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: Gaurav Pradhan

London Met ID: 23050283

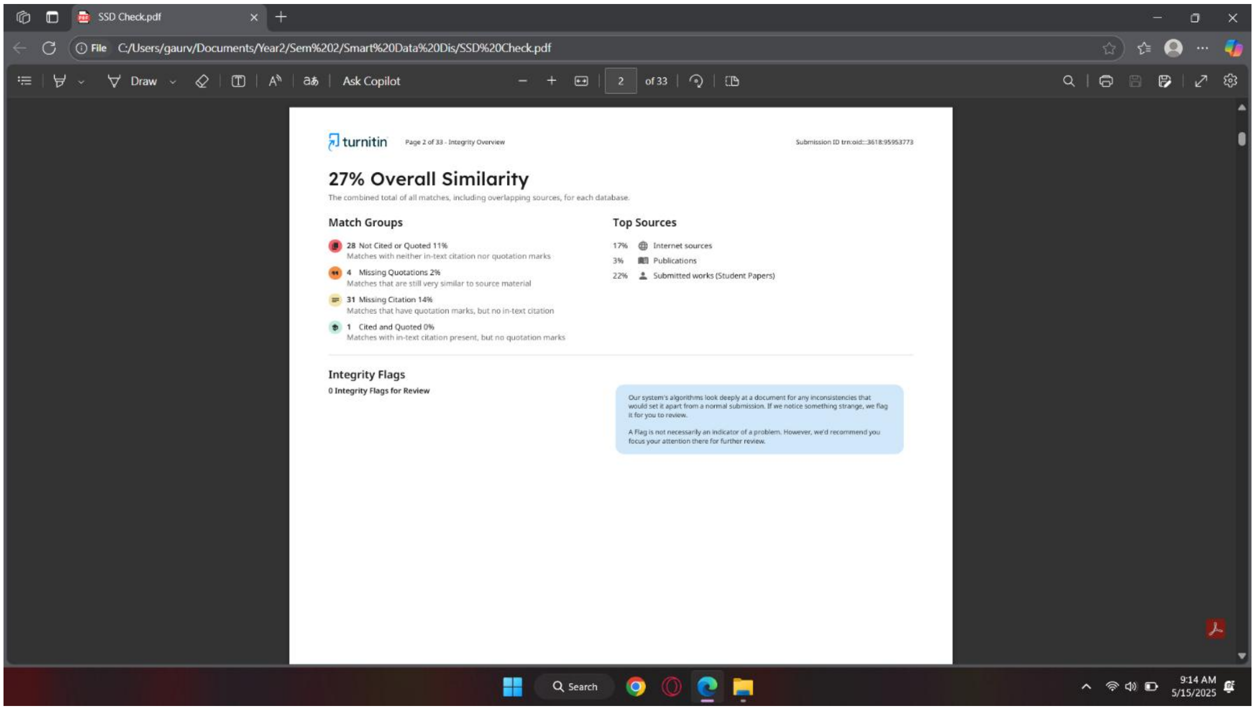
College ID: NP 01CP4A230277

Assignment Due Date: Thursday, May 15, 2025

Assignment Submission Date: Thursday, May 15, 2025

Submitted To: Mr. Dipeshor Silwal

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.



This similarity report was done with the removal of data understanding table and without the removal this is still at 27% with all the questions and with the removal of all the question and table was less than 20%.

Table of content

Table of figure

Table of table

1. Data understanding

Data understanding is one of the most important steps in data analysis process. Which helps in correct decision making in analysis and shows the strength of data. It gives us quality data by ensuring accuracy and reliable result. In simple words it the process of getting familiarity with the data by exploring its structure, quality, and context. (coskun, 2023)

The provided dataset is titled Customer service requests which shows the comprehensive records of 311 calls which is received in New York City. It includes very detailed accounts of 311 New York City resident citizen complaints. They encompass a range of customer service grievances from noise, illegal parking, sanitation, water leaks, etc. Each account contains significant data like category of the agency responsible for handling the complaint, location, and timestamp showing when the complaint was opened and closed through created and closed date.

This information offers further research, and temporal trends, frequency, and type analysis of complaints by region, agency response time measurement, and identification of potential service inequities in the city can be explored.

Column name	**Description**	**Data Type**
Unique Key	A unique that identifies the assigned to each service request.	Integer
Created Date	Date and time of creation	Object
Closed Date	The date and time of resolution or closure.	Object
Agency	The person of the city agency who is responsible for complaint handling.	Object
Agency Name	Full name of the agency responsible for addressing the issue.	Object
Complaint Type	A broad category that categorizes the nature of the complaint.	Object

Descriptor	Specific, detailed information about the specific type of complaint.	Object
Location Type	Location where the issue occurred like Zip code and address	Object
Incident Zip	The part of city where the complaint came from .	Integer
Incident Address	A notable landmark situated in proximity to the incident.	Object
Street Name	The specific type of facility, such as a school, park, or other relevant establishment, associated with the complaint.	Object
Cross Street 1	The cross street which is near to the first occurrence.	Object
Cross Street 2	The cross street which is near to the second occurrence.	Object
Intersection Street 1	One of the intersecting roads at the place.	Object
Intersection Street 2	The other road intersecting at the place.	Object
Address Type	Address format or type provided	Object
City	City where the complaint was lodged.	Object
Landmark	Major landmark near the occurrence	Object
Facility Type	The facility which is related to the complaint	Object
Status	Request Status weather its active or not	Object
Due Date	Expected date of fixing the request.	Object
Resolution Description	Explanation of how the issue was fixed.	Object
Resolution Action Updated Date	Data on last updated resolution.	Object
Community Board	Administrative district board pertaining to the area.	Object

Borough	One of NYC's five boroughs where the request was submitted.	Object
X Coordinate	X coordinate within the States planed coordinate system in NYC.	Float
Y Coordinate	Y coordinate within state planed coordinate System in NYC.	Float
Park Facility Name	Name of park involved	Object
Park Borough	Borough where the park is located.	Object
School Name	Name of the school pertinent to the complaint.	Object
School Number	Official school number.	Object
School Region	Region code within the NYC Department of Education.	Object
School Code	School code assigned.	Object
School Phone Number	School contact phone number.	Object
School Address	School street address.	Object
School City	Place in the city where the school is located.	Object
School Zip	School ZIP code.	Integer
School Not Found	Whether the school was found or not	Object
School or Citywide Complaint	If the issue is school or citywide.	Object
Vehicle Type	Vehicle type that was involved in complaint	Object
Taxi Company Borough	Taxi company borough	Object
Taxi Pick Up Location	Location in which taxi picks someone up.	Object
Bridge Highway Name	Name of bridge or highway	Object
Bridge Highway Direction	Direction of the bridge or highway	Object
Road Ramp	Single road ramp within the scope of the complaint.	Object
Bridge Highway Segment	Individual bridge or highway segment.	Object

Garage Lot Name	Name of a parking garage within the scope of the request.	Object
Ferry Direction	Travel direction of the ferry.	Object
Ferry Terminal Name	ferry terminal name within the scope of the issue.	Object
Latitude	Geographic latitude of the incident location.	Float
Longitude	Geographic longitude of the incident location.	Float
Location	Blended latitude and longitude data	Object

Table 1 Data dinery

2. Data preparation

2.1. Importing dataset

Q. Provide your insight on the information and details that the provided dataset carries.

A. Before we import the data set, we had to import a python library like 'pandas'. Now to import the dataset we executed this code:

"df = pd.read_csv("Customer Service_Requests_from_2010_to_Present.csv")" here this code used panda to load, access the CSV file to the pandas data frame. (geeksforgeeks.org, 2024)

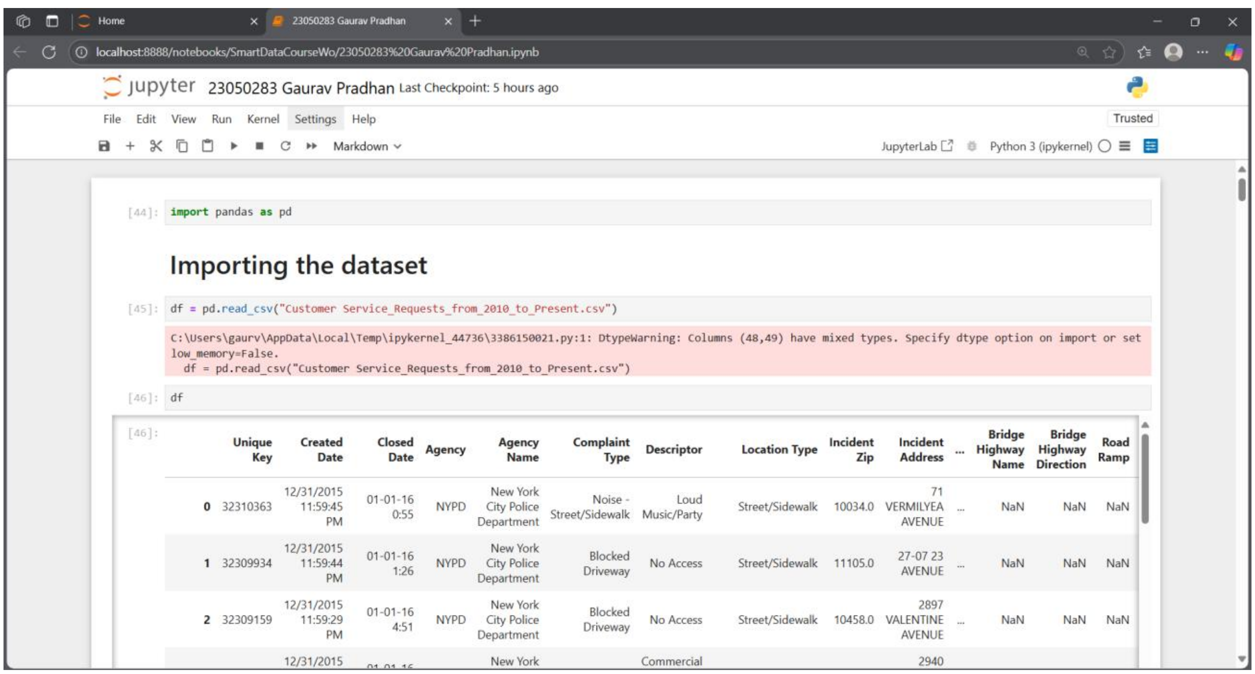


Figure 1 Importing dataset

As we can see there is an error popping up so fix the error, we edited the code to: “df=pd.read_csv("CustomerService_Requests_from_2010_to_Present.csv",low_memory = False)” and the error was fixed and to check if our csv file was uploaded or nor we used “df” to check. Which helps to enhance the efficiency and accuracy of the data with larger dataset and which consist column with custom data type. (geeksforgeeks.org, 2025)

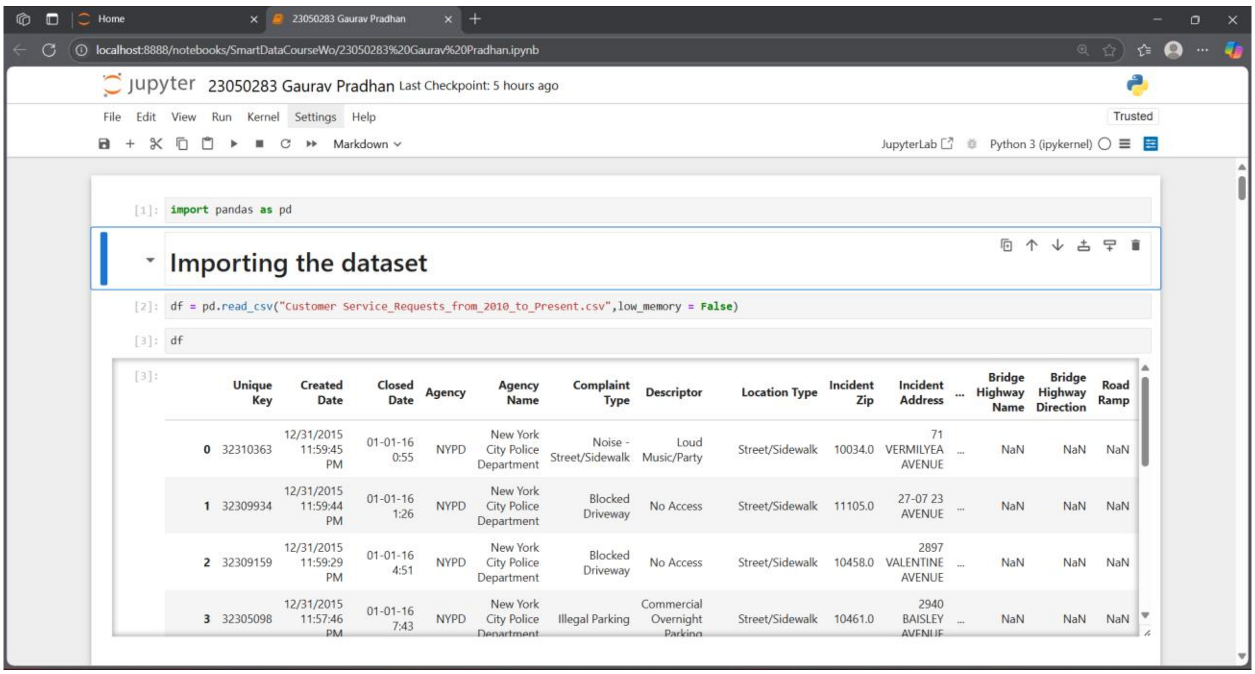


Figure 2 importing dataset without errors

2.2. Providing inside on the Dataset

Q. Convert the columns "Created Date" and "Closed Date" to datetime datatype and create a new column "Request_Closing_Time" as the time elapsed between request creation and request closing.

A. To see the get some inside on the dataset we executed this following code:

“df.info()” and “df.head()”.

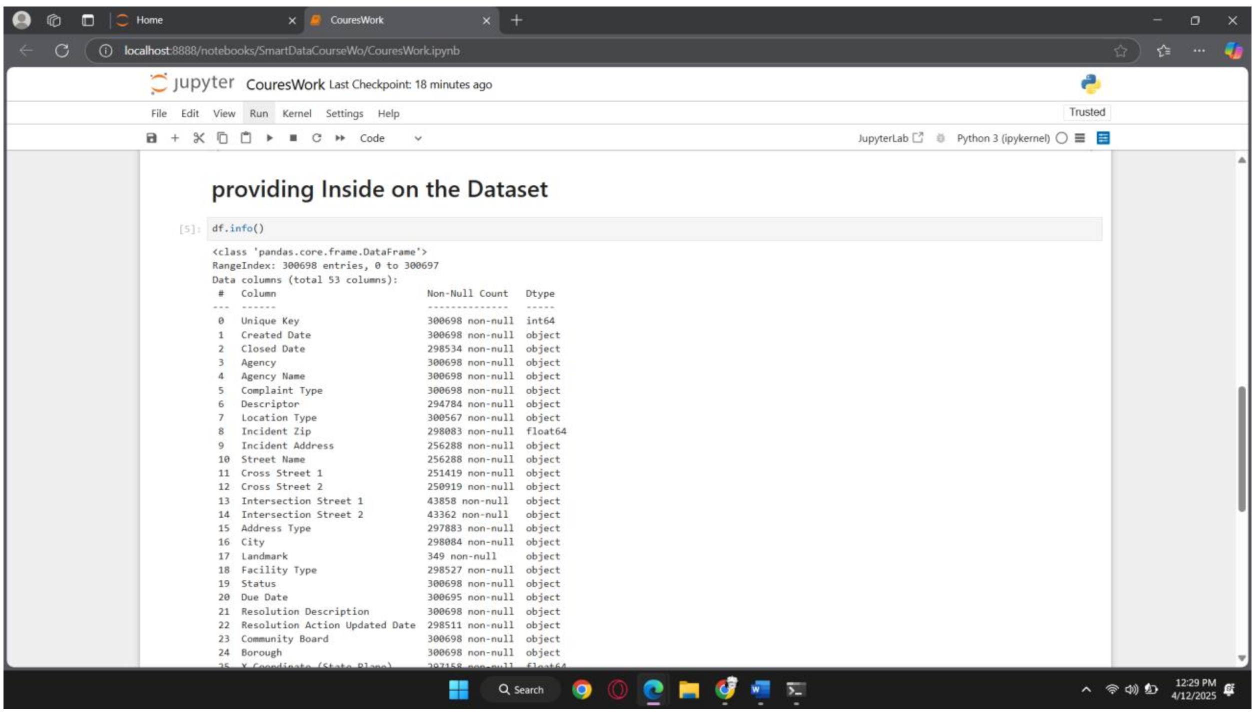


Figure 3 Providing inside on Dataset with .info ()

This is the result of df.info () which prints the information about the data frame, that may consist number of columns , data type, the number of cell in each column and non-null values.

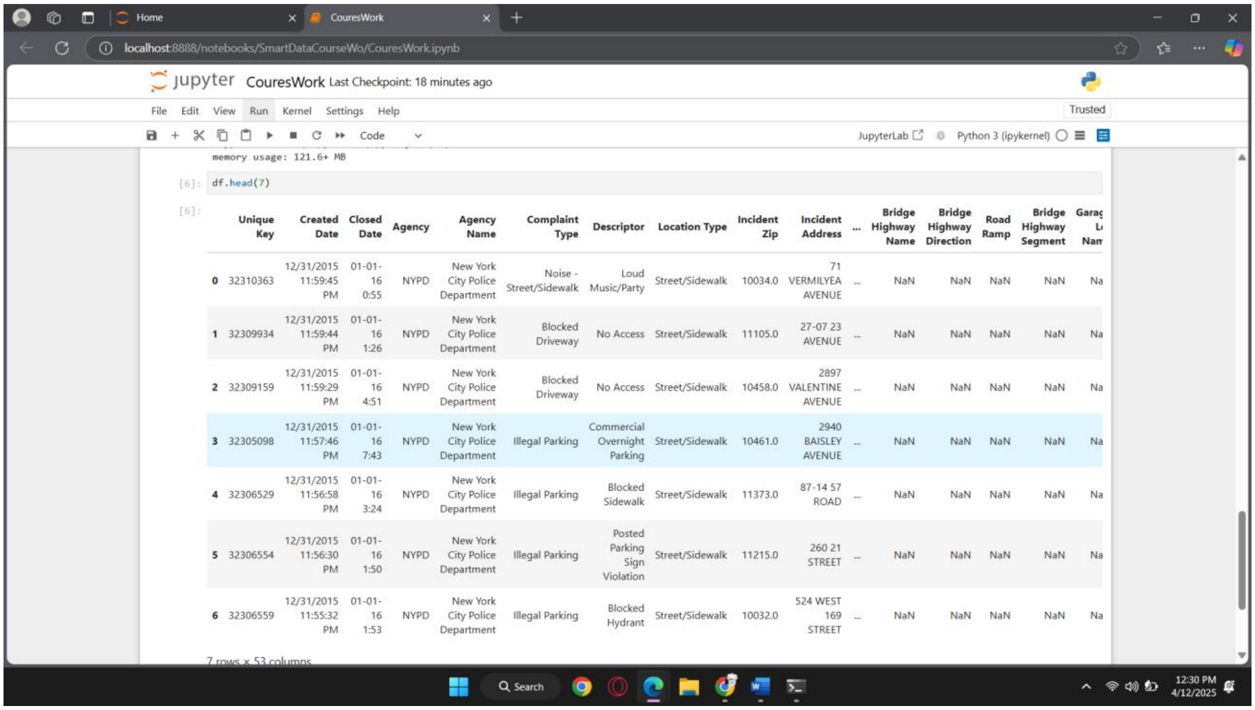


Figure 4 Providing inside on Dataset with. head ()

This image shows the result of df.head () which shows a specific number of rows from the top we can specific the number oof rows by giving the number inside the bracket.

2.3. Converting the column and creating a new column

Q. Convert the columns "Created Date" and "Closed Date" to datetime datatype and create a new column "Request_Closing_Time" as the time elapsed between request creation and request closing

A. To convert the column, I executed the following code:

“df['Created Date'] = pd.to_datetime(df['Created Date'])”

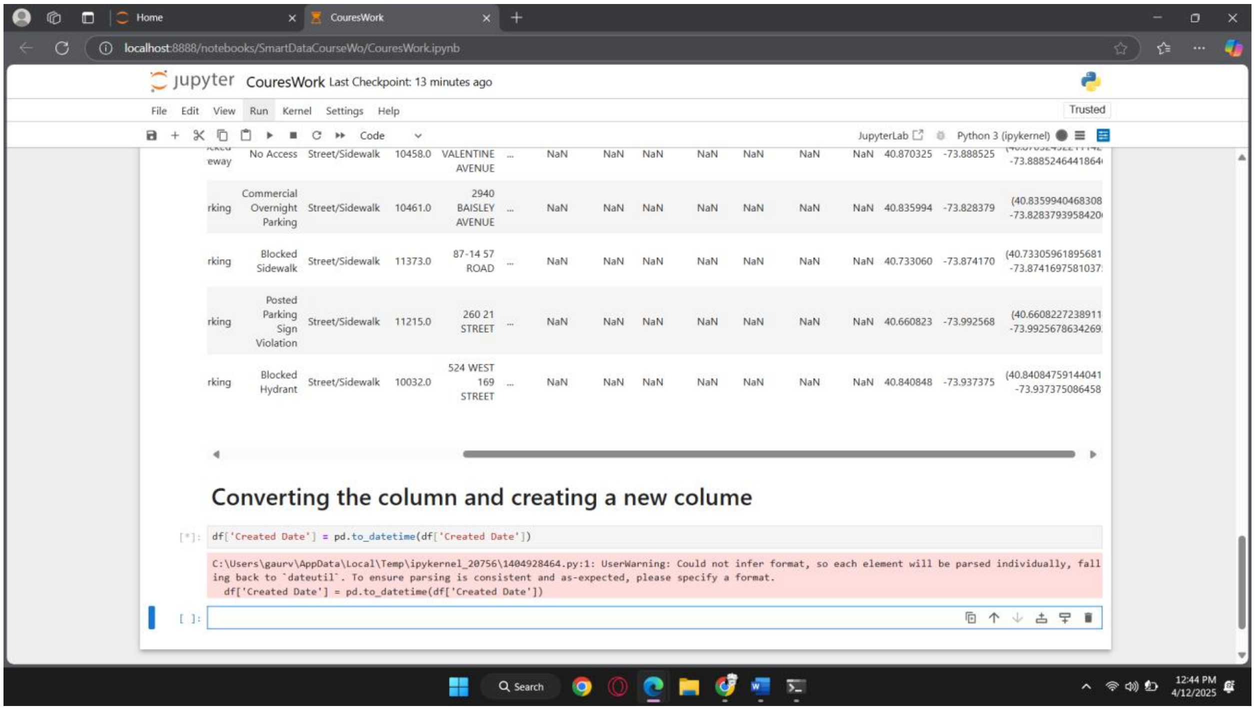


Figure 5 converting column for created date

This code came with an error which required proper formatting and doing that would bring some feature errors to ensure that there will not be any feature errors I modified the code.

The code:

“df ['Created Date'] = pd.to_datetime(df['Created Date'], errors = 'coerce')

df ['Closed Date'] = pd.to_datetime(df['Closed Date'],errors='coerce')”

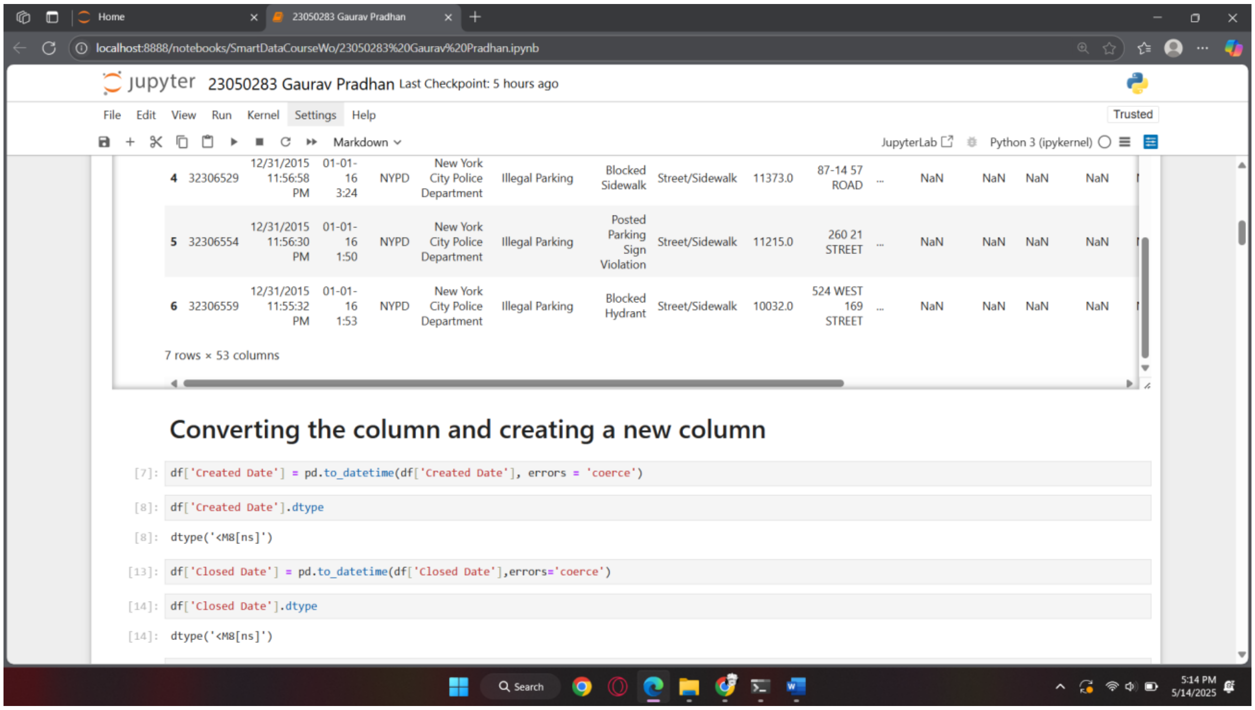


Figure 6 To check the conversion for Closed date and created date

This shows the result of converting column created date to shift the string format to datetime to insures parsing faster, with the data type of both create date and closed date.

To create a new column ‘Request_Closed_Time’ I executed the following code:

“df ['Request_Closing_Time'] = df['Closed Date'] - df['Created Date']”, here we defined a new column Request_Closing_Time and provided the data.

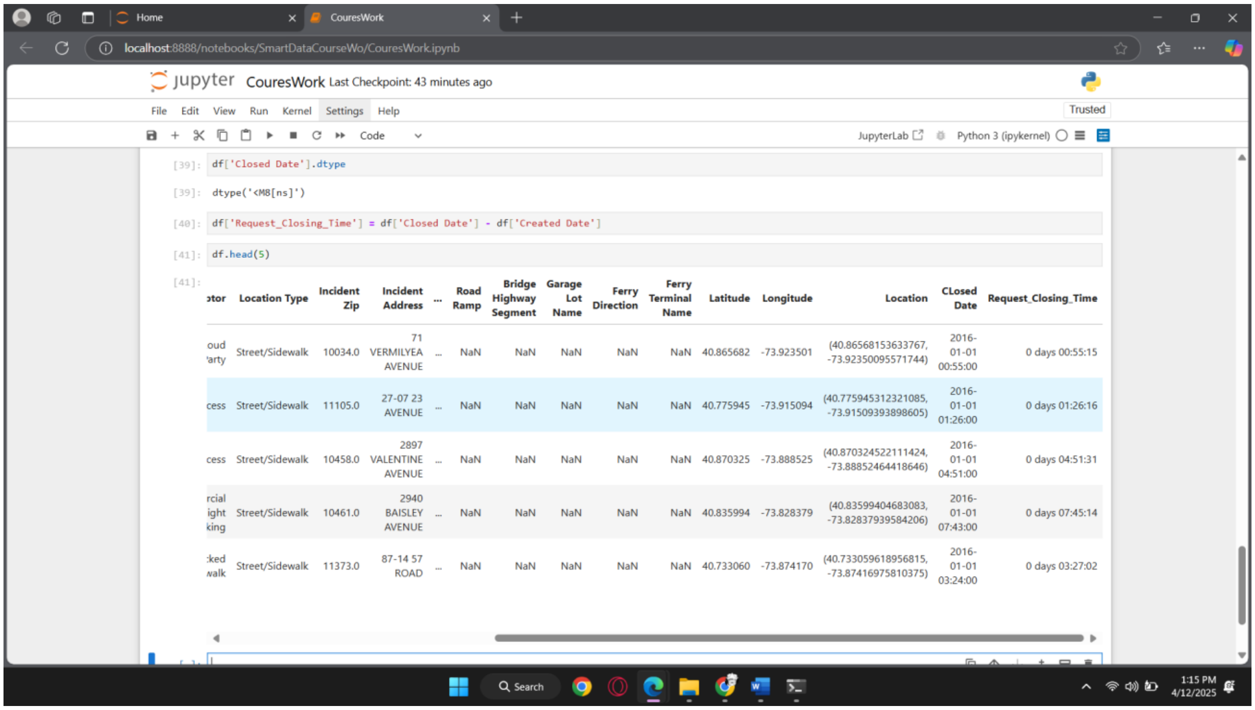


Figure 7 creating Request_Closing_Time column

This shows the result of creating a new column which takes the data from previous two columns and we used df.head to see if the column was created or not.

2.4. Dropping Irreverent Columns

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

A. To drop some irreverent columns which was provide in the question I executed this code:

"columns_to_drop =['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.drop(columns = columns_to_drop, axis = 1, inplace = True)"

where we defined the column, we wanted to drop and also with axis = 1 we defined the axis also with inplace = True makes the change directly to the original Data Frame df without needing to assign it to a new variable. (Harris, 2021)

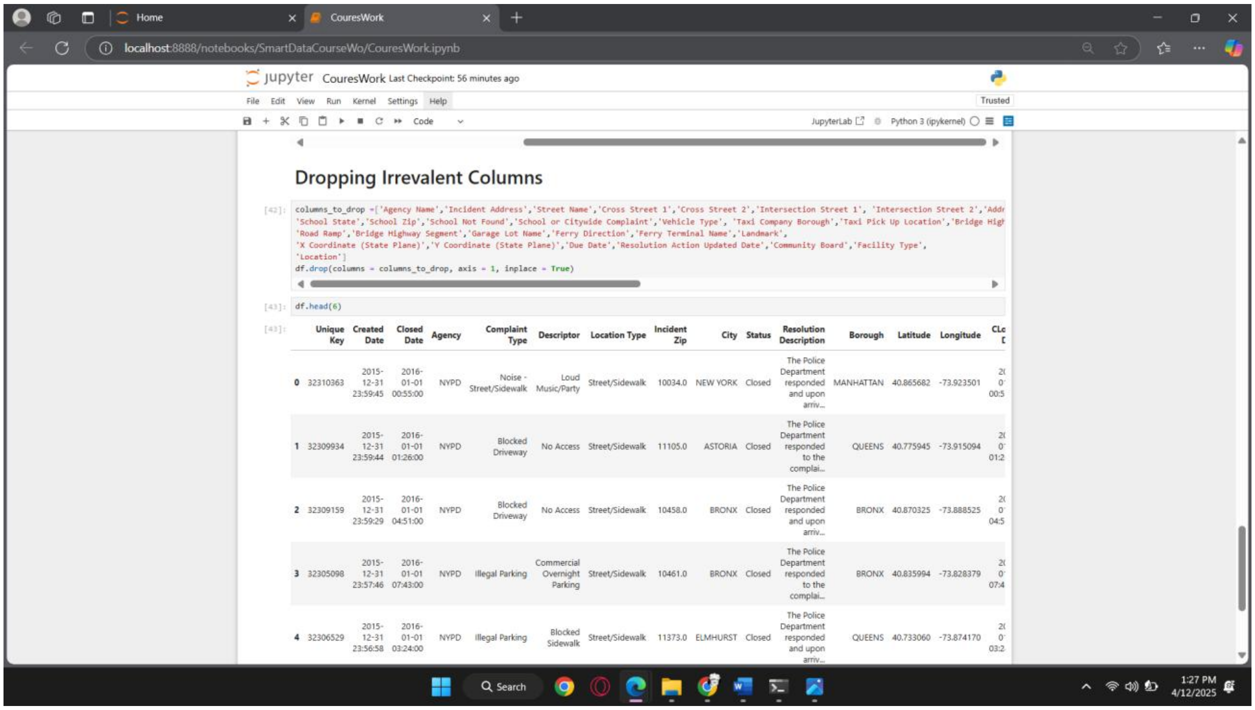


Figure 8 Dropping Irreverent Columns

This shows the result to removing some irreverent columns. And it also insures that change which is done in the data frame does not create a new one by making the modification on the data frame and axis = 1 makes sure that drop row is zero or column one.

2.5. Removing the NaN missing values from updated data frame

Q. Write a python program to remove the NaN missing values from updated dataframe.

A. To remove NaN missing value, I executed this code:

“df_clean = df.dropna()” and to check if it deleted then I used “df.head(4)” .

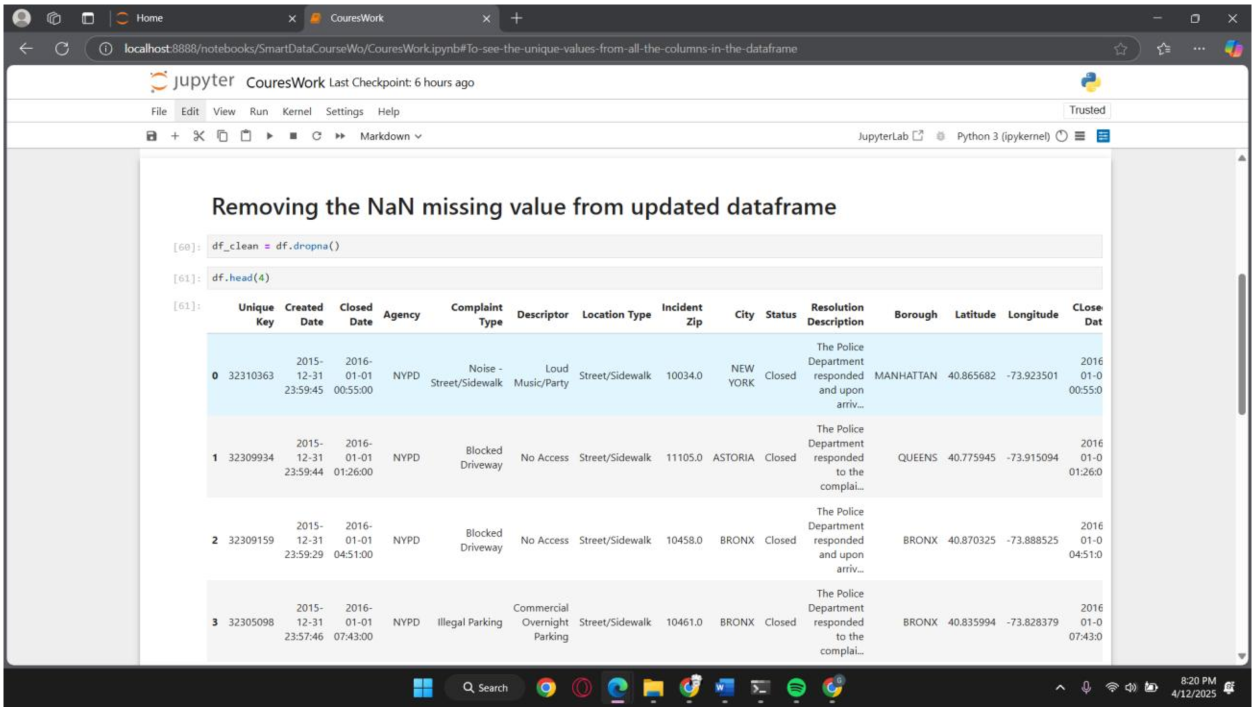


Figure 9 removing the NaN value

Here to drop nan value I used df_clean.dropna() and then I used df.head(4)to see four row to check whether it was removed or not.

2.6. To see the unique values from all the columns in the data frame

Q. Write a python program to see the unique values from all the columns in the dataframe.

A. To see the unique values, I executed following code:

“for col in df_clean.columns:

print(df_clean[col].unique())”

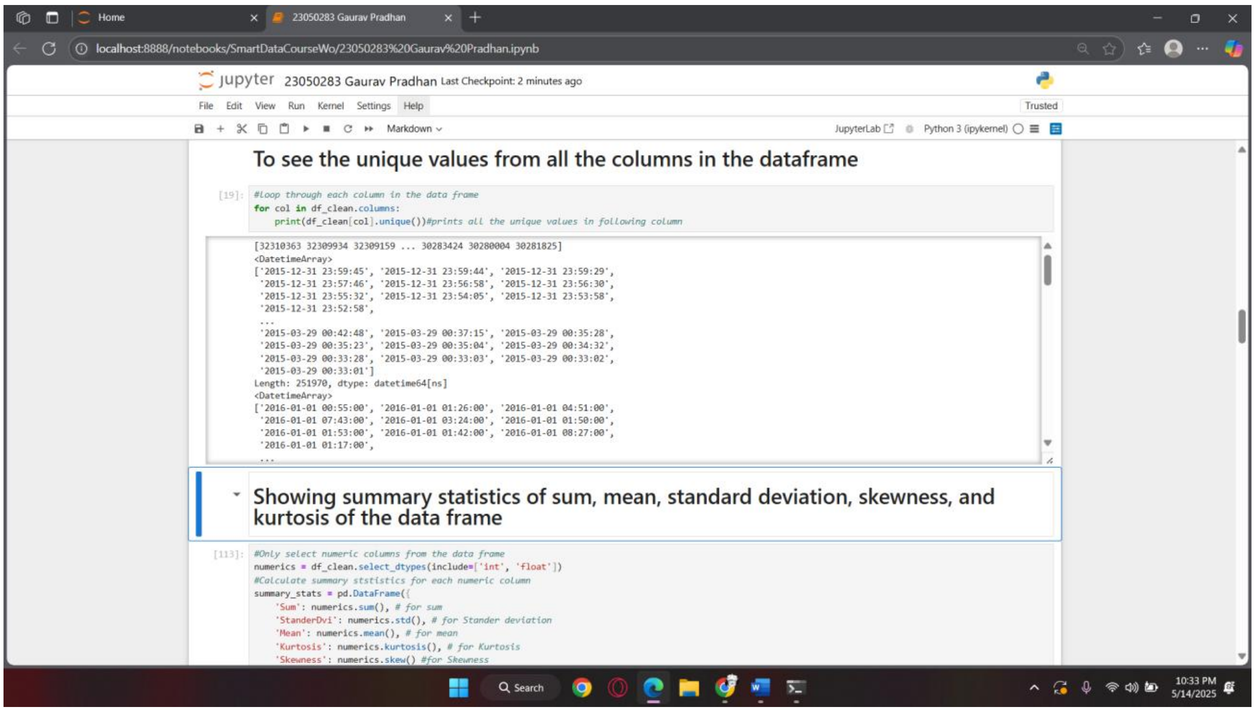


Figure 10 unique values from all the columns in the data frame

Here we can see the output which shows the unique value from the data frame.

3. Data Analysis

3.1. Showing the summary of statistics which include Sum, Mean, Standard Deviation, Skewness, and Kurtosis of the data frame

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

A. To show the summary statistics of the requirement I used the following code:

```
“numerics = df_clean.select_dtypes(include=['int', 'float'])
 summary_stats = pd.DataFrame({
    'Sum': numerics.sum(),
    'StanderDvi': numerics.std(),
    'Mean': numerics.mean(),
    'Kurtosis': numerics.kurtosis(),
    'Skewness': numerics.skew()


})
summary_stats_df=pd.DataFrame(summary_stats)
summary_stats_df”
```

Here we ca see the Sum, Mean, Standard deviation, Kurtosis and skewness of unique key, incident zip, latitude, and longitude. In which select_dtype filters from cleaned dataFrame and int and float ate integer and float which is stored in numerics. And summary_stats_df=pd.DataFrame(summary_stats) makes sure the output is kept in proper fomat.

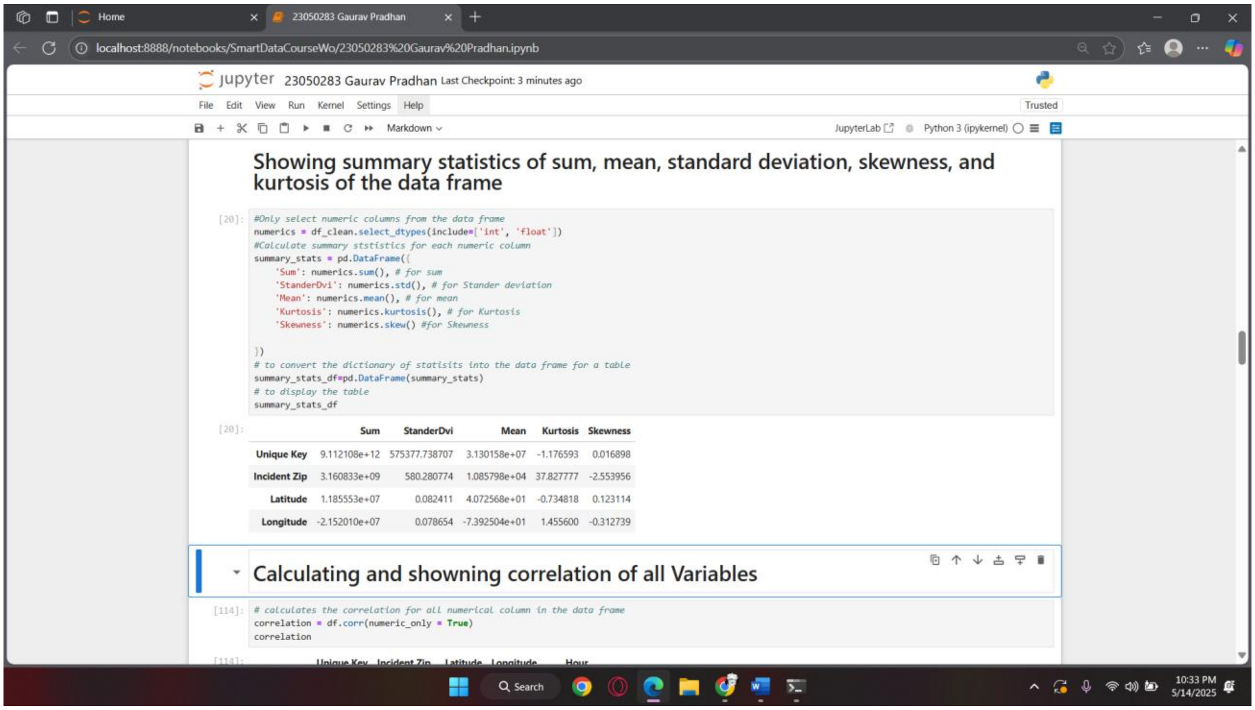


Figure 11 Showing the summary statistics of the data frame

The summarizing the description statistics for the numerical column using five key metrics they are:

Sum: It's the total of all values in a row or column

Standard Deviation: A measure of how data is dispersed or spread out.

Mena: It's the average of given set of data or values.

Kurtosis: It's describing the shape of probability. (sciencedirect, 2025)

Skewness: A measure of the asymmetry of the probability distribution of real values. (Turney, 2022)

The findings:

Highest variable in unique key.

Negative Skewness in incident Zip.

Kurtosis Observation.

3.2. Calculating and Showing correlation of all variables

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

A. To show the calculated value of the correlation of all variables I executed this code:

“correlation = df.corr(numeric_only = True)

correlation” where we defined a variable called correlation and the df is the csv file and corr is a method that calculates the pairwise correlation between numeric columns (w3schools, 2025), and numeric_only = True ensures that only numerical value is considered. (StevenSwiniarski, 2023)

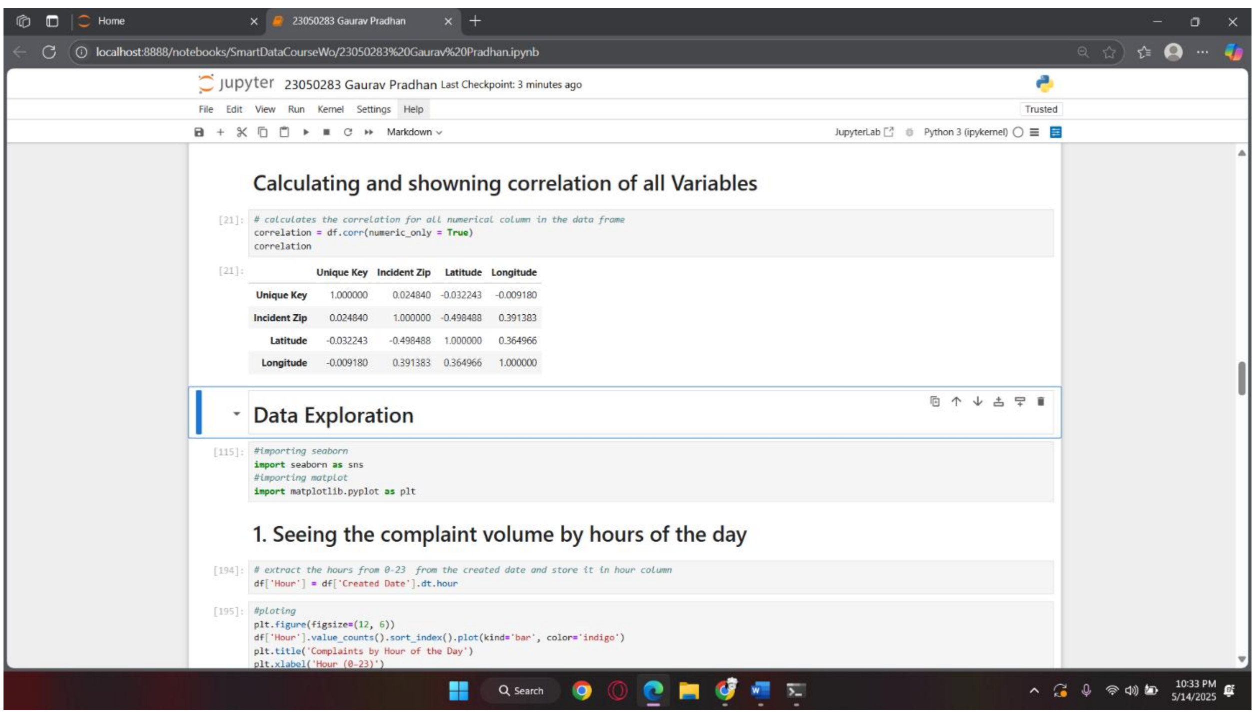


Figure 12 Calculating and showing correlation of all variables

4. Data Exploration

Data exploration is one of the beginning steps of data analysis that involves the use of data visualization and statical techniques to uncover data set. As humans are visual learners who can process visual data more easily this helps in understanding in more effective manner. (Robinson, 2025). Here are some question which requires visualization to show the data in a more understandable manner.

Q. Provide four major insights through visualization that you come up after data mining.

A. To provide four major insights through visualization which comes up after data mining I choose to show for:

4.1. Seeing the complaint volume by hours of the day

“import seaborn as sns

import matplotlib.pyplot as plt”

Before running the programs, we had to import some python libraries like seaborn which helps to show data in visual from and matplotib.pyplot which helps to create static and also visualization.

To see this bar graph the code I executed is:

“df['Created Date'] = pd.to_datetime(df['Created Date'])

df['Hour'] = df['Created Date'].dt.hour”

“plt.figure(figsize=(12, 6))

df['Hour'].value_counts().sort_index().plot(kind='bar', color='indigo') plt.title('Complaints by Hour of the Day')

plt.xlabel('Hour (0–23)')

plt.ylabel('Number of Complaints')

plt.tight_layout()”

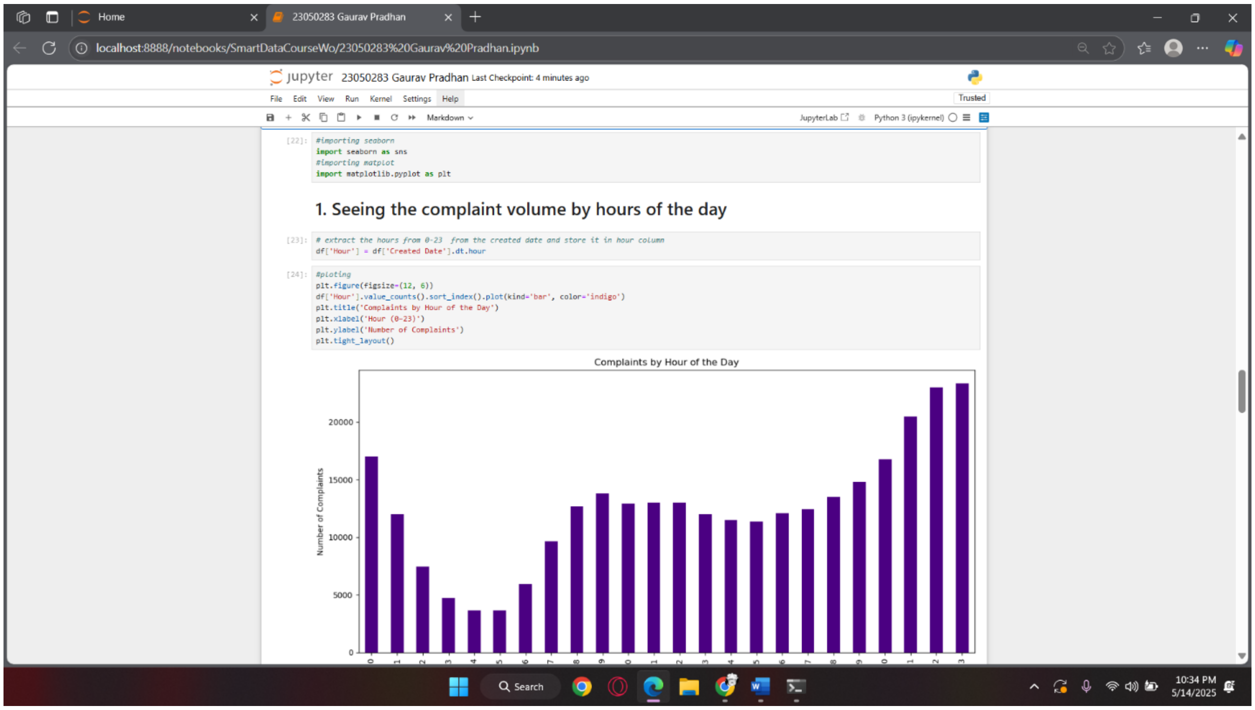


Figure 13 Seeing the complaint volume by hours

The above figure shows the frequency of complaints during a day inn hours format which shows the greatest number of complaints is done in 23 hour which can be also 12 in the morning.

The findings:

This result shows the frequency of complaint during a day.

Shows a time of complaint.

Shows the numbers of complaints.

4.2. Top ten Complaint’s type and frequency

To show the result the code I executed is:

“top_comp = df[‘Created Date’].valus_counts().head(10)

plt.figure(figsize=(12,6))

plt.bar(top_comp.index, top_comp.values, color = ‘skyblue’)

plt.title('Top ten Compliant Type and frequency')

plt.ylabel('Number of complaints')

plt.xlabel('Complaint Type')

plt.xtrick(rotation = 45)

plt.tight_layout()”

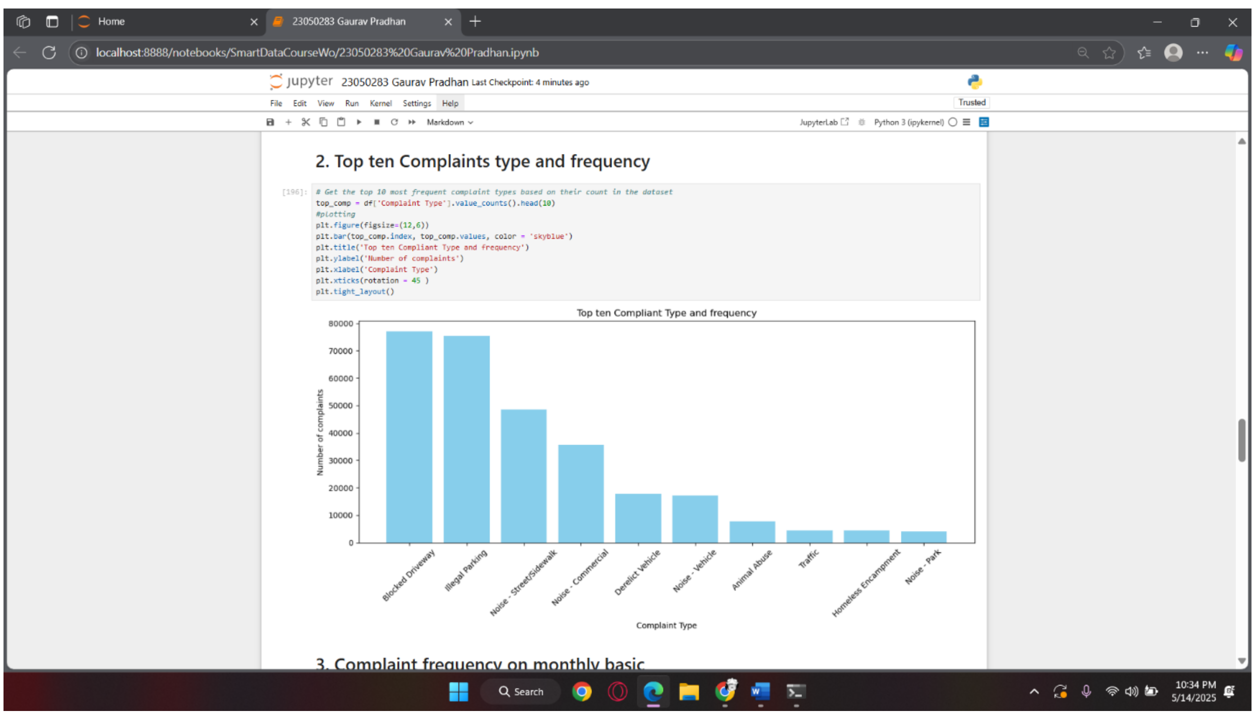


Figure 14 Top ten Complaint’s type and frequency

This code shows the visual representation of the top ten complaints with frequency also where blocked driveway having the most amount of complaint and noise- park being the list among there ten complaints.

The findings:

This bar graph shows the top ten complaints.

Here we can see the number of complaints with the most frequent and lest among these ten.

This provides a view on what is the most amount of problem faced by any citizen of the city.

4.3. Complaint frequency on monthly basic

To show the outcome for complaint frequency on monthly basis I executed this code:

This code makes sure that create date data is in the correct format and date_monthly gets the date from the month end with the help of “ME”.

“plt.figure(figsize = (12,6))

date_month.plot(kind= 'line',marker='o',color = 'green')

plt.title('Monthly Complaint frequency')

plt.xlabel('Months')

plt.ylabel('Complaint Frequency')

plt.tight_layout()

plt.show()”

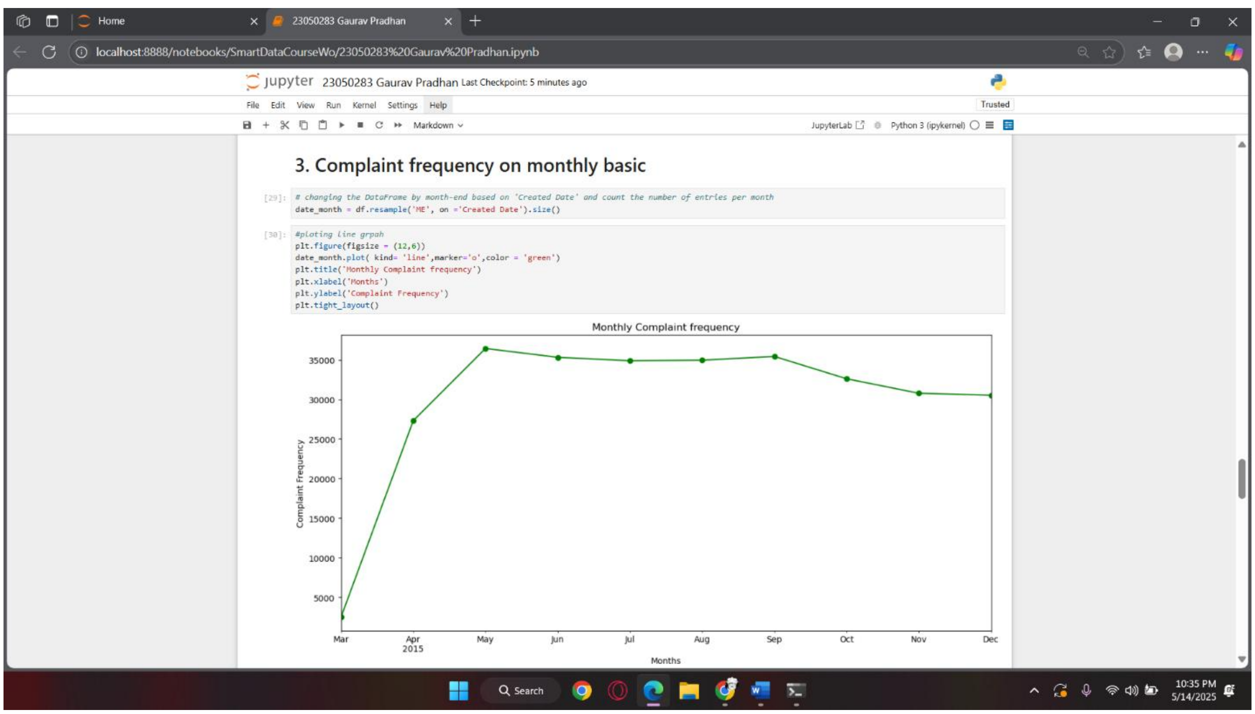


Figure 15 Complaint frequency on monthly basic

This line graph shows the frequency of complaints done with a large scale where it shows the duration in months where it shows the highest time being between May and July and lowest being march. Here the markers pinpoint every month.

The findings:

This showed the number of complaints done within a year.

Shows the month with the highest number of complaints.

Shows the month with the lowest number of complaints.

4.4. Average request closing time according to compliant

To show the average request closing time in a visual format I executed this code:

This line of code makes sure that any trailing whitespace is removed from the column and also convert the data

“df.columns = df.columns.str.strip()
df['Create Date'] = pd.to_datetime(df['Create Date'], errors = 'ignore')
df['Closed Date'] = pd.to_datetime(df['Closed Date'], errors = 'ignore')”

This following code ensures that null value is removed while calculating the difference of closing date and create date to get request closing time which is then converted to days for one day is equal to 86400 second also ensure that the data which cannot be converted will be removed to calculate the average duration.

“df['Day of Week'] = df['Created Date'].dt.day_name()”

“day_counts = df['Day of Week'].value_counts().reindex(['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'])”

“df_day_counts = pd.DataFrame({'Day': day_counts.index, 'Count': day_counts.values})
plt.figure(figsize=(12, 6))
sns.barplot(data=df_day_counts, x='Day', y='Count', hue='Day', palette="deep", legend=False)
plt.title('Number of Complaints by Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Number of Complaints')
plt.tight_layout()
plt.show()”

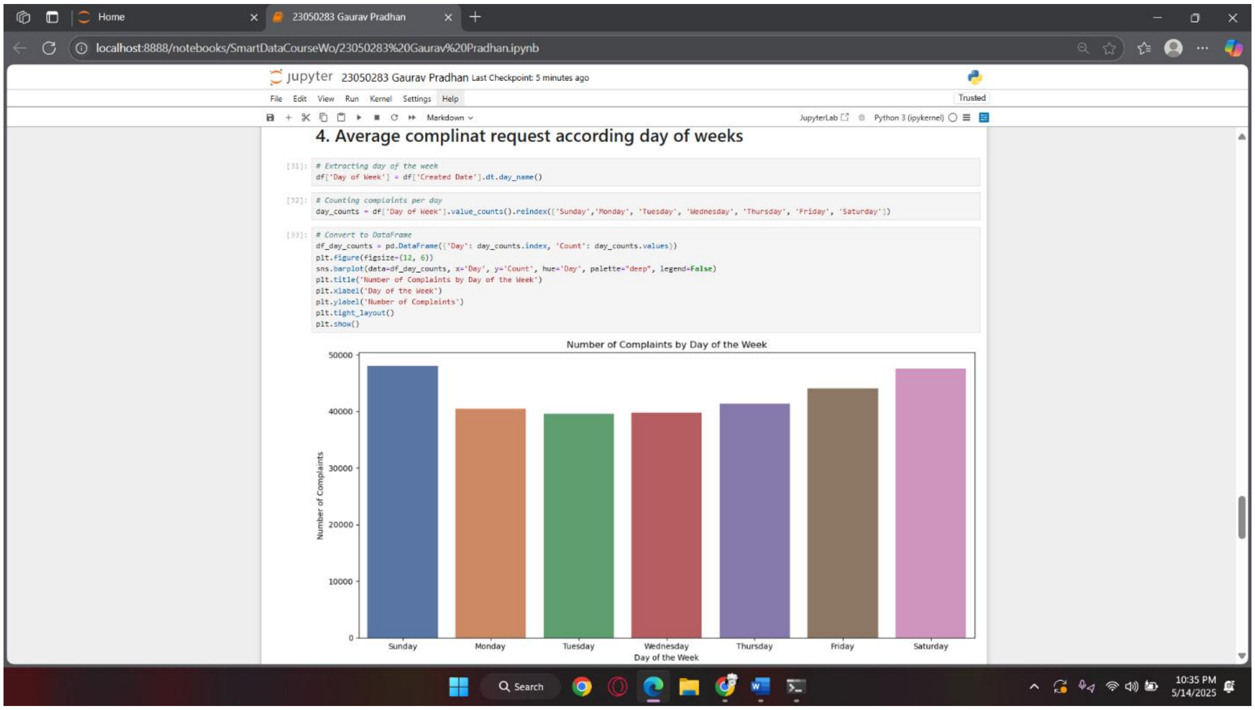


Figure 16 Average request closing time according to compliant 1

This shows the visual representation of the complaints within a week in bar graph format where it starts from Sunday and goes till Saturday and shoes that Saturday is the day where there is the greatest number of complaints and Tuesday has the least number of complaints.

The findings:

Shows the frequency of complaints within a week's duration.

Shows the highest number of complaints denoting a day.

Shows the lowest number of complaints denoting a day.

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

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

A. To show the average closing time according to complaint type with graph the code executed is:

"df['Request_Closing_Time'] = (df['Closed Date'] - df['Created Date']).dt.total_seconds() / 3600

top_5_complaints = df['Complaint Type'].value_counts().head(5).index.tolist()

filtered_df = df[df['Complaint Type'].isin(top_5_complaints)]

avg_closing_time = filtered_df.groupby(['Complaint Type', 'Location Type'])['Request_Closing_Time'].mean().reset_index()

pivot_data = avg_closing_time.pivot(index='Complaint Type', columns='Location Type', values='Request_Closing_Time')

pivot_data.plot(kind='bar', figsize=(14, 8))

plt.title('Average Request Closing Time of Top 5 Complaint Types by Location Type')

plt.ylabel('Average Request Closing Time (hours)')

plt.xlabel('Complaint Type')

plt.xticks(rotation=45, ha='right')

plt.legend(title='Location Type',loc='upper left')

plt.tight_layout()"

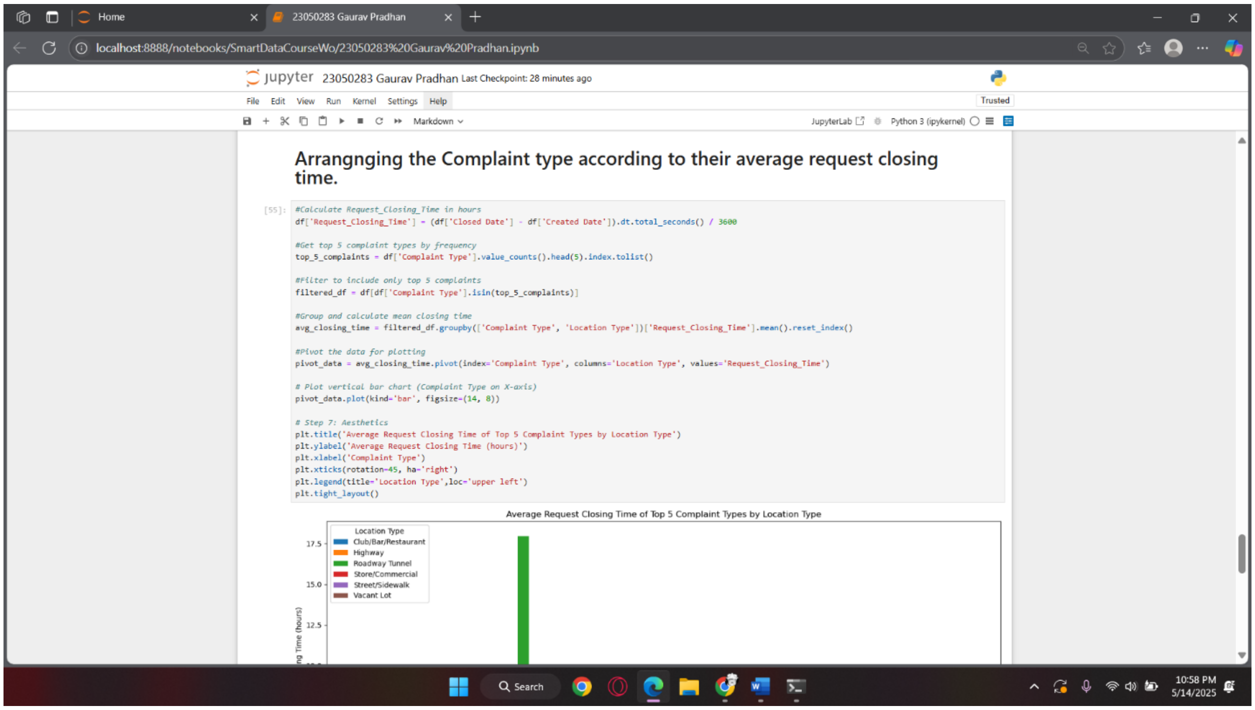


Figure 17 Average 'Request_Closing_Time', categorized

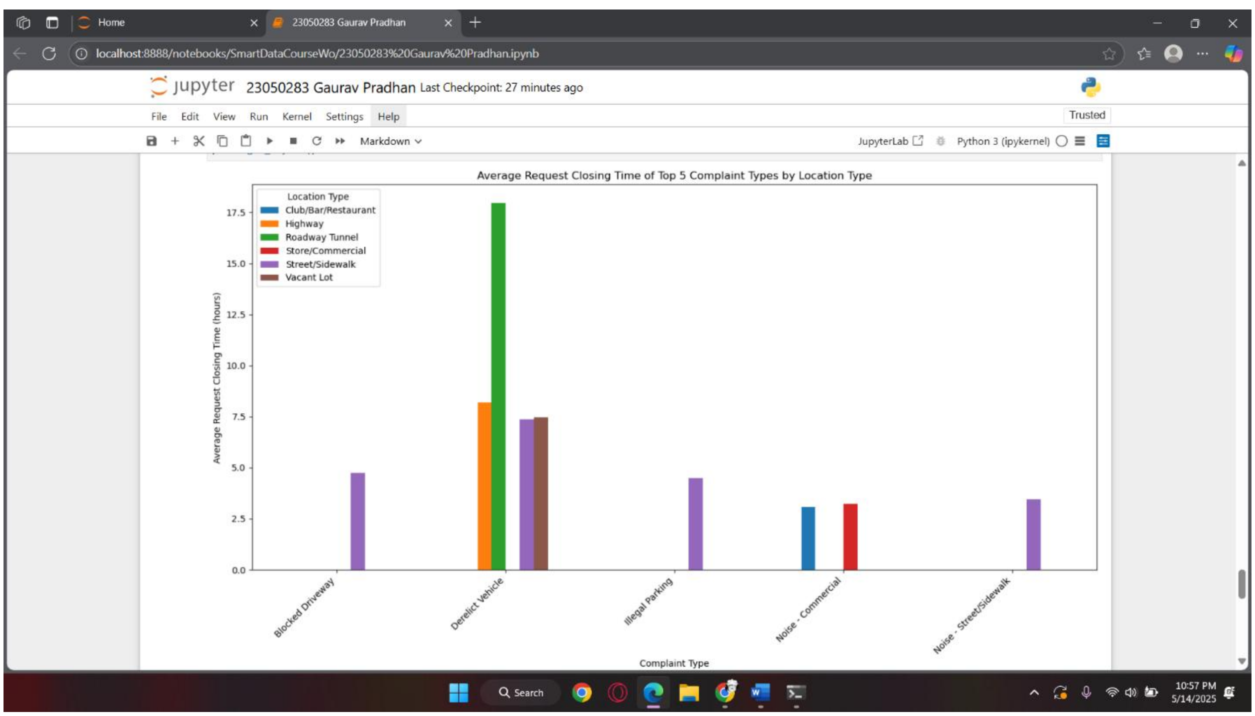


Figure 18 Result of request_closing _time

This diagram shows the average request closing time in a bar graph format which show the complaint type and location with average duration also shows.

5. Statistical Testing

Statistical testing is assumed as null hypothesis of no relationship or no difference between groups. It determines whether the observed data fall outside the range of value predicted of null hypothesis. (Bevans, 2020)

Test 1: One-way ANOVA

ANOVA, which stands for Analysis of Variance, is a type of statistical test which is used to analyse the different between the means of two or more groups. A one one-way ANOVA test is used to find the independent variable, it is used to collect data about one categorical independent variable and one quantitative dependent variable. (Bevans, 2024)

Q. Whether the average response time across complaint types is similar or not.

State the Null Hypothesis (H0) and Alternate Hypothesis (H1).

Perform the statistical test and provide the p-value.

Interpret the results to accept or reject the Null Hypothesis.

Code for testing:

“from sicpy.stats import f_oneway”

“# Filter necessary columns and drop NA

filtered_df = df[['Complaint Type', 'Request_Closing_Time']].dropna()

Top_complaints = filtered_df['Complaint Type'].value_counts().head(5).index

data = [filtered_df[filtered_df['Complaint Type'] == c]['Request_Closing_Time']for c in Top_complaints]

Perform ANOVA

f_stats, p_value = f_oneway(*data)

Display results

print("ANOVA Test:")

print('F-statical:', f_stats)

print('p-value:' ,p_value)"

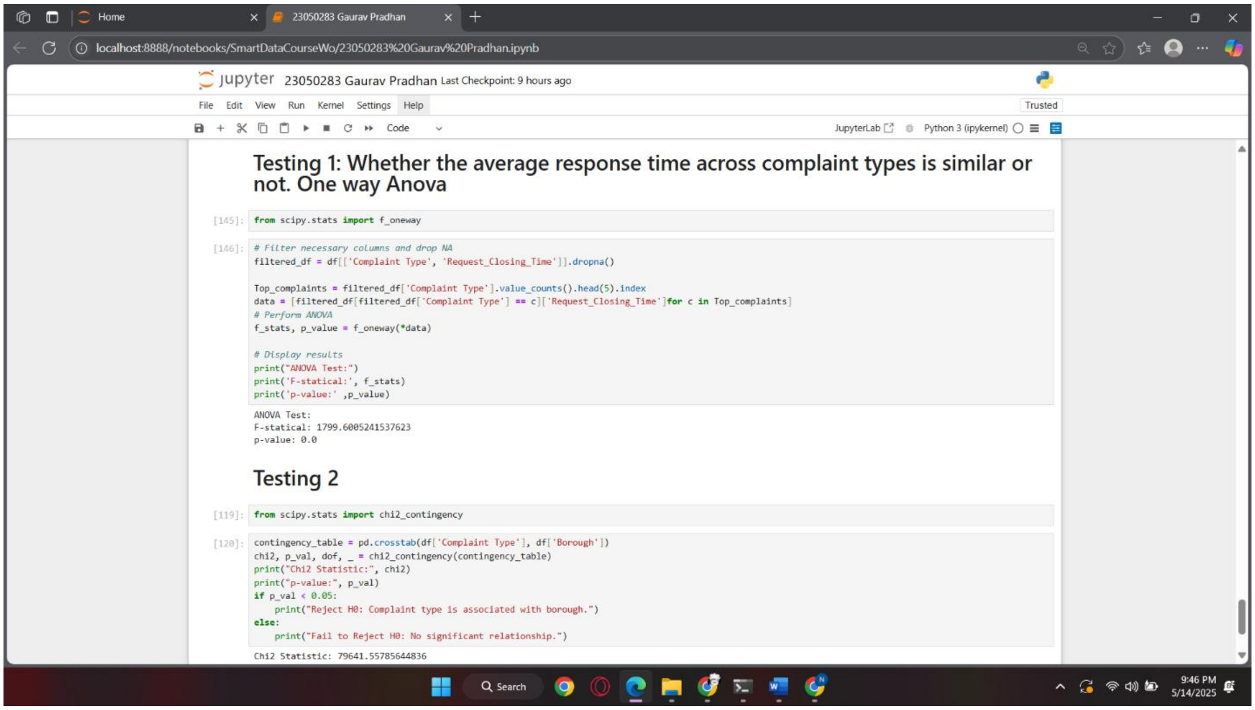


Figure 19 Statical Testing 1

This code completes the sequence to perform the statistical test and provide the p-value, with average P-value and interpret the results to accept or reject the Null Hypothesis with the proper message.

The findings:

This test shows the value of probability p- value

This shows the ration of mean sequence for groups between divided by the mean sequence within the group F-statical.

Test 2: Chi-Squared test

The chi-squared test is a statistical hypothesis test used to analyse the categorical values to determine whether observed data are different from exception it is commonly used nonparametric test which means that it doesn't assume the distribution of the data involved. (McClenaghan, 2024)

Q. Whether the type of complaint or service requested, and location are related.

State the Null Hypothesis (H0) and Alternate Hypothesis (H1).
Perform the statistical test and provide the p-value.

Interpret the results to accept or reject the Null Hypothesis.

Answer:

Stating the null hypothesis (H0): In this hypothesis there is no association between the variables.

Alternate Hypothesis(H1): In this hypothesis there is an association of any kind.

Code for testing 2:

“from scipy.stats import chi2_contingency”

“contingency_table = pd.crosstab(df['Complaint Type'], df['Borough'])

chi2, p_val, dof, _ = chi2_contingency(contingency_table)

print("Chi2 Statistic:", chi2)

print("p-value:", p_val)

if p_val < 0.05:

    print("Reject H0: Complaint type is associated with borough.")

else:

    print("Fail to Reject H0: No significant relationship.")"

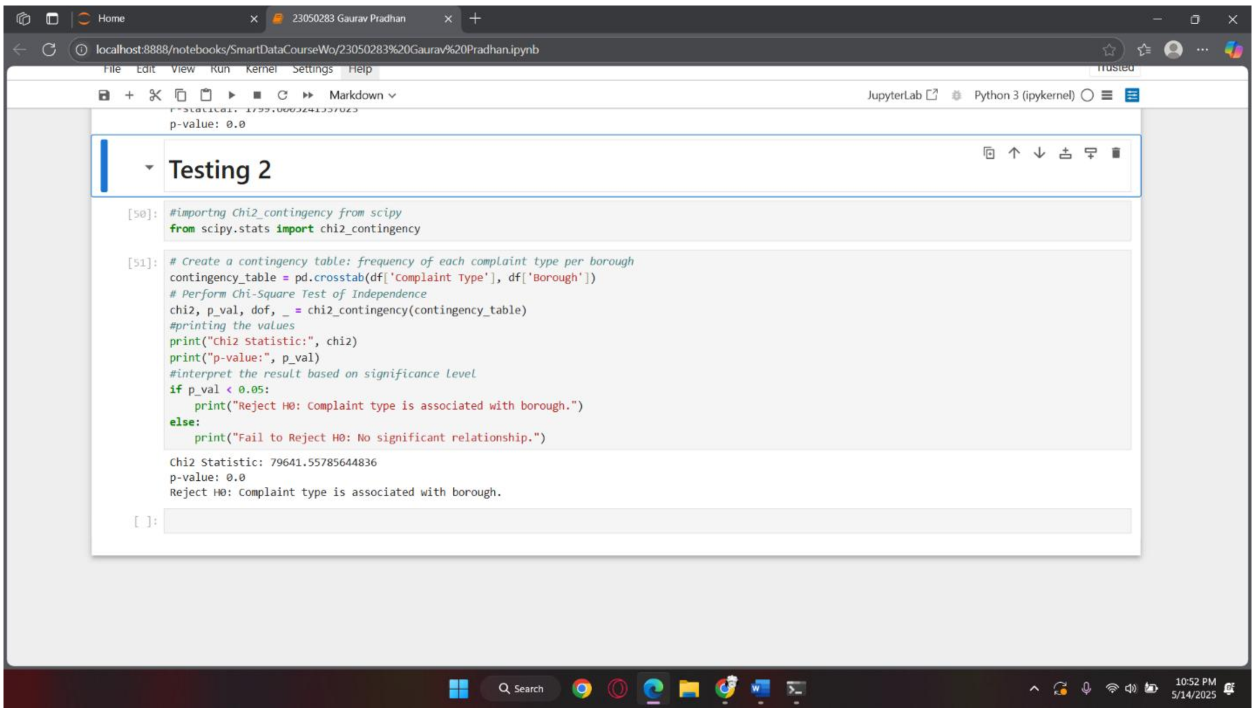


Figure 20 Statical testing 2

This code makes sure the given statements are fulfilled such as perform the statistical test and provide the p-value with the value, Interpret the results to accept or reject the Null Hypothesis.

The findings:

This test shows the value of probability p- value.

This showed the chi-2 statistic where it checks little difference between what was observed and what would be expected.

Conclusion

During the completion of this course work I got to experience how it feels to analysis a CSV file which contained the information of customer service request which was from the city of New York. With a systematic approach involving data understanding, preparation, analysis, and visualization we got serval key findings. Where the dataset included multiple columns which held information on the nature, geolocation, timing, and the resolution of certain complaints. Data quality, analysis accuracy, and general management was optimized by pre-processing steps.

These included handling missing values, dropping irrelevant fields, and converting dates, with that correlation and summary statistics helped identify various data patterns and relationship and thought how to add, update, and remove any unwanted columns, also how to create different graph with ANOVA testing and chi-square testing. This provided an importance of effective data cleaning and exploration in a large-scale and meaning full pattern.

6. References

Bevans, R., 2020. *www.scribbr.com.* [Online] Available at: https://www.scribbr.com/statistics/statistical-tests/
[Accessed 03 05 2025].

Bevans, R., 2024. *https://www.scribbr.com/.* [Online] Available at: https://www.scribbr.com/statistics/one-way-anova/
[Accessed 10 05 2025].

coskun, O., 2023. *https://medium.com/.* [Online] Available at: https://medium.com/@onurcoskun1983/why-is-data-understanding-is-crucial-step-in-data-analytics-c41a898792be#:~:text=%2D%20It%20enables%20correct%20decisions%20to,before%20and%20during%20data%20analysis.
[Accessed 12 4 2025].

geeksforgeeks.org, 2024. *https://www.geeksforgeeks.org/.* [Online] Available at: https://www.geeksforgeeks.org/python-read-csv-using-pandas-read_csv/
[Accessed 28 04 2025].

geeksforgeeks.org, 2025. *https://www.geeksforgeeks.org/.* [Online] Available at: https://www.geeksforgeeks.org/pandas-read_csv-low_memory-and-dtype-options/
[Accessed 01 05 2025].

Harris, S., 2021. *https://medium.com/.* [Online] Available at: https://medium.com/data-science/why-you-should-probably-never-use-pandas-inplace-true-9f9f211849e4#:~:text=Using%20the%20inplace%3DTrue%20keyword,(more%20on%20that%20later).
[Accessed 12 4 2025].

McClenaghan, E., 2024. *https://www.technologynetworks.com/.* [Online] Available at: https://www.technologynetworks.com/informatics/articles/the-chi-squared-test-

368882#:~:text=The%20null%20hypothesis%20(H0)%20is,an%20association%20of%20any%20kind.
[Accessed 09 05 2025].

Robinson, S., 2025. *https://www.techtarget.com/.* [Online] Available at: https://www.techtarget.com/searchbusinessanalytics/definition/data-exploration
[Accessed 03 05 2025].

sciencedirect, 2025. *https://www.sciencedirect.com/.* [Online] Available at: https://www.sciencedirect.com/topics/neuroscience/kurtosis#:~:text=Kurtosis%20is%20a%20statistical%20measure,Statistics%20(Second%20Edition)%2C%202018
[Accessed 13 05 2025].

StevenSwiniarski, 2023. *www.codecademy.co.* [Online] Available at: https://www.codecademy.com/resources/docs/pandas/groupby/sum
[Accessed 12 4 2025].

Turney, S., 2022. *https://www.scribbr.com/.* [Online] Available at: https://www.scribbr.com/statistics/skewness/#:~:text=Skewness%20is%20a%20measure%20of,negative)%2C%20or%20zero%20skewness.
[Accessed 13 05 2025].

w3schools, 2025. *https://www.w3schools.com/.* [Online] Available at: https://www.w3schools.com/python/pandas/pandas_correlations.asp#:~:text=A%20great%20aspect%20of%20the,%3A%20'data.csv'.
[Accessed 12 4 2025].