1-Outbrain-ValidationSetSplit

January 21, 2022

[1]:
```
OUTPUT_BUCKET_FOLDER = "gs://akhilbucket/outbrain-click-prediction/output/"
DATA_BUCKET_FOLDER = "gs://akhilbucket/data/"
```

[2]:
```
from pyspark.sql.types import *
import pyspark.sql.functions as F
from pyspark.shell import spark
```

```
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
setLogLevel(newLevel).

Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.2
      /_/

Using Python version 3.8.12 (default, Oct 12 2021 21:59:51)
Spark context Web UI available at http://cluster-2039-m.us-central1-b.c.thesis-
practical-work.internal:41751
Spark context available as 'sc' (master = yarn, app id =
application_1642270375158_0007).
SparkSession available as 'spark'.
```

0.1 Loading data

[3]:
```
truncate_day_from_timestamp_udf = F.udf(lambda ts: int(ts / 1000 / 60 / 60 /␣
↪24), IntegerType())
```

[4]:
```
events_schema = StructType(
                    [StructField("display_id", IntegerType(), True),
                    StructField("uuid_event", StringType(), True),
                    StructField("document_id_event", IntegerType(), True),
                    StructField("timestamp_event", IntegerType(), True),
                    StructField("platform_event", IntegerType(), True),
                    StructField("geo_location_event", StringType(), True)]
                    )
```

```
events_df = spark.read.schema(events_schema).options(header='true',␣
↪inferschema='false', nullValue='\\N') \
                .csv(DATA_BUCKET_FOLDER + "events.csv") \
                .withColumn('day_event',␣
↪truncate_day_from_timestamp_udf('timestamp_event')) \
                .alias('events')
```

[ ]:

[5]:
```
promoted_content_schema = StructType(
                    [StructField("ad_id", IntegerType(), True),
                    StructField("document_id_promo", IntegerType(), True),
                    StructField("campaign_id", IntegerType(), True),
                    StructField("advertiser_id", IntegerType(), True)]
                    )

promoted_content_df = spark.read.schema(promoted_content_schema).
↪options(header='true', inferschema='false', nullValue='\\N') \
                .csv(DATA_BUCKET_FOLDER+"promoted_content.csv") \
                .alias('promoted_content')
```

[6]:
```
clicks_train_schema = StructType(
                    [StructField("display_id", IntegerType(), True),
                    StructField("ad_id", IntegerType(), True),
                    StructField("clicked", IntegerType(), True)]
                    )

clicks_train_df = spark.read.schema(clicks_train_schema).options(header='true',␣
↪inferschema='false', nullValue='\\N') \
                .csv(DATA_BUCKET_FOLDER+"clicks_train.csv") \
                .alias('clicks_train')
```

[7]:
```
clicks_train_joined_df = clicks_train_df \
                        .join(promoted_content_df, on='ad_id', how='left') \
                        .join(events_df, on='display_id', how='left')
clicks_train_joined_df.createOrReplaceTempView('clicks_train_joined')
```

[8]:
```
validation_display_ids_df = clicks_train_joined_df.
↪select('display_id','day_event').distinct() \
                                .sampleBy("day_event", fractions={0: 0.2, 1: 0.
↪2, 2: 0.2, 3: 0.2, 4: 0.2, \
```

```
                                                                  5: 0.2, 6: 0.2,␣
↪7: 0.2, 8: 0.2, 9: 0.2, 10: 0.2, \
                                                                 11: 1.0, 12: 1.
↪0}, seed=0)
validation_display_ids_df.createOrReplaceTempView("validation_display_ids")
```

[9]:
```
validation_set_df = spark.sql('''SELECT display_id, ad_id, uuid_event,␣
↪day_event, timestamp_event,
                                        document_id_promo, platform_event,␣
↪geo_location_event FROM clicks_train_joined t
            WHERE EXISTS (SELECT display_id FROM validation_display_ids
                          WHERE display_id = t.display_id)''')
```

[10]:
```
validation_set_gcs_output = "validation_set.parquet"
validation_set_df.write.parquet(OUTPUT_BUCKET_FOLDER+validation_set_gcs_output,␣
↪mode='overwrite')
```

[11]:
```
validation_set_df.take(5)
```

[11]:
```
[Row(display_id=65, ad_id=57971, uuid_event='68eea10a276986', day_event=0,
timestamp_event=5163, document_id_promo=998835, platform_event=3,
geo_location_event='US>CO>751'),
 Row(display_id=65, ad_id=137800, uuid_event='68eea10a276986', day_event=0,
timestamp_event=5163, document_id_promo=1055593, platform_event=3,
geo_location_event='US>CO>751'),
 Row(display_id=392, ad_id=50452, uuid_event='c148fad0872fcb', day_event=0,
timestamp_event=29288, document_id_promo=906372, platform_event=3,
geo_location_event='US>NY>532'),
 Row(display_id=392, ad_id=153427, uuid_event='c148fad0872fcb', day_event=0,
timestamp_event=29288, document_id_promo=987161, platform_event=3,
geo_location_event='US>NY>532'),
 Row(display_id=392, ad_id=186872, uuid_event='c148fad0872fcb', day_event=0,
timestamp_event=29288, document_id_promo=1444890, platform_event=3,
geo_location_event='US>NY>532')]
```