team five

Georgia Institute of Technology
Data Science Boot Camp, Team: Five

05.18.19 | Project 02:
Consumer shopping by location & pattern, and crime

team five

Megan

Ivan

Dan

EXTRACT
TRANSFORM
LOAD

team five

OBJECTIVE

Extract, transform, load, three database functions that are combined into one tool to pull data out of one database and place it into another database.

EXTRACT
is the process of reading data from a database. In this stage, the data is collected, often from multiple and different types of sources.

TRANSFORM
is the process of converting the extracted data from its previous form into the form it needs to be in so that it can be placed into another database. Transformation occurs by using rules or lookup tables or by combining the data with other data.

LOAD
is the process of writing the data into the target database.

EXTRACT
TRANSFORM
LOAD
ETL

team five

RELIABLE DATA SOURCES UTILIZED

Our data sets are thoughtfully chosen in an effort to combine consumer shopping by location and pattern with crime activity in specific areas as a potential indicator of consumer behavior in proximity to criminal activity versus relying upon the assumptions of convenience and price.

CONSUMER & VISITOR INSIGHTS FOR NEIGHBORHOODS

- SAFEGRAPH
- SAFEGRAPH PLACES
- SAFEGRAPH PLACES PATTERNS

CRIME AND INCARCERATION IN THE UNITED STATES

- THE BUREAU OF JUSTICE STATISTICS ADMINISTERS THE NATIONAL PRISONERS STATISTICS PROGRAM (NPS)
- THE UNIFORM CRIME REPORT (UCR)
- THE FBI'S PRIMARY NATIONAL DATA COLLECTION TOOL
- CRIME REPORTING CHANGE FIELD
- STATE CRIME AND POPULATION STATISTICS PUBLISHED BY THE FBI UNIFORM CRIME REPORTING (UCR) PROGRAM

SAFEGRAPH

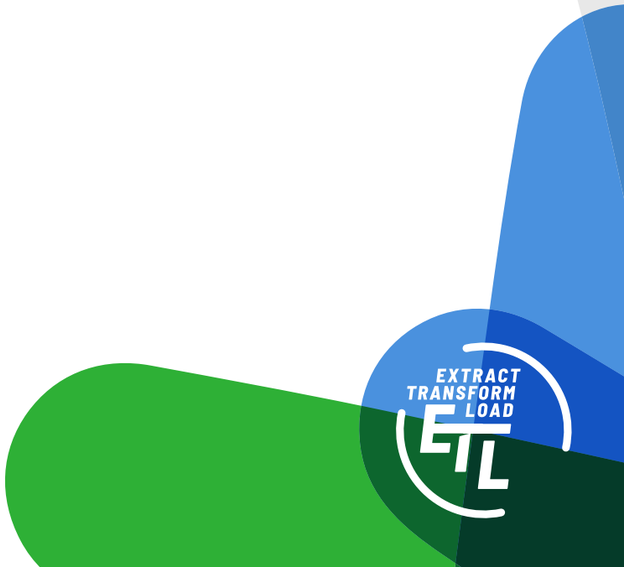
EXTRACT TRANSFORM LOAD

team five

TRANSFORMATION TYPE

THE TYPE OF TRANSFORMATION NEEDED FOR THIS DATA (CLEANING, JOINING, FILTERING, AGGREGATING, ETC)

- Cleaning
- Deleting the null values
- Changing strings that originally were attempting to represent dictionaries and lists into actual dictionaries and lists
- Part of the data munging process was to structure the dictionaries and lists into separated rows assigned with a unique identifier
- Specific to the 'CBG - Dataframe' (consumer shopping information) had over 220,000 rows of data entered and as a team we tactically isolated the first 10,000 rows for this class assignment due to the highest and best use of concentrated time
- The way that the coded structure allows for completing the whole dataset with a little modification to eventually combine all of the rows could be done with better/faster processing capacity, example (Hadoop, Pickle, etc.)
- Specific to the 'US Crime DF' our team had to drop nulls, keep only 2016 values,
- To augment the 'US Crime DF', we also had to locate and merge the US FIPS State Codes with the original DF, and then rename column to match FIPS DF (for link to CBG DF)
- Matched the formats of both datasets; state codes into strings



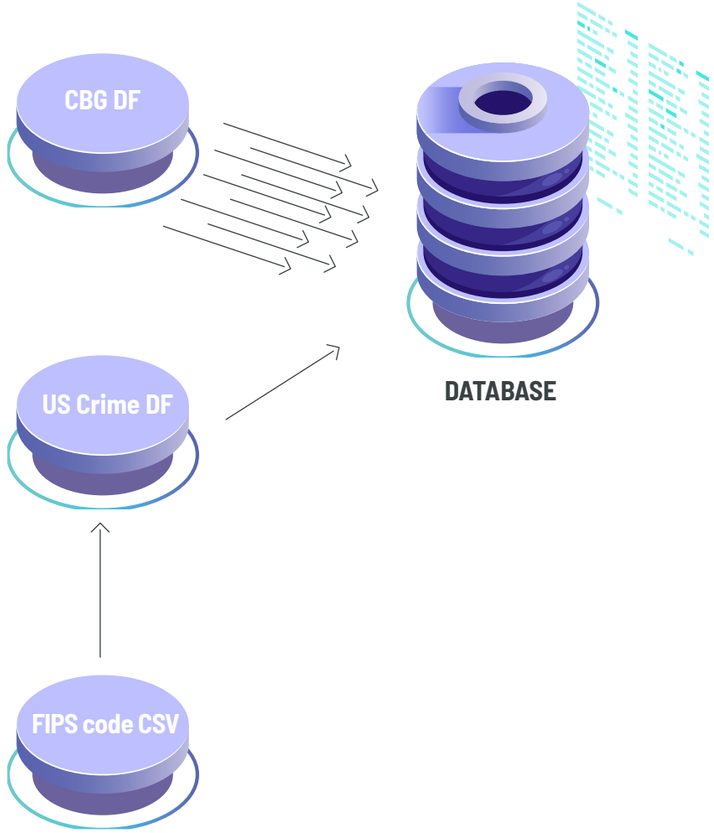
team five

FINAL DATABASE TYPE

- The type of final production database to load the data into was relational
- We utilized SQL Alchemy to accomplish this

TABLES/COLLECTIONS

- The dataframes from each of the two datasets were merged utilizing Pandas
- An end user could utilize SQL Alchemy to further merge or to make connections to all of the available tables within the database



EXTRACT
TRANSFORM
LOAD

team five

FINAL REPORT

EXTRACT TRANSFORM LOAD

team five

EXTRACT

WE UTILIZED THREE DIFFERENT CSV FILES

1. FIPS code CSV

https://catalog.data.gov/dataset/fips-state-codes

2. Crime CSV file

https://www.kaggle.com/christophercorrea/prisoners-and-crime-in-united-states

3. Consumer Block Group (CBG) CSV file

https://www.kaggle.com/safegraph/visit-patterns-by-census-block-group

```
df2 = pd.concat([df['census_block_group'], df['State Code'], df['visitor_home_cbgs'], df['visitor_work_cbgs'], df['related_same_day_brand'], df['related_same_month_brand'], df['top_brands'], df['popu
df2.head()
```

	census_block_group	State Code	visitor_home_cbgs	visitor_work_cbgs	related_same_day_brand	related_same_month_brand	top_brands	
0	1.005951e+10	01	{"010059501003":127,"010059509001":111,"010059...	{"010059501003":109,"010810407002":62,"0108104...	["Chick-fil-A","mcdonalds","Marathon Petroleum...	["walmart","mcdonalds","Dollar General","Chick...	["CrossFit","Health Mart","Coldwell Banker"]	[2617,24
1	1.009051e+10	01	{"010730113021":210,"010090506022":205,"010090...	{"010890111001":271,"010730045001":269,"010439...	["Shell Oil","mcdonalds","Chick-fil-A","Chevron"]	["walmart","mcdonalds","Shell Oil","Chick-fil-...	[]	[6556,633
2	1.047957e+10	01	{"010479567011":67,"010479567021":60}	{"010479567021":52}	["Dollar General"]	["walmart","Dollar General","mcdonalds","Chevr...	["Dollar General"]	[807,7
3	1.069040e+10	01	{"010690402013":370,"010690402011":322,"010690...	{"010690402024":313,"010690415004":203,"010450...	["Chick-fil-A","Sam's Club","Dollar General","...	["walmart","Dollar General","mcdonalds","Marat...	["Chick-fil-A","Sam's Club","Olive Garden","mc...	[2121,1
4	1.073011e+10	01	{"010090507001":183,"010730113021":167,"010730...	{"010730045001":140,"010730027001":123,"010730...	["Chevron","Daylight Donuts","walmart"]	["walmart","Chevron","Dollar General","Shell O...	["Chevron","CrossFit"]	[3804,37

= These entries contain dictionaries to be split into newer dataframes arranged by their unique CBG number

TRANSFORM

DATA CLEANING OR TRANSFORMATION WAS REQUIRED

To clean and modify the data so that all three files are compatible

```
w2 = []
x2 = []
y2 = []
z2 = []

for i in range(0,10000):
    mydict = ast.literal_eval(df2.iloc[i,3])
    b = len(mydict)
    w2 =w2+[df2.iloc[i,0]]*b
    x2 =x2+[df2.iloc[i,1]]*b
    y2 =y2+list(mydict.keys())
    z2 =z2+list(mydict.values())
    if (i%1000 == 0):
        ts = time.time()
        st = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')
        print (str(i) + " " + st)
```

```
0 2019-05-18 11:24:15
1000 2019-05-18 11:24:15
2000 2019-05-18 11:24:15
3000 2019-05-18 11:24:15
4000 2019-05-18 11:24:16
5000 2019-05-18 11:24:16
6000 2019-05-18 11:24:17
7000 2019-05-18 11:24:18
8000 2019-05-18 11:24:19
9000 2019-05-18 11:24:20
```

```
df4 = pd.DataFrame({'Census Block Group':w2,'State Code':x2,'Visitor Work CBGS Key':y2,'Visitor Work CBGS Value':z2})
df4.head()
```

	Census Block Group	State Code	Visitor Work CBGS Key	Visitor Work CBGS Value
0	1.005951e+10	01	010059501003	109
1	1.005951e+10	01	010810407002	62
2	1.005951e+10	01	010810420061	55
3	1.005951e+10	01	010690402024	54
4	1.009051e+10	01	010890111001	271

Pandas showing how the entries were split into newer dataframes arranged by their unique CBG number

LOAD

FINAL DATABASE, TABLES/COLLECTIONS, AND WHY THIS WAS CHOSEN

We chose a relational database format because all of the data was structured

```
from sqlalchemy import create_engine
database_path = "project2.sqlite"
engine1 = create_engine(f"sqlite:///{database_path}")
```

```
engine1.table_names()
```

```
['US Crime', 'US_Crime']
```

```
merge_table.to_sql(name='US_Crime', con=engine1, if_exists='replace', index=False)
```

```
pd.read_sql_query('select * from US_Crime', con=engine1).head()
```

	State (FIPS)	jurisdiction	includes_jails	year	prisoner_count	crime_reporting_change	crimes_estimated	state_population	violent_crime_total	murder_manslaughter
0	9.0	CONNECTICUT	1.0	2016.0	15040.0	0.0	0.0	3587685.0	8169.0	79.0
1	23.0	MAINE	0.0	2016.0	2356.0	0.0	0.0	1330232.0	1649.0	20.0
2	25.0	MASSACHUSETTS	0.0	2016.0	9038.0	0.0	0.0	6823721.0	25975.0	135.0
3	33.0	NEW HAMPSHIRE	0.0	2016.0	2599.0	0.0	0.0	1335015.0	2668.0	19.0
4	44.0	RHODE ISLAND	1.0	2016.0	2887.0	0.0	0.0	1057566.0	2529.0	29.0

EXTRACT TRANSFORM LOAD

team five

CONSIDERATION

"For several decades, tough laws and long sentences have created the illusion that public safety is best served when we treat all offenders the same way: arrest, convict, incarcerate..."

- Hon. Kamala D. Harris,
Attorney General of California

EXTRACT TRANSFORM LOAD

team five

THANK YOU

EXTRACT
TRANSFORM
LOAD