00b_S3_Setup_Final

April 14, 2023

1 ADS-508-01-SP23 Team 8: Final Project

2 Setup Database and Athena Tables

Much of the code is modified from `Fregly, C., & Barth, A. (2021). Data science on AWS: Implementing end-to-end, continuous AI and machine learning pipelines. O'Reilly.`

2.1 Install missing dependencies

PyAthena is a Python DB API 2.0 (PEP 249) compliant client for Amazon Athena.

[2]:
```
!pip install --disable-pip-version-check -q PyAthena==2.1.0
```

```
WARNING: The directory '/root/.cache/pip' or its parent directory is not
owned or is not writable by the current user. The cache has been disabled. Check
the permissions and owner of that directory. If executing pip with sudo, you
should use sudo's -H flag.
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
```

2.2 Globally import libraries

[3]:
```
import boto3
from botocore.client import ClientError
import sagemaker
import pandas as pd
from pyathena import connect
from IPython.core.display import display, HTML

%matplotlib inline
```

2.3 Instantiate AWS SageMaker session

[4]:
```
session = boto3.session.Session()
region = session.region_name
sagemaker_session = sagemaker.Session()
def_bucket = sagemaker_session.default_bucket()
bucket = 'sagemaker-us-east-ads508-sp23-t8'

s3 = boto3.Session().client(service_name="s3",
                            region_name=region)
```

[5]:
```
print(f"Default bucket: {def_bucket}")
print(f"Public T8 bucket: {bucket}")
```

```
Default bucket: sagemaker-us-east-1-657724983756
Public T8 bucket: sagemaker-us-east-ads508-sp23-t8
```

2.4 Create Athena Database Table

[6]:
```
database_name = "ads508_t8"
```

[7]:
```
# Set S3 staging directory -- this is a temporary directory used for Athena␣
 ↪queries
s3_staging_dir = f"s3://{def_bucket}/team_8_data/athena/staging"
print(s3_staging_dir)
```

```
s3://sagemaker-us-east-1-657724983756/team_8_data/athena/staging
```

[8]:
```
conn = connect(region_name=region,
               s3_staging_dir=s3_staging_dir)
```

2.5 Define custom function to create tables in existing database

[9]:
```
def create_athena_tbl_tsv(conn=None,
                          db=None,
                          tbl_name=None,
                          fields='',
                          s3_path=None,
                          delim=',',
                          ret='',
                          comp='',
                          skip=''):
    # Set Athena parameters

    # SQL statement to execute
    drop_tsv_tbl_stmnt = f"""DROP TABLE IF EXISTS {db}.{tbl_name}"""

    create_tsv_tbl_stmnt = f"""
```

```
        CREATE EXTERNAL TABLE IF NOT EXISTS {db}.{tbl_name}({fields})
        ROW FORMAT DELIMITED
            FIELDS
                TERMINATED BY '{delim}'
            LINES
                TERMINATED BY '{ret}\\n'
        LOCATION '{s3_path}'
        TBLPROPERTIES ({comp}{skip})
        """

    print(f'Create table statement:\n{create_tsv_tbl_stmnt}')

    pd.read_sql(drop_tsv_tbl_stmnt,
                conn)

    pd.read_sql(create_tsv_tbl_stmnt,
                conn)

    # Verify The Table Has Been Created Succesfully
    show_tsv_tbl_stmnt = f"SHOW TABLES IN {db}"

    df_show = pd.read_sql(show_tsv_tbl_stmnt,
                          conn)
    display(df_show.head(17))

    if tbl_name in df_show.values:
        ingest_create_athena_table_tsv_passed = True

    print(f'\nDataframe contains records:␣
↪{ingest_create_athena_table_tsv_passed}')
```

2.6 Create Athena Table from Local TSV File - census_block_loc.csv

2.6.1 Dataset columns

[10]:
```
ceb_tsv_tbl_name = 'census_block'
ceb_tsv_field_list = """
latitude double,
longitude double,
blockCode string,
county string
"""
ceb_tsv_s3_raw_data_path = f"s3://{def_bucket}/team_8_data/raw_data/
↪census_block"
print(ceb_tsv_s3_raw_data_path)

create_athena_tbl_tsv(conn=conn,
```

```
                        db=database_name,
                        tbl_name=ceb_tsv_tbl_name,
                        fields=ceb_tsv_field_list,
                        s3_path=ceb_tsv_s3_raw_data_path,
                        comp='',
                        skip="'skip.header.line.count'='1'")
```

```
s3://sagemaker-us-east-1-657724983756/team_8_data/raw_data/census_block
Create table statement:

        CREATE EXTERNAL TABLE IF NOT EXISTS ads508_t8.census_block(
latitude double,
longitude double,
blockCode string,
county string
)
        ROW FORMAT DELIMITED
            FIELDS
                TERMINATED BY ','
            LINES
                TERMINATED BY '\n'
        LOCATION 's3://sagemaker-us-
east-1-657724983756/team_8_data/raw_data/census_block'
        TBLPROPERTIES ('skip.header.line.count'='1')


        tab_name
0         census
1   census_block
2          crime
3      crime_pqt
4      evictions
5  grad_outcomes
6        hs_info
7           jobs


Dataframe contains records: True
```

2.6.2 Run A Sample Query

[11]:
```
ceb_select_dbn_stmnt01 = f"""
SELECT
    substr(blockCode,1,11) AS blockCode,
    count(*),
    min(latitude) AS min_lat,
    max(latitude) AS max_lat,
    min(longitude) AS min_long,
```

```
    max(longitude) AS max_long
FROM {database_name}.{ceb_tsv_tbl_name}
GROUP BY substr(blockCode,1,11)
ORDER BY count(*) DESC
LIMIT 50000
"""

print(ceb_select_dbn_stmnt01)

ceb_df01_s01 = pd.read_sql(ceb_select_dbn_stmnt01,
                           conn)

print(ceb_df01_s01.shape)
display(ceb_df01_s01.head(15))
```

```
SELECT
    substr(blockCode,1,11) AS blockCode,
    count(*),
    min(latitude) AS min_lat,
    max(latitude) AS max_lat,
    min(longitude) AS min_long,
    max(longitude) AS max_long
FROM ads508_t8.census_block
GROUP BY substr(blockCode,1,11)
ORDER BY count(*) DESC
LIMIT 50000

(2995, 6)
```

	blockCode	_col1	min_lat	max_lat	min_long	max_long
0	36081990100	1816	40.491307	40.584020	-74.039397	-73.757638
1	36085990100	1198	40.480000	40.604372	-74.257839	-74.036231
2	34025990000	917	40.480000	40.525226	-74.093216	-73.887437
3	36059990400	690	40.534271	40.579497	-73.767136	-73.650000
4	36059301000	412	40.819196	40.877990	-73.751307	-73.653166
5	36081107202	366	40.586281	40.645075	-73.852613	-73.767136
6	36047070203	327	40.579497	40.642814	-73.890603	-73.833618
7	34017012700	305	40.712915	40.776231	-74.143869	-74.077387
8	34013980200	297	40.674472	40.715176	-74.200854	-74.115377
9	36081071600	286	40.622462	40.663166	-73.830452	-73.748141
10	34039035400	275	40.593065	40.640553	-74.261005	-74.200854
11	36047990100	260	40.552362	40.604372	-74.039397	-73.928593
12	36059300100	252	40.798844	40.841809	-73.773467	-73.713317
13	34039039800	251	40.645075	40.688040	-74.197688	-74.140704
14	36005050400	240	40.839548	40.884774	-73.820955	-73.751307

2.7 Review the New Athena Table in the Glue Catalog

[12]:
```
display(
    HTML(
        f'<b>Review <a target="top" href="https://console.aws.amazon.com/glue/
 ↪home?region={region}#">AWS Glue Catalog</a></b>'
    )
)
```

```
<IPython.core.display.HTML object>
```

2.8 Store Variables for the Next Notebooks

[13]:
```
%store
```

```
Stored variables and their in-db values:
balance_dataset                                 -> True
balanced_bias_data_jsonlines_s3_uri             -> 's3://sagemaker-us-
east-1-657724983756/bias-detect
balanced_bias_data_s3_uri                       -> 's3://sagemaker-us-
east-1-657724983756/bias-detect
bias_data_s3_uri                                -> 's3://sagemaker-us-
east-1-657724983756/bias-detect
experiment_name                                 -> 'Amazon-Customer-
Reviews-BERT-Experiment-168013737
feature_group_name                              -> 'reviews-feature-
group-1680137375'
feature_store_offline_prefix                    -> 'reviews-feature-
store-1680137375'
ingest_create_athena_db_passed                  -> True
ingest_create_athena_table_parquet_passed       -> True
ingest_create_athena_table_tsv_passed           -> True
max_seq_length                                  -> 64
processed_test_data_s3_uri                      -> 's3://sagemaker-us-
east-1-657724983756/sagemaker-s
processed_train_data_s3_uri                     -> 's3://sagemaker-us-
east-1-657724983756/sagemaker-s
processed_validation_data_s3_uri                -> 's3://sagemaker-us-
east-1-657724983756/sagemaker-s
raw_input_data_s3_uri                           -> 's3://sagemaker-us-
east-1-657724983756/amazon-revi
s3_private_path_tsv                             -> 's3://sagemaker-us-
east-1-657724983756/team_8_data
s3_public_path_tsv                              -> 's3://sagemaker-us-
east-ads508-sp23-t8'
setup_dependencies_passed                       -> True
setup_iam_roles_passed                          -> True
setup_instance_check_passed                     -> True
setup_s3_bucket_passed                          -> True
```

```
test_split_percentage                                -> 0.05
train_split_percentage                               -> 0.9
trial_name                                           -> 'trial-1680137374'
validation_split_percentage                          -> 0.05
```

2.9 Release Resources

[14]:
```
%%html

<p><b>Shutting down your kernel for this notebook to release resources.</b></p>
<button class="sm-command-button" data-commandlinker-command="kernelmenu:
 ↪shutdown" style="display:none;">Shutdown Kernel</button>

<script>
try {
    els = document.getElementsByClassName("sm-command-button");
    els[0].click();
}
catch(err) {
    // NoOp
}
</script>
```

```
<IPython.core.display.HTML object>
```

[15]:
```
%%javascript

try {
    Jupyter.notebook.save_checkpoint();
    Jupyter.notebook.session.delete();
}
catch(err) {
    // NoOp
}
```

```
<IPython.core.display.Javascript object>
```