02_Setup_Join_Final

April 14, 2023

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

2 Setup Database Joins to Achieve the Analytics Base Table (ABT)

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
!pip install missingno
```

```
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
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.
Requirement already satisfied: missingno in /opt/conda/lib/python3.7/site-
packages (0.5.2)
Requirement already satisfied: seaborn in /opt/conda/lib/python3.7/site-packages
(from missingno) (0.10.0)
Requirement already satisfied: scipy in /opt/conda/lib/python3.7/site-packages
```

```
(from missingno) (1.4.1)
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.7/site-
packages (from missingno) (3.1.3)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages
(from missingno) (1.21.6)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->missingno) (1.1.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->missingno) (2.4.6)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-
packages (from matplotlib->missingno) (0.10.0)
Requirement already satisfied: python-dateutil>=2.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->missingno) (2.8.2)
Requirement already satisfied: pandas>=0.22.0 in /opt/conda/lib/python3.7/site-
packages (from seaborn->missingno) (1.3.5)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages
(from cycler>=0.10->matplotlib->missingno) (1.14.0)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.7/site-
packages (from kiwisolver>=1.0.1->matplotlib->missingno) (59.3.0)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.7/site-
packages (from pandas>=0.22.0->seaborn->missingno) (2019.3)
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
import missingno as msno
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import make_pipeline, Pipeline
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.model_selection import train_test_split, cross_val_score,␣
 ↪GridSearchCV
import datetime as dt

%matplotlib inline
```

2.3 Instantiate AWS SageMaker and S3 sessions

[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 Query Athena Database

[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)
```

[9]:
```
cen_tsv_tbl_name = 'census'
ceb_tsv_tbl_name = 'census_block'
evi_tsv_tbl_name = 'evictions'
cri_tsv_tbl_name = 'crime'
cri_pqt_tbl_name = 'crime_pqt'
grd_tsv_tbl_name = 'grad_outcomes'
hsi_tsv_tbl_name = 'hs_info'
job_tsv_tbl_name = 'jobs'
```

2.4.1 Experiment with join using Athena

[10]:
```
abt_select_to_join_stmnt01 = f"""
SELECT
    cen.censustract,
    cen.borough,
    cen.totalpop,
    cen.men,
```

```
    cen.women,
    cen.hispanic,
    cen.white,
    cen.black,
    cen.native,
    cen.asian,
    cen.citizen,
    cen.income,
    cen.poverty,
    cen.childpoverty,
    cen.professional,
    cen.service,
    cen.office,
    cen.construction,
    cen.production,
    cen.drive,
    cen.carpool,
    cen.transit,
    cen.walk,
    cen.othertransp,
    cen.workathome,
    cen.meancommute,
    cen.employed,
    cen.privatework,
    cen.publicwork,
    cen.selfemployed,
    cen.familywork,
    cen.unemployment,
    cvi.blockCode,
    cvi.eviction_count_x_lat_long
FROM {database_name}.{cen_tsv_tbl_name} AS cen
LEFT JOIN (
    SELECT
        ceb.blockCode AS blockCode,
        SUM(evi.eviction_count_x_lat_long) AS eviction_count_x_lat_long
    FROM (
        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
        ) AS ceb
```

```
    INNER JOIN (
        SELECT
            CAST(latitude AS DOUBLE) AS latitude,
            CAST(longitude AS DOUBLE) AS longitude,
            COUNT(*) AS eviction_count_x_lat_long
        FROM {database_name}.{evi_tsv_tbl_name}
        WHERE latitude != ''
        GROUP BY latitude, longitude
        ORDER BY COUNT(*) DESC
        ) AS evi
    ON evi.latitude >= ceb.min_lat
        AND evi.latitude <= ceb.max_lat
        AND evi.longitude >= ceb.min_long
        AND evi.longitude <= ceb.max_long
    GROUP BY ceb.blockCode
    LIMIT 50000
    ) AS cvi
    ON cen.censustract = cvi.blockCode
ORDER BY cen.censustract
"""

print(abt_select_to_join_stmnt01)

abt_select_to_join_df01 = pd.read_sql(abt_select_to_join_stmnt01,
                                      conn)

print(abt_select_to_join_df01.shape)
display(abt_select_to_join_df01.head(15))
```

```
SELECT
    cen.censustract,
    cen.borough,
    cen.totalpop,
    cen.men,
    cen.women,
    cen.hispanic,
    cen.white,
    cen.black,
    cen.native,
    cen.asian,
    cen.citizen,
    cen.income,
    cen.poverty,
    cen.childpoverty,
    cen.professional,
    cen.service,
    cen.office,
```

```
    cen.construction,
    cen.production,
    cen.drive,
    cen.carpool,
    cen.transit,
    cen.walk,
    cen.othertransp,
    cen.workathome,
    cen.meancommute,
    cen.employed,
    cen.privatework,
    cen.publicwork,
    cen.selfemployed,
    cen.familywork,
    cen.unemployment,
    cvi.blockCode,
    cvi.eviction_count_x_lat_long
FROM ads508_t8.census AS cen
LEFT JOIN (
    SELECT
        ceb.blockCode AS blockCode,
        SUM(evi.eviction_count_x_lat_long) AS eviction_count_x_lat_long
    FROM (
        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
        ) AS ceb
    INNER JOIN (
        SELECT
            CAST(latitude AS DOUBLE) AS latitude,
            CAST(longitude AS DOUBLE) AS longitude,
            COUNT(*) AS eviction_count_x_lat_long
        FROM ads508_t8.evictions
        WHERE latitude != ''
        GROUP BY latitude, longitude
        ORDER BY COUNT(*) DESC
        ) AS evi
    ON evi.latitude >= ceb.min_lat
        AND evi.latitude <= ceb.max_lat
        AND evi.longitude >= ceb.min_long
        AND evi.longitude <= ceb.max_long
```

```
    GROUP BY ceb.blockCode
    LIMIT 50000
    ) AS cvi
    ON cen.censustract = cvi.blockCode
ORDER BY cen.censustract
```

(2167, 34)

	censustract	borough	totalpop	men	women	hispanic	white	black	\
0	36005000100	Bronx	7703	7133	570	29.9	6.1	60.9	
1	36005000200	Bronx	5403	2659	2744	75.8	2.3	16.0	
2	36005000400	Bronx	5915	2896	3019	62.7	3.6	30.7	
3	36005001600	Bronx	5879	2558	3321	65.1	1.6	32.4	
4	36005001900	Bronx	2591	1206	1385	55.4	9.0	29.0	
5	36005002000	Bronx	8516	3301	5215	61.1	1.6	31.1	
6	36005002300	Bronx	4774	2130	2644	62.3	0.2	36.5	
7	36005002400	Bronx	150	109	41	0.0	52.0	48.0	
8	36005002500	Bronx	5355	2338	3017	76.5	1.5	18.9	
9	36005002701	Bronx	3016	1375	1641	68.0	0.0	31.2	
10	36005002702	Bronx	4778	2427	2351	71.3	1.6	26.2	
11	36005002800	Bronx	5299	2292	3007	23.0	0.2	71.4	
12	36005003100	Bronx	1466	769	697	72.3	0.6	24.6	
13	36005003300	Bronx	3912	1824	2088	65.6	1.0	30.6	
14	36005003500	Bronx	3948	1921	2027	73.5	0.7	25.9	

	native	asian	…	workathome	meancommute	employed	privatework	\
0	0.2	1.6	…	NaN	NaN	0	NaN	
1	0.0	4.2	…	0.0	43.0	2308	80.8	
2	0.0	0.3	…	2.1	45.0	2675	71.7	
3	0.0	0.0	…	1.7	38.8	2120	75.0	
4	0.0	2.1	…	6.2	45.4	1083	76.8	
5	0.3	3.3	…	0.0	46.0	2508	71.0	
6	1.0	0.0	…	4.1	42.7	1191	74.2	
7	0.0	0.0	…	0.0	NaN	113	62.8	
8	0.0	3.0	…	2.7	35.5	1691	85.1	
9	0.0	0.0	…	1.6	42.8	1102	86.9	
10	0.0	0.0	…	0.5	44.0	1559	75.0	
11	0.0	1.7	…	2.7	47.3	2394	61.9	
12	0.0	2.2	…	0.0	40.1	722	79.2	
13	1.7	0.0	…	1.8	42.5	1113	77.2	
14	0.0	0.0	…	4.7	41.6	1360	83.2	

	publicwork	selfemployed	familywork	unemployment	blockCode	\
0	NaN	NaN	NaN	NaN	None	
1	16.2	2.9	0.0	7.7	36005000200	
2	25.3	2.5	0.6	9.5	36005000400	
3	21.3	3.8	0.0	8.7	36005001600	
4	15.5	7.7	0.0	19.2	36005001900	

```
5         21.3           7.7         0.0          17.2  36005002000
6         16.1           9.7         0.0          18.9         None
7         37.2           0.0         0.0           0.0  36005002400
8          8.3           6.1         0.5           9.4  36005002500
9          8.5           4.5         0.0          15.2         None
10        14.0          11.0         0.0          10.6  36005002702
11        37.4           0.6         0.0          12.8  36005002800
12        10.2          10.5         0.0           6.6  36005003100
13        16.9           5.9         0.0          18.5         None
14        13.4           3.4         0.0          11.8  36005003500

    eviction_count_x_lat_long
0                         NaN
1                        31.0
2                        46.0
3                        10.0
4                       230.0
5                        69.0
6                         NaN
7                       169.0
8                        22.0
9                         NaN
10                       32.0
11                       45.0
12                        1.0
13                        NaN
14                       57.0

[15 rows x 34 columns]
```

[11]:
```
ceb_select_to_join_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
"""

print(ceb_select_to_join_stmnt01)

ceb_select_to_join_df01 = pd.read_sql(ceb_select_to_join_stmnt01,
                                      conn)
```

```
print(ceb_select_to_join_df01.shape)
display(ceb_select_to_join_df01.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
```

```
(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

2.4.2 SELECT statements to prepare for full join: evictions table

[12]:
```
# Display full table for review
evi_full_select_stmnt01 = f"""
SELECT * FROM {database_name}.{evi_tsv_tbl_name}
WHERE executed_date <> ''
LIMIT 100
"""

# Display SQL statement
print(evi_full_select_stmnt01)

# Run SQL statement against Athena table
```

```
evi_full_select_df01 = pd.read_sql(evi_full_select_stmnt01,
                                   conn)

# Display results
print(evi_full_select_df01.shape)
display(evi_full_select_df01.head(11))
```

```
SELECT * FROM ads508_t8.evictions
WHERE executed_date <> ''
LIMIT 100
```

```
(100, 20)
```

	court_index_number	docket_number	eviction_address	eviction_apartment_number	executed_date	marshal_first_name
0	56037/17	339568	547 EAST 168TH STREET	3H	02/26/2018	Thomas
1	B047517/19	409031	4014 CARPENTER AVENUE	4B	11/16/2022	Richard
2	15068/17	334442	655 EAST 224TH STREET	1	09/29/2017	Thomas
3	58273/18	025388	1551 DEAN STREET	1ST FLOOR	07/12/2018	Gary
4	14866/19A	097278	718 PENFIELD STREET	2-F	10/24/2019	Justin
5	66703/18BX	090391	2032 EAST 177TH ST A /K/A 2032 CROSS BRONX EXP…	1E	07/30/2019	Justin
6	98925/17	075402	175 WOODRUFF AVENUE	GARDEN APARTMENT	06/01/2018	Justin
7	304057/20	107717	555 TENTH AVENUE			
8	210706/18	085502	2201 FIRST AVENUE			
9	B806500/18	396012	281 EAST 143RD STREET			
10	83995/16	464985	1-11 MARBLE HILL AVE NUE			

```
7                       32I    04/18/2022            Justin
8                       05B    03/14/2019             Henry
9                       07A    01/17/2019           Richard
10                       3F    03/17/2017             Danny

   marshal_last_name residential_or_commercial    borough eviction_postcode  \
0                Bia                Residential      BRONX             10456
1              McCoy                Residential      BRONX             10466
2                Bia                Residential      BRONX             10467
3               Rose                Residential   BROOKLYN             11213
4           Grossman                Residential      BRONX             10470
5           Grossman                Residential      BRONX             10472
6           Grossman                Residential   BROOKLYN             11226
7           Grossman                Residential  MANHATTAN             10018
8              Daley                Residential  MANHATTAN             10029
9              McCoy                Residential      BRONX             10451
10          Weinheim                Residential  MANHATTAN             10463

           ejectment eviction_or_legal_possession   latitude   longitude  \
0   Not an Ejectment                   Possession  40.830857  -73.905191
1   Not an Ejectment                   Possession  40.889878  -73.862686
2   Not an Ejectment                   Possession  40.887599  -73.862391
3   Not an Ejectment                   Possession  40.676166  -73.936661
4   Not an Ejectment                   Possession  40.904888  -73.849089
5   Not an Ejectment                   Possession  40.831685  -73.856168
6   Not an Ejectment                   Possession  40.654641  -73.960291
7   Not an Ejectment                   Possession  40.758888  -73.996022
8   Not an Ejectment                   Possession  40.794176  -73.936754
9   Not an Ejectment                   Possession  40.814845  -73.924083
10  Not an Ejectment                   Possession  40.874862  -73.910845

   community_board council_district census_tract      bin         bbl  \
0                3                16          145  2004227  2026100065
1               12                12          408  2063060  2048280031
2               12                12          394  2062985  2048260028
3                8                36          311  3388499  3013400049
4               12                11          442  2071873  2051130039
5                9                18           78  2026230  2038030019
6               14                40        50803  3115933  3050540052
7                4                 3          117  1089722  1010697501
8               11                 8          180  1081091  1016840001
9                1                 8           51  2091116  2023240001
10               8                10          309  1064643  1022150465

                                    nta
0                  Claremont-Bathgate
1               Williamsbridge-Olinville
2               Williamsbridge-Olinville
```

```
3                          Crown Heights North
4                           Woodlawn-Wakefield
5                        Westchester-Unionport
6                                    Flatbush
7   Hudson Yards-Chelsea-Flatiron-Union Square
8                           East Harlem North
9                       Mott Haven-Port Morris
10                          Marble Hill-Inwood
```

[13]:
```
# Aggregate table based on borough and relative data year
evi_borough_year_stmnt01 = f"""
SELECT
    LOWER(borough) AS borough,
    CAST(YEAR(DATE_PARSE(executed_date, '%m/%d/%Y')) AS INT) - 2022 AS␣
 ↪relative_data_year,
    COUNT(*) AS annual_evictions_x_borough
FROM {database_name}.{evi_tsv_tbl_name}
WHERE executed_date <> ''
    AND CAST(YEAR(DATE_PARSE(executed_date, '%m/%d/%Y')) AS INT) BETWEEN 2018␣
 ↪AND 2022
GROUP BY borough, YEAR(DATE_PARSE(executed_date, '%m/%d/%Y'))
ORDER BY borough, YEAR(DATE_PARSE(executed_date, '%m/%d/%Y'))
LIMIT 10000
"""

# Display SQL statement
print(evi_borough_year_stmnt01)

# Run SQL statement against Athena table
evi_borough_year_df01 = pd.read_sql(evi_borough_year_stmnt01,
                                    conn)

# Display results
print(evi_borough_year_df01.shape)
display(evi_borough_year_df01.head(11))

# Create pivot table
evi_borough_year_df02 = evi_borough_year_df01.pivot_table(index = 'borough',
                                                          columns =␣
 ↪'relative_data_year',
                                                          values =␣
 ↪'annual_evictions_x_borough',
                                                          aggfunc = 'sum',
                                                          fill_value = 0)

print(evi_borough_year_df02.shape)
display(evi_borough_year_df02.head(35))
```

```
SELECT
    LOWER(borough) AS borough,
    CAST(YEAR(DATE_PARSE(executed_date, '%m/%d/%Y')) AS INT) - 2022 AS
relative_data_year,
    COUNT(*) AS annual_evictions_x_borough
FROM ads508_t8.evictions
WHERE executed_date <> ''
    AND CAST(YEAR(DATE_PARSE(executed_date, '%m/%d/%Y')) AS INT) BETWEEN 2018
AND 2022
GROUP BY borough, YEAR(DATE_PARSE(executed_date, '%m/%d/%Y'))
ORDER BY borough, YEAR(DATE_PARSE(executed_date, '%m/%d/%Y'))
LIMIT 10000
```

(25, 3)

	borough	relative_data_year	annual_evictions_x_borough
0	bronx	-4	7140
1	bronx	-3	6244
2	bronx	-2	1088
3	bronx	-1	29
4	bronx	0	1174
5	brooklyn	-4	6157
6	brooklyn	-3	5312
7	brooklyn	-2	1005
8	brooklyn	-1	100
9	brooklyn	0	1864
10	manhattan	-4	3390

(5, 5)

relative_data_year	-4	-3	-2	-1	0
borough					
bronx	7140	6244	1088	29	1174
brooklyn	6157	5312	1005	100	1864
manhattan	3390	2818	521	68	930
queens	4452	3705	696	36	811
staten island	691	636	112	35	271

[14]:
```
# Aggregate table based on census_tract and year
evi_ceb_join_select_stmnt01 = f"""
SELECT
    ceb.blockCode AS census_tract,
    evi.year,
    SUM(evi.eviction_count_x_lat_long) AS annual_evictions_x_census_tract
FROM (
    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
    ) AS ceb
INNER JOIN (
    SELECT
        CAST(latitude AS DOUBLE) AS latitude,
        CAST(longitude AS DOUBLE) AS longitude,
        CAST(YEAR(DATE_PARSE(executed_date, '%m/%d/%Y')) AS INT) AS year,
        COUNT(*) AS eviction_count_x_lat_long
    FROM {database_name}.{evi_tsv_tbl_name}
    WHERE latitude != ''
    GROUP BY latitude, longitude, YEAR(DATE_PARSE(executed_date, '%m/%d/%Y'))
    ORDER BY COUNT(*) DESC
    ) AS evi
    ON evi.latitude >= ceb.min_lat
        AND evi.latitude <= ceb.max_lat
        AND evi.longitude >= ceb.min_long
        AND evi.longitude <= ceb.max_long
GROUP BY ceb.blockCode, evi.year
ORDER BY ceb.blockCode, evi.year
LIMIT 50000
"""

# Display SQL statement
print(evi_ceb_join_select_stmnt01)

evi_ceb_join_select_df01 = pd.read_sql(evi_ceb_join_select_stmnt01,
                           conn)

# Display results
print(evi_ceb_join_select_df01.shape)
display(evi_ceb_join_select_df01.head(11))

# Create pivot table
evi_ceb_join_select_df02 = evi_ceb_join_select_df01.pivot_table(index =␣
 ↪'census_tract',
                                                                 columns =␣
 ↪'year',
                                                                 values =␣
 ↪'annual_evictions_x_census_tract',
                                                                 aggfunc = 'sum',
                                                                 fill_value = 0)
```

```
print(evi_ceb_join_select_df02.shape)
display(evi_ceb_join_select_df02.head(11))
```

```
SELECT
    ceb.blockCode AS census_tract,
    evi.year,
    SUM(evi.eviction_count_x_lat_long) AS annual_evictions_x_census_tract
FROM (
    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
    ) AS ceb
INNER JOIN (
    SELECT
        CAST(latitude AS DOUBLE) AS latitude,
        CAST(longitude AS DOUBLE) AS longitude,
        CAST(YEAR(DATE_PARSE(executed_date, '%m/%d/%Y')) AS INT) AS year,
        COUNT(*) AS eviction_count_x_lat_long
    FROM ads508_t8.evictions
    WHERE latitude != ''
    GROUP BY latitude, longitude, YEAR(DATE_PARSE(executed_date, '%m/%d/%Y'))
    ORDER BY COUNT(*) DESC
    ) AS evi
    ON evi.latitude >= ceb.min_lat
        AND evi.latitude <= ceb.max_lat
        AND evi.longitude >= ceb.min_long
        AND evi.longitude <= ceb.max_long
GROUP BY ceb.blockCode, evi.year
ORDER BY ceb.blockCode, evi.year
LIMIT 50000
```

```
(4870, 3)
```

	census_tract	year	annual_evictions_x_census_tract
0	34003013001	2017	9
1	34003013001	2018	11
2	34003013001	2019	6
3	34003013001	2021	1
4	34003013001	2022	6

```
5    34003013001  2023                                1
6    34003016000  2017                               14
7    34003016000  2018                               13
8    34003016000  2019                               16
9    34003016000  2020                                2
10   34003016000  2021                                2
```

```
(1306, 7)
```

year	2017	2018	2019	2020	2021	2022	2023
census_tract							
34003013001	9	11	6	0	1	6	1
34003016000	14	13	16	2	2	1	0
34017010800	2	2	5	0	0	0	0
36005000200	9	10	10	0	0	1	1
36005000400	10	16	15	0	0	5	0
36005001600	3	2	4	0	0	1	0
36005001900	74	67	53	9	0	22	5
36005002000	20	14	29	3	0	3	0
36005002400	50	43	65	5	0	6	0
36005002500	8	2	9	0	0	3	0
36005002702	10	14	5	1	0	2	0

2.4.3 SELECT statements to prepare for full join: crime_pqt table

[15]:
```
# Display full table for review
cri_full_select_stmnt01 = f"""
SELECT * FROM {database_name}.{cri_pqt_tbl_name}
LIMIT 10000
"""

# Display SQL statement
print(cri_full_select_stmnt01)

# Run SQL statement against Athena table
cri_full_select_df01 = pd.read_sql(cri_full_select_stmnt01,
                                   conn)

# Display results
print(cri_full_select_df01.shape)
display(cri_full_select_df01.head(5))
```

```
SELECT * FROM ads508_t8.crime_pqt
LIMIT 10000

(10000, 35)
```

```
  cmplnt_num cmplnt_fr_dt cmplnt_fr_tm cmplnt_to_dt cmplnt_to_tm addr_pct_cd  \
```

```
0  615895978   02/04/2016     06:44:00                                    44
1  753741689   04/03/2017     23:20:00   04/03/2017     23:30:00          42
2  157660368   06/01/2017     19:00:00   08/08/2017     12:00:00          49
3  638104805   10/04/2014     10:30:00   10/04/2014     14:30:00          43
4  330952034   12/24/2020     16:00:00   12/24/2020     17:00:00          44

       rpt_dt ky_cd      ofns_desc pd_cd  …            latitude  \
0  02/12/2016   578  HARRASSMENT 2   638  …        40.834209452
1  04/03/2017   578  HARRASSMENT 2   637  …        40.830145224
2  08/15/2017   578  HARRASSMENT 2   638  …        40.845844356
3  10/05/2014   578  HARRASSMENT 2   638  …        40.824130774
4  12/28/2020   578  HARRASSMENT 2   638  …  40.826524240000026

            longitude                                    lat_lon  \
0       -73.925706819             (40.834209452, -73.925706819)
1       -73.891878074             (40.830145224, -73.891878074)
2       -73.851790951             (40.845844356, -73.851790951)
3       -73.869872717             (40.824130774, -73.869872717)
4  -73.92154563799994  (40.826524240000026, -73.92154563799994)

         patrol_boro    station_name vic_age_group        vic_race vic_sex  \
0  PATROL BORO BRONX                         25-44  WHITE HISPANIC       F
1  PATROL BORO BRONX  FREEMAN STREET         25-44  WHITE HISPANIC       M
2  PATROL BORO BRONX                         25-44  BLACK HISPANIC       M
3  PATROL BORO BRONX                         25-44           BLACK       F
4  PATROL BORO BRONX                         25-44  BLACK HISPANIC       M

  law_cat_cd borough
0  VIOLATION   BRONX
1  VIOLATION   BRONX
2  VIOLATION   BRONX
3  VIOLATION   BRONX
4  VIOLATION   BRONX

[5 rows x 35 columns]
```

[16]:
```
# Aggregate table based on borough, relative data year, & law_cat_cd
cri_borough_year_type_stmnt01 = f"""
SELECT
    LOWER(borough) AS borough,
    CAST(YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) AS INT) - 2021 AS␣
 ↪relative_data_year,
    law_cat_cd AS complaint_type,
    COUNT(*) AS annual_complaint_counts
FROM {database_name}.{cri_pqt_tbl_name}
WHERE cmplnt_fr_dt <> ''
    AND YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) BETWEEN 2017 AND 2021
```

```
GROUP BY borough, YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')), law_cat_cd
ORDER BY borough, YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')), law_cat_cd
LIMIT 100000
"""

# Display SQL statement
print(cri_borough_year_type_stmnt01)

# Run SQL statement against Athena table
cri_borough_year_type_df01 = pd.read_sql(cri_borough_year_type_stmnt01,
                                         conn)

# Display results
print(cri_borough_year_type_df01.shape)
display(cri_borough_year_type_df01.head(35))
```

```
SELECT
    LOWER(borough) AS borough,
    CAST(YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) AS INT) - 2021 AS
relative_data_year,
    law_cat_cd AS complaint_type,
    COUNT(*) AS annual_complaint_counts
FROM ads508_t8.crime_pqt
WHERE cmplnt_fr_dt <> ''
    AND YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) BETWEEN 2017 AND 2021
GROUP BY borough, YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')), law_cat_cd
ORDER BY borough, YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')), law_cat_cd
LIMIT 100000

(81, 4)
```

	borough	relative_data_year	complaint_type	annual_complaint_counts
0	bronx	-4	FELONY	583
1	bronx	-4	MISDEMEANOR	1179
2	bronx	-4	VIOLATION	289
3	bronx	-3	FELONY	540
4	bronx	-3	MISDEMEANOR	1133
5	bronx	-3	VIOLATION	338
6	bronx	-2	FELONY	558
7	bronx	-2	MISDEMEANOR	1114
8	bronx	-2	VIOLATION	328
9	bronx	-1	FELONY	529
10	bronx	-1	MISDEMEANOR	954
11	bronx	-1	VIOLATION	322
12	bronx	0	FELONY	617
13	bronx	0	MISDEMEANOR	910
14	bronx	0	VIOLATION	320

15	brooklyn	-4	FELONY	862
16	brooklyn	-4	MISDEMEANOR	1512
17	brooklyn	-4	VIOLATION	430
18	brooklyn	-3	FELONY	976
19	brooklyn	-3	MISDEMEANOR	1473
20	brooklyn	-3	VIOLATION	423
21	brooklyn	-2	FELONY	909
22	brooklyn	-2	MISDEMEANOR	1371
23	brooklyn	-2	VIOLATION	435
24	brooklyn	-1	FELONY	803
25	brooklyn	-1	MISDEMEANOR	1168
26	brooklyn	-1	VIOLATION	384
27	brooklyn	0	FELONY	834
28	brooklyn	0	MISDEMEANOR	1185
29	brooklyn	0	VIOLATION	475
30	manhattan	-4	FELONY	701
31	manhattan	-4	MISDEMEANOR	1306
32	manhattan	-4	VIOLATION	267
33	manhattan	-3	FELONY	736
34	manhattan	-3	MISDEMEANOR	1276

[17]:
```
# Aggregate table based on borough, relative data year, & law_cat_cd
cri_borough_year_type_stmnt02 = f"""
SELECT
    LOWER(borough) AS borough,
    CONCAT(CAST(YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) AS VARCHAR), ' - ',␣
 ↪law_cat_cd) AS year_w_complaint,
    COUNT(*) AS annual_complaint_counts
FROM {database_name}.{cri_pqt_tbl_name}
WHERE cmplnt_fr_dt <> ''
    AND YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) >= 2017
GROUP BY borough, CONCAT(CAST(YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) AS␣
 ↪VARCHAR), ' - ', law_cat_cd)
ORDER BY borough, CONCAT(CAST(YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) AS␣
 ↪VARCHAR), ' - ', law_cat_cd)
LIMIT 100000
"""

# Display SQL statement
print(cri_borough_year_type_stmnt02)

# Run SQL statement against Athena table
cri_borough_year_type_df12 = pd.read_sql(cri_borough_year_type_stmnt02,
                                         conn)

# Display results
print(cri_borough_year_type_df12.shape)
```

```
display(cri_borough_year_type_df12.head(11))

# Create pivot table
cri_borough_year_type_df13 = cri_borough_year_type_df12.pivot_table(index =␣
 ↪'borough',
                                                                    columns =␣
 ↪'year_w_complaint',
                                                                    values =␣
 ↪'annual_complaint_counts',
                                                                    aggfunc =␣
 ↪'sum',
                                                                    fill_value␣
 ↪= 0)

print(cri_borough_year_type_df13.shape)
display(cri_borough_year_type_df13.head(11))
```

```
SELECT
    LOWER(borough) AS borough,
    CONCAT(CAST(YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) AS VARCHAR), ' - ',
law_cat_cd) AS year_w_complaint,
    COUNT(*) AS annual_complaint_counts
FROM ads508_t8.crime_pqt
WHERE cmplnt_fr_dt <> ''
    AND YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) >= 2017
GROUP BY borough, CONCAT(CAST(YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) AS
VARCHAR), ' - ', law_cat_cd)
ORDER BY borough, CONCAT(CAST(YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) AS
VARCHAR), ' - ', law_cat_cd)
LIMIT 100000
```

```
(81, 3)
```

	borough	year_w_complaint	annual_complaint_counts
0	bronx	2017 - FELONY	583
1	bronx	2017 - MISDEMEANOR	1179
2	bronx	2017 - VIOLATION	289
3	bronx	2018 - FELONY	540
4	bronx	2018 - MISDEMEANOR	1133
5	bronx	2018 - VIOLATION	338
6	bronx	2019 - FELONY	558
7	bronx	2019 - MISDEMEANOR	1114
8	bronx	2019 - VIOLATION	328
9	bronx	2020 - FELONY	529
10	bronx	2020 - MISDEMEANOR	954

```
(5, 15)
```

```
year_w_complaint  2017 - FELONY  2017 - MISDEMEANOR  2017 - VIOLATION  \
borough
bronx                       583                1179               289
brooklyn                    862                1512               430
manhattan                   701                1306               267
queens                      610                 932               299
staten island               100                 249               112

year_w_complaint  2018 - FELONY  2018 - MISDEMEANOR  2018 - VIOLATION  \
borough
bronx                       540                1133               338
brooklyn                    976                1473               423
manhattan                   736                1276               311
queens                      576                 877               309
staten island               107                 239               109

year_w_complaint  2019 - FELONY  2019 - MISDEMEANOR  2019 - VIOLATION  \
borough
bronx                       558                1114               328
brooklyn                    909                1371               435
manhattan                   665                1302               317
queens                      604                 970               292
staten island               101                 196                61

year_w_complaint  2020 - FELONY  2020 - MISDEMEANOR  2020 - VIOLATION  \
borough
bronx                       529                 954               322
brooklyn                    803                1168               384
manhattan                   605                1085               248
queens                      607                 911               290
staten island               104                 175                57

year_w_complaint  2021 - FELONY  2021 - MISDEMEANOR  2021 - VIOLATION
borough
bronx                       617                 910               320
brooklyn                    834                1185               475
manhattan                   708                1145               295
queens                      646                 995               320
staten island                96                 174                75
```

[18]:
```
# Aggregate table based on census_tract and year
cri_ceb_join_select_stmnt01 = f"""
SELECT
    ceb.blockCode AS census_tract,
    cri.year,
    SUM(cri.complaint_count_x_lat_long) AS annual_complaints_x_census_tract
FROM (
```

```
    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
    ) AS ceb
INNER JOIN (
    SELECT
        CAST(latitude AS DOUBLE) AS latitude,
        CAST(longitude AS DOUBLE) AS longitude,
        CONCAT(CAST(YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) AS VARCHAR), ' -␣
 ↪', law_cat_cd) AS year,
        COUNT(*) AS complaint_count_x_lat_long
    FROM {database_name}.{cri_pqt_tbl_name}
    WHERE cmplnt_fr_dt <> ''
        AND latitude != ''
        AND YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) >= 2017
    GROUP BY latitude, longitude, CONCAT(CAST(YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/
 ↪%d/%Y')) AS VARCHAR), ' - ', law_cat_cd)
    ORDER BY COUNT(*) DESC
    ) AS cri
    ON cri.latitude >= ceb.min_lat
        AND cri.latitude <= ceb.max_lat
        AND cri.longitude >= ceb.min_long
        AND cri.longitude <= ceb.max_long
GROUP BY ceb.blockCode, cri.year
ORDER BY ceb.blockCode, cri.year
LIMIT 100000
"""

# Display SQL statement
print(cri_ceb_join_select_stmnt01)

# Run SQL statement against Athena table
cri_ceb_join_select_df01 = pd.read_sql(cri_ceb_join_select_stmnt01,
                                       conn)

# Display results
print(cri_ceb_join_select_df01.shape)
display(cri_ceb_join_select_df01.head(15))
```

```
# Create pivot table
cri_ceb_join_select_df02 = cri_ceb_join_select_df01.pivot_table(index =␣
 ↪'census_tract',
                                                                 columns =␣
 ↪'year',
                                                                 values =␣
 ↪'annual_complaints_x_census_tract',
                                                                 aggfunc = 'sum',
                                                                 fill_value = 0)

print(cri_ceb_join_select_df02.shape)
display(cri_ceb_join_select_df02.head(35))
```

```
SELECT
    ceb.blockCode AS census_tract,
    cri.year,
    SUM(cri.complaint_count_x_lat_long) AS annual_complaints_x_census_tract
FROM (
    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
    ) AS ceb
INNER JOIN (
    SELECT
        CAST(latitude AS DOUBLE) AS latitude,
        CAST(longitude AS DOUBLE) AS longitude,
        CONCAT(CAST(YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) AS VARCHAR), ' -
', law_cat_cd) AS year,
        COUNT(*) AS complaint_count_x_lat_long
    FROM ads508_t8.crime_pqt
    WHERE cmplnt_fr_dt <> ''
        AND latitude != ''
        AND YEAR(DATE_PARSE(cmplnt_fr_dt, '%m/%d/%Y')) >= 2017
    GROUP BY latitude, longitude, CONCAT(CAST(YEAR(DATE_PARSE(cmplnt_fr_dt,
'%m/%d/%Y')) AS VARCHAR), ' - ', law_cat_cd)
    ORDER BY COUNT(*) DESC
    ) AS cri
    ON cri.latitude >= ceb.min_lat
        AND cri.latitude <= ceb.max_lat
```

```
        AND cri.longitude >= ceb.min_long
        AND cri.longitude <= ceb.max_long
GROUP BY ceb.blockCode, cri.year
ORDER BY ceb.blockCode, cri.year
LIMIT 100000
```

(9555, 3)

	census_tract	year	annual_complaints_x_census_tract
0	34003013001	2017 - FELONY	1
1	34003013001	2017 - MISDEMEANOR	6
2	34003013001	2017 - VIOLATION	4
3	34003013001	2018 - FELONY	3
4	34003013001	2018 - MISDEMEANOR	5
5	34003013001	2019 - FELONY	1
6	34003013001	2019 - MISDEMEANOR	6
7	34003013001	2019 - VIOLATION	1
8	34003013001	2020 - FELONY	3
9	34003013001	2020 - MISDEMEANOR	2
10	34003013001	2020 - VIOLATION	1
11	34003013001	2021 - FELONY	6
12	34003013001	2021 - MISDEMEANOR	5
13	34003013001	2021 - VIOLATION	2
14	34003016000	2017 - MISDEMEANOR	4

(1358, 15)

year	2017 - FELONY	2017 - MISDEMEANOR	2017 - VIOLATION	\
census_tract				
34003013001	1	6	4	
34003016000	0	4	0	
34017010800	0	1	0	
36005000100	16	10	0	
36005000200	0	1	1	
36005000400	0	1	1	
36005001600	0	0	2	
36005001900	14	20	5	
36005002000	3	10	0	
36005002400	13	36	8	
36005002500	0	3	0	
36005002702	1	1	0	
36005002800	1	1	0	
36005003100	1	0	0	
36005003500	0	1	0	
36005003900	0	2	1	
36005004100	2	5	0	
36005004200	4	11	6	
36005004300	1	1	0	
36005004400	1	2	0	

36005005100	2	23	2
36005005200	0	1	0
36005005902	0	2	0
36005006000	0	1	0
36005006100	0	1	0
36005006200	2	1	1
36005006300	1	16	2
36005006500	3	4	0
36005006700	0	0	1
36005006800	1	1	0
36005007100	7	17	1
36005007200	0	0	0
36005007500	2	4	1
36005007600	0	0	1
36005007800	0	1	0

year census_tract	2018 - FELONY	2018 - MISDEMEANOR	2018 - VIOLATION \
34003013001	3	5	0
34003016000	3	0	2
34017010800	0	0	0
36005000100	5	10	0
36005000200	3	3	1
36005000400	2	3	2
36005001600	0	0	1
36005001900	7	15	5
36005002000	1	1	0
36005002400	7	14	7
36005002500	0	2	0
36005002702	0	1	0
36005002800	1	0	1
36005003100	0	0	0
36005003500	1	0	1
36005003900	2	1	0
36005004100	1	5	1
36005004200	1	9	5
36005004300	1	1	0
36005004400	3	3	0
36005005100	10	11	1
36005005200	1	1	1
36005005902	1	0	1
36005006000	0	0	0
36005006100	0	0	1
36005006200	2	1	0
36005006300	13	19	3
36005006500	4	8	2
36005006700	0	4	0
36005006800	0	0	0

36005007100	6	9	5
36005007200	0	0	1
36005007500	0	3	0
36005007600	2	0	0
36005007800	0	2	0

year	2019 - FELONY	2019 - MISDEMEANOR	2019 - VIOLATION	\
census_tract				
34003013001	1	6	1	
34003016000	3	1	0	
34017010800	0	0	0	
36005000100	10	4	1	
36005000200	1	2	2	
36005000400	3	1	1	
36005001600	0	0	0	
36005001900	11	25	12	
36005002000	3	7	2	
36005002400	8	25	6	
36005002500	3	4	1	
36005002702	0	2	0	
36005002800	0	1	0	
36005003100	0	0	0	
36005003500	1	0	0	
36005003900	1	3	1	
36005004100	4	4	1	
36005004200	5	11	1	
36005004300	1	4	0	
36005004400	0	2	1	
36005005100	9	11	1	
36005005200	0	0	0	
36005005902	1	5	1	
36005006000	0	2	0	
36005006100	0	2	0	
36005006200	0	5	1	
36005006300	2	13	4	
36005006500	1	5	2	
36005006700	0	0	0	
36005006800	0	2	0	
36005007100	3	16	4	
36005007200	0	1	0	
36005007500	0	3	0	
36005007600	1	1	0	
36005007800	0	0	2	

year	2020 - FELONY	2020 - MISDEMEANOR	2020 - VIOLATION	\
census_tract				
34003013001	3	2	1	
34003016000	0	0	0	

34017010800	0	1	0
36005000100	3	2	0
36005000200	3	0	0
36005000400	3	2	0
36005001600	1	1	0
36005001900	9	11	7
36005002000	3	3	0
36005002400	12	18	6
36005002500	0	3	0
36005002702	0	0	1
36005002800	1	2	0
36005003100	2	0	0
36005003500	1	1	0
36005003900	2	1	0
36005004100	0	1	2
36005004200	3	6	4
36005004300	1	1	0
36005004400	2	4	3
36005005100	5	13	4
36005005200	0	1	0
36005005902	0	2	0
36005006000	2	0	0
36005006100	0	0	0
36005006200	2	1	4
36005006300	7	13	3
36005006500	4	4	2
36005006700	0	2	0
36005006800	0	1	0
36005007100	3	8	2
36005007200	0	1	0
36005007500	2	3	1
36005007600	0	1	1
36005007800	1	0	0

year	2021 - FELONY	2021 - MISDEMEANOR	2021 - VIOLATION
census_tract			
34003013001	6	5	2
34003016000	3	1	0
34017010800	0	0	0
36005000100	2	1	0
36005000200	3	2	1
36005000400	3	2	1
36005001600	1	1	0
36005001900	8	13	3
36005002000	0	1	1
36005002400	8	15	2
36005002500	0	1	2
36005002702	1	2	0

36005002800	1	5	3
36005003100	0	0	0
36005003500	0	3	2
36005003900	2	1	0
36005004100	2	3	0
36005004200	0	12	3
36005004300	1	1	0
36005004400	2	1	3
36005005100	2	9	3
36005005200	0	1	0
36005005902	1	0	1
36005006000	0	0	0
36005006100	0	0	0
36005006200	2	2	2
36005006300	10	14	3
36005006500	3	2	1
36005006700	2	1	1
36005006800	1	2	0
36005007100	9	10	5
36005007200	0	1	0
36005007500	0	2	1
36005007600	1	0	0
36005007800	0	0	0

2.4.4 SELECT statements to prepare for full join: grad_outcomes table

[19]:
```
# Run query to review a sample of records
grd_full_select_stmnt01 = f"""
SELECT
    grd.dbn,
    grd.school_name,
    grd.cohort,
    grd.total_grads_n,
    grd.dropped_out_n,
    hsi.borough,
    hsi.census_tract,
    hsi.bin
FROM {database_name}.{grd_tsv_tbl_name} AS grd
INNER JOIN {database_name}.{hsi_tsv_tbl_name} AS hsi
    ON grd.dbn = hsi.dbn
WHERE census_tract IS NOT NULL
ORDER BY hsi.census_tract ASC
LIMIT 100000
"""

# Display SQL statement
print(grd_full_select_stmnt01)
```

```
# Run SQL statement against Athena table
grd_full_select_df01 = pd.read_sql(grd_full_select_stmnt01,
                                   conn)

# Display results
print(grd_full_select_df01.shape)
display(grd_full_select_df01.head(7))
```

```
SELECT
    grd.dbn,
    grd.school_name,
    grd.cohort,
    grd.total_grads_n,
    grd.dropped_out_n,
    hsi.borough,
    hsi.census_tract,
    hsi.bin
FROM ads508_t8.grad_outcomes AS grd
INNER JOIN ads508_t8.hs_info AS hsi
    ON grd.dbn = hsi.dbn
WHERE census_tract IS NOT NULL
ORDER BY hsi.census_tract ASC
LIMIT 100000
```

```
(0, 8)
```

```
Empty DataFrame
Columns: [dbn, school_name, cohort, total_grads_n, dropped_out_n, borough,␣
 ↪census_tract, bin]
Index: []
```

[20]:
```
# Run query to review a sample of records
grd_select_borough_stmnt01 = f"""
SELECT
    LOWER(hsi.borough) AS borough,
    CAST(grd.cohort AS INT) - 2006 AS relative_data_year,
    SUM(CAST(grd.total_grads_n AS DOUBLE)) AS annual_grad_n,
    SUM(CAST(grd.dropped_out_n AS DOUBLE)) AS annual_dropped_out_n
FROM {database_name}.{grd_tsv_tbl_name} AS grd
LEFT JOIN {database_name}.{hsi_tsv_tbl_name} AS hsi
    ON grd.dbn = hsi.dbn
WHERE total_grads_n <> 's'
    AND cohort != '2006 Aug'
    AND borough IS NOT null
    AND CAST(grd.cohort AS INT) BETWEEN 2002 AND 2006
GROUP BY hsi.borough, grd.cohort
```

```
ORDER BY hsi.borough, grd.cohort
LIMIT 100000
"""

# Display SQL statement
print(grd_select_borough_stmnt01)

# Run SQL statement against Athena table
grd_select_borough_df01 = pd.read_sql(grd_select_borough_stmnt01,
                                      conn)

# Display results
print(grd_select_borough_df01.shape)
display(grd_select_borough_df01.head(50))

# Create pivot table
grd_select_borough_df02 = grd_select_borough_df01.pivot_table(index = 'borough',
                                                              columns =␣
 ↪'relative_data_year',
                                                              values =␣
 ↪['annual_grad_n', 'annual_dropped_out_n'],
                                                              aggfunc = 'sum',
                                                              fill_value = 0)

print(grd_select_borough_df02.shape)
display(grd_select_borough_df02.head(35))
```

```
SELECT
    LOWER(hsi.borough) AS borough,
    CAST(grd.cohort AS INT) - 2006 AS relative_data_year,
    SUM(CAST(grd.total_grads_n AS DOUBLE)) AS annual_grad_n,
    SUM(CAST(grd.dropped_out_n AS DOUBLE)) AS annual_dropped_out_n
FROM ads508_t8.grad_outcomes AS grd
LEFT JOIN ads508_t8.hs_info AS hsi
    ON grd.dbn = hsi.dbn
WHERE total_grads_n <> 's'
    AND cohort != '2006 Aug'
    AND borough IS NOT null
    AND CAST(grd.cohort AS INT) BETWEEN 2002 AND 2006
GROUP BY hsi.borough, grd.cohort
ORDER BY hsi.borough, grd.cohort
LIMIT 100000

(25, 4)
```

	borough	relative_data_year	annual_grad_n	annual_dropped_out_n
0	bronx	-4	17130.0	2833.0

```
1           bronx                  -3        22123.0                4494.0
2           bronx                  -2        27594.0                4974.0
3           bronx                  -1        31643.0                5112.0
4           bronx                   0        33597.0                6251.0
5        brooklyn                  -4        38539.0                8505.0
6        brooklyn                  -3        44230.0                8488.0
7        brooklyn                  -2        51985.0                8323.0
8        brooklyn                  -1        53783.0                6934.0
9        brooklyn                   0        56436.0                7878.0
10      manhattan                  -4        28721.0                2741.0
11      manhattan                  -3        30950.0                3404.0
12      manhattan                  -2        34333.0                3079.0
13      manhattan                  -1        38817.0                2545.0
14      manhattan                   0        41153.0                3211.0
15         queens                  -4        47027.0               11220.0
16         queens                  -3        50268.0               10593.0
17         queens                  -2        53307.0                9598.0
18         queens                  -1        57757.0                9165.0
19         queens                   0        61196.0                8777.0
20  staten island                  -4        16133.0                2236.0
21  staten island                  -3        16651.0                1852.0
22  staten island                  -2        16391.0                1799.0
23  staten island                  -1        18054.0                1657.0
24  staten island                   0        19483.0                2035.0

(5, 10)

                   annual_dropped_out_n                          \
relative_data_year                   -4     -3    -2    -1     0
borough
bronx                              2833   4494  4974  5112  6251
brooklyn                           8505   8488  8323  6934  7878
manhattan                          2741   3404  3079  2545  3211
queens                            11220  10593  9598  9165  8777
staten island                      2236   1852  1799  1657  2035

                   annual_grad_n
relative_data_year            -4     -3     -2     -1      0
borough
bronx                      17130  22123  27594  31643  33597
brooklyn                   38539  44230  51985  53783  56436
manhattan                  28721  30950  34333  38817  41153
queens                     47027  50268  53307  57757  61196
staten island              16133  16651  16391  18054  19483
```

[21]:
```
# Run query to review a sample of records
grd_select_ct_stmnt01 = f"""
SELECT
```

```
    hsi.census_tract,
    SUM(CAST(grd.total_grads_n AS DOUBLE)) AS annual_grad_n,
    SUM(CAST(grd.dropped_out_n AS DOUBLE)) AS annual_dropped_out_n
FROM {database_name}.{grd_tsv_tbl_name} AS grd
LEFT JOIN {database_name}.{hsi_tsv_tbl_name} AS hsi
    ON grd.dbn = hsi.dbn
WHERE total_grads_n <> 's'
GROUP BY hsi.census_tract
ORDER BY hsi.census_tract
LIMIT 100000
"""

# Display SQL statement
print(grd_select_ct_stmnt01)

# Run SQL statement against Athena table
grd_select_ct_df01 = pd.read_sql(grd_select_ct_stmnt01,
                                 conn)

# Display results
print(grd_select_ct_df01.shape)
display(grd_select_ct_df01.head(50))
```

```
SELECT
    hsi.census_tract,
    SUM(CAST(grd.total_grads_n AS DOUBLE)) AS annual_grad_n,
    SUM(CAST(grd.dropped_out_n AS DOUBLE)) AS annual_dropped_out_n
FROM ads508_t8.grad_outcomes AS grd
LEFT JOIN ads508_t8.hs_info AS hsi
    ON grd.dbn = hsi.dbn
WHERE total_grads_n <> 's'
GROUP BY hsi.census_tract
ORDER BY hsi.census_tract
LIMIT 100000

(1, 3)
```

	census_tract	annual_grad_n	annual_dropped_out_n
0	None	1489323.0	308140.0

2.4.5 SELECT statements to prepare for full join: census table

[22]:
```
# Display full table for review
cen_full_select_stmnt01 = f"""
SELECT
    censustract AS census_tract,
    LOWER(borough) AS borough,
```

```
    totalpop,
    men,
    women,
    hispanic,
    white,
    black,
    native,
    asian,
    citizen,
    income,
    incomeerr,
    incomepercap,
    incomepercaperr,
    poverty,
    childpoverty,
    professional,
    service,
    office,
    construction,
    production,
    drive,
    carpool,
    transit,
    walk,
    othertransp,
    workathome,
    meancommute,
    employed,
    privatework,
    publicwork,
    selfemployed,
    familywork,
    unemployment
FROM {database_name}.{cen_tsv_tbl_name}
WHERE childpoverty IS NOT NULL
LIMIT 10000
"""

# Display SQL statement
print(cen_full_select_stmnt01)

# Run SQL statement against Athena table
cen_full_select_df01 = pd.read_sql(cen_full_select_stmnt01,
                                   conn)

# Display results
print(cen_full_select_df01.shape)
```

```
display(cen_full_select_df01.head(11))
```

```
SELECT
    censustract AS census_tract,
    LOWER(borough) AS borough,
    totalpop,
    men,
    women,
    hispanic,
    white,
    black,
    native,
    asian,
    citizen,
    income,
    incomeerr,
    incomepercap,
    incomepercaperr,
    poverty,
    childpoverty,
    professional,
    service,
    office,
    construction,
    production,
    drive,
    carpool,
    transit,
    walk,
    othertransp,
    workathome,
    meancommute,
    employed,
    privatework,
    publicwork,
    selfemployed,
    familywork,
    unemployment
FROM ads508_t8.census
WHERE childpoverty IS NOT NULL
LIMIT 10000

(2107, 35)
```

	census_tract	borough	totalpop	men	women	hispanic	white	black	\
0	36005000200	bronx	5403	2659	2744	75.8	2.3	16.0	
1	36005000400	bronx	5915	2896	3019	62.7	3.6	30.7	

```
2   36005001600   bronx      5879  2558   3321      65.1    1.6   32.4
3   36005001900   bronx      2591  1206   1385      55.4    9.0   29.0
4   36005002000   bronx      8516  3301   5215      61.1    1.6   31.1
5   36005002300   bronx      4774  2130   2644      62.3    0.2   36.5
6   36005002400   bronx       150   109     41       0.0   52.0   48.0
7   36005002500   bronx      5355  2338   3017      76.5    1.5   18.9
8   36005002701   bronx      3016  1375   1641      68.0    0.0   31.2
9   36005002702   bronx      4778  2427   2351      71.3    1.6   26.2
10  36005002800   bronx      5299  2292   3007      23.0    0.2   71.4

    native  asian  …  walk  othertransp  workathome  meancommute  employed  \
0      0.0    4.2  …   2.9          0.0         0.0         43.0      2308
1      0.0    0.3  …   1.4          0.5         2.1         45.0      2675
2      0.0    0.0  …   8.6          1.6         1.7         38.8      2120
3      0.0    2.1  …   3.0          2.4         6.2         45.4      1083
4      0.3    3.3  …   4.3          1.0         0.0         46.0      2508
5      1.0    0.0  …  14.0          1.5         4.1         42.7      1191
6      0.0    0.0  …   0.0          0.0         0.0          NaN       113
7      0.0    3.0  …  17.7          1.8         2.7         35.5      1691
8      0.0    0.0  …  18.0          0.0         1.6         42.8      1102
9      0.0    0.0  …   7.1          0.7         0.5         44.0      1559
10     0.0    1.7  …   2.0          0.6         2.7         47.3      2394

    privatework  publicwork  selfemployed  familywork  unemployment
0          80.8        16.2           2.9         0.0           7.7
1          71.7        25.3           2.5         0.6           9.5
2          75.0        21.3           3.8         0.0           8.7
3          76.8        15.5           7.7         0.0          19.2
4          71.0        21.3           7.7         0.0          17.2
5          74.2        16.1           9.7         0.0          18.9
6          62.8        37.2           0.0         0.0           0.0
7          85.1         8.3           6.1         0.5           9.4
8          86.9         8.5           4.5         0.0          15.2
9          75.0        14.0          11.0         0.0          10.6
10         61.9        37.4           0.6         0.0          12.8

[11 rows x 35 columns]
```

2.5 Setup ABT version 1 by joining census table with pivot tables of other tables (evictions, crime_pqt, and grad_outcomes) based on borough feature

[23]:
```
evi_borough_year_df03 = evi_borough_year_df02.reset_index()
cri_borough_year_type_df03 = cri_borough_year_type_df13.reset_index()
grd_select_borough_df03 = grd_select_borough_df02.reset_index()

display(cen_full_select_df01.head(11))
```

```
display(evi_borough_year_df03.head(5))
display(cri_borough_year_type_df03.head(5))
display(grd_select_borough_df03.head(5))

abt_df01 = pd.merge(cen_full_select_df01, evi_borough_year_df03,
                    on='borough')

abt_df01 = pd.merge(abt_df01, cri_borough_year_type_df03,
                    on='borough')

abt_df01 = pd.merge(abt_df01, grd_select_borough_df03,
                    on='borough')

display(abt_df01)
```

	census_tract	borough	totalpop	men	women	hispanic	white	black	\
0	36005000200	bronx	5403	2659	2744	75.8	2.3	16.0	
1	36005000400	bronx	5915	2896	3019	62.7	3.6	30.7	
2	36005001600	bronx	5879	2558	3321	65.1	1.6	32.4	
3	36005001900	bronx	2591	1206	1385	55.4	9.0	29.0	
4	36005002000	bronx	8516	3301	5215	61.1	1.6	31.1	
5	36005002300	bronx	4774	2130	2644	62.3	0.2	36.5	
6	36005002400	bronx	150	109	41	0.0	52.0	48.0	
7	36005002500	bronx	5355	2338	3017	76.5	1.5	18.9	
8	36005002701	bronx	3016	1375	1641	68.0	0.0	31.2	
9	36005002702	bronx	4778	2427	2351	71.3	1.6	26.2	
10	36005002800	bronx	5299	2292	3007	23.0	0.2	71.4	

	native	asian	…	walk	othertransp	workathome	meancommute	employed	\
0	0.0	4.2	…	2.9	0.0	0.0	43.0	2308	
1	0.0	0.3	…	1.4	0.5	2.1	45.0	2675	
2	0.0	0.0	…	8.6	1.6	1.7	38.8	2120	
3	0.0	2.1	…	3.0	2.4	6.2	45.4	1083	
4	0.3	3.3	…	4.3	1.0	0.0	46.0	2508	
5	1.0	0.0	…	14.0	1.5	4.1	42.7	1191	
6	0.0	0.0	…	0.0	0.0	0.0	NaN	113	
7	0.0	3.0	…	17.7	1.8	2.7	35.5	1691	
8	0.0	0.0	…	18.0	0.0	1.6	42.8	1102	
9	0.0	0.0	…	7.1	0.7	0.5	44.0	1559	
10	0.0	1.7	…	2.0	0.6	2.7	47.3	2394	

	privatework	publicwork	selfemployed	familywork	unemployment
0	80.8	16.2	2.9	0.0	7.7
1	71.7	25.3	2.5	0.6	9.5
2	75.0	21.3	3.8	0.0	8.7
3	76.8	15.5	7.7	0.0	19.2
4	71.0	21.3	7.7	0.0	17.2
5	74.2	16.1	9.7	0.0	18.9

```
6          62.8        37.2           0.0         0.0           0.0
7          85.1         8.3           6.1         0.5           9.4
8          86.9         8.5           4.5         0.0          15.2
9          75.0        14.0          11.0         0.0          10.6
10         61.9        37.4           0.6         0.0          12.8

[11 rows x 35 columns]
```

relative_data_year	borough	-4	-3	-2	-1	0
0	bronx	7140	6244	1088	29	1174
1	brooklyn	6157	5312	1005	100	1864
2	manhattan	3390	2818	521	68	930
3	queens	4452	3705	696	36	811
4	staten island	691	636	112	35	271

year_w_complaint	borough	2017 - FELONY	2017 - MISDEMEANOR
0	bronx	583	1179
1	brooklyn	862	1512
2	manhattan	701	1306
3	queens	610	932
4	staten island	100	249

year_w_complaint	2017 - VIOLATION	2018 - FELONY	2018 - MISDEMEANOR
0	289	540	1133
1	430	976	1473
2	267	736	1276
3	299	576	877
4	112	107	239

year_w_complaint	2018 - VIOLATION	2019 - FELONY	2019 - MISDEMEANOR
0	338	558	1114
1	423	909	1371
2	311	665	1302
3	309	604	970
4	109	101	196

year_w_complaint	2019 - VIOLATION	2020 - FELONY	2020 - MISDEMEANOR
0	328	529	954
1	435	803	1168
2	317	605	1085
3	292	607	911
4	61	104	175

year_w_complaint	2020 - VIOLATION	2021 - FELONY	2021 - MISDEMEANOR
0	322	617	910
1	384	834	1185
2	248	708	1145
3	290	646	995
4	57	96	174

```
year_w_complaint  2021 - VIOLATION
0                              320
1                              475
2                              295
3                              320
4                               75
```

```
                          borough annual_dropped_out_n                    \
relative_data_year                                     -4     -3    -2    -1
0                            bronx                   2833   4494  4974  5112
1                         brooklyn                   8505   8488  8323  6934
2                        manhattan                   2741   3404  3079  2545
3                           queens                  11220  10593  9598  9165
4                    staten island                   2236   1852  1799  1657

                         annual_grad_n
relative_data_year     0            -4     -3     -2     -1      0
0                   6251         17130  22123  27594  31643  33597
1                   7878         38539  44230  51985  53783  56436
2                   3211         28721  30950  34333  38817  41153
3                   8777         47027  50268  53307  57757  61196
4                   2035         16133  16651  16391  18054  19483
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:18: FutureWarning:
merging between different levels is deprecated and will be removed in a future
version. (1 levels on the left,2 on the right)
/opt/conda/lib/python3.7/site-packages/pandas/core/generic.py:4150:
PerformanceWarning: dropping on a non-lexsorted multi-index without a level
parameter may impact performance.
  obj = obj._drop_axis(labels, axis, level=level, errors=errors)
```

```
     census_tract        borough  totalpop   men  women  hispanic  white  \
0     36005000200          bronx      5403  2659   2744      75.8    2.3
1     36005000400          bronx      5915  2896   3019      62.7    3.6
2     36005001600          bronx      5879  2558   3321      65.1    1.6
3     36005001900          bronx      2591  1206   1385      55.4    9.0
4     36005002000          bronx      8516  3301   5215      61.1    1.6
…           …             …        …    …     …        …    …
2102  36085030301  staten island      4895  2371   2524      30.7   40.2
2103  36085030302  staten island      6279  3093   3186      35.8   28.7
2104  36085031901  staten island      2550   953   1597      27.1    6.2
2105  36085031902  staten island      4611  2043   2568      20.9   14.7
2106  36085032300  staten island      1131   597    534      45.5   24.0

      black  native  asian  …  (annual_dropped_out_n, -4)  \
0      16.0     0.0    4.2  …                        2833
1      30.7     0.0    0.3  …                        2833
2      32.4     0.0    0.0  …                        2833
```

```
3      29.0     0.0    2.1  …                          2833
4      31.1     0.3    3.3  …                          2833
…      …     …    …  …                                …
2102   11.6     0.0   16.0  …                          2236
2103   17.6     0.0   14.3  …                          2236
2104   60.4     0.0    6.3  …                          2236
2105   61.9     0.0    0.9  …                          2236
2106   29.7     0.0    0.0  …                          2236

      (annual_dropped_out_n, -3)  (annual_dropped_out_n, -2)  \
0                           4494                        4974
1                           4494                        4974
2                           4494                        4974
3                           4494                        4974
4                           4494                        4974
…                           …                           …
2102                        1852                        1799
2103                        1852                        1799
2104                        1852                        1799
2105                        1852                        1799
2106                        1852                        1799

      (annual_dropped_out_n, -1)  (annual_dropped_out_n, 0)  \
0                           5112                        6251
1                           5112                        6251
2                           5112                        6251
3                           5112                        6251
4                           5112                        6251
…                           …                           …
2102                        1657                        2035
2103                        1657                        2035
2104                        1657                        2035
2105                        1657                        2035
2106                        1657                        2035

      (annual_grad_n, -4)  (annual_grad_n, -3)  (annual_grad_n, -2)  \
0                   17130                22123                27594
1                   17130                22123                27594
2                   17130                22123                27594
3                   17130                22123                27594
4                   17130                22123                27594
…                   …                    …                    …
2102                16133                16651                16391
2103                16133                16651                16391
2104                16133                16651                16391
2105                16133                16651                16391
2106                16133                16651                16391
```

```
      (annual_grad_n, -1)  (annual_grad_n, 0)
0                   31643               33597
1                   31643               33597
2                   31643               33597
3                   31643               33597
4                   31643               33597
…                   …                   …
2102                18054               19483
2103                18054               19483
2104                18054               19483
2105                18054               19483
2106                18054               19483

[2107 rows x 65 columns]
```

2.6 Setup ABT version 2 by joining census table with pivot tables of other tables (evictions, crime_pqt, and grad_outcomes) based on census_tract or borough features

[24]:
```
evi_ceb_join_select_df03 = evi_ceb_join_select_df02.reset_index()
cri_ceb_join_select_df03 = cri_ceb_join_select_df02.reset_index()
grd_select_borough_df03 = grd_select_borough_df02.reset_index()

display(cen_full_select_df01.head(11))

display(evi_ceb_join_select_df03.head(5))
display(cri_ceb_join_select_df03.head(5))
display(grd_select_borough_df03.head(5))

abt_df02 = pd.merge(cen_full_select_df01,
                    evi_ceb_join_select_df03,
                    how='left',
                    on='census_tract')

abt_df02 = pd.merge(abt_df02,
                    cri_ceb_join_select_df03,
                    how='left',
                    on='census_tract')

abt_df02 = pd.merge(abt_df02,
                    grd_select_borough_df03,
                    on='borough')
display(abt_df02)
```

```
   census_tract borough  totalpop   men  women  hispanic  white  black  \
0   36005000200   bronx      5403  2659   2744      75.8    2.3   16.0
1   36005000400   bronx      5915  2896   3019      62.7    3.6   30.7
2   36005001600   bronx      5879  2558   3321      65.1    1.6   32.4
```

```
3   36005001900   bronx      2591  1206   1385      55.4    9.0   29.0
4   36005002000   bronx      8516  3301   5215      61.1    1.6   31.1
5   36005002300   bronx      4774  2130   2644      62.3    0.2   36.5
6   36005002400   bronx       150   109     41       0.0   52.0   48.0
7   36005002500   bronx      5355  2338   3017      76.5    1.5   18.9
8   36005002701   bronx      3016  1375   1641      68.0    0.0   31.2
9   36005002702   bronx      4778  2427   2351      71.3    1.6   26.2
10  36005002800   bronx      5299  2292   3007      23.0    0.2   71.4

    native  asian  …  walk  othertransp  workathome  meancommute  employed  \
0      0.0    4.2  …   2.9          0.0         0.0         43.0      2308
1      0.0    0.3  …   1.4          0.5         2.1         45.0      2675
2      0.0    0.0  …   8.6          1.6         1.7         38.8      2120
3      0.0    2.1  …   3.0          2.4         6.2         45.4      1083
4      0.3    3.3  …   4.3          1.0         0.0         46.0      2508
5      1.0    0.0  …  14.0          1.5         4.1         42.7      1191
6      0.0    0.0  …   0.0          0.0         0.0          NaN       113
7      0.0    3.0  …  17.7          1.8         2.7         35.5      1691
8      0.0    0.0  …  18.0          0.0         1.6         42.8      1102
9      0.0    0.0  …   7.1          0.7         0.5         44.0      1559
10     0.0    1.7  …   2.0          0.6         2.7         47.3      2394

    privatework  publicwork  selfemployed  familywork  unemployment
0          80.8        16.2           2.9         0.0           7.7
1          71.7        25.3           2.5         0.6           9.5
2          75.0        21.3           3.8         0.0           8.7
3          76.8        15.5           7.7         0.0          19.2
4          71.0        21.3           7.7         0.0          17.2
5          74.2        16.1           9.7         0.0          18.9
6          62.8        37.2           0.0         0.0           0.0
7          85.1         8.3           6.1         0.5           9.4
8          86.9         8.5           4.5         0.0          15.2
9          75.0        14.0          11.0         0.0          10.6
10         61.9        37.4           0.6         0.0          12.8

[11 rows x 35 columns]

year census_tract  2017  2018  2019  2020  2021  2022  2023
0     34003013001     9    11     6     0     1     6     1
1     34003016000    14    13    16     2     2     1     0
2     34017010800     2     2     5     0     0     0     0
3     36005000200     9    10    10     0     0     1     1
4     36005000400    10    16    15     0     0     5     0

year census_tract  2017 - FELONY  2017 - MISDEMEANOR  2017 - VIOLATION  \
0     34003013001              1                   6                 4
1     34003016000              0                   4                 0
2     34017010800              0                   1                 0
3     36005000100             16                  10                 0
```

```
4     36005000200              0                   1                1

year  2018 - FELONY  2018 - MISDEMEANOR  2018 - VIOLATION  2019 - FELONY  \
0                 3                   5                 0              1
1                 3                   0                 2              3
2                 0                   0                 0              0
3                 5                  10                 0             10
4                 3                   3                 1              1

year  2019 - MISDEMEANOR  2019 - VIOLATION  2020 - FELONY  2020 - MISDEMEANOR  \
0                      6                 1              3                   2
1                      1                 0              0                   0
2                      0                 0              0                   1
3                      4                 1              3                   2
4                      2                 2              3                   0

year  2020 - VIOLATION  2021 - FELONY  2021 - MISDEMEANOR  2021 - VIOLATION
0                    1              6                   5                 2
1                    0              3                   1                 0
2                    0              0                   0                 0
3                    0              2                   1                 0
4                    0              3                   2                 1

                         borough annual_dropped_out_n                    \
relative_data_year                                   -4     -3    -2    -1
0                           bronx                  2833   4494  4974  5112
1                        brooklyn                  8505   8488  8323  6934
2                       manhattan                  2741   3404  3079  2545
3                          queens                 11220  10593  9598  9165
4                  staten island                  2236   1852  1799  1657

                        annual_grad_n
relative_data_year     0            -4     -3     -2     -1      0
0                   6251         17130  22123  27594  31643  33597
1                   7878         38539  44230  51985  53783  56436
2                   3211         28721  30950  34333  38817  41153
3                   8777         47027  50268  53307  57757  61196
4                   2035         16133  16651  16391  18054  19483

/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:23: FutureWarning:
merging between different levels is deprecated and will be removed in a future
version. (1 levels on the left,2 on the right)
/opt/conda/lib/python3.7/site-packages/pandas/core/generic.py:4150:
PerformanceWarning: dropping on a non-lexsorted multi-index without a level
parameter may impact performance.
  obj = obj._drop_axis(labels, axis, level=level, errors=errors)

     census_tract        borough  totalpop   men  women  hispanic  white  \
0     36005000200          bronx      5403  2659   2744      75.8    2.3
```

```
1     36005000400          bronx      5915  2896   3019      62.7    3.6
2     36005001600          bronx      5879  2558   3321      65.1    1.6
3     36005001900          bronx      2591  1206   1385      55.4    9.0
4     36005002000          bronx      8516  3301   5215      61.1    1.6
…           …              …        …    …    …        …    …
2102  36085030301  staten island      4895  2371   2524      30.7   40.2
2103  36085030302  staten island      6279  3093   3186      35.8   28.7
2104  36085031901  staten island      2550   953   1597      27.1    6.2
2105  36085031902  staten island      4611  2043   2568      20.9   14.7
2106  36085032300  staten island      1131   597    534      45.5   24.0

      black  native  asian  …  (annual_dropped_out_n, -4)  \
0      16.0     0.0    4.2  …                        2833
1      30.7     0.0    0.3  …                        2833
2      32.4     0.0    0.0  …                        2833
3      29.0     0.0    2.1  …                        2833
4      31.1     0.3    3.3  …                        2833
…      …      …     …   …                          …
2102   11.6     0.0   16.0  …                        2236
2103   17.6     0.0   14.3  …                        2236
2104   60.4     0.0    6.3  …                        2236
2105   61.9     0.0    0.9  …                        2236
2106   29.7     0.0    0.0  …                        2236

      (annual_dropped_out_n, -3)  (annual_dropped_out_n, -2)  \
0                           4494                        4974
1                           4494                        4974
2                           4494                        4974
3                           4494                        4974
4                           4494                        4974
…                           …                          …
2102                        1852                        1799
2103                        1852                        1799
2104                        1852                        1799
2105                        1852                        1799
2106                        1852                        1799

      (annual_dropped_out_n, -1)  (annual_dropped_out_n, 0)  \
0                           5112                       6251
1                           5112                       6251
2                           5112                       6251
3                           5112                       6251
4                           5112                       6251
…                           …                          …
2102                        1657                       2035
2103                        1657                       2035
2104                        1657                       2035
2105                        1657                       2035
```

```
2106                        1657                        2035

      (annual_grad_n, -4)  (annual_grad_n, -3)  (annual_grad_n, -2)  \
0                   17130                22123                27594
1                   17130                22123                27594
2                   17130                22123                27594
3                   17130                22123                27594
4                   17130                22123                27594
…                     …                    …                    …
2102                16133                16651                16391
2103                16133                16651                16391
2104                16133                16651                16391
2105                16133                16651                16391
2106                16133                16651                16391

      (annual_grad_n, -1)  (annual_grad_n, 0)
0                   31643               33597
1                   31643               33597
2                   31643               33597
3                   31643               33597
4                   31643               33597
…                     …                   …
2102                18054               19483
2103                18054               19483
2104                18054               19483
2105                18054               19483
2106                18054               19483

[2107 rows x 67 columns]
```

2.7 Setup ABT version 3 (FINAL) by joining census table with pivot tables of other tables (evictions, crime_pqt, and grad_outcomes) based on borough feature with or without relative_data_year

[25]:
```
cen_full_select_df02 = cen_full_select_df01.drop(['census_tract'],
                                                 axis=1)

display(cen_full_select_df02.head(11))
print(cen_full_select_df02.shape)

display(evi_borough_year_df01.head(11))
print(evi_borough_year_df01.shape)
display(cri_borough_year_type_df01.head(11))
print(cri_borough_year_type_df01.shape)
display(grd_select_borough_df01.head(11))
print(grd_select_borough_df01.shape)
```

```
   borough  totalpop   men  women  hispanic  white  black  native  asian  \
```

```
0    bronx      5403  2659   2744      75.8    2.3   16.0     0.0    4.2
1    bronx      5915  2896   3019      62.7    3.6   30.7     0.0    0.3
2    bronx      5879  2558   3321      65.1    1.6   32.4     0.0    0.0
3    bronx      2591  1206   1385      55.4    9.0   29.0     0.0    2.1
4    bronx      8516  3301   5215      61.1    1.6   31.1     0.3    3.3
5    bronx      4774  2130   2644      62.3    0.2   36.5     1.0    0.0
6    bronx       150   109     41       0.0   52.0   48.0     0.0    0.0
7    bronx      5355  2338   3017      76.5    1.5   18.9     0.0    3.0
8    bronx      3016  1375   1641      68.0    0.0   31.2     0.0    0.0
9    bronx      4778  2427   2351      71.3    1.6   26.2     0.0    0.0
10   bronx      5299  2292   3007      23.0    0.2   71.4     0.0    1.7

    citizen  …  walk  othertransp  workathome  meancommute  employed  \
0      3639  …   2.9          0.0         0.0         43.0      2308
1      4100  …   1.4          0.5         2.1         45.0      2675
2      3536  …   8.6          1.6         1.7         38.8      2120
3      1557  …   3.0          2.4         6.2         45.4      1083
4      5436  …   4.3          1.0         0.0         46.0      2508
5      3056  …  14.0          1.5         4.1         42.7      1191
6        41  …   0.0          0.0         0.0          NaN       113
7      2509  …  17.7          1.8         2.7         35.5      1691
8      1456  …  18.0          0.0         1.6         42.8      1102
9      2365  …   7.1          0.7         0.5         44.0      1559
10     4056  …   2.0          0.6         2.7         47.3      2394

    privatework  publicwork  selfemployed  familywork  unemployment
0          80.8        16.2           2.9         0.0           7.7
1          71.7        25.3           2.5         0.6           9.5
2          75.0        21.3           3.8         0.0           8.7
3          76.8        15.5           7.7         0.0          19.2
4          71.0        21.3           7.7         0.0          17.2
5          74.2        16.1           9.7         0.0          18.9
6          62.8        37.2           0.0         0.0           0.0
7          85.1         8.3           6.1         0.5           9.4
8          86.9         8.5           4.5         0.0          15.2
9          75.0        14.0          11.0         0.0          10.6
10         61.9        37.4           0.6         0.0          12.8

[11 rows x 34 columns]

(2107, 34)

      borough  relative_data_year  annual_evictions_x_borough
0       bronx                  -4                        7140
1       bronx                  -3                        6244
2       bronx                  -2                        1088
3       bronx                  -1                          29
4       bronx                   0                        1174
5    brooklyn                  -4                        6157
```

```
6    brooklyn                  -3                        5312
7    brooklyn                  -2                        1005
8    brooklyn                  -1                         100
9    brooklyn                   0                        1864
10  manhattan                  -4                        3390
```

```
(25, 3)
```

```
   borough  relative_data_year complaint_type  annual_complaint_counts
0    bronx                  -4         FELONY                      583
1    bronx                  -4    MISDEMEANOR                     1179
2    bronx                  -4      VIOLATION                      289
3    bronx                  -3         FELONY                      540
4    bronx                  -3    MISDEMEANOR                     1133
5    bronx                  -3      VIOLATION                      338
6    bronx                  -2         FELONY                      558
7    bronx                  -2    MISDEMEANOR                     1114
8    bronx                  -2      VIOLATION                      328
9    bronx                  -1         FELONY                      529
10   bronx                  -1    MISDEMEANOR                      954
```

```
(81, 4)
```

```
      borough  relative_data_year  annual_grad_n  annual_dropped_out_n
0       bronx                  -4        17130.0                2833.0
1       bronx                  -3        22123.0                4494.0
2       bronx                  -2        27594.0                4974.0
3       bronx                  -1        31643.0                5112.0
4       bronx                   0        33597.0                6251.0
5    brooklyn                  -4        38539.0                8505.0
6    brooklyn                  -3        44230.0                8488.0
7    brooklyn                  -2        51985.0                8323.0
8    brooklyn                  -1        53783.0                6934.0
9    brooklyn                   0        56436.0                7878.0
10  manhattan                  -4        28721.0                2741.0
```

```
(25, 4)
```

[26]:
```
abt_df03 = pd.merge(evi_borough_year_df01,
                    cri_borough_year_type_df01,
                    how='inner',
                    left_on=['borough', 'relative_data_year'],
                    right_on=['borough', 'relative_data_year'])

abt_df03 = pd.merge(abt_df03,
                    grd_select_borough_df01,
                    how='inner',
                    left_on=['borough', 'relative_data_year'],
                    right_on=['borough', 'relative_data_year'])
```

```
abt_df03 = pd.merge(abt_df03,
                    cen_full_select_df02,
                    how='inner',
                    on='borough')

print(abt_df03.shape)
display(abt_df03.head(11))
```

```
(31605, 40)
```

	borough	relative_data_year	annual_evictions_x_borough	complaint_type	\
0	bronx	-4	7140	FELONY	
1	bronx	-4	7140	FELONY	
2	bronx	-4	7140	FELONY	
3	bronx	-4	7140	FELONY	
4	bronx	-4	7140	FELONY	
5	bronx	-4	7140	FELONY	
6	bronx	-4	7140	FELONY	
7	bronx	-4	7140	FELONY	
8	bronx	-4	7140	FELONY	
9	bronx	-4	7140	FELONY	
10	bronx	-4	7140	FELONY	

	annual_complaint_counts	annual_grad_n	annual_dropped_out_n	totalpop	\
0	583	17130.0	2833.0	5403	
1	583	17130.0	2833.0	5915	
2	583	17130.0	2833.0	5879	
3	583	17130.0	2833.0	2591	
4	583	17130.0	2833.0	8516	
5	583	17130.0	2833.0	4774	
6	583	17130.0	2833.0	150	
7	583	17130.0	2833.0	5355	
8	583	17130.0	2833.0	3016	
9	583	17130.0	2833.0	4778	
10	583	17130.0	2833.0	5299	

	men	women	…	walk	othertransp	workathome	meancommute	employed	\
0	2659	2744	…	2.9	0.0	0.0	43.0	2308	
1	2896	3019	…	1.4	0.5	2.1	45.0	2675	
2	2558	3321	…	8.6	1.6	1.7	38.8	2120	
3	1206	1385	…	3.0	2.4	6.2	45.4	1083	
4	3301	5215	…	4.3	1.0	0.0	46.0	2508	
5	2130	2644	…	14.0	1.5	4.1	42.7	1191	
6	109	41	…	0.0	0.0	0.0	NaN	113	
7	2338	3017	…	17.7	1.8	2.7	35.5	1691	
8	1375	1641	…	18.0	0.0	1.6	42.8	1102	
9	2427	2351	…	7.1	0.7	0.5	44.0	1559	
10	2292	3007	…	2.0	0.6	2.7	47.3	2394	

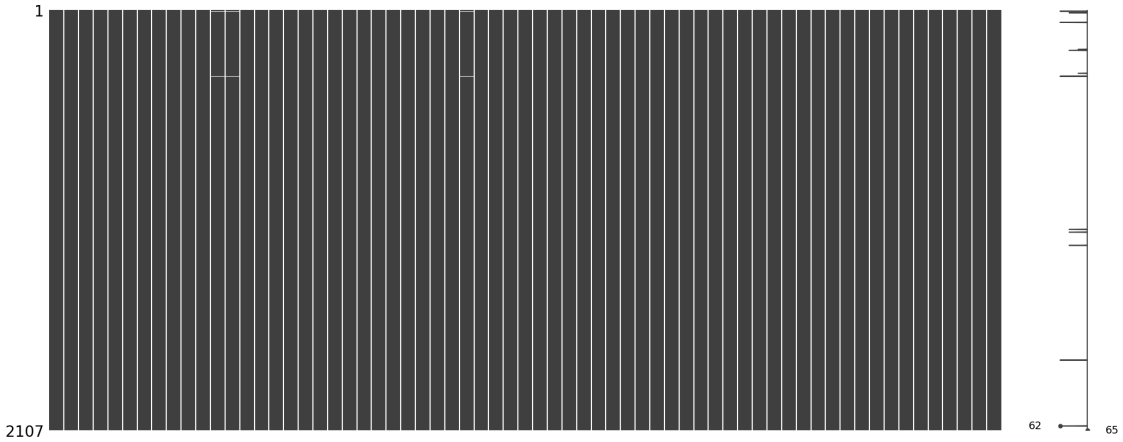
	privatework	publicwork	selfemployed	familywork	unemployment
0	80.8	16.2	2.9	0.0	7.7
1	71.7	25.3	2.5	0.6	9.5
2	75.0	21.3	3.8	0.0	8.7
3	76.8	15.5	7.7	0.0	19.2
4	71.0	21.3	7.7	0.0	17.2
5	74.2	16.1	9.7	0.0	18.9
6	62.8	37.2	0.0	0.0	0.0
7	85.1	8.3	6.1	0.5	9.4
8	86.9	8.5	4.5	0.0	15.2
9	75.0	14.0	11.0	0.0	10.6
10	61.9	37.4	0.6	0.0	12.8

[11 rows x 40 columns]

2.7.1 Check missing values for resulting ABTs

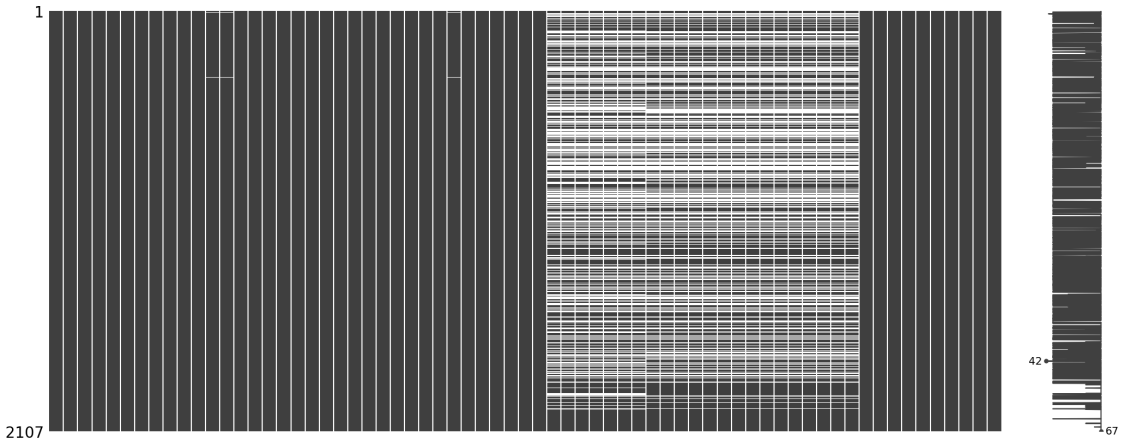
[27]:
```
# Visualize missing values in each column
msno.matrix(abt_df01)
```

[27]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe7fa33eb50>



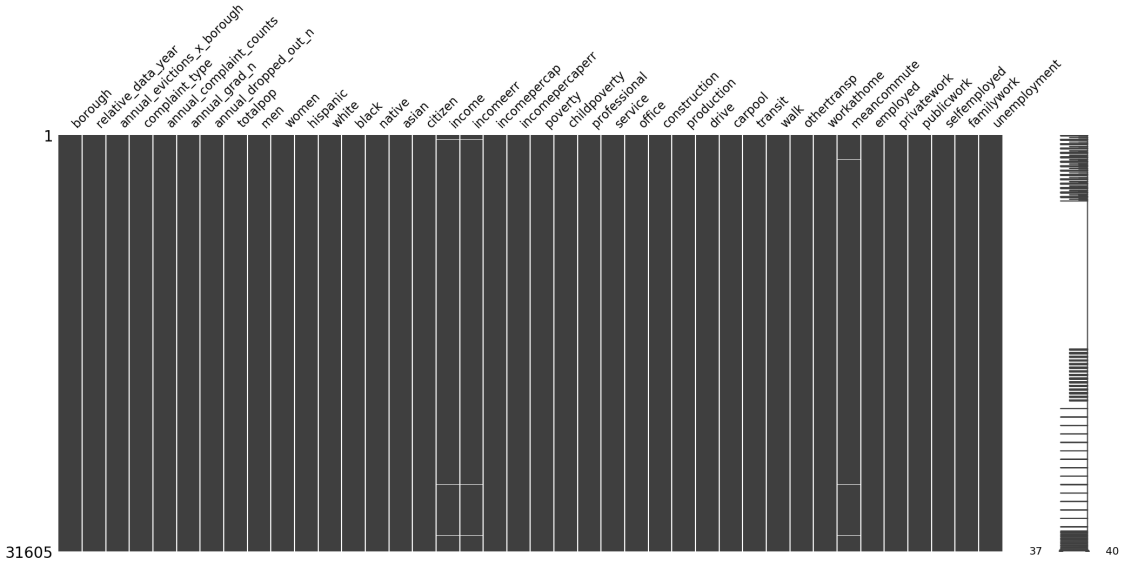
[28]:
```
# Visualize missing values in each column
msno.matrix(abt_df02)
```

[28]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe7fabc62d0>



[29]:
```
# Visualize missing values in each column
msno.matrix(abt_df03)
```

[29]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe7faf91c90>



[30]:
```
# Remove any features for which the number of null vals exceed a threshold--
#-- (5% of total N)
abt_df02_null_summ01 = pd.DataFrame(abt_df02.isnull().sum(),
                                    columns=['null_count'])

abt_df02_null_summ02 = abt_df02_null_summ01.
 ↪loc[(abt_df02_null_summ01['null_count'] != 0)].sort_values('null_count',
```

```
                                                                        ␣
  ↪                                    ascending=False)
abt_df02_null_summ03 = abt_df02_null_summ02.reset_index()
print(abt_df02_null_summ03)

abt_df02_null_summ04 = abt_df02_null_summ03.
  ↪loc[abt_df02_null_summ03['null_count'] > (len(abt_df02)*.05)]
print('\n', abt_df02_null_summ04)

abt_df02_null_summ04_remove_lst01 = list(abt_df02_null_summ04['index'])
print('\n', abt_df02_null_summ04_remove_lst01)
```

```
                index  null_count
0                2017         857
1                2018         857
2                2019         857
3                2020         857
4                2021         857
5                2022         857
6                2023         857
7     2017 - VIOLATION         812
8        2019 - FELONY         812
9   2021 - MISDEMEANOR         812
10       2021 - FELONY         812
11    2020 - VIOLATION         812
12  2020 - MISDEMEANOR         812
13       2020 - FELONY         812
14    2019 - VIOLATION         812
15  2019 - MISDEMEANOR         812
16    2021 - VIOLATION         812
17    2018 - VIOLATION         812
18  2018 - MISDEMEANOR         812
19       2018 - FELONY         812
20  2017 - MISDEMEANOR         812
21       2017 - FELONY         812
22           incomeerr          10
23              income          10
24        meancommute           7

                index  null_count
0                2017         857
1                2018         857
2                2019         857
3                2020         857
4                2021         857
5                2022         857
6                2023         857
```

```
7     2017 - VIOLATION         812
8        2019 - FELONY         812
9   2021 - MISDEMEANOR         812
10       2021 - FELONY         812
11    2020 - VIOLATION         812
12  2020 - MISDEMEANOR         812
13       2020 - FELONY         812
14    2019 - VIOLATION         812
15  2019 - MISDEMEANOR         812
16    2021 - VIOLATION         812
17    2018 - VIOLATION         812
18  2018 - MISDEMEANOR         812
19       2018 - FELONY         812
20  2017 - MISDEMEANOR         812
21       2017 - FELONY         812
```

```
 [2017, 2018, 2019, 2020, 2021, 2022, 2023, '2017 - VIOLATION', '2019 - FELONY',
'2021 - MISDEMEANOR', '2021 - FELONY', '2020 - VIOLATION', '2020 - MISDEMEANOR',
'2020 - FELONY', '2019 - VIOLATION', '2019 - MISDEMEANOR', '2021 - VIOLATION',
'2018 - VIOLATION', '2018 - MISDEMEANOR', '2018 - FELONY', '2017 - MISDEMEANOR',
'2017 - FELONY']
```

2.7.2 Create pipeline for One Hot Encoding

[31]:
```
'''Setup pipelne citation:
OpenAI. (2021). ChatGPT [Computer software]. https://openai.com/'''

print(abt_df03.shape)
display(abt_df03.head(11))

# Define a ColumnTransformer to apply the OneHotEncoder to the selected columns
cols_to_encode = ['borough',
                  'relative_data_year',
                  'complaint_type']

ct = ColumnTransformer(transformers=[('encoder',
                                      OneHotEncoder(),
                                      cols_to_encode)],
                       remainder='passthrough'
)

# Define a Pipeline to apply the ColumnTransformer
abt_pipe = Pipeline(steps=[('preprocessor',
                            ct)])

# Fit and transform the Pipeline to one-hot encode the selected columns
abt_encoded_df01 = pd.DataFrame(abt_pipe.fit_transform(abt_df03))
```

```
# Get the names of the one-hot encoded columns from the OneHotEncoder object
encoder = abt_pipe.named_steps['preprocessor'].named_transformers_['encoder']
print(encoder)

encoded_cols = encoder.get_feature_names(cols_to_encode)
print(encoded_cols)

# Get the names of the non-encoded columns by removing the columns that were␣
 ↪encoded
non_encoded_cols = [col for col in abt_df03.columns if col not in␣
 ↪cols_to_encode]
print(non_encoded_cols)

# Concatenate the one-hot encoded columns with the non-encoded columns to␣
 ↪obtain the new DataFrame with the desired column names
abt_encoded_df01.columns = list(encoded_cols) + non_encoded_cols
print(abt_encoded_df01.head(11))

display(abt_encoded_df01.head(11))
```

```
(31605, 40)
```

	borough	relative_data_year	annual_evictions_x_borough	complaint_type	\
0	bronx	-4	7140	FELONY	
1	bronx	-4	7140	FELONY	
2	bronx	-4	7140	FELONY	
3	bronx	-4	7140	FELONY	
4	bronx	-4	7140	FELONY	
5	bronx	-4	7140	FELONY	
6	bronx	-4	7140	FELONY	
7	bronx	-4	7140	FELONY	
8	bronx	-4	7140	FELONY	
9	bronx	-4	7140	FELONY	
10	bronx	-4	7140	FELONY	

	annual_complaint_counts	annual_grad_n	annual_dropped_out_n	totalpop	\
0	583	17130.0	2833.0	5403	
1	583	17130.0	2833.0	5915	
2	583	17130.0	2833.0	5879	
3	583	17130.0	2833.0	2591	
4	583	17130.0	2833.0	8516	
5	583	17130.0	2833.0	4774	
6	583	17130.0	2833.0	150	
7	583	17130.0	2833.0	5355	
8	583	17130.0	2833.0	3016	
9	583	17130.0	2833.0	4778	
10	583	17130.0	2833.0	5299	

```
     men  women  …  walk  othertransp  workathome  meancommute  employed  \
0   2659   2744  …   2.9          0.0         0.0         43.0      2308
1   2896   3019  …   1.4          0.5         2.1         45.0      2675
2   2558   3321  …   8.6          1.6         1.7         38.8      2120
3   1206   1385  …   3.0          2.4         6.2         45.4      1083
4   3301   5215  …   4.3          1.0         0.0         46.0      2508
5   2130   2644  …  14.0          1.5         4.1         42.7      1191
6    109     41  …   0.0          0.0         0.0          NaN       113
7   2338   3017  …  17.7          1.8         2.7         35.5      1691
8   1375   1641  …  18.0          0.0         1.6         42.8      1102
9   2427   2351  …   7.1          0.7         0.5         44.0      1559
10  2292   3007  …   2.0          0.6         2.7         47.3      2394

    privatework  publicwork  selfemployed  familywork  unemployment
0          80.8        16.2           2.9         0.0           7.7
1          71.7        25.3           2.5         0.6           9.5
2          75.0        21.3           3.8         0.0           8.7
3          76.8        15.5           7.7         0.0          19.2
4          71.0        21.3           7.7         0.0          17.2
5          74.2        16.1           9.7         0.0          18.9
6          62.8        37.2           0.0         0.0           0.0
7          85.1         8.3           6.1         0.5           9.4
8          86.9         8.5           4.5         0.0          15.2
9          75.0        14.0          11.0         0.0          10.6
10         61.9        37.4           0.6         0.0          12.8

[11 rows x 40 columns]
```

```
OneHotEncoder(categories='auto', drop=None, dtype=<class 'numpy.float64'>,
              handle_unknown='error', sparse=True)
['borough_bronx' 'borough_brooklyn' 'borough_manhattan' 'borough_queens'
 'borough_staten island' 'relative_data_year_-4' 'relative_data_year_-3'
 'relative_data_year_-2' 'relative_data_year_-1' 'relative_data_year_0'
 'complaint_type_FELONY' 'complaint_type_MISDEMEANOR'
 'complaint_type_VIOLATION']
['annual_evictions_x_borough', 'annual_complaint_counts', 'annual_grad_n',
'annual_dropped_out_n', 'totalpop', 'men', 'women', 'hispanic', 'white',
'black', 'native', 'asian', 'citizen', 'income', 'incomeerr', 'incomepercap',
'incomepercaperr', 'poverty', 'childpoverty', 'professional', 'service',
'office', 'construction', 'production', 'drive', 'carpool', 'transit', 'walk',
'othertransp', 'workathome', 'meancommute', 'employed', 'privatework',
'publicwork', 'selfemployed', 'familywork', 'unemployment']
    borough_bronx  borough_brooklyn  borough_manhattan  borough_queens  \
0             1.0               0.0                0.0             0.0
1             1.0               0.0                0.0             0.0
2             1.0               0.0                0.0             0.0
3             1.0               0.0                0.0             0.0
```

```
4              1.0                0.0                0.0              0.0
5              1.0                0.0                0.0              0.0
6              1.0                0.0                0.0              0.0
7              1.0                0.0                0.0              0.0
8              1.0                0.0                0.0              0.0
9              1.0                0.0                0.0              0.0
10             1.0                0.0                0.0              0.0

    borough_staten island  relative_data_year_-4  relative_data_year_-3  \
0                     0.0                    1.0                    0.0
1                     0.0                    1.0                    0.0
2                     0.0                    1.0                    0.0
3                     0.0                    1.0                    0.0
4                     0.0                    1.0                    0.0
5                     0.0                    1.0                    0.0
6                     0.0                    1.0                    0.0
7                     0.0                    1.0                    0.0
8                     0.0                    1.0                    0.0
9                     0.0                    1.0                    0.0
10                    0.0                    1.0                    0.0

    relative_data_year_-2  relative_data_year_-1  relative_data_year_0  …  \
0                     0.0                    0.0                   0.0  …
1                     0.0                    0.0                   0.0  …
2                     0.0                    0.0                   0.0  …
3                     0.0                    0.0                   0.0  …
4                     0.0                    0.0                   0.0  …
5                     0.0                    0.0                   0.0  …
6                     0.0                    0.0                   0.0  …
7                     0.0                    0.0                   0.0  …
8                     0.0                    0.0                   0.0  …
9                     0.0                    0.0                   0.0  …
10                    0.0                    0.0                   0.0  …

    walk  othertransp  workathome  meancommute  employed  privatework  \
0    2.9          0.0         0.0         43.0    2308.0         80.8
1    1.4          0.5         2.1         45.0    2675.0         71.7
2    8.6          1.6         1.7         38.8    2120.0         75.0
3    3.0          2.4         6.2         45.4    1083.0         76.8
4    4.3          1.0         0.0         46.0    2508.0         71.0
5   14.0          1.5         4.1         42.7    1191.0         74.2
6    0.0          0.0         0.0          NaN     113.0         62.8
7   17.7          1.8         2.7         35.5    1691.0         85.1
8   18.0          0.0         1.6         42.8    1102.0         86.9
9    7.1          0.7         0.5         44.0    1559.0         75.0
10   2.0          0.6         2.7         47.3    2394.0         61.9

    publicwork  selfemployed  familywork  unemployment
```

```
0         16.2           2.9         0.0           7.7
1         25.3           2.5         0.6           9.5
2         21.3           3.8         0.0           8.7
3         15.5           7.7         0.0          19.2
4         21.3           7.7         0.0          17.2
5         16.1           9.7         0.0          18.9
6         37.2           0.0         0.0           0.0
7          8.3           6.1         0.5           9.4
8          8.5           4.5         0.0          15.2
9         14.0          11.0         0.0          10.6
10        37.4           0.6         0.0          12.8

[11 rows x 50 columns]
    borough_bronx  borough_brooklyn  borough_manhattan  borough_queens  \
0             1.0               0.0                0.0             0.0
1             1.0               0.0                0.0             0.0
2             1.0               0.0                0.0             0.0
3             1.0               0.0                0.0             0.0
4             1.0               0.0                0.0             0.0
5             1.0               0.0                0.0             0.0
6             1.0               0.0                0.0             0.0
7             1.0               0.0                0.0             0.0
8             1.0               0.0                0.0             0.0
9             1.0               0.0                0.0             0.0
10            1.0               0.0                0.0             0.0

    borough_staten island  relative_data_year_-4  relative_data_year_-3  \
0                     0.0                    1.0                    0.0
1                     0.0                    1.0                    0.0
2                     0.0                    1.0                    0.0
3                     0.0                    1.0                    0.0
4                     0.0                    1.0                    0.0
5                     0.0                    1.0                    0.0
6                     0.0                    1.0                    0.0
7                     0.0                    1.0                    0.0
8                     0.0                    1.0                    0.0
9                     0.0                    1.0                    0.0
10                    0.0                    1.0                    0.0

    relative_data_year_-2  relative_data_year_-1  relative_data_year_0  …  \
0                     0.0                    0.0                   0.0  …
1                     0.0                    0.0                   0.0  …
2                     0.0                    0.0                   0.0  …
3                     0.0                    0.0                   0.0  …
4                     0.0                    0.0                   0.0  …
5                     0.0                    0.0                   0.0  …
6                     0.0                    0.0                   0.0  …
```

```
7                    0.0                    0.0                    0.0  …
8                    0.0                    0.0                    0.0  …
9                    0.0                    0.0                    0.0  …
10                   0.0                    0.0                    0.0  …

    walk  othertransp  workathome  meancommute  employed  privatework  \
0    2.9          0.0         0.0         43.0    2308.0         80.8
1    1.4          0.5         2.1         45.0    2675.0         71.7
2    8.6          1.6         1.7         38.8    2120.0         75.0
3    3.0          2.4         6.2         45.4    1083.0         76.8
4    4.3          1.0         0.0         46.0    2508.0         71.0
5   14.0          1.5         4.1         42.7    1191.0         74.2
6    0.0          0.0         0.0          NaN     113.0         62.8
7   17.7          1.8         2.7         35.5    1691.0         85.1
8   18.0          0.0         1.6         42.8    1102.0         86.9
9    7.1          0.7         0.5         44.0    1559.0         75.0
10   2.0          0.6         2.7         47.3    2394.0         61.9

    publicwork  selfemployed  familywork  unemployment
0         16.2           2.9         0.0           7.7
1         25.3           2.5         0.6           9.5
2         21.3           3.8         0.0           8.7
3         15.5           7.7         0.0          19.2
4         21.3           7.7         0.0          17.2
5         16.1           9.7         0.0          18.9
6         37.2           0.0         0.0           0.0
7          8.3           6.1         0.5           9.4
8          8.5           4.5         0.0          15.2
9         14.0          11.0         0.0          10.6
10        37.4           0.6         0.0          12.8

[11 rows x 50 columns]
```

2.7.3 Save ABT to S3

[32]:
```
s3_abt_csv_path = f"s3://{def_bucket}/team_8_data/abt/abt_encoded_df01.csv"
abt_encoded_df01.to_csv(s3_abt_csv_path, index=False, header=True)
```

2.8 Show the Tables

[33]:
```
show_tbl_stmnt = f"SHOW TABLES in {database_name}"
```

[34]:
```
df_tables = pd.read_sql(show_tbl_stmnt,
                        conn)

df_tables.head(17)
```

[34]:
```
         tab_name
0          census
1   census_block
2           crime
3       crime_pqt
4       evictions
5  grad_outcomes
6         hs_info
7            jobs
```

2.9 Review the New Athena Table in the Glue Catalog

[35]:
```
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>
```

2.10 Store Variables for the Next Notebooks

[36]:
```
%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
```

2.11 Release Resources

[37]:
```
%%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>
```

[38]:
```
%%javascript

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

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