01_Data_Ingestion

April 13, 2022

1 Data Ingestion and Preparation

1.1 Create Athena Database Schema

[1]:

```
import pandas as pd
import boto3
import sagemaker

sess = sagemaker.Session()
bucket = sess.default_bucket()
role = sagemaker.get_execution_role()
region = boto3.Session().region_name
```

1.2 Import PyAthena

[2]:

```
!pip install --disable-pip-version-check -q PyAthena==2.1.0
from pyathena import connect
```

```
/opt/conda/lib/python3.7/site-packages/secretstorage/dhcrypto.py:16:
CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes
instead
  from cryptography.utils import int_from_bytes
/opt/conda/lib/python3.7/site-packages/secretstorage/util.py:25:
CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes
instead
  from cryptography.utils import int_from_bytes
WARNING: Running pip as the 'root' user can result in broken permissions

and conflicting behaviour with the system package manager. It is recommended to

use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

1.3 Set Private bucket

[3]:

```
s3_private_path = "s3://ads508-team4-raw"
```

1.4 List files in the Private bucket

[4]:

```
!aws s3 ls $s3_private_path
```

```
                           PRE assets/
                           PRE demographics/
                           PRE plays/
                           PRE psychographics/
                           PRE users/
```

1.5 Create Athena Database and Tables

[5]:

```
database_name = "ads508team4"
table_1 = "assets"
table_2 = "plays"
table_3 = "users"
table_4 = "demographics"
table_5 = "psychographics"
```

[6]:

```
# Set S3 staging directory -- this is a temporary directory used for Athena␣
 ↪queries
s3_staging_dir = "s3://{0}/athena/staging".format(bucket)
```

[7]:

```
conn = connect(region_name=region, s3_staging_dir=s3_staging_dir)
```

[8]:

```
statement = "CREATE DATABASE IF NOT EXISTS {}".format(database_name)
```

[9]:

```
pd.read_sql(statement, conn)
```

[9]:

```
Empty DataFrame
Columns: []
Index: []
```

1.6 Make sure the database is created

[10]:

```
statement = "SHOW DATABASES"

df_show = pd.read_sql(statement, conn)
df_show.head(5)
```

[10]:

```
  database_name
0   ads508team4
1       default
2        dsoaws
```

[11]:

```
# Create table assets
create_table_1 = """CREATE EXTERNAL TABLE IF NOT EXISTS ads508team4.assets
```

```
(
         showtype string,
         genre string,
         running_minutes int,
         source_language string,
         asset_id int,
         season_id int,
         series_id int,
         studio_id int

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LINES TERMINATED BY '\\n'
STORED AS TEXTFILE
LOCATION 's3://ads508-team4-raw/assets/'
TBLPROPERTIES ('skip.header.line.count'='1')""".format(database_name, table_1)

pd.read_sql(create_table_1, conn)
pd.read_sql("""SELECT * FROM ads508team4.assets LIMIT 10""", conn)
```

[11]:

```
  showtype                      genre  running_minutes source_language  \
0   Movies                     Sci-Fi              146         English
1       TV  Documentary and Biography               43         English
2       TV                    Reality               22         English
3       TV                    Reality               22         English
4       TV                    Reality               22         English
5       TV                     Comedy               23         English
6       TV                     Comedy               23         English
7       TV                     Comedy               23         English
8       TV                       Kids               12         English
9       TV                     Comedy               19         English

   asset_id  season_id  series_id  studio_id
0         1        NaN        NaN        325
1         2        4.0        5.0          7
2         3       15.0       22.0        442
3         4       15.0       22.0        442
4         5       15.0       22.0        442
5         6       12.0       20.0        397
6         7       13.0       20.0        397
7         8       13.0       20.0        397
8         9       50.0        6.0         47
9        10       35.0       41.0        442
```

[12]:

```
# Create table plays
create_table_2 = """CREATE EXTERNAL TABLE IF NOT EXISTS ads508team4.plays
```

```
(
         user_id double,
         platform string,
         asset_id int,
         minutes_viewed int

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LINES TERMINATED BY '\\n'
STORED AS TEXTFILE
LOCATION 's3://ads508-team4-raw/plays/'
TBLPROPERTIES ('skip.header.line.count'='1')""".format(database_name, table_2)

pd.read_sql(create_table_2, conn)
pd.read_sql("""SELECT * FROM ads508team4.plays LIMIT 10""", conn)
```

[12]:

```
        user_id platform  asset_id  minutes_viewed
0  7.650000e+11  android     13758              28
1  4.120000e+11  android     13825              28
2  1.500000e+12      iOS        93             105
3  4.900000e+11  android      6226               7
4  6.871948e+10  android      3762               1
5  2.580000e+11  android      4673              44
6  1.240000e+12  android     10526               1
7  1.080000e+12  android     14441               0
8  1.220000e+12  android      4808              28
9  7.560000e+11  android     15019              11
```

[13]:

```
# Create table users
create_table_3 = """CREATE EXTERNAL TABLE IF NOT EXISTS ads508team4.users

(
         user_id double,
         country_code string

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LINES TERMINATED BY '\\n'
STORED AS TEXTFILE
LOCATION 's3://ads508-team4-raw/users/'
TBLPROPERTIES ('skip.header.line.count'='1')""".format(database_name, table_3)

pd.read_sql(create_table_3, conn)
pd.read_sql("""SELECT * FROM ads508team4.users LIMIT 10""", conn)
```

```
[13]:         user_id country_code
      0  7.816840e+11           ID
      1  7.816840e+11           MY
      2  7.816840e+11           ID
      3  7.816840e+11           ID
      4  7.816840e+11           ID
      5  7.816840e+11           ID
      6  7.816840e+11           ID
      7  7.816840e+11           ID
      8  7.816840e+11           MY
      9  7.816840e+11           ID
```

```
[14]: # Create table demographics
      create_table_4 = """CREATE EXTERNAL TABLE IF NOT EXISTS ads508team4.demographics

      (
               user_id double,
               platform string,
               level_1 string,
               level_2 string,
               level_3 string,
               confidence_score float

      )

      ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
      LINES TERMINATED BY '\\n'
      STORED AS TEXTFILE
      LOCATION 's3://ads508-team4-raw/demographics/'
      TBLPROPERTIES ('skip.header.line.count'='1')""".format(database_name, table_4)

      pd.read_sql(create_table_4, conn)
      pd.read_sql("""SELECT * FROM ads508team4.demographics LIMIT 10""", conn)
```

```
[14]:         user_id platform       level_1 level_2 level_3  confidence_score
      0  1.720000e+11  android  Demographics  Income  Medium               1.0
      1  3.260000e+11  android  Demographics  Income  Medium               1.0
      2  1.717987e+10  android  Demographics  Income  Medium               1.0
      3  9.960000e+11  android  Demographics  Income     Low               1.0
      4  1.610000e+12  android  Demographics  Income     Low               1.0
      5  1.280000e+12      iOS  Demographics  Income    High               1.0
      6  1.280000e+12  android  Demographics  Income    High               1.0
      7  8.589935e+09  android  Demographics  Income     Low               1.0
      8  1.560000e+12  android  Demographics  Income     Low               1.0
      9  9.020000e+11  android  Demographics  Income     Low               1.0
```

[15]:

```
# Create table psychographics
create_table_5 = """CREATE EXTERNAL TABLE IF NOT EXISTS ads508team4.
 ↪psychographics

(
         user_id double,
         platform string,
         level_1 string,
         level_2 string,
         level_3 string,
         confidence_score float

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LINES TERMINATED BY '\\n'
STORED AS TEXTFILE
LOCATION 's3://ads508-team4-raw/psychographics/'
TBLPROPERTIES ('skip.header.line.count'='1')""".format(database_name, table_5)

pd.read_sql(create_table_5, conn)
pd.read_sql("""SELECT * FROM ads508team4.psychographics LIMIT 10""", conn)
```

[15]:

```
        user_id platform         level_1             level_2          level_3  \
0  4.290000e+11  android  Psychographics  Mobile Enthusiasts  High Data Users
1  3.090000e+11  android  Psychographics  Mobile Enthusiasts  High Data Users
2  8.680000e+11  android  Psychographics  Mobile Enthusiasts  High Data Users
3  1.380000e+12  android  Psychographics  Mobile Enthusiasts  High Data Users
4  1.280000e+12  android  Psychographics  Mobile Enthusiasts  High Data Users
5  1.550000e+11  android  Psychographics  Mobile Enthusiasts  High Data Users
6  1.020000e+12  android  Psychographics  Mobile Enthusiasts  High Data Users
7  1.420000e+12  android  Psychographics  Mobile Enthusiasts  High Data Users
8  1.430000e+12  android  Psychographics  Mobile Enthusiasts  High Data Users
9  1.370000e+11  android  Psychographics  Mobile Enthusiasts  High Data Users

   confidence_score
0              0.93
1              0.39
2              0.16
3              0.74
4              0.08
5              0.77
6              0.31
7              0.59
8              0.76
9              0.59
```

1.7 Check and make sure the tables are created

[16]:

```
statement = "SHOW TABLES in {}".format(database_name)

df_show = pd.read_sql(statement, conn)
df_show.head(5)
```

[16]:

```
        tab_name
0         assets
1   demographics
2          plays
3 psychographics
4          users
```

1.8 Create a sub-master table for demographics (includes everything except for psychographics info)

[32]:

```
plays = """SELECT * FROM {}.plays""".format(database_name)

users = """SELECT * FROM {}.users""".format(database_name)

assets = """SELECT * FROM {}.assets""".format(database_name)

demographics = """SELECT * FROM {}.demographics""".format(database_name)

psychographics = """SELECT * FROM {}.psychographics""".format(database_name)

df_plays = pd.read_sql(plays, conn)
df_users = pd.read_sql(users, conn)
df_assets = pd.read_sql(assets, conn)
df_demographics = pd.read_sql(demographics, conn)
df_psychographics = pd.read_sql(psychographics, conn)

result_1 = pd.merge(df_demographics, df_users, how='inner', on='user_id')
result_2 = pd.merge(result_1, df_plays, how='inner', on='user_id')
```

[36]:

```
result_demo = pd.merge(result_2, df_assets, how='inner', on='asset_id')
```

[37]:

```
result_demo.head()
```

[37]:

```
        user_id platform_x       level_1 level_2 level_3  confidence_score  \
0  1.717987e+10    android  Demographics  Income  Medium          1.000000
1  3.435974e+10    android  Demographics  Income  Medium          1.000000
2  1.717987e+10    android  Demographics  Income     Low          1.000000
3  1.717987e+10    android  Demographics     Age 25 - 34          1.000000
4  1.717987e+10    android  Demographics  Gender    Male          0.993641
```

```
  country_code platform_y  asset_id  minutes_viewed showtype  \
0           PH    android     14707              55   Movies
1           PH    android     14707              92   Movies
2           PH    android     14707              76   Movies
3           PH    android     14707              76   Movies
4           PH    android     14707              76   Movies

                  genre  running_minutes source_language  season_id  \
0  Action and Adventure              103         Tagalog        NaN
1  Action and Adventure              103         Tagalog        NaN
2  Action and Adventure              103         Tagalog        NaN
3  Action and Adventure              103         Tagalog        NaN
4  Action and Adventure              103         Tagalog        NaN

   series_id  studio_id
0        NaN      448.0
1        NaN      448.0
2        NaN      448.0
3        NaN      448.0
4        NaN      448.0
```

1.9 Create a sub-master table for psychographics

[38]:

```
result_a = pd.merge(df_psychographics, df_users, how='inner', on='user_id')
result_b = pd.merge(result_a, df_plays, how='inner', on='user_id')
result_psych = pd.merge(result_b, df_assets, how='inner', on='asset_id')

result_psych.head()
```

[38]:

```
        user_id platform_x         level_1        level_2  \
0  8.589935e+10  web-embed  Psychographics  Movies Lovers
1  8.589935e+10  web-embed  Psychographics  Movies Lovers
2  2.576980e+10    android  Psychographics  Movies Lovers
3  2.576980e+10    android  Psychographics      TV Lovers
4  2.576980e+10    android  Psychographics      TV Lovers

                  level_3  confidence_score country_code platform_y  asset_id  \
0      Horror Movies Fans              0.07           ID  web-embed     10377
1  Indonesian Movies Fans              0.03           ID  web-embed     10377
2     Romance Movies Fans              0.52           ID    android     10377
3            Kids TV Fans              0.61           ID    android     10377
4           Drama TV Fans              0.60           ID    android     10377

   minutes_viewed showtype   genre  running_minutes source_language  \
0               1   Movies  Horror               87      Indonesian
1               1   Movies  Horror               87      Indonesian
2               3   Movies  Horror               87      Indonesian
```

```
3               3   Movies  Horror              87      Indonesian
4               3   Movies  Horror              87      Indonesian

   season_id  series_id  studio_id
0        NaN        NaN      350.0
1        NaN        NaN      350.0
2        NaN        NaN      350.0
3        NaN        NaN      350.0
4        NaN        NaN      350.0
```

[38]:
```
result_a = pd.merge(df_psychographics, df_users, how='inner', on='user_id')
result_b = pd.merge(result_a, df_plays, how='inner', on='user_id')
result_psych = pd.merge(result_b, df_assets, how='inner', on='asset_id')

result_psych.head()
```

[38]:
```
        user_id platform_x         level_1        level_2  \
0  8.589935e+10  web-embed  Psychographics  Movies Lovers
1  8.589935e+10  web-embed  Psychographics  Movies Lovers
2  2.576980e+10    android  Psychographics  Movies Lovers
3  2.576980e+10    android  Psychographics      TV Lovers
4  2.576980e+10    android  Psychographics      TV Lovers

                 level_3  confidence_score country_code platform_y  asset_id  \
0      Horror Movies Fans              0.07           ID  web-embed     10377
1  Indonesian Movies Fans              0.03           ID  web-embed     10377
2     Romance Movies Fans              0.52           ID    android     10377
3            Kids TV Fans              0.61           ID    android     10377
4           Drama TV Fans              0.60           ID    android     10377

   minutes_viewed showtype   genre  running_minutes source_language  \
0               1   Movies  Horror              87      Indonesian
1               1   Movies  Horror              87      Indonesian
2               3   Movies  Horror              87      Indonesian
3               3   Movies  Horror              87      Indonesian
4               3   Movies  Horror              87      Indonesian

   season_id  series_id  studio_id
0        NaN        NaN      350.0
1        NaN        NaN      350.0
2        NaN        NaN      350.0
3        NaN        NaN      350.0
4        NaN        NaN      350.0
```

1.10 Create a new S3 bucket to upload our 2 master file: result_demo and result_psych

[39]:

```
!aws s3 mb s3://ads508-team4-master
```

```
make_bucket: ads508-team4-master
```

[45]:

```
from io import StringIO

bucket = 'ads508-team4-master'
csv_buffer1 = StringIO()
csv_buffer2 = StringIO()

result_demo.to_csv(csv_buffer1)
result_psych.to_csv(csv_buffer2)
s3_resource = boto3.resource('s3')
s3_resource.Object(bucket, 'result_demo.csv').put(Body=csv_buffer1.getvalue())
s3_resource.Object(bucket, 'result_psych.csv').put(Body=csv_buffer2.getvalue())
```

[45]:

```
{'ResponseMetadata': {'RequestId': 'MT8MSJJHP2HNN6VT',
  'HostId':
'sX+T7AjA4+LIy1xU6OwP1tegL7mdfNa4WzrUBJd5o55pwDSOv4Wc3kM+P5zmqM4iwLeOn9Fmkfc=',
  'HTTPStatusCode': 200,
  'HTTPHeaders': {'x-amz-id-2':
'sX+T7AjA4+LIy1xU6OwP1tegL7mdfNa4WzrUBJd5o55pwDSOv4Wc3kM+P5zmqM4iwLeOn9Fmkfc=',
   'x-amz-request-id': 'MT8MSJJHP2HNN6VT',
   'date': 'Mon, 21 Mar 2022 20:40:22 GMT',
   'etag': '"151c915b09d52d5b8a652dd15cb78f70"',
   'server': 'AmazonS3',
   'content-length': '0'},
  'RetryAttempts': 0},
 'ETag': '"151c915b09d52d5b8a652dd15cb78f70"'}
```