_posts.withColumn("movie\_name", lit(row["movie\_name"]))

mov\_posts = mov\_posts.withColumn("body\_esc", regexp\_replace("body", "[^A-Za-z0-9\_\s]", ""));

all\_posts = all\_posts.union(mov\_posts)

all\_posts = all\_posts.drop('body')

%pyspark

all\_posts.coalesce(1).write.csv("s3://msbabana2952020/movies\_may\_2015\_posts.csv", header="true")