311 SERVICE REQUESTS - Data Curation Project

Susmitha Marripalapu (A20489531)
Pranava Brunda Manuguri (A20472679)
Yogith Reddy Venkanagari (A20493317)

Group-7

1 INTRODUCTION

Data curation helps us to access and look for useful information for analysis by creating, organizing and maintaining datasets. Cleaned data can be used to support business decision-making, academic needs, scientific research and other purposes.

2 UNDERSTANDING DATA

311 Service Requests received by the City of Chicago. This dataset includes requests created after the launch of the new 311 system on 12/18/2018 and some records from the previous system, indicated in the LEGACY_RECORD column. The dataset contains 37 columns in total with 5.95 million records. For this project, we have taken service requests between 04/01/2021 till 04/01/2022. Total requests between this time period are 865,209. The main motivation for selecting this dataset was to explore geo data and APIs available for curation.

AttributeName	Datatype
SR_NUMBER	string
SR_TYPE	string
SR_SHORT_CODE	string
OWNER_DEPARTMENT	string
STATUS	string
CREATED_DATE	Date
LAST_MODIFIED_DATE	Date
CLOSED_DATE	Date
STREET_ADDRESS	string
CITY	string
STATE	string
ZIP_CODE	string
STREET_NUMBER	Number
STREET_DIRECTION	string
STREET_NAME	string
STREET_TYPE	string
DUPLICATE	string
LEGACY_RECORD	string
LEGACY_SR_NUMBER	string
PARENT_SR_NUMBER	string
COMMUNITY_AREA	Number
WARD	Number
ELECTRICAL_DISTRICT	Number
ELECTRICITY_GRID	Number
POLICE_SECTOR	Number
POLICE_BEAT	Number
POLICE_DISTRICT	Number
PRECIENT	Number
SANITATION_DIVISION_DAYS	Number
CREATED_HOUR	Number
CREATED_DAY_OF_WEEK	Number
CREATED_MONTH	Number
X_COORDINATE	Number
Y_COORDINATE	Number
LATITUDE	Number
LONGITUDE	Number
LOCATION	Number

3 IDENTIFYING ISSUES

To identify the issues we have used several methods.

- The visualization tool in City of Chicago portal was used to identify the inconsistencies in data and redundant data.
- Basic python code is used to check for null values in the data.
- To check the correctness of the address and geo related data, geopy APIs in python are used.
  For example:

Address in data	Real address
2711 E 31 ST	2711 E 31st ST

4 CHALLENGES

Major steps to clean data are to remove duplicate or irrelevant observations, fix structural errors, filter unwanted outliers, handle missing data and validate the data. Below are the challenges to clean data for this project:

- To eliminate duplicate records, eliminate data redundancy by deleting unwanted columns containing similar information and filling the gaps for missing values.
- Obtaining and validating the valid location data and street addresses.
- Identifying auxiliary datasets to fill gaps in police, electricity, community area and ward data.
- Decoding MULTIPOLYGON Location border data.

5 PROBLEMS

5.1 NULL Values:

Attribute	Null values
SR_NUMBER	0
SR_TYPE	0
SR_SHORT_CODE	0
OWNER_DEPARTMENT	0
STATUS	0
CREATED_DATE	0
LAST_MODIFIED_DATE	0
CLOSED_DATE	0
STREET_ADDRESS	1,574
CITY	757,537
STATE	757,537
ZIP_CODE	3,688
DUPLICATE	0
LEGACY_RECORD	0
LEGACY_SR_NUMBER	0
PARENT_SR_NUMBER	0
COMMUNITY_AREA	2,378
WARD	2,271
POLICE_SECTOR	2,288
POLICE_BEAT	2,288
POLICE_DISTRICT	2,288
PRECIENT	2,780
X_COORDINATE	1,500
Y_COORDINATE	1,500

Attribute	Null values
LATITUDE	1,571
LONGITUDE	1,571

5.2 Redundant Data:

Some columns in the dataset can be omitted as the meaning conveyed is same as other columns. Below are the redundancies observed in service requests data.

- STREET ADDRESS is split into STREET NUMBER, STREET DIRECTION, STREET NAME, STREET TYPE. If we have street address, we can omit the four columns with out loss of information.
- CREATED DATE is split into CREATED HOUR, CREATED DAY OF WEEK, CREATED MONTH. If we have CREATED DATE, we can omit the three columns with out loss of information.
- LATITUDE and LONGITUDE are combined to represent LOCATION attribute. We can omit LOCATION attribute with out loss of information.

5.3 Inconsistent Data:

Some columns like CITY and STATE have inconsistent data.

- CITY is represented as CHICAGO, Chicago or chicago.
- STATE is represented as IL or ILLINOIS.

5.4 Invalid Data:

Some columns like STREET_ADDRESS have invalid address data. Some examples include addresses like 0, 1, 0 Don't know etc.

6 DATA CLEANING

Remove Redundant columns: Firstly, to reduce the redundancy, we drop the columns STREET NUMBER, STREET DIRECTION, STREET NAME, STREET TYPE, CREATED HOUR, CREATED DAY OF WEEK, CREATED MONTH and LOCATION.

Make data consistant: To make CITY and STATE consistent, we assign the CITY values to "CHICAGO" and STATE values to "ILLINOIS".

Check for invalid street address: To get the missing STREET_ADDRESS and check if existing address is correct, we use geoPy libraries. The code looks as below:

```
from geopy.geocoders import Nominatim
locator = Nominatim()
location = locator.geocode(
        sr['STREET_ADDRESS'],
        timeout=10,
        exactly_one=False)
```

If the location gets 'None' as return value, that implies that street address is invalid. Fig 1 explains the flow of logic to curate STREET ADDRESS.

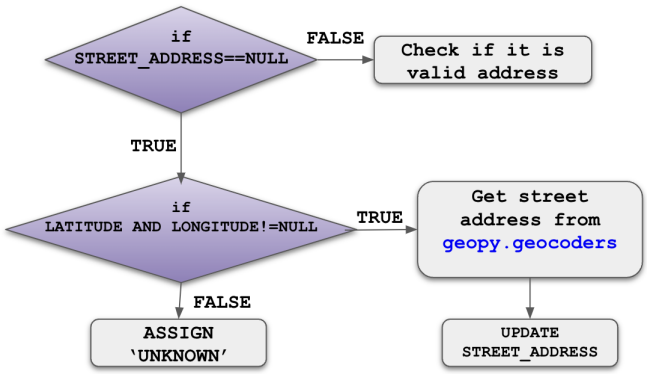

Fig 1 Street Address Curation

Get missing location data: geopy is a Python client for several popular geocoding web services. geopy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources. Latitude and longitude are retrieved using this package.

To get x coordinate and y coordinate, we are using UTM package in python. The Universal Transverse Mercator (UTM) is a map projection system for assigning coordinates to locations on the surface of the Earth.

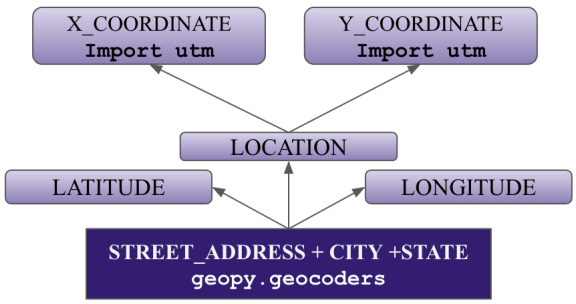

Fig 2 Location data Curation

Below is a sample code used o get latitude and longitude from address:

```
from geopy.geocoders import Nominatim
locator = Nominatim()
geocode = RateLimiter(locator.geocode
        ,min_delay_seconds=1)
df['GEOADR'] = df['ADDRESS'].
            apply(geocode)

# create longitude, latitude
df['LOCATION'] = df['GEOADR'].apply(
                lambda loc: tuple(
                loc.point)[:2]
                if loc else None)

# split point column into latitude,
# longitude
df[['LATITUDE','LONGITUDE']] =
                pd.DataFrame(
                df['LOCATION']
                .tolist(),
                index=df.index)
```

Get missing data from auxiliary datasets: Derive missing community areas using Community_area dataset from City of Chicago Data portal, wards using WARD dataset from City of Chicago Data portal and police data using Police Beats dataset from City of Chicago Data portal.

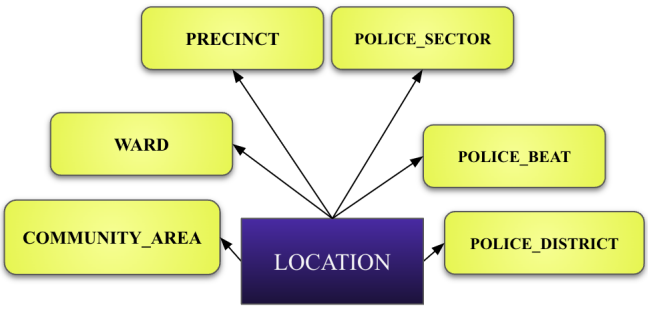

Fig 3 Data Curation

Using boundary data, MULTIPOLYGON GEO DATA we have derived missing values for COMMUNITY_AREA, WARD, POLICE_SECTOR, POLICE_DISTRICT, POLICE_BEATS and PRICINCTS. geopandas, geoseries packages are used to get if a location (point) is within the borders. Below is sample code for finding police data.

```
from shapely.geometry import shape
import math
import geopandas

for ind1,sr in df.iterrows():
  #Get missing police information
  #from policeBeat data
  for ind2,rec in policeBeats.iterrows():
    P = shapely.wkt.loads(rec['the_geom'])
    pt = Point(sr['LONGITUDE'],
        sr['LATITUDE'])
```

```
s = geopandas.GeoSeries([P])
if(math.isnan(sr['POLICE_BEAT'])
    and pt.within(P)):
  df.loc[ind1,['POLICE_DISTRICT']]
    = rec['DISTRICT']
  df.loc[ind1,['POLICE_SECTOR']]
    = rec['SECTOR']
  df.loc[ind1,['POLICE_BEAT']]
    = rec['BEAT_NUM']
```

7 VALIDATIONS

We have implemented the logic in python to collect all the invalid records in a validation report. This logic includes checks for null values in all the columns, Checking if CLOSED DATE is present even if the request status is still open. Check if the columns like COMMUNITY AREA, WARD, POLICE related attributes have right numerical values between the range specified.

8 EVALUATION

- Minimized number of records in validation report.
- Current invalid records for the 2021 service requests is less than 5% after curation.
- Made the data consistent.
- Eliminated data redundancy.

FUTURE WORKS: Time taken for address check using geopy libraries for 1 million records is larger than expected which can be improved.

References

[1] Geopy libraries documentation LINK

[2] Geopandas multipolygon data documentation LINK