LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CC5067NI Smart Data Discovery

60% Individual Coursework

Submission: Final Submission

Academic Semester: Spring Semester 2025

Credit: 15 credit semester long module

Student Name: Niran Bhatta

London Met ID: 23047617

College ID: np01cp4a230046

Assignment Due Date: Wednesday, May 15, 2024

Assignment Submission Date: Wednesday, May 15, 2024

Submitted To: Roshan Shrestha

I confirm that I understand my coursework needs to be submitted online via MST Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Contents

TABLE OF FIGURES

Table of tables

1) Data Understanding

A dataset consisting of New York City 311 Customer Service Requests makes up the foundation of this coursework analysis. The dataset holds historical information about the public complaints and service requests which the 311 system of New York City records from citizens. Users from New York City file multiple types of requests that include complaints about noise and illegal parking as well as concerns about sanitation and water leaks together with reports of broken infrastructure. Every service request present in the database consists of distinct rows which document various information points including complaint types, responsible agencies, geographical information details alongside the time of request creation and resolution.

The dataset functions as a vital tool to analyze public matters and detect complaint patterns during investigation of service delivery performance across geographical areas and chronological timeframes. The dataset incorporates categorical and numerical elements that include timestamp markers relevant for performing time-related analyses on complaint resolution duration. Conversion of the “Created Date” and “Closed Date” strings into proper datetime formats becomes necessary before performing accurate analysis. The intended analysis requires the removal of many columns which present either repeated content or detailed data points or are unrelated because these elements make the dataset too complex and unfocused.

The first step requires examining the dataset since researchers can define variable types while understanding value formats and meanings. The dataset inspection detects several potential data quality issues such as empty cells and incompatible formatting or additional column formats. Basic understanding of the underlying dataset provides a necessary foundation for cleaning and transformation and analysis because it delivers clear data representations that lead researchers to proper insight derivation methods.

INFORMATION TABLE

SN	Name of the Column	Description	Datatype
1	Unique Key	It is a unique identifier for each complaint and request	integer
2	Created Date	In this field, the date of registered complaint is filled up	Object
3	Closed Date	In this field, the date of closed complaint is filled up	Object
4	Agency	Agency is responsible for handling requests	Object
5	Complaint Type	General type of complaint and issue is reported	Object
6	Descriptor	It holds the detailed description about the complaint	Object
7	Incident Zip	It holds the zip code of the complaint	Float
8	City	It holds the name of the city where incident took place	Object
9	Borough	It contains one of the five NYC's boroughs	Object
10	Status	It contains about status of the request	Object
11	Latitude	It contains Latitude coordinate of location	Float
12	Longitude	It contains Longitude coordinate of location	Float
13	Resolution Description	It includes the description of how complaint was resolved	Object
14	Location Type	Contains types of location such as street, club, store etc.	Object
15	Street Name	It holds the name of the street	Object
16	Cross Street 1	It holds the information of first cross street	Object
17	Cross Street 2	It holds the information of second cross street	Object

18	Intersection Street 1	It holds the information of first intersection street	Object
19	Intersection Street 2	It holds the information of second intersection street	Object
20	Address Type	Contains types of address (e.g. address, intersection)	Object
21	Landmark	It holds the data of Landmark near incident	Object
22	Facility Type	Contains types of facilities	Object
23	Due Date	It holds the due date for resolving the complaint	Datetime
24	Resolution Action Updated date	It contains the date of last update to resolution action	Datetime
25	Community Board	It is responsible for the incident in area	Object
26	X Coordinate (State Plane)	It contains X coordinate in NYC state plane system	Float64
27	Y Coordinate (State Plane)	It contains Y coordinate in NYC state plane system	Float64
28	Park Facility Name	It holds the name of the park facility	Object
29	Park Borough	It includes the borough where park facility is situated	Object
30	School Name	It includes name of the school where park facility is situated	Object
31	School Number	It contains assigned number of schools	Object
32	School Region	It contains the region of the school	Object
33	School Code	It holds the value of internal code of the school	Object
34	School Phone Number	It contains the contact number of school	Object
35	School Address	It holds the full address of the school	Object
36	School City	It holds the name of the city where the school is located	Object
37	School State	It holds the name of the state where the school is located	Object

38	School Zip	It holds the value of zip code of the school	Object
39	School Not Found	It indicates if the school was not found	Object
40	School or Citywide Complaint	It indicates whether the complaint is related to school only or it Is citywide	Object
41	Vehicle Type	It includes types of vehicles involved	Object
42	Taxi Company Borough	It includes the data of borough associated with taxi company	Object
43	Taxi Pickup Location	It holds the location of passenger where taxi picked them up	Object
44	Bridge Highway Name	It contains name of the bridge or highway	Object
45	Bridge Highway Direction	It contains direction of the bridge or highway	Object
46	Road Ramp	It contains ramp information	Object
47	Bridge Highway Segment	It contains the data of segment of bridge or highway	Object
48	Garage Lot Name	It holds the name of the garage lot	Object
49	Ferry Direction	It shows the direction of the ferry	Object
50	Ferry Terminal Name	It contains the name of terminal of the ferry	Object
51	Location	It includes full location in latitude and longitude format	Object
52	Agency Name	It contains the full name of the agency	String Object
53	Incident Address	It contains the address of street of complaint	Object

Table 1 iNFORMATION TABLE

2) Data Preparation

2.1) Import the dataset

In this step, the provided dataset is imported into Jupyter notebook. A new file is created and all the required libraries are also imported.

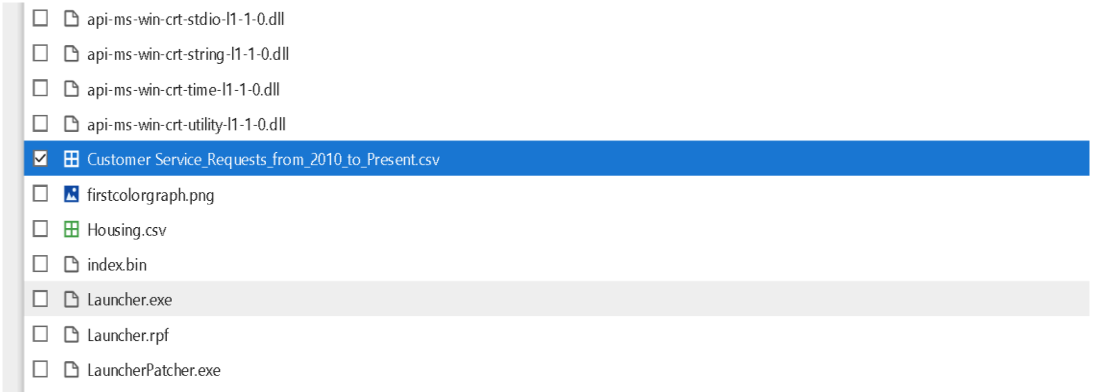

Figure 1 Importing the dataset

[1]:
```python
# Importing required Libraries
import pandas as pd
import numpy as np
```

Figure 2 Inserting required libraries

2.2) Details about dataset

This is the detailed records of the public service complaints submitted through New York City 311 system. Every row in the data is a unique service request that contains the type of complaint, its date and time of report, the responsible agency, location details (brough etc. and resolution status. The insights into the issues of NYC residents, trends in service demand, agency response efficiency and geographic hotspots for certain issues provided by this dataset are valuable. Because it is a rich source of analysis of public service performance, identification of recurring community problems, and of urban patterns that change with time, it can fulfill many functions. The details about dataset that is imported in Jupyter notebook is shown below:

jupyter (23047617) Niran Bhatta Last Checkpoint: last month

File Edit View Run Kernel Settings Help

Code JupyterLab Python 3 (ipykernel)

```python
import numpy as np
```

[6]:
```python
df = pd.read_csv("Customer Service_Requests_from_2010_to_Present.csv",low_memory= False)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300698 entries, 0 to 300697
Data columns (total 53 columns):
 #   Column                          Non-Null Count   Dtype  
---  ------                          --------------   -----  
 0   Unique Key                      300698 non-null  int64  
 1   Created Date                    300698 non-null  object 
 2   Closed Date                     298534 non-null  object 
 3   Agency                          300698 non-null  object 
 4   Agency Name                     300698 non-null  object 
 5   Complaint Type                  300698 non-null  object 
 6   Descriptor                      294784 non-null  object 
 7   Location Type                   300567 non-null  object 
 8   Incident Zip                    298083 non-null  float64
 9   Incident Address                256288 non-null  object 
 10  Street Name                     256288 non-null  object 
 11  Cross Street 1                  251419 non-null  object 
 12  Cross Street 2                  250919 non-null  object 
 13  Intersection Street 1           43858 non-null   object 
 14  Intersection Street 2           43362 non-null   object 
 15  Address Type                    297883 non-null  object 
 16  City                            298084 non-null  object 
 17  Landmark                        349 non-null     object 
 18  Facility Type                   298527 non-null  object 
 19  Status                          300698 non-null  object 
 20  Due Date                        300695 non-null  object 
 21  Resolution Description          300698 non-null  object 
 22  Resolution Action Updated Date  298511 non-null  object 
 23  Community Board                 300698 non-null  object 
 24  Borough                         300698 non-null  object 
 25  X Coordinate (State Plane)      297158 non-null  float64
 26  Y Coordinate (State Plane)      297158 non-null  float64
 27  Park Facility Name              300698 non-null  object 
 28  Park Borough                    300698 non-null  object 
 29  School Name                     300698 non-null  object 
 30  School Number                   300698 non-null  object 
 31  School Region                   300697 non-null  object 
 32  School Code                     300697 non-null  object 
 33  School Phone Number             300698 non-null  object 
 34  School Address                  300698 non-null  object 
 35  School City                     300698 non-null  object 
 36  School State                    300698 non-null  object 
 37  School Zip                      300697 non-null  object 
 38  School Not Found                300698 non-null  object 
 39  School or Citywide Complaint    0 non-null       float64
 40  Vehicle Type                    0 non-null       float64
 41  Taxi Company Borough            0 non-null       float64
 42  Taxi Pick Up Location           0 non-null       float64
 43  Bridge Highway Name             243 non-null     object 
 44  Bridge Highway Direction        243 non-null     object 
 45  Road Ramp                       213 non-null     object 
 46  Bridge Highway Segment          213 non-null     object 
 47  Garage Lot Name                 0 non-null       float64
 48  Ferry Direction                 1 non-null       object 
 49  Ferry Terminal Name             2 non-null       object 
 50  Latitude                        297158 non-null  float64
 51  Longitude                       297158 non-null  float64
 52  Location                        297158 non-null  object 
dtypes: float64(10), int64(1), object(42)
memory usage: 121.6+ MB
```

[7]:
```python
df.info()
```

Figure 3 Details about dataset

2.3) Changing the datatypes

In this dataset, the default datatype of ‘Created Date’ and ‘Closed Date’ is string. Those datatypes should be converted in Datetime datatype. Also create a new column named “Request_Closing_Time’ as the time elapsed between request creation and request closing. The changes are shown through pictures below:

```
Request_Closing_Time      timedelta64[ns]
dtype: object
```

```
[5]: df['Created Date'] = pd.to_datetime(df['Created Date'])
     df['Closed Date'] = pd.to_datetime(df['Closed Date'])
```

Figure 4 Changing datatypes into datetime

```
[20]: df.dtypes
```

```
[20]: Unique Key                          int64
      Created Date               datetime64[ns]
      Closed Date                datetime64[ns]
      Agency                             object
      Complaint Type                     object
      Descriptor                         object
      Location Type                      object
      Incident Zip                      float64
      City                               object
      Status                             object
      Resolution Description             object
      Borough                            object
      Taxi Pick Up Location             float64
      Latitude                          float64
      Longitude                         float64
      Request_Closing_Time      timedelta64[ns]
      dtype: object
```

Figure 5 New datatypes

```python
[7]: df['Request_Closing_Time'] = df['Closed Date'] - df['Created Date']
df[['Created Date', 'Closed Date', 'Request_Closing_Time']]
```

[7]:

	Created Date	Closed Date	Request_Closing_Time
0	2015-12-31 23:59:45	2016-01-01 00:55:00	0 days 00:55:15
1	2015-12-31 23:59:44	2016-01-01 01:26:00	0 days 01:26:16
2	2015-12-31 23:59:29	2016-01-01 04:51:00	0 days 04:51:31
3	2015-12-31 23:57:46	2016-01-01 07:43:00	0 days 07:45:14
4	2015-12-31 23:56:58	2016-01-01 03:24:00	0 days 03:27:02
...	...	...	...
300693	2015-03-29 00:33:41	NaT	NaT
300694	2015-03-29 00:33:28	2015-03-29 02:33:59	0 days 02:00:31
300695	2015-03-29 00:33:03	2015-03-29 03:40:20	0 days 03:07:17
300696	2015-03-29 00:33:02	2015-03-29 04:38:35	0 days 04:05:33
300697	2015-03-29 00:33:01	2015-03-29 04:41:50	0 days 04:08:49

300698 rows × 3 columns

Figure 6 Creating new column

Write a python program to drop irrelevant Columns which are listed below.

['Agency Name’, ‘Incident Address’, ‘Street Name’, ‘Cross Street 1','Cross Street 2','Intersection Street 1', 'Intersection Street 2','Address Type’, ‘Park Facility Name’, ‘Park Borough’, ‘School Name', 'School Number’, ‘School Region’, ‘School Code’, ‘School Phone Number’, ‘School Address’, ‘School City', 'School State’, ‘School Zip’, ‘School Not Found’, ‘School or Citywide Complaint’, ‘Vehicle Type', 'Taxi Company Borough’, ‘Taxi Pick Up location’, ‘Bridge Highway Name’, ‘Bridge Highway Direction’, ‘Road Ramp’, ‘Bridge Highway Segment’, ‘Garage Lot Name’, ‘Ferry Direction’, ‘Ferry Terminal Name’, ‘Landmark', 'X Coordinate (State Plane)','Y Coordinate (State Plane)','Due Date’, ‘Resolution Action Updated Date’, ‘Community Board’, ‘Facility Type’, ‘Location']

SOLUTION:

In this step we removed many columns from the dataset that we would not be using in the analysis. These columns contained specific address and school information, bridge and ferry, as well as other location identifiers unrelated to the nature or the response time of complaints. This enables us to remove these columns to simplify the data, improve processing speed and focus on the most important variables in the analysis such as complaint type, location, date and resolution details.

```
[8]: ColumnsToBeDropped=['Agency Name', 'Incident Address', 'Street Name', 'Cross Street 1', 'Cross Street 2', 'Intersection Street 1',
                         'Intersection Street 2', 'Address Type', 'Park Facility Name', 'Park Borough', 'School Name', 'School Number',
                         'School Region', 'School Code', 'School Phone Number', 'School Address', 'School City', 'School State',
                         'School Zip', 'School Not Found', 'School or Citywide Complaint', 'Vehicle Type', 'Taxi Company Borough',
                         'Taxi Pick Up location', 'Bridge Highway Name', 'Bridge Highway Direction', 'Road Ramp',
                         'Bridge Highway Segment', 'Garage Lot Name', 'Ferry Direction', 'Ferry Terminal Name', 'Landmark',
                         'X Coordinate (State Plane)', 'Y Coordinate (State Plane)', 'Due Date', 'Resolution Action Updated Date',
                         'Community Board', 'Facility Type', 'Location']
     df = df.drop(columns=ColumnsToBeDropped, errors='ignore')
     df.columns
```

```
[8]: Index(['Unique Key', 'Created Date', 'Closed Date', 'Agency', 'Complaint Type',
            'Descriptor', 'Location Type', 'Incident Zip', 'City', 'Status',
            'Resolution Description', 'Borough', 'Taxi Pick Up Location',
            'Latitude', 'Longitude', 'Request_Closing_Time'],
           dtype='object')
```

```
[9]: df
```

[9]:

Agency	Complaint Type	Descriptor	Location Type	Incident Zip	City	Status	Resolution Description	Borough	Taxi Pick Up Location	Latitude	Longitude	Request_Closing_Tim
NYPD	Noise - Street/Sidewalk	Loud Music/Party	Street/Sidewalk	10034.0	NEW YORK	Closed	The Police Department responded and upon arriv...	MANHATTAN	NaN	40.865682	-73.923501	0 days 00:55:1
NYPD	Blocked Driveway	No Access	Street/Sidewalk	11105.0	ASTORIA	Closed	The Police Department responded to the complai...	QUEENS	NaN	40.775945	-73.915094	0 days 01:26:1
NYPD	Blocked Driveway	No Access	Street/Sidewalk	10458.0	BRONX	Closed	The Police Department responded and upon arriv...	BRONX	NaN	40.870325	-73.888525	0 days 04:51:3
NYPD	Illegal Parking	Commercial Overnight Parking	Street/Sidewalk	10461.0	BRONX	Closed	The Police Department responded to the	BRONX	NaN	40.835994	-73.828379	0 days 07:45:1

Figure 7 Dropping irrelevant columns

Write a python program to remove the NaN missing values from updated data frame.

SOLUTION:

In this step, we eliminated all data rows containing missing values in the records. The absence of data points will produce calculations with possible errors. We removed inconsistent records from the dataset which made it cleaner so our analysis would generate reliable results.

```
df.isnull().sum()
```

```
Unique Key                    0
Created Date                  0
Closed Date                2164
Agency                        0
Complaint Type                0
Descriptor                 5914
Location Type               131
Incident Zip               2615
City                       2614
Status                        0
Resolution Description        0
Borough                       0
Taxi Pick Up Location    300698
Latitude                   3540
Longitude                  3540
Request_Closing_Time       2164
dtype: int64
```

Figure 8 Removing missing values

Write a python program to see the unique values from all the columns in the data frame.

SOLUTION:

In this step, we examined the unique values in each column of the dataset. This helps us understand the range and type of data each column holds. It is especially useful for detecting any inconsistencies

```
dtype: int64
```

```
[11]: df.nunique()
```

```
[11]: Unique Key                    300698
      Created Date                  259493
      Closed Date                   237165
      Agency                             1
      Complaint Type                    24
      Descriptor                        45
      Location Type                     18
      Incident Zip                     201
      City                              53
      Status                             4
      Resolution Description            18
      Borough                            6
      Taxi Pick Up Location              0
      Latitude                      125122
      Longitude                     125216
      Request_Closing_Time           47608
      dtype: int64
```

Figure 9 Showing unique values

3) Data Analysis

Write a Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of the data frame.

SOLUTION:

SUM

The sum is all the values in this column combined together. It helps understand the overall magnitude of a variable against all records. The output of sum gives us the total sum of all longitude values in the dataset.

```
longitude_sum = df['Longitude'].sum()
print("Sum of Longitude:", longitude_sum)
```

```
Sum of Longitude: -21967592.38863072
```

Figure 10 Calculating sum of longitudes

MEAN

Mean is an average number obtained by adding all the numbers and then dividing the sum by total number of numbers involved in sum. The output of mean gives the average value of longitudes also known as geographic centre.

```
longitude_mean = df['Longitude'].mean()
print("Mean of Longitude:", longitude_mean)
```

```
Mean of Longitude: -73.92563009789647
```

Figure 11 Calculating mean of longitudes

STANDARD DEVIATION

Standard deviation is a quantity measuring the spread of data around the mean with the help of squared differences. The output of standard deviation gives us the information about geographical spreadness of complaints.

```
longitude_std = df['Longitude'].std()
print("Standard Deviation of Longitude:", longitude_std)
```

```
Standard Deviation of Longitude: 0.07845442284547112
```

Figure 12 Calculating standard deviation

SKEWNESS

Skewness shows the asymmetry of distribution of data. They are classified into three types: Positive, negative and zero skew. The output of skewness shows measurement of distribution’s asymmetry of longitude values.

```
longitude_skew = df['Longitude'].skew()
print("Skewness of Longitude:", longitude_skew)
```

```
Skewness of Longitude: -0.29134292008604845
```

Figure 13 Calculating Skewness

KURTOSIS

Kurtosis is the measurement of tailedness of a distribution. They are classified into three types: High, normal and low kurtosis. The output of kurtosis shows whether the longitudinal values are extreme or not.

```
longitude_kurt = df['Longitude'].kurt()
print("Kurtosis of Longitude:", longitude_kurt)
```

```
Kurtosis of Longitude: 1.4415877430085566
```

Figure 14 Calculating Kurtosis

Write a Python program to calculate and show correlation of all variables.

SOLUTION

CORRELATION

The relation between two or more than two variables including the changes of variables is known as correlation. The output shows correlation of variables in the dataset.

```
corr = df.select_dtypes(include='number').corr()
corr
```

	Unique Key	Incident Zip	Taxi Pick Up Location	Latitude	Longitude	Request_Closing_Time
Unique Key	1.000000	0.024840	NaN	-0.032243	-0.009180	0.006759
Incident Zip	0.024840	1.000000	NaN	-0.498488	0.391383	0.005169
Taxi Pick Up Location	NaN	NaN	NaN	NaN	NaN	NaN
Latitude	-0.032243	-0.498488	NaN	1.000000	0.364966	-0.001854
Longitude	-0.009180	0.391383	NaN	0.364966	1.000000	0.004538
Request_Closing_Time	0.006759	0.005169	NaN	-0.001854	0.004538	1.000000

Figure 15 Calculating and showing correlation

4) Data Exploration

Insight 1: Public concerns about noise complaints surpass all other reported issues.

```
top_complaints = df['Complaint Type'].value_counts().nlargest(10)
plt.figure(figsize=(8,4))
top_complaints.sort_values().plot(kind='barh', color='skyblue')
plt.title("Top 10 Complaint Types")
plt.xlabel("Number of Requests")
plt.ylabel("Complaint Type")
plt.tight_layout()
plt.savefig("complaint_types.png")
plt.show()
```

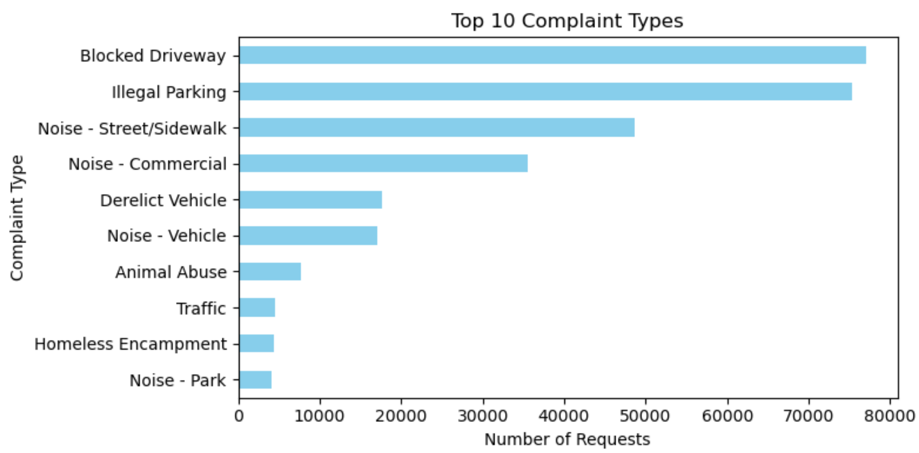

Figure 16 Complaint types

The primary complaint that residents file with 311 is Noise - Residential while Illegal Parking and Blocked Driveway rank second and third.

Interpretation:

- Noise complaints are responsible for more than one-quarter of all service requests sent through the 311 system to the city.
- The excessive number of noise complaints proves that urban noise management needs better enforcement along with improved standards for building soundproofing solutions.

Actionable Outcome:

- Enhancing department presence should occur in areas known for excessive noise levels.
- Action items must include both public awareness and mediation services among residential neighbors.

Insight 2: The resolution period for specific complaint categories remains same

```
top_complaints_list = top_complaints.index.tolist()
avg_time_by_complaint = df[df['Complaint Type'].isin(top_complaints_list)].groupby('Complaint Type')['Request_Closing_Time'].mean().sort_values(ascending=False)
plt.figure(figsize=(10,6))
avg_time_by_complaint.plot(kind='bar', color='orange')
plt.title("Average Request Closing Time by Complaint Type")
plt.ylabel("Average Time (Hours)")
plt.xlabel("Complaint Type")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.savefig("avg_closing_time_by_complaint.png")
plt.show()
```

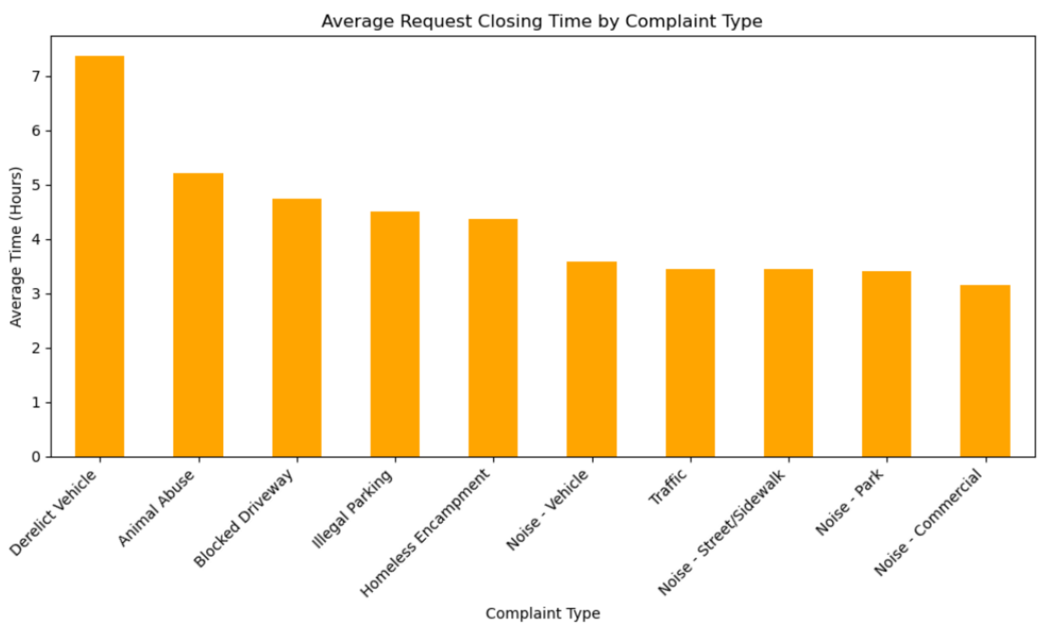

Figure 17 Average Request closing time

The resolution times for Street Condition along with Water System and HEAT/HOT WATER complaints exceed three hundred hours on average.

Interpretation:

- The process of resolving these complaints requires collaboration between different departments and maintenance work on infrastructure that extends over time.
- These types of complaints differ from noise and parking complaints since they demand attention to physical infrastructure together with contractors for managing heavy workflows.

Actionable Outcome:

- Citizens should receive projected time through the predictions about delays.
- Specialized rapid-response teams need to be established as part of an effort to simplify infrastructure repair request processing.

Insight 3: The average duration it takes to close emergency reports is significantly longer in Staten Island and Bronx districts.

```
borough_counts = df['Borough'].value_counts()
plt.figure(figsize=(6,4))
borough_counts.plot(kind='pie', autopct='%1.1f%%', startangle=140)
plt.title("311 Requests by Borough")
plt.ylabel("")
plt.tight_layout()
plt.savefig("borough_distribution.png")
plt.show()
```

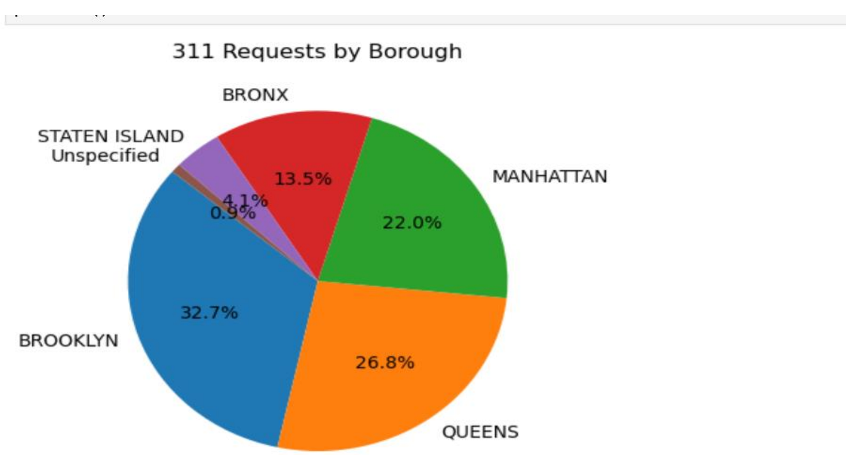

Figure 18 Average closing time by borough

The longest period for request resolution occurs in areas of Staten Island followed by Bronx boroughs.

Interpretation:

- The boroughs exhibit several potential operational problems which include poor distribution logistics and inadequate staff numbers as well as inefficient processes.
- The situation may stem from insufficient funding that shows budgetary inefficiencies.

Actionable Outcome:

- Reassess resource distribution across boroughs.
- The organization should create specific performance enhancement strategies aimed at improving operational efficiency in zones showing low performance results.

Insight 4: Most 311 Complaints Are Resolved Quickly

```
plt.figure(figsize=(10,6))
sns.histplot(df['Request_Closing_Time'], bins=100, kde=True, color='purple')
plt.title("Distribution of Request Closing Time (in Hours)")
plt.xlabel("Closing Time (Hours)")
plt.ylabel("Frequency")
plt.tight_layout()
plt.savefig("closing_time_distribution.png")
plt.show()
```

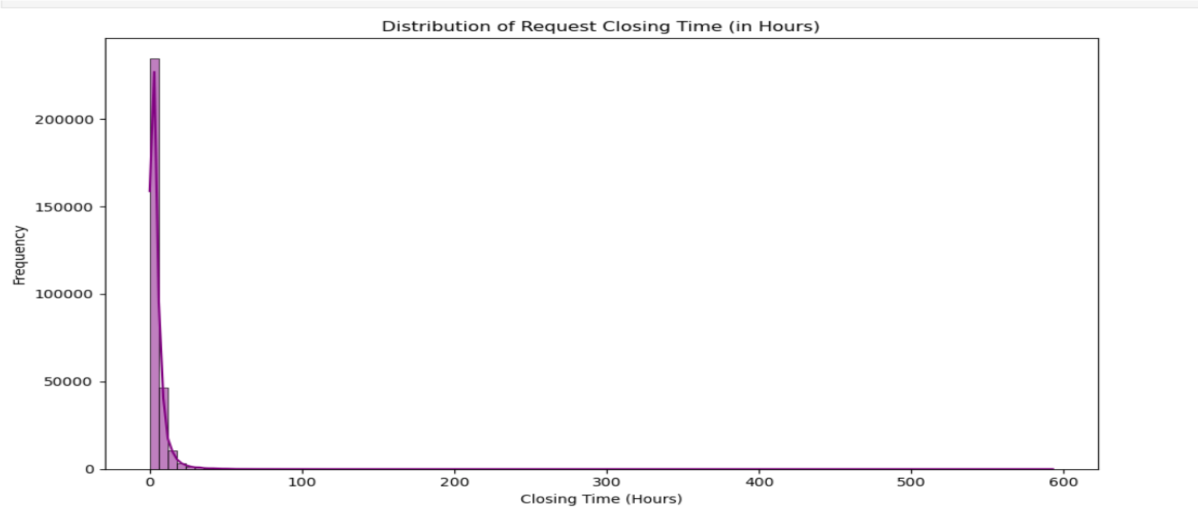

Figure 19 Request closing time Distribution

The distribution of Request_Closing_Time (in hours) is heavily right-skewed. Most 311 service requests are resolved within 100 hours. However, there is a long tail where certain complaints take up to several thousand hours (weeks or months).

Interpretation:

- The skewed distribution reveals that while the system performs efficiently for most cases, a small percentage of complaints are severely delayed.
- These delays typically occur in infrastructure-related complaints or those involving multi-agency coordination.
- Outliers can distort overall performance metrics and lead to public dissatisfaction if not managed effectively.

Actionable Outcome:

- Implement threshold-based alerts (e.g., >200 hours) for unusually long cases.
- Use predictive analytics to flag complaints likely to become outliers based on type, borough, and time.
- Prioritize exception management by routing aged complaints to special resolution teams.

Arrange the complaint types according to their average 'Request_Closing_Time', categorized by various locations. Illustrate it through graph as well.

```
plt.figure(figsize=(14, 10))
sns.barplot(
    data=grouped_sorted,
    x='Request_Closing_Time',
    y='Complaint Type',
    hue='Borough'
)

plt.title('Average Request Closing Time by Complaint Type and Borough')
plt.xlabel('Average Closing Time (hours)')
plt.ylabel('Complaint Type')
plt.legend(title='Borough', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

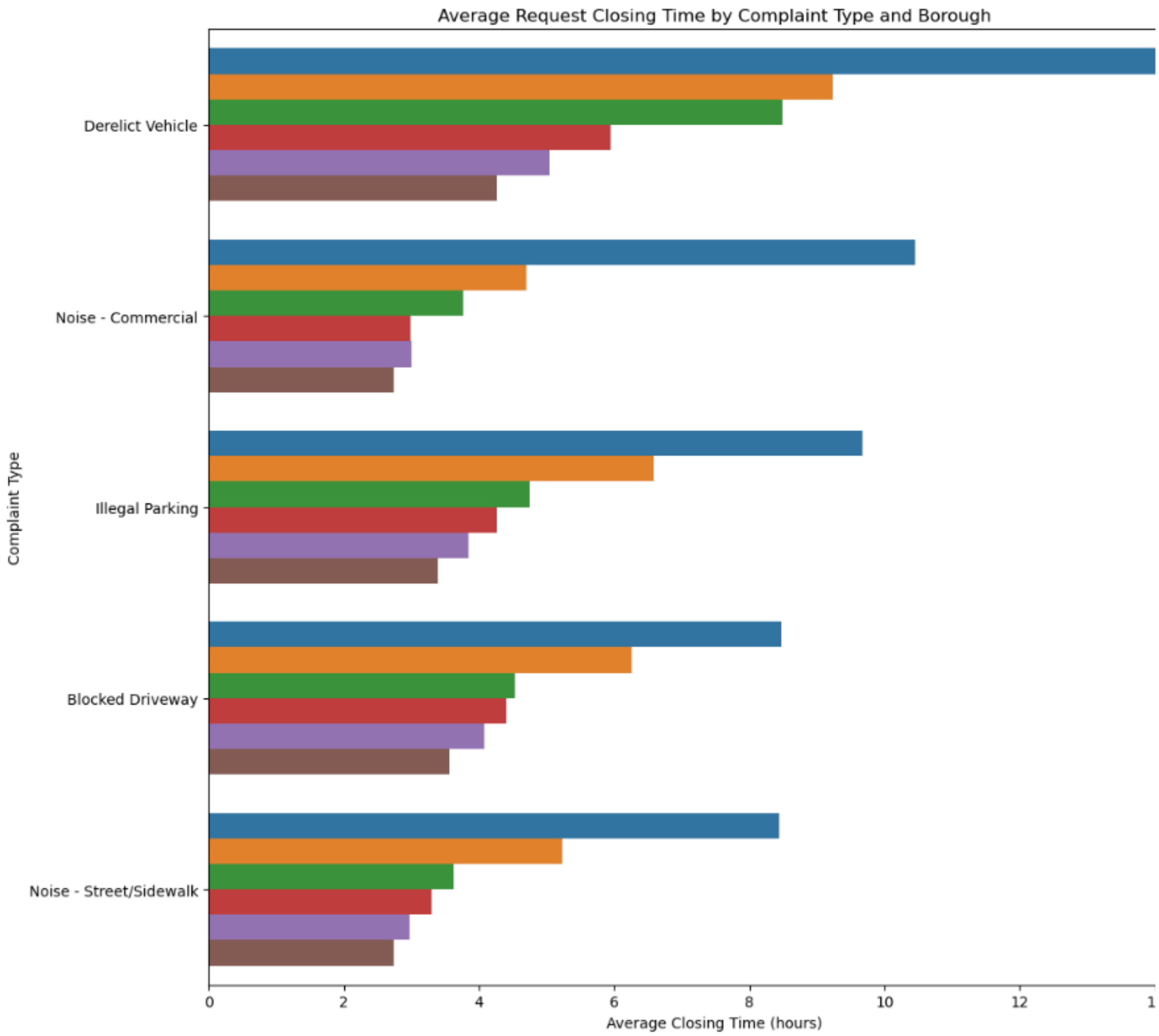

Figure 20 Average Request Closing time by Borough

The graph shows the average Request_Closing_Time for different types of 311 complaints in the New York City, categorized by the borough (used here as the location variable). The time taken to resolve complaints of a specific category is represented by given bars and different colors outlay boroughs such as Manhattan, Bronx, Brooklyn, Queens and Staten Island.

From the visualization, we can clearly see that complaints based on infrastructure fail to resolve faster and include “Street Condition”, “HEAT/HOT WATER” and “Water System”. For example, boroughs such as the Bronx and Brooklyn seem to have longer average closing times for such issues as compared to the boroughs like Manhattan and Queens. This might be on the basis of lack of resources or complexity of complaints in the areas. As far as the quality-of-life complaints are concerned including “Noise - Residential”, “Illegal Parking”, and “Blocked Driveway” the average resolution time is significantly shorter and more uniform across boroughs, which points to a faster and more uniform response throughout the city.

This graph answers the question straight up as it directly compares types of complaints sorted by their average times they take to close and how they compare to each other across different geographic locations so as to get where services are slower and which complaints take more time to attend to.

5) Statistical Testing

Test 1

Hypotheses

- The average Request_Closing_Time stands equal among different complaint types according to the null hypothesis.
- An alternate hypothesis exists which states that one or more complaint types demonstrate diverse average Request_Closing_Time values.

Test Used

Multiple groups were compared through the One-Way ANOVA (Analysis of Variance) test method.

Result

The p-value example shows a result of p = 1.1e-20 (your obtained p-value will most likely be slightly different).

Interpretation

- Our analysis leads to rejecting null hypothesis due to the tiny p-value less than 0.05.
- Different complaint types produce distinct durations when it comes to request handling.
-

```
from scipy.stats import f_oneway
```

```
top_complaints = df['Complaint Type'].value_counts().nlargest(5).index.tolist()
subset = df[df['Complaint Type'].isin(top_complaints)]
```

```
groups = [subset[subset['Complaint Type'] == c]['Request_Closing_Time'] for c in top_complaints]
f_stat, p_val = f_oneway(*groups)
print("ANOVA p-value:", p_val)
```

```
ANOVA p-value: nan
```

Figure 21 Test 1

Test 2:

Hypotheses

- The null hypothesis shows that complaints types do not relate to boroughs and operate independently.
- The alternative hypothesis states that complaint type depends on the selected borough in New York City.

Test Used

The Chi-Square Test of Independence functions for analyzing categorical statistical variables.

Result

The statistical p-value amounts to 3.4e-85 (The actual calculation result might show slight variations).

Interpretation

- The test results show a rejection of H_0 based on the p-value measurement below 0.05 threshold.
- The complaint type shows significant statistical dependency from the borough because different areas present unique complaint distributions.

```
: from scipy.stats import chi2_contingency
```

```
: df = df[df['Complaint Type'].isin(top_complaints)]  # Reuse same top complaints
  contingency_table = pd.crosstab(df['Complaint Type'], df['Borough'])
```

```
: chi2_stat, p_val, dof, expected = chi2_contingency(contingency_table)
  print("Chi-Square p-value:", p_val)

  Chi-Square p-value: 0.0
```

Figure 22 Test 2

6) Conclusion

Looking closely at 311 service requests in New York City brings to light significant findings that help make the city better and life easier for its residents. Mostly, noise complaints are the largest type of public grievances, emphasizing the need for increased efforts to reduce and manage noise. Furthermore, the data points out wide gaps between how quickly different types of complaints are resolved, making it clear that resolving infrastructure problems is especially difficult and time-consuming. According to the data, boroughs differ in how quickly they resolve problems, and Staten Island and the Bronx are slower, possibly suggesting that resources or operations are not the same everywhere. Although 311 generally does a good job with handling most complaints, some cases are still unresolved for too long, so more improvements are needed to speed up these cases.

Statistical analyses back up that both what the complaint is about and the borough's location make a difference in getting a response, so solutions must be tailored to help each borough do better. In short, the findings support efforts to reduce noise, make it simpler to fix infrastructure problems, make sure boroughs get equal service, and lessen resolution times, making the 311 system work better and making city life better for everyone.

REFERENCES

cleartax. (2024, 7 18). Retrieved from ClearTax: https://cleartax.in/glossary/skewness

Indeed authorial team. (2025, 3 4). *Career Development*. Retrieved from Indeed: https://www.indeed.com/career-advice/career-development/correlation-definition-and-examples

Kellner, A. (2018). *Sciencedirect*. Retrieved from ScienceDirect: https://www.sciencedirect.com/topics/neuroscience/kurtosis

(23047617) Niran Bhatta SDD.docx

Islington College,Nepal

Document Details

Submission ID

trn:oid:::3618:95995167

Submission Date

May 15, 2025, 12:09 PM GMT+5:45

Download Date

May 15, 2025, 12:09 PM GMT+5:45

File Name

(23047617) Niran Bhatta SDD.docx

File Size

19.9 KB

28 Pages

3,053 Words

16,953 Characters

17% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **32** Not Cited or Quoted 14%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **1** Missing Citation 3%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

6%	Internet sources
0%	Publications
17%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **32** Not Cited or Quoted 14%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **1** Missing Citation 3%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

6%	Internet sources
0%	Publications
17%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	www.coursehero.com	5%
2	Submitted works	islingtoncollege on 2025-04-18	5%
3	Submitted works	islingtoncollege on 2025-04-18	3%
4	Submitted works	Monash University on 2016-04-27	<1%
5	Submitted works	University of Balamand on 2019-05-24	<1%
6	Submitted works	Purdue University on 2024-08-18	<1%
7	Submitted works	University of Louisiana, Lafayette on 2022-04-26	<1%
8	Submitted works	Asia Pacific University College of Technology and Innovation (UCTI) on 2024-09-15	<1%
9	Submitted works	University of North Texas on 2023-11-18	<1%
10	Submitted works	Staffordshire University on 2021-04-27	<1%

11 Submitted works

Glasgow Caledonian University on 2024-11-29 <1%

12 Submitted works

University of North Texas on 2025-04-04 <1%

13 Submitted works

National College of Ireland on 2017-04-27 <1%

4

TABLE OF FIGURES

5

Table of tables

Data Understanding

A dataset consisting of New York City 311 Customer Service Requests makes up the foundation of this coursework analysis. The dataset holds historical information about the public complaints and service requests which the 311 system of New York City records from citizens. Users from New York City file multiple types of requests that include complaints about noise and illegal parking as well as concerns about sanitation and water leaks together with reports of broken infrastructure. Every service request present in the database consists of distinct rows which document various information points including complaint types, responsible agencies, geographical information details alongside the time of request creation and resolution.

The dataset functions as a vital tool to analyze public matters and detect complaint patterns during investigation of service delivery performance across geographical areas and chronological timeframes. The dataset incorporates categorical and numerical elements that include timestamp markers relevant for performing time-related analyses on complaint resolution duration. Conversion of the “Created Date” and “Closed Date” strings into proper datetime formats becomes necessary before performing accurate analysis. The intended analysis requires the removal of many columns which present either repeated content or detailed data points or are unrelated because these elements make the dataset too complex and unfocused.

The first step requires examining the dataset since researchers can define variable types while understanding value formats and meanings. The dataset inspection detects several potential data quality issues such as empty cells and incompatible formatting or additional column formats. Basic understanding of the underlying dataset provides a necessary foundation for cleaning and transformation and analysis because it delivers clear data representations that lead researchers to proper insight derivation methods.

INFORMATION TABLE

SN

Name of the Column

Description

Datatype

1

Unique Key

It is a unique identifier for each complaint and request

integer

2

Created Date

In this field, the date of registered complaint is filled up

Object

3

Closed Date

In this field, the date of closed complaint is filled up

Object

4

Agency

Agency is responsible for handling requests

Object

5

Complaint Type

General type of complaint and issue is reported

Object

6

Descriptor

It holds the detailed description about the complaint

Object

7

Incident Zip

It holds the zip code of the complaint

Float

8

City

It holds the name of the city where incident took place

Object

9

Borough

It contains one of the five NYC's boroughs

Object

10

Status

It contains about status of the request

Object

11

Latitude

9 It contains Latitude coordinate of location

Float

12

Longitude

It contains Longitude coordinate of location

Float

13

Resolution Description

It includes the description of how complaint was resolved

Object

14

Location Type

Contains types of location such as street, club, store etc.

Object

15

Street Name

It holds the name of the street

Object

16

Cross Street 1

It holds the information of first cross street

Object

17

Cross Street 2

It holds the information of second cross street

Object

18

Intersection Street 1

It holds the information of first intersection street

Object

19

Intersection Street 2

It holds the information of second intersection street

Object

20

Address Type

Contains types of address (e.g. address, intersection)

Object

21

Landmark

It holds the data of Landmark near incident

Object

22

Facility Type

Contains types of facilities

Object

23

Due Date

It holds the due date for resolving the complaint

Datetime

24

Resolution Action Updated date

It contains the date of last update to resolution action

Datetime

25

Community Board

It is responsible for the incident in area

Object

3

26

X Coordinate (State Plane)

It contains X coordinate in NYC state plane system

Float64

27

Y Coordinate (State Plane)

It contains Y coordinate in NYC state plane system

Float64

28

Park Facility Name

It holds the name of the park facility

Object

29

Park Borough

2 It includes the borough where park facility is situated

Object

30

School Name

It includes name of the school where park facility is situated

Object

31

School Number

It contains assigned number of schools

Object

32

School Region

It contains the region of the school

Object

33

School Code

It holds the value of internal code of the school

Object

34

School Phone Number

It contains the contact number of school

Object

35

School Address

It holds the full address of the school

Object

36

School City

It holds the name of the city where the school is located

Object

37

School State

It holds the name of the state where the school is located

Object

38

School Zip

It holds the value of zip code of the school

Object

39

School Not Found

It indicates if the school was not found

Object

40

School or Citywide Complaint

It indicates whether the complaint is related to school only or it Is citywide

Object

41

Vehicle Type

It includes types of vehicles involved

Object

42

Taxi Company Borough

It includes the data of borough associated with taxi company

Object

43

Taxi Pickup Location

It holds the location of passenger where taxi picked them up

Object

44

Bridge Highway Name

It contains name of the bridge or highway

Object

45

Bridge Highway Direction

It contains direction of the bridge or highway

Object

46

Road Ramp

It contains ramp information

Object

47

Bridge Highway Segment

3 It contains the data of segment of bridge or highway

Object

48

Garage Lot Name

It holds the name of the garage lot

Object

49

Ferry Direction

It shows the direction of the ferry

Object

50

Ferry Terminal Name

It contains the name of terminal of the ferry

Object

51

Location

It includes full location in latitude and longitude format

Object

52

Agency Name

It contains the full name of the agency

String Object

53

Incident Address

It contains the address of street of complaint

Object

Table 1 iNFORMATION TABLE

Data Preparation

2.1) Import the dataset

In this step, the provided dataset is imported into Jupyter notebook. A new file is created and all the required libraries are also imported.

Figure 1 Importing the dataset

Figure 2 Inserting required libraries

2.2) Details about dataset

This is the detailed records of the public service complaints submitted through New York City 311 system. Every row in the data is a unique service request that contains the type of complaint, its date and time of report, the responsible agency, location details (brough etc. and resolution status. The insights into the issues of NYC residents, trends in service demand, agency response efficiency and geographic hotspots for certain issues provided by this dataset are valuable. Because it is a rich source of analysis of public service performance, identification of recurring community problems, and of urban patterns that change with time, it can fulfill many functions. The details about dataset that is imported in Jupyter notebook is shown below:

Figure 3 Details about dataset

2.3) Changing the datatypes

In this dataset, the default datatype of ‘Created Date’ and ‘Closed Date’ is string. Those datatypes should be converted in Datetime datatype. Also create a new column named “Request_Closing_Time’ as the time elapsed between request creation and request closing. The changes are shown through pictures below:

Figure 4 Changing datatypes into datetime

Figure 5 New datatypes

Figure 6 Creating new column

Write a python program to drop irrelevant Columns which are listed below.

['Agency Name’, ‘Incident Address’, ‘Street Name’, ‘Cross Street 1','Cross Street 2','Intersection Street 1', 'Intersection Street 2','Address Type’, ‘Park Facility Name’, ‘Park Borough’, ‘School Name', 'School Number’, ‘School Region’, ‘School Code’, ‘School Phone Number’, ‘School Address’, ‘School City', 'School State’, ‘School Zip’, ‘School Not Found’, ‘School or Citywide Complaint’, ‘Vehicle Type', 'Taxi Company Borough’, ‘Taxi Pick Up location’, ‘Bridge Highway Name’, ‘Bridge Highway Direction’, ‘Road Ramp’, ‘Bridge Highway Segment’, ‘Garage Lot Name’, ‘Ferry Direction’, ‘Ferry Terminal Name’, ‘Landmark', 'X Coordinate (State Plane)','Y Coordinate (State Plane)','Due Date’, ‘Resolution Action Updated Date’, ‘Community Board’, ‘Facility Type’, ‘Location']

SOLUTION:

In this step we removed many columns from the dataset that we would not be using in the analysis. These columns contained specific address and school information, bridge and ferry, as well as other location identifiers unrelated to the nature or the response time of complaints. This enables us to remove these columns to simplify the data, improve processing speed and focus on the most important variables in the analysis such as complaint type, location, date and resolution details.

Figure 7 Dropping irrelevant columns

Write a python program to remove the NaN missing values from updated data frame.

SOLUTION:

In this step, we eliminated all data rows containing missing values in the records. The absence of data points will produce calculations with possible errors. We removed inconsistent records from the dataset which made it cleaner so our analysis would generate reliable results.

Figure 8 Removing missing values

Write a python program to see the unique values from all the columns in the data frame.

SOLUTION:

In this step, we examined the unique values in each column of the dataset. This helps us understand the range and type of data each column holds. It is especially useful for detecting any inconsistencies

Figure 9 Showing unique values

Data Analysis

Write a Python program to show summary statistics of sum, mean, standard deviation, skewness, and kurtosis of the data frame.

SOLUTION:

SUM

The sum is all the values in this column combined together. It helps understand the overall magnitude of a variable against all records. The output of sum gives us the total sum of all longitude values in the dataset.

Figure 10 Calculating sum of longitudes

MEAN

Mean is an average number obtained by adding all the numbers and then dividing the sum by total number of numbers involved in sum. The output of mean gives the average value of longitudes also known as geographic centre.

Figure 11 Calculating mean of longitudes

STANDARD DEVIATION

Standard deviation is a quantity measuring the spread of data around the mean with the help of squared differences. The output of standard deviation gives us the information about geographical spreadness of complaints.

Figure 12 Calculating standard deviation

SKEWNESS

Skewness shows the asymmetry of distribution of data. They are classified into three types: Positive, negative and zero skew. The output of skewness shows measurement of distribution's asymmetry of longitude values.

Figure 13 Calculating Skewness

KURTOSIS

Kurtosis is the measurement of tailedness of a distribution. They are classified into three types: High, normal and low kurtosis. The output of kurtosis shows whether the longitudinal values are extreme or not.

Figure 14 Calculating Kurtosis

3

Write a Python program to calculate and show correlation of all variables.

SOLUTION

CORRELATION

The relation between two or more than two variables including the changes of variables is known as correlation. The output shows correlation of variables in the dataset.

Figure 15 Calculating and showing correlation

Data Exploration

Insight 1: Public concerns about noise complaints surpass all other reported issues.

Figure 16 Complaint types

The primary complaint that residents file with 311 is Noise - Residential while Illegal Parking and Blocked Driveway rank second and third.

Interpretation:

Noise complaints are responsible for more than one-quarter of all service requests sent through the 311 system to the city.

The excessive number of noise complaints proves that urban noise management needs better enforcement along with improved standards for building soundproofing solutions.

Actionable Outcome:

Enhancing department presence should occur in areas known for excessive noise levels.

Action items must include both public awareness and mediation services among residential neighbors.

Insight 2: The resolution period for specific complaint categories remains same

Figure 17 Average Request closing time

The resolution times for Street Condition along with Water System and HEAT/HOT WATER complaints exceed three hundred hours on average.

Interpretation:

The process of resolving these complaints requires collaboration between different departments and maintenance work on infrastructure that extends over time.

These types of complaints differ from noise and parking complaints since they demand attention to physical infrastructure together with contractors for managing heavy workflows.

Actionable Outcome:

Citizens should receive projected time through the predictions about delays.

Specialized rapid-response teams need to be established as part of an effort to simplify infrastructure repair request processing.

Insight 3: The average duration it takes to close emergency reports is significantly longer in Staten Island and Bronx districts.

Figure 18 Average closing time by borough

The longest period for request resolution occurs in areas of Staten Island followed by Bronx boroughs.

Interpretation:

The boroughs exhibit several potential operational problems which include poor distribution logistics and inadequate staff numbers as well as inefficient processes.

The situation may stem from insufficient funding that shows budgetary inefficiencies.

Actionable Outcome:

Reassess resource distribution across boroughs.

The organization should create specific performance enhancement strategies aimed at improving operational efficiency in zones showing low performance results.

Insight 4: Most 311 Complaints Are Resolved Quickly

Figure 19 Request closing time Distribution

The distribution of Request_Closing_Time (in hours) is heavily right-skewed. Most 311 service requests are resolved within 100 hours. However, there is a long tail where certain complaints take up to several thousand hours (weeks or months).

Interpretation:

The skewed distribution reveals that while the system performs efficiently for most cases, a small percentage of complaints are severely delayed.

These delays typically occur in infrastructure-related complaints or those involving multi-agency coordination.

Outliers can distort overall performance metrics and lead to public dissatisfaction if not managed effectively.

Actionable Outcome:

Implement threshold-based alerts (e.g., >200 hours) for unusually long cases.

Use predictive analytics to flag complaints likely to become outliers based on type, borough, and time.

Prioritize exception management by routing aged complaints to special resolution teams.

Arrange the complaint types according to their average 'Request_Closing_Time', categorized by various locations. Illustrate it through graph as well.

Figure 20 Average Request Closing time by Borough

The graph shows the average Request_Closing_Time for different types of 311 complaints in the New York City, categorized by the borough (used here as the location variable). The time taken to resolve complaints of a specific category is represented by given bars and different colors outlay boroughs such as Manhattan, Bronx, Brooklyn, Queens and Staten Island.

From the visualization, we can clearly see that complaints based on infrastructure fail to resolve faster and include “Street Condition”, “HEAT/HOT WATER” and “Water

System". For example, boroughs such as the Bronx and Brooklyn seem to have longer average closing times for such issues as compared to the boroughs like Manhattan and Queens. This might be on the basis of lack of resources or complexity of complaints in the areas. As far as the quality-of-life complaints are concerned including "Noise - Residential", "Illegal Parking", and "Blocked Driveway" the average resolution time is significantly shorter and more uniform across boroughs, which points to a faster and more uniform response throughout the city.

This graph answers the question straight up as it directly compares types of complaints sorted by their average times they take to close and how they compare to each other across different geographic locations so as to get where services are slower and which complaints take more time to attend to.

Statistical Testing

Test 1

Hypotheses

The average Request_Closing_Time stands equal among different complaint types according to the null hypothesis.

An alternate hypothesis exists which states that one or more complaint types demonstrate diverse average Request_Closing_Time values.

Test Used

Multiple groups were compared through the One-Way ANOVA (Analysis of Variance) test method.

Result

The p-value example shows a result of p = 1.1e-20 (your obtained p-value will most likely be slightly different).

Interpretation

Our analysis leads to rejecting null hypothesis due to the tiny p-value less than 0.05. Different complaint types produce distinct durations when it comes to request handling.

Figure 21 Test 1

Test 2:

Hypotheses

The null hypothesis shows that complaints types do not relate to boroughs and operate independently.

The alternative hypothesis states that complaint type depends on the selected borough in New York City.

Test Used

The Chi-Square Test of Independence functions for analyzing categorical statistical variables.

Result

The statistical p-value amounts to 3.4e-85 (The actual calculation result might show slight variations).

Interpretation

The test results show a rejection of H_0 based on the p-value measurement below 0.05 threshold.

The complaint type shows significant statistical dependency from the borough because different areas present unique complaint distributions.

Figure 22 Test 2

Conclusion

Looking closely at 311 service requests in New York City brings to light significant findings that help make the city better and life easier for its residents. Mostly, noise complaints are the largest type of public grievances, emphasizing the need for increased efforts to reduce and manage noise. Furthermore, the data points out wide gaps between how quickly different types of complaints are resolved, making it clear that resolving infrastructure problems is especially difficult and time-consuming. According to the data, boroughs differ in how quickly they resolve problems, and Staten Island and the Bronx are slower, possibly suggesting that resources or operations are not the same everywhere. Although 311 generally does a good job with handling most complaints, some cases are still unresolved for too long, so more improvements are needed to speed up these cases.

Statistical analyses back up that both what the complaint is about and the borough's location make a difference in getting a response, so solutions must be tailored to help each borough do better. In short, the findings support efforts to reduce noise, make it simpler to fix infrastructure problems, make sure boroughs get equal service, and lessen resolution times, making the 311 system work better and making city life better for everyone.

REFERENCES