import html

schema = "polarity FLOAT, id LONG, date\_time TIMESTAMP, query STRING, user STRING, text STRING"

timestampformat = "EEE MMM dd HH:mm:ss zzz yyyy"

spark.sql("set spark.sql.legacy.timeParserPolicy=LEGACY")

IN\_PATH\_RAW = "/kaggle/…”

IN\_PATH\_TEST = "/kaggle/…"

*#OUT\_PATH\_CLEAN = "CLEAN"*

spark\_reader = spark.read.schema(schema)

user\_regex = r"(@\w{1,15})"

hashtag\_regex = "(#\w{1,})"

url\_regex=r"((https?|ftp|file):\/{2,3})+([-\w+&@#/%=~|$?!:,.]\*)|(www.)+([-\w+&@#/%=~|$?!:,.]\*)"

email\_regex=r"[\w.-]+@[\w.-]+\.[a-zA-Z]{1,}"

@f.udf

def html\_unescape(s: str):

if isinstance(s, str):

return html.unescape(s)

return s

def clean\_data(df):

df = (

df

.withColumn("original\_text", f.col("text"))

.withColumn("text", f.regexp\_replace(f.col("text"), url\_regex, ""))

.withColumn("text", f.regexp\_replace(f.col("text"), email\_regex, ""))

.withColumn("text", f.regexp\_replace(f.col("text"), user\_regex, ""))

.withColumn("text", f.regexp\_replace(f.col("text"), "#", " "))

.withColumn("text", html\_unescape(f.col("text")))

.filter("text != ''")

)

return df

df\_train\_raw = spark\_reader.csv(IN\_PATH\_RAW, timestampFormat=timestampformat)

df\_train\_clean = clean\_data(df\_train\_raw)

df\_test\_raw = spark\_reader.csv(IN\_PATH\_TEST, timestampFormat=timestampformat)

df\_test\_clean = clean\_data(df\_test\_raw)