In [2]:
```
import pandas as pd
import requests
import numpy as np
import matplotlib.pyplot as plt
```

```
/Users/sanhitha/Library/Python/3.9/lib/python/site-packages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports OpenSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. See: https://github.com/urllib3/urllib3/issues/3020
  warnings.warn(
```

In [3]:
```
pd.set_option('display.max_columns',35)
pd.set_option('display.max_rows',35)
```

In [4]:
```
limit=1000
offset=0
params={
    '$limit':limit,
    '$offset':offset
}
```

In [5]:
```
url='https://data.buffalony.gov/resource/d6g9-xbgu.json'
```

In [6]:
```
df_list=[]
while True:
    params={
        '$limit':limit,
        '$offset':offset
    }
    response=requests.get(url,params=params)
    data=response.json()
    df_page=pd.DataFrame(data)
    if df_page.empty:
        break
    df_list.append(df_page)
    offset+=limit
df=pd.concat(df_list,ignore_index=True)
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In[6], line 7
      2 while True:
      3     params={
      4         '$limit':limit,
      5         '$offset':offset
      6     }
----> 7     response=requests.get(url,params=params)
      8     data=response.json()
      9     df_page=pd.DataFrame(data)

File ~/Library/Python/3.9/lib/python/site-packages/requests/api.py:73, in get(url, pa
rams, **kwargs)
     62 def get(url, params=None, **kwargs):
     63     r"""Sends a GET request.
     64
     65     :param url: URL for the new :class:`Request` object.
   (...)
     70     :rtype: requests.Response
     71     """
---> 73     return request("get", url, params=params, **kwargs)

File ~/Library/Python/3.9/lib/python/site-packages/requests/api.py:59, in request(met
hod, url, **kwargs)
     55 # By using the 'with' statement we are sure the session is closed, thus we
     56 # avoid leaving sockets open which can trigger a ResourceWarning in some
     57 # cases, and look like a memory leak in others.
     58 with sessions.Session() as session:
---> 59     return session.request(method=method, url=url, **kwargs)

File ~/Library/Python/3.9/lib/python/site-packages/requests/sessions.py:589, in Sessi
on.request(self, method, url, params, data, headers, cookies, files, auth, timeout, a
llow_redirects, proxies, hooks, stream, verify, cert, json)
    584 send_kwargs = {
    585     "timeout": timeout,
    586     "allow_redirects": allow_redirects,
    587 }
    588 send_kwargs.update(settings)
--> 589 resp = self.send(prep, **send_kwargs)
    591 return resp

File ~/Library/Python/3.9/lib/python/site-packages/requests/sessions.py:703, in Sessi
on.send(self, request, **kwargs)
    700 start = preferred_clock()
    702 # Send the request
--> 703 r = adapter.send(request, **kwargs)
    705 # Total elapsed time of the request (approximately)
    706 elapsed = preferred_clock() - start

File ~/Library/Python/3.9/lib/python/site-packages/requests/adapters.py:667, in HTTPA
dapter.send(self, request, stream, timeout, verify, cert, proxies)
    664     timeout = TimeoutSauce(connect=timeout, read=timeout)
    666 try:
--> 667     resp = conn.urlopen(
    668         method=request.method,
    669         url=url,
    670         body=request.body,
    671         headers=request.headers,
    672         redirect=False,
```

```
    673         assert_same_host=False,
    674         preload_content=False,
    675         decode_content=False,
    676         retries=self.max_retries,
    677         timeout=timeout,
    678         chunked=chunked,
    679     )
    681 except (ProtocolError, OSError) as err:
    682     raise ConnectionError(err, request=request)

File ~/Library/Python/3.9/lib/python/site-packages/urllib3/connectionpool.py:789, in
HTTPConnectionPool.urlopen(self, method, url, body, headers, retries, redirect, asser
t_same_host, timeout, pool_timeout, release_conn, chunked, body_pos, preload_content,
decode_content, **response_kw)
    786 response_conn = conn if not release_conn else None
    788 # Make the request on the HTTPConnection object
--> 789 response = self._make_request(
    790     conn,
    791     method,
    792     url,
    793     timeout=timeout_obj,
    794     body=body,
    795     headers=headers,
    796     chunked=chunked,
    797     retries=retries,
    798     response_conn=response_conn,
    799     preload_content=preload_content,
    800     decode_content=decode_content,
    801     **response_kw,
    802 )
    804 # Everything went great!
    805 clean_exit = True

File ~/Library/Python/3.9/lib/python/site-packages/urllib3/connectionpool.py:466, in
HTTPConnectionPool._make_request(self, conn, method, url, body, headers, retries, tim
eout, chunked, response_conn, preload_content, decode_content, enforce_content_lengt
h)
    463 try:
    464     # Trigger any extra validation we need to do.
    465     try:
--> 466         self._validate_conn(conn)
    467     except (SocketTimeout, BaseSSLError) as e:
    468         self._raise_timeout(err=e, url=url, timeout_value=conn.timeout)

File ~/Library/Python/3.9/lib/python/site-packages/urllib3/connectionpool.py:1095, in
HTTPSConnectionPool._validate_conn(self, conn)
   1093 # Force connect early to allow us to validate the connection.
   1094 if conn.is_closed:
-> 1095     conn.connect()
   1097 # TODO revise this, see https://github.com/urllib3/urllib3/issues/2791
   1098 if not conn.is_verified and not conn.proxy_is_verified:

File ~/Library/Python/3.9/lib/python/site-packages/urllib3/connection.py:730, in HTTP
SConnection.connect(self)
    727     # Remove trailing '.' from fqdn hostnames to allow certificate validation
    728     server_hostname_rm_dot = server_hostname.rstrip(".")
--> 730     sock_and_verified = _ssl_wrap_socket_and_match_hostname(
    731         sock=sock,
    732         cert_reqs=self.cert_reqs,
    733         ssl_version=self.ssl_version,
```

```
    734         ssl_minimum_version=self.ssl_minimum_version,
    735         ssl_maximum_version=self.ssl_maximum_version,
    736         ca_certs=self.ca_certs,
    737         ca_cert_dir=self.ca_cert_dir,
    738         ca_cert_data=self.ca_cert_data,
    739         cert_file=self.cert_file,
    740         key_file=self.key_file,
    741         key_password=self.key_password,
    742         server_hostname=server_hostname_rm_dot,
    743         ssl_context=self.ssl_context,
    744         tls_in_tls=tls_in_tls,
    745         assert_hostname=self.assert_hostname,
    746         assert_fingerprint=self.assert_fingerprint,
    747     )
    748     self.sock = sock_and_verified.socket
    750 # If an error occurs during connection/handshake we may need to release
    751 # our lock so another connection can probe the origin.

File ~/Library/Python/3.9/lib/python/site-packages/urllib3/connection.py:909, in _ssl
_wrap_socket_and_match_hostname(sock, cert_reqs, ssl_version, ssl_minimum_version, ss
l_maximum_version, cert_file, key_file, key_password, ca_certs, ca_cert_dir, ca_cert_
data, assert_hostname, assert_fingerprint, server_hostname, ssl_context, tls_in_tls)
    906     if is_ipaddress(normalized):
    907         server_hostname = normalized
--> 909 ssl_sock = ssl_wrap_socket(
    910     sock=sock,
    911     keyfile=key_file,
    912     certfile=cert_file,
    913     key_password=key_password,
    914     ca_certs=ca_certs,
    915     ca_cert_dir=ca_cert_dir,
    916     ca_cert_data=ca_cert_data,
    917     server_hostname=server_hostname,
    918     ssl_context=context,
    919     tls_in_tls=tls_in_tls,
    920 )
    922 try:
    923     if assert_fingerprint:

File ~/Library/Python/3.9/lib/python/site-packages/urllib3/util/ssl_.py:469, in ssl_w
rap_socket(sock, keyfile, certfile, cert_reqs, ca_certs, server_hostname, ssl_versio
n, ciphers, ssl_context, ca_cert_dir, key_password, ca_cert_data, tls_in_tls)
    465         context.load_cert_chain(certfile, keyfile, key_password)
    467 context.set_alpn_protocols(ALPN_PROTOCOLS)
--> 469 ssl_sock = _ssl_wrap_socket_impl(sock, context, tls_in_tls, server_hostname)
    470 return ssl_sock

File ~/Library/Python/3.9/lib/python/site-packages/urllib3/util/ssl_.py:513, in _ssl_
wrap_socket_impl(sock, ssl_context, tls_in_tls, server_hostname)
    510     SSLTransport._validate_ssl_context_for_tls_in_tls(ssl_context)
    511     return SSLTransport(sock, ssl_context, server_hostname)
--> 513 return ssl_context.wrap_socket(sock, server_hostname=server_hostname)

File /Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/Version
s/3.9/lib/python3.9/ssl.py:500, in SSLContext.wrap_socket(self, sock, server_side, do
_handshake_on_connect, suppress_ragged_eofs, server_hostname, session)
    494 def wrap_socket(self, sock, server_side=False,
    495                  do_handshake_on_connect=True,
    496                  suppress_ragged_eofs=True,
    497                  server_hostname=None, session=None):
```

```
    498     # SSLSocket class handles server_hostname encoding before it calls
    499     # ctx._wrap_socket()
--> 500     return self.sslsocket_class._create(
    501         sock=sock,
    502         server_side=server_side,
    503         do_handshake_on_connect=do_handshake_on_connect,
    504         suppress_ragged_eofs=suppress_ragged_eofs,
    505         server_hostname=server_hostname,
    506         context=self,
    507         session=session
    508     )

File /Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/Versions/3.9/lib/python3.9/ssl.py:1040, in SSLSocket._create(cls, sock, server_side, do_handshake_on_connect, suppress_ragged_eofs, server_hostname, context, session)
   1037         if timeout == 0.0:
   1038             # non-blocking
   1039             raise ValueError("do_handshake_on_connect should not be specified for non-blocking sockets")
-> 1040         self.do_handshake()
   1041 except (OSError, ValueError):
   1042     self.close()

File /Library/Developer/CommandLineTools/Library/Frameworks/Python3.framework/Versions/3.9/lib/python3.9/ssl.py:1309, in SSLSocket.do_handshake(self, block)
   1307     if timeout == 0.0 and block:
   1308         self.settimeout(None)
-> 1309     self._sslobj.do_handshake()
   1310 finally:
   1311     self.settimeout(timeout)

KeyboardInterrupt:
```

In [7]:
```
df
```

Out[7]:

	case_number	incident_datetime	incident_type_primary	incident_description	parent_incident_t
0	16-1660403	2016-06-14T01:20:00.000	ASSAULT	ASSAULT	Ass
1	16-3480266	2016-12-13T05:00:00.000	LARCENY/THEFT	LARCENY/THEFT	T
2	20-2010167	2020-07-19T03:09:00.000	ASSAULT	Buffalo Police are investigating this report o...	Ass
3	14-3210732	2014-11-17T08:08:00.000	LARCENY/THEFT	LARCENY/THEFT	T
4	15-1100268	2015-04-20T10:22:00.000	LARCENY/THEFT	LARCENY/THEFT	T
...	...	...	...	...	
315863	24-2600569	2024-09-16T12:39:48.000	LARCENY/THEFT	Buffalo Police are investigating this report o...	T
315864	24-2551037	2024-09-11T19:43:44.000	UUV	Buffalo Police are investigating this report o...	Theft of Veh
315865	24-2900880	2024-10-16T18:30:00.000	ROBBERY	Buffalo Police are investigating this report o...	Robb
315866	24-2550623	2024-09-11T13:58:19.000	LARCENY/THEFT	Buffalo Police are investigating this report o...	T
315867	24-3070416	2024-10-31T16:00:00.000	BURGLARY	Buffalo Police are investigating this report o...	Breaking & Ente

315868 rows × 35 columns

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 315868 entries, 0 to 315867
Data columns (total 35 columns):
 #   Column                       Non-Null Count   Dtype 
---  ------                       --------------   ----- 
 0   case_number                  315868 non-null  object
 1   incident_datetime            315868 non-null  object
 2   incident_type_primary        315868 non-null  object
 3   incident_description         315868 non-null  object
 4   parent_incident_type         315868 non-null  object
 5   hour_of_day                  315868 non-null  object
 6   day_of_week                  315868 non-null  object
 7   address_1                    315834 non-null  object
 8   city                         315868 non-null  object
 9   state                        315868 non-null  object
 10  location                     309589 non-null  object
 11  latitude                     314784 non-null  object
 12  longitude                    314784 non-null  object
 13  zip_code                     313445 non-null  object
 14  neighborhood                 312493 non-null  object
 15  council_district             313394 non-null  object
 16  council_district_2011        313445 non-null  object
 17  census_tract                 312493 non-null  object
 18  census_block_group           312493 non-null  object
 19  census_block                 312493 non-null  object
 20  census_tract_2010            312493 non-null  object
 21  census_block_group_2010      312493 non-null  object
 22  census_block_2010            312493 non-null  object
 23  police_district              312493 non-null  object
 24  tractce20                    312632 non-null  object
 25  geoid20_tract                312632 non-null  object
 26  geoid20_blockgroup           312632 non-null  object
 27  geoid20_block                312632 non-null  object
 28  :@computed_region_jdfw_hhbp  308270 non-null  object
 29  :@computed_region_h7a8_iwt4  308236 non-null  object
 30  :@computed_region_ff6v_jbaa  308280 non-null  object
 31  :@computed_region_vsen_jbmg  308072 non-null  object
 32  :@computed_region_nmyf_6jtp  308482 non-null  object
 33  :@computed_region_yg52_574g  308130 non-null  object
 34  created_at                   72642 non-null   object
dtypes: object(35)
memory usage: 84.3+ MB
```

In [9]:
```
df['incident_datetime']=pd.to_datetime(df['incident_datetime'],format='%Y-%m-%dT%H:%M:
```

In [10]:
```
df['incident_description'].value_counts()
```

Out[10]:
```
incident_description
Buffalo Police are investigating this report of a crime.  It is important to note tha
t this is very preliminary information and further investigation as to the facts and
circumstances of this report may be necessary.    306941
Buffalo Police are investigating this report of a crime. It is important to note that
this is very preliminary information and further investigation as to the facts and ci
rcumstances of this report may be necessary.       5175
LARCENY/THEFT
1811
BURGLARY
960
ASSAULT
603
SEXUAL ABUSE
122
UUV
103
RAPE
78
ROBBERY
34
CRIM NEGLIGENT HOMICIDE
20
THEFT OF SERVICES
13
AGG ASSAULT ON P/OFFICER
4
MURDER
2
AGGR ASSAULT
2
Name: count, dtype: int64
```

In [11]:
```
df['incident_description'] = df['incident_description'].str.replace(r'\s+', ' ', regex
```

In [12]:
```
df['incident_description']=df['incident_description'].str.replace('Buffalo Police are
```

In [13]:
```
df['incident_description']=df['incident_description'].str.replace('Buffalo Police are
```

In [14]:
```
df['incident_description'].value_counts()
```

Out[14]:
```
incident_description
under investigation         312116
LARCENY/THEFT                 1811
BURGLARY                       960
ASSAULT                        603
SEXUAL ABUSE                   122
UUV                            103
RAPE                            78
ROBBERY                         34
CRIM NEGLIGENT HOMICIDE         20
THEFT OF SERVICES               13
AGG ASSAULT ON P/OFFICER         4
MURDER                           2
AGGR ASSAULT                     2
Name: count, dtype: int64
```

In [16]:
```
df.isnull().sum()
```

Out[16]:

```
case_number                    0
incident_datetime              0
incident_type_primary          0
incident_description           0
parent_incident_type           0
hour_of_day                    0
day_of_week                    0
address_1                     34
city                           0
state                          0
location                    6279
latitude                    1084
longitude                   1084
zip_code                    2423
neighborhood                3375
council_district            2474
council_district_2011       2423
census_tract                3375
census_block_group          3375
census_block                3375
census_tract_2010           3375
census_block_group_2010     3375
census_block_2010           3375
police_district             3375
tractce20                   3236
geoid20_tract               3236
geoid20_blockgroup          3236
geoid20_block               3236
:@computed_region_jdfw_hhbp    7598
:@computed_region_h7a8_iwt4    7632
:@computed_region_ff6v_jbaa    7588
:@computed_region_vsen_jbmg    7796
:@computed_region_nmyf_6jtp    7386
:@computed_region_yg52_574g    7738
created_at                  243226
dtype: int64
```

In [17]:
```
df=df.replace('UNKNOWN',np.nan)
```

In [18]:
```
df.isnull().sum()
```

Out[18]:
```
case_number                    0
incident_datetime              0
incident_type_primary          0
incident_description           0
parent_incident_type           0
hour_of_day                    0
day_of_week                    0
address_1                     51
city                           0
state                          0
location                    6279
latitude                    6279
longitude                   6279
zip_code                    3687
neighborhood                6401
council_district            2474
council_district_2011       3781
census_tract                6281
census_block_group          6281
census_block                6281
census_tract_2010          19637
census_block_group_2010    19688
census_block_2010          19638
police_district             6287
tractce20                   6281
geoid20_tract               6281
geoid20_blockgroup          6281
geoid20_block               6281
:@computed_region_jdfw_hhbp    7598
:@computed_region_h7a8_iwt4    7632
:@computed_region_ff6v_jbaa    7588
:@computed_region_vsen_jbmg    7796
:@computed_region_nmyf_6jtp    7386
:@computed_region_yg52_574g    7738
created_at                  243226
dtype: int64
```

In [19]:
```
df=df.sort_values(by='incident_datetime')
```

In [20]:
```
# df.to_csv('crime_dataset_buffalo.csv')
```

In [21]:
```
df['hour_of_day']=pd.to_datetime(df['hour_of_day'],format='%H')
df['hour_of_day']=df['hour_of_day'].dt.hour
```

In [22]:
```
df.isnull().sum()
```

Out[22]:
```
case_number                     0
incident_datetime               0
incident_type_primary           0
incident_description            0
parent_incident_type            0
hour_of_day                     0
day_of_week                     0
address_1                      51
city                            0
state                           0
location                     6279
latitude                     6279
longitude                    6279
zip_code                     3687
neighborhood                 6401
council_district             2474
council_district_2011        3781
census_tract                 6281
census_block_group           6281
census_block                 6281
census_tract_2010           19637
census_block_group_2010     19688
census_block_2010           19638
police_district              6287
tractce20                    6281
geoid20_tract                6281
geoid20_blockgroup           6281
geoid20_block                6281
:@computed_region_jdfw_hhbp  7598
:@computed_region_h7a8_iwt4  7632
:@computed_region_ff6v_jbaa  7588
:@computed_region_vsen_jbmg  7796
:@computed_region_nmyf_6jtp  7386
:@computed_region_yg52_574g  7738
created_at                 243226
dtype: int64
```

In [23]:
```
df=df[df['incident_datetime']>='2009']
```

In [24]:
```
df
```

Out[24]:

	case_number	incident_datetime	incident_type_primary	incident_description	parent_incident_t
109038	09-1740322	2009-01-01 00:00:00	LARCENY/THEFT	under investigation	T
110550	09-1210237	2009-01-01 00:00:00	LARCENY/THEFT	under investigation	T
28850	09-3010604	2009-01-01 00:00:00	RAPE	under investigation	Sexual Ass
71497	09-0830328	2009-01-01 00:00:00	SEXUAL ABUSE	under investigation	Other Sexual Offe
88235	09-1060143	2009-01-01 00:00:00	LARCENY/THEFT	under investigation	T
...	...	...	...	...	
315759	24-3090156	2024-11-04 07:03:51	ASSAULT	under investigation	Ass
313387	24-3090236	2024-11-04 08:33:26	LARCENY/THEFT	under investigation	T
315752	24-3090255	2024-11-04 08:58:45	LARCENY/THEFT	under investigation	T
315414	24-3090287	2024-11-04 09:27:27	LARCENY/THEFT	under investigation	T
315850	24-3090308	2024-11-04 09:40:00	LARCENY/THEFT	under investigation	T

260207 rows × 35 columns

In [25]:
```
df.isnull().sum()
```

Out[25]:
```
case_number                    0
incident_datetime              0
incident_type_primary          0
incident_description           0
parent_incident_type           0
hour_of_day                    0
day_of_week                    0
address_1                     34
city                           0
state                          0
location                    6027
latitude                    6027
longitude                   6027
zip_code                    3333
neighborhood                6023
council_district            2351
council_district_2011       3415
census_tract                5938
census_block_group          5938
census_block                5938
census_tract_2010          19278
census_block_group_2010    19316
census_block_2010          19279
police_district             5952
tractce20                   5938
geoid20_tract               5938
geoid20_blockgroup          5938
geoid20_block               5938
:@computed_region_jdfw_hhbp    7108
:@computed_region_h7a8_iwt4    7130
:@computed_region_ff6v_jbaa    7100
:@computed_region_vsen_jbmg    7281
:@computed_region_nmyf_6jtp    6884
:@computed_region_yg52_574g    7205
created_at                  187652
dtype: int64
```

In [26]:
```
df=df.reset_index(drop=True)
```

In [27]:
```
df['incident_type_primary']=df['incident_type_primary'].str.lower()
```

In [28]:
```
df['incident_type_primary'].value_counts()
```

Out[28]:
```
incident_type_primary
larceny/theft                113930
assault                       52731
burglary                      45211
uuv                           25120
robbery                       15699
rape                           2361
sexual abuse                   2234
theft of services              1795
murder                          830
breaking & entering              81
aggr assault                     75
crim negligent homicide          61
theft                            34
manslaughter                     19
agg assault on p/officer         13
sexual assault                    6
theft of vehicle                  4
other sexual offense              2
sodomy                            1
Name: count, dtype: int64
```

In [29]:
```
df.columns
```

Out[29]:
```
Index(['case_number', 'incident_datetime', 'incident_type_primary',
       'incident_description', 'parent_incident_type', 'hour_of_day',
       'day_of_week', 'address_1', 'city', 'state', 'location', 'latitude',
       'longitude', 'zip_code', 'neighborhood', 'council_district',
       'council_district_2011', 'census_tract', 'census_block_group',
       'census_block', 'census_tract_2010', 'census_block_group_2010',
       'census_block_2010', 'police_district', 'tractce20', 'geoid20_tract',
       'geoid20_blockgroup', 'geoid20_block', ':@computed_region_jdfw_hhbp',
       ':@computed_region_h7a8_iwt4', ':@computed_region_ff6v_jbaa',
       ':@computed_region_vsen_jbmg', ':@computed_region_nmyf_6jtp',
       ':@computed_region_yg52_574g', 'created_at'],
      dtype='object')
```

In [30]:
```
df['latitude']=df['latitude'].astype('float64')
df['longitude']=df['longitude'].astype('float64')
```

In [31]:
```
df.isnull().sum()
```

Out[31]:
```
case_number                    0
incident_datetime              0
incident_type_primary          0
incident_description           0
parent_incident_type           0
hour_of_day                    0
day_of_week                    0
address_1                     34
city                           0
state                          0
location                    6027
latitude                    6027
longitude                   6027
zip_code                    3333
neighborhood                6023
council_district            2351
council_district_2011       3415
census_tract                5938
census_block_group          5938
census_block                5938
census_tract_2010          19278
census_block_group_2010    19316
census_block_2010          19279
police_district             5952
tractce20                   5938
geoid20_tract               5938
geoid20_blockgroup          5938
geoid20_block               5938
:@computed_region_jdfw_hhbp    7108
:@computed_region_h7a8_iwt4    7130
:@computed_region_ff6v_jbaa    7100
:@computed_region_vsen_jbmg    7281
:@computed_region_nmyf_6jtp    6884
:@computed_region_yg52_574g    7205
created_at                  187652
dtype: int64
```

In [32]:
```
df_filtered=df.drop(columns=['created_at'])
```

In [33]:
```
df_filtered.dropna(axis='index',inplace=True)
```

In [34]:
```
# df_filtered.to_csv('filtered_data.csv')
```

In [35]:
```
df_filtered['address_1']=df_filtered['address_1'].str.lower()
```

In [36]:
```
df['incident_type_primary'].unique()
```

Out[36]:
```
array(['larceny/theft', 'rape', 'sexual abuse', 'burglary', 'uuv',
       'theft of services', 'assault', 'robbery', 'murder',
       'manslaughter', 'theft', 'theft of vehicle', 'breaking & entering',
       'sexual assault', 'other sexual offense', 'aggr assault',
       'agg assault on p/officer', 'crim negligent homicide', 'sodomy'],
      dtype=object)
```

In [37]:
```
df_filtered.isnull().sum()
```

Out[37]:
```
case_number                 0
incident_datetime           0
incident_type_primary       0
incident_description        0
parent_incident_type        0
hour_of_day                 0
day_of_week                 0
address_1                   0
city                        0
state                       0
location                    0
latitude                    0
longitude                   0
zip_code                    0
neighborhood                0
council_district            0
council_district_2011       0
census_tract                0
census_block_group          0
census_block                0
census_tract_2010           0
census_block_group_2010     0
census_block_2010           0
police_district             0
tractce20                   0
geoid20_tract               0
geoid20_blockgroup          0
geoid20_block               0
:@computed_region_jdfw_hhbp 0
:@computed_region_h7a8_iwt4 0
:@computed_region_ff6v_jbaa 0
:@computed_region_vsen_jbmg 0
:@computed_region_nmyf_6jtp 0
:@computed_region_yg52_574g 0
dtype: int64
```

In [38]:
```
# Convert dicts to strings
df_filtered['location'] = df_filtered['location'].apply(lambda x: str(x) if isinstance
```

In [39]:
```
duplicate_rows_df = df_filtered[df_filtered.duplicated()]
print("number of duplicate rows: ", duplicate_rows_df)
```

```
number of duplicate rows:         case_number   incident_datetime incident_type_prima
ry  \
5       09-0830328 2009-01-01 00:00:00          sexual abuse
8       09-1320602 2009-01-01 00:00:00                  rape
22      09-0640645 2009-01-01 00:00:00         larceny/theft
25      09-1740322 2009-01-01 00:00:00         larceny/theft
29      17-2770266 2009-01-01 00:01:00          sexual abuse
...            ...                 ...                   ...
260176  24-3080099 2024-11-03 01:00:00         larceny/theft
260181  24-3080156 2024-11-03 02:52:58              burglary
260183  24-3080255 2024-11-03 06:52:49                   uuv
260185  24-3080296 2024-11-03 08:03:06                   uuv
260188  24-3080408 2024-11-03 10:15:00         larceny/theft

       incident_description  parent_incident_type  hour_of_day day_of_week  \
5       under investigation  Other Sexual Offense            0    Thursday
8       under investigation        Sexual Assault            0    Thursday
22      under investigation                 Theft            0    Thursday
25      under investigation                 Theft            0    Thursday
29      under investigation  Other Sexual Offense            0    Thursday
...                     ...                   ...          ...         ...
260176  under investigation                 Theft            1      Sunday
260181  under investigation   Breaking & Entering            2      Sunday
260183  under investigation      Theft of Vehicle            6      Sunday
260185  under investigation      Theft of Vehicle            8      Sunday
260188  under investigation                 Theft           10      Sunday

                     address_1     city state  \
5       1000 block w delavan av  Buffalo    NY
8          1 block devereaux av  Buffalo    NY
22         300 block loepere st  Buffalo    NY
25           1600 block main st  Buffalo    NY
29            200 block lawn av  Buffalo    NY
...                         ...      ...   ...
260176    600 block delaware av  Buffalo    NY
260181        600 block ohio st  Buffalo    NY
260183   300 block e amherst st  Buffalo    NY
260185       0 block camelot ct  Buffalo    NY
260188       900 block smith st  Buffalo    NY

                                                location  latitude  \
5       {'type': 'Point', 'coordinates': [-78.861, 42....    42.922
8       {'type': 'Point', 'coordinates': [-78.83, 42.9...    42.956
22      {'type': 'Point', 'coordinates': [-78.834, 42....    42.902
25      {'type': 'Point', 'coordinates': [-78.863, 42....    42.917
29      {'type': 'Point', 'coordinates': [-78.888, 42....    42.950
...                                                   ...       ...
260176  {'type': 'Point', 'coordinates': [-78.873, 42....    42.902
260181  {'type': 'Point', 'coordinates': [-78.867, 42....    42.859
260183  {'type': 'Point', 'coordinates': [-78.825, 42....    42.942
260185  {'type': 'Point', 'coordinates': [-78.826, 42....    42.946
260188  {'type': 'Point', 'coordinates': [-78.844, 42....    42.889

        longitude zip_code        neighborhood council_district  \
5          -78.861    14209     Elmwood Bidwell         ELLICOTT
8          -78.830    14214  University Heights       UNIVERSITY
22         -78.834    14211   Broadway Fillmore         ELLICOTT
25         -78.863    14209         Masten Park         ELLICOTT
29         -78.888    14207         West Hertel            NORTH
...            ...      ...                 ...              ...
```

```
260176    -78.873    14201           Allentown         FILLMORE
260181    -78.867    14203             Central            SOUTH
260183    -78.825    14215  University Heights       UNIVERSITY
260185    -78.826    14214  University Heights       UNIVERSITY
260188    -78.844    14212   Broadway Fillmore         FILLMORE

       council_district_2011 census_tract census_block_group census_block  \
5                   ELLICOTT          169                   4         4002
8                 UNIVERSITY        46.01                   2         2001
22                  FILLMORE        27.04                   1         1005
25                  ELLICOTT       168.02                   1         1017
29                     NORTH           56                   4         4000
...                      ...          ...                 ...          ...
260176              FILLMORE        68.02                   3         3000
260181                 SOUTH            5                   1         1049
260183            UNIVERSITY        47.02                   2         2004
260185            UNIVERSITY        47.02                   2         2004
260188              FILLMORE           15                   2         2011

       census_tract_2010 census_block_group_2010 census_block_2010  \
5                    169                       4              4000
8                  46.01                       2              2001
22                 27.02                       1              1005
25                   168                       3              3015
29                    56                       4              4000
...                  ...                     ...               ...
260176                68                       4              4000
260181                 5                       1              1044
260183                47                       4              4008
260185                47                       4              4004
260188                15                       2              2022

       police_district tractce20 geoid20_tract geoid20_blockgroup  \
5           District D    016900   36029016900       360290002004
8           District E    004601   36029004601       360290001102
22          District C    002704   36029002704       360290001101
25          District E    016802   36029016802       360290001101
29          District D    005600   36029005600       360290002004
...                ...       ...           ...                ...
260176      District B    006802   36029006802       360290068023
260181      District A    000500   36029000500       360290005001
260183      District E    004702   36029004702       360290047022
260185      District E    004702   36029004702       360290047022
260188      District C    001500   36029001500       360290015002

          geoid20_block :@computed_region_jdfw_hhbp  \
5       360290002004002                          14
8       360290033022001                          20
22      360290002001005                          17
25      360290165001017                          14
29      360290002004000                          18
...                 ...                         ...
260176  360290068023000                          19
260181  360290005001049                          16
260183  360290047022004                          11
260185  360290047022004                          20
260188  360290015002011                           2

       :@computed_region_h7a8_iwt4 :@computed_region_ff6v_jbaa  \
5                                4                          10
```

```
8                                7                                91
22                               2                                42
25                               4                                10
29                              10                                79
...                            ...                               ...
260176                           2                                15
260181                           9                                14
260183                           7                                46
260185                           7                                46
260188                           2                                75

       :@computed_region_vsen_jbmg :@computed_region_nmyf_6jtp  \
5                                2                                14
8                                4                                34
22                               1                                35
25                               2                                14
29                               2                                19
...                            ...                               ...
260176                           5                                23
260181                           3                                16
260183                           4                                34
260185                           4                                34
260188                           1                                35

       :@computed_region_yg52_574g
5                                3
8                                9
22                               3
25                               3
29                               7
...                            ...
260176                           6
260181                           4
260183                           9
260185                           9
260188                           6

[60694 rows x 34 columns]
```

In [40]:
```
# Find duplicate rows after fixing the dict issue
duplicate_rows_df = df_filtered[df_filtered.duplicated()]
print("Number of duplicate rows: ", len(duplicate_rows_df))
```

```
Number of duplicate rows:  60694
```

In [41]:
```
# Show all the duplicate rows
duplicate_rows_df = df_filtered[df_filtered.duplicated()]
print(duplicate_rows_df)
```

```
       case_number   incident_datetime incident_type_primary  \
5       09-0830328 2009-01-01 00:00:00          sexual abuse
8       09-1320602 2009-01-01 00:00:00                 rape
22      09-0640645 2009-01-01 00:00:00        larceny/theft
25      09-1740322 2009-01-01 00:00:00        larceny/theft
29      17-2770266 2009-01-01 00:01:00          sexual abuse
...            ...                 ...                   ...
260176  24-3080099 2024-11-03 01:00:00        larceny/theft
260181  24-3080156 2024-11-03 02:52:58             burglary
260183  24-3080255 2024-11-03 06:52:49                  uuv
260185  24-3080296 2024-11-03 08:03:06                  uuv
260188  24-3080408 2024-11-03 10:15:00        larceny/theft

       incident_description  parent_incident_type  hour_of_day day_of_week  \
5       under investigation  Other Sexual Offense            0    Thursday
8       under investigation        Sexual Assault            0    Thursday
22      under investigation                 Theft            0    Thursday
25      under investigation                 Theft            0    Thursday
29      under investigation  Other Sexual Offense            0    Thursday
...                     ...                   ...          ...         ...
260176  under investigation                 Theft            1      Sunday
260181  under investigation   Breaking & Entering            2      Sunday
260183  under investigation      Theft of Vehicle            6      Sunday
260185  under investigation      Theft of Vehicle            8      Sunday
260188  under investigation                 Theft           10      Sunday

                    address_1     city state  \
5       1000 block w delavan av  Buffalo    NY
8          1 block devereaux av  Buffalo    NY
22         300 block loepere st  Buffalo    NY
25           1600 block main st  Buffalo    NY
29            200 block lawn av  Buffalo    NY
...                         ...      ...   ...
260176    600 block delaware av  Buffalo    NY
260181        600 block ohio st  Buffalo    NY
260183   300 block e amherst st  Buffalo    NY
260185       0 block camelot ct  Buffalo    NY
260188       900 block smith st  Buffalo    NY

                                                location  latitude  \
5       {'type': 'Point', 'coordinates': [-78.861, 42....    42.922
8       {'type': 'Point', 'coordinates': [-78.83, 42.9...    42.956
22      {'type': 'Point', 'coordinates': [-78.834, 42....    42.902
25      {'type': 'Point', 'coordinates': [-78.863, 42....    42.917
29      {'type': 'Point', 'coordinates': [-78.888, 42....    42.950
...                                                   ...       ...
260176  {'type': 'Point', 'coordinates': [-78.873, 42....    42.902
260181  {'type': 'Point', 'coordinates': [-78.867, 42....    42.859
260183  {'type': 'Point', 'coordinates': [-78.825, 42....    42.942
260185  {'type': 'Point', 'coordinates': [-78.826, 42....    42.946
260188  {'type': 'Point', 'coordinates': [-78.844, 42....    42.889

        longitude zip_code        neighborhood council_district  \
5          -78.861    14209     Elmwood Bidwell         ELLICOTT
8          -78.830    14214  University Heights       UNIVERSITY
22         -78.834    14211   Broadway Fillmore         ELLICOTT
25         -78.863    14209         Masten Park         ELLICOTT
29         -78.888    14207         West Hertel            NORTH
...            ...      ...                 ...              ...
260176     -78.873    14201           Allentown         FILLMORE
```

```
260181    -78.867    14203            Central           SOUTH
260183    -78.825    14215  University Heights       UNIVERSITY
260185    -78.826    14214  University Heights       UNIVERSITY
260188    -78.844    14212   Broadway Fillmore         FILLMORE

       council_district_2011 census_tract census_block_group census_block  \
5                  ELLICOTT          169                  4         4002
8                UNIVERSITY        46.01                  2         2001
22                 FILLMORE        27.04                  1         1005
25                 ELLICOTT       168.02                  1         1017
29                    NORTH           56                  4         4000
...                     ...          ...                ...          ...
260176             FILLMORE        68.02                  3         3000
260181                SOUTH            5                  1         1049
260183           UNIVERSITY        47.02                  2         2004
260185           UNIVERSITY        47.02                  2         2004
260188             FILLMORE           15                  2         2011

       census_tract_2010 census_block_group_2010 census_block_2010  \
5                    169                        4              4000
8                  46.01                        2              2001
22                 27.02                        1              1005
25                   168                        3              3015
29                    56                        4              4000
...                  ...                      ...               ...
260176                68                        4              4000
260181                 5                        1              1044
260183                47                        4              4008
260185                47                        4              4004
260188                15                        2              2022

       police_district tractce20 geoid20_tract geoid20_blockgroup  \
5           District D    016900   36029016900       360290002004
8           District E    004601   36029004601       360290001102
22          District C    002704   36029002704       360290001101
25          District E    016802   36029016802       360290001101
29          District D    005600   36029005600       360290002004
...                ...       ...           ...                ...
260176      District B    006802   36029006802       360290068023
260181      District A    000500   36029000500       360290005001
260183      District E    004702   36029004702       360290047022
260185      District E    004702   36029004702       360290047022
260188      District C    001500   36029001500       360290015002

         geoid20_block :@computed_region_jdfw_hhbp  \
5       360290002004002                          14
8       360290033022001                          20
22      360290002001005                          17
25      360290165001017                          14
29      360290002004000                          18
...                 ...                         ...
260176  360290068023000                          19
260181  360290005001049                          16
260183  360290047022004                          11
260185  360290047022004                          20
260188  360290015002011                           2

       :@computed_region_h7a8_iwt4 :@computed_region_ff6v_jbaa  \
5                                4                          10
8                                7                          91
```

```
22                                2                                42
25                                4                                10
29                               10                                79
...                             ...                               ...
260176                            2                                15
260181                            9                                14
260183                            7                                46
260185                            7                                46
260188                            2                                75

       :@computed_region_vsen_jbmg :@computed_region_nmyf_6jtp  \
5                                 2                                14
8                                 4                                34
22                                1                                35
25                                2                                14
29                                2                                19
...                             ...                               ...
260176                            5                                23
260181                            3                                16
260183                            4                                34
260185                            4                                34
260188                            1                                35

       :@computed_region_yg52_574g
5                                 3
8                                 9
22                                3
25                                3
29                                7
...                             ...
260176                            6
260181                            4
260183                            9
260185                            9
260188                            6

[60694 rows x 34 columns]
```

In [42]:
```
df_non_duplicates = df_filtered.drop_duplicates(keep=False)
```

In [43]:
```
df_non_duplicates.incident_type_primary.unique()
```

Out[43]:
```
array(['larceny/theft', 'rape', 'sexual abuse', 'burglary', 'uuv',
       'theft of services', 'assault', 'robbery', 'murder',
       'manslaughter', 'theft', 'theft of vehicle', 'breaking & entering',
       'sexual assault', 'aggr assault', 'agg assault on p/officer',
       'crim negligent homicide'], dtype=object)
```

In [44]:
```
# Crime categories grouping
sexual_crimes = ['other sexual offense', 'sexual assault', 'rape', 'sexual abuse', 'sc
assault_crimes = ['agg assault on p/officer', 'aggr assault', 'assault']
vehicle_crimes = ['theft of vehicles', 'uuv', 'theft of vehicle']
theft_crimes = ['burglary', 'larceny/theft', 'robbery', 'theft of services', 'theft',
murder_crimes = ['crim negligent homicide', 'homicide', 'manslaughter', 'murder']

# Replace crime types with broader categories
df_non_duplicates['incident_type_primary'] = df_non_duplicates['incident_type_primary'
df_non_duplicates['incident_type_primary'] = df_non_duplicates['incident_type_primary'
df_non_duplicates['incident_type_primary'] = df_non_duplicates['incident_type_primary'
df_non_duplicates['incident_type_primary'] = df_non_duplicates['incident_type_primary'
```

```
df_non_duplicates['incident_type_primary'] = df_non_duplicates['incident_type_primary'
```

```
/var/folders/89/hbkg9qpn41q375z67mzxbytc0000gn/T/ipykernel_23067/4266199688.py:9: Set
tingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df_non_duplicates['incident_type_primary'] = df_non_duplicates['incident_type_prima
ry'].replace(sexual_crimes, 'Sexual Crime')
/var/folders/89/hbkg9qpn41q375z67mzxbytc0000gn/T/ipykernel_23067/4266199688.py:10: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df_non_duplicates['incident_type_primary'] = df_non_duplicates['incident_type_prima
ry'].replace(assault_crimes, 'Assault Crime')
/var/folders/89/hbkg9qpn41q375z67mzxbytc0000gn/T/ipykernel_23067/4266199688.py:11: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df_non_duplicates['incident_type_primary'] = df_non_duplicates['incident_type_prima
ry'].replace(vehicle_crimes, 'Vehicle Crime')
/var/folders/89/hbkg9qpn41q375z67mzxbytc0000gn/T/ipykernel_23067/4266199688.py:12: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df_non_duplicates['incident_type_primary'] = df_non_duplicates['incident_type_prima
ry'].replace(theft_crimes, 'Theft Crime')
/var/folders/89/hbkg9qpn41q375z67mzxbytc0000gn/T/ipykernel_23067/4266199688.py:13: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df_non_duplicates['incident_type_primary'] = df_non_duplicates['incident_type_prima
ry'].replace(murder_crimes, 'Murder Crime')
```

In [45]:
```
df_graph1 = df_non_duplicates.groupby('incident_type_primary')['case_number'].nunique(
```

In [ ]:
```
df_non_duplicates['Year'] = pd.to_datetime(df_non_duplicates['incident_datetime']).dt.
df_non_duplicates['month'] = pd.to_datetime(df_non_duplicates['incident_datetime']).dt
```

```
/var/folders/89/hbkg9qpn41q375z67mzxbytc0000gn/T/ipykernel_23067/1075928847.py:1: Set
tingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df_non_duplicates['Year'] = pd.to_datetime(df_non_duplicates['incident_datetime']).
dt.year
```

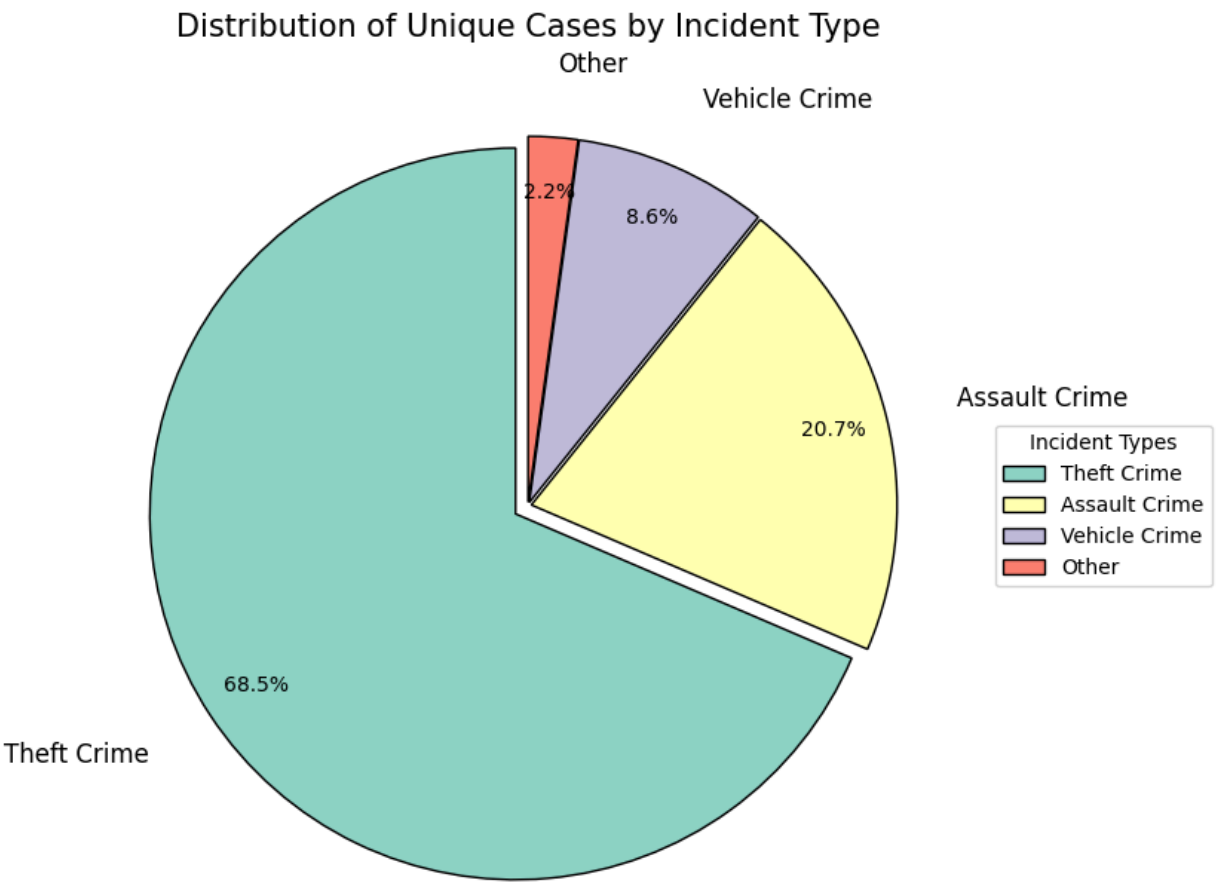
In [53]:
```
df_non_duplicates
```

Out[53]:

	case_number	incident_datetime	incident_type_primary	incident_description	parent_incident_t
1	09-1210237	2009-01-01 00:00:00	Theft Crime	under investigation	T
2	09-3010604	2009-01-01 00:00:00	Sexual Crime	under investigation	Sexual Ass
4	09-1060143	2009-01-01 00:00:00	Theft Crime	under investigation	T
6	09-0770421	2009-01-01 00:00:00	Theft Crime	under investigation	T
11	09-0010819	2009-01-01 00:00:00	Theft Crime	under investigation	T
...	...	...	...	...	
260202	24-3090156	2024-11-04 07:03:51	Assault Crime	under investigation	Ass
260203	24-3090236	2024-11-04 08:33:26	Theft Crime	under investigation	T
260204	24-3090255	2024-11-04 08:58:45	Theft Crime	under investigation	T
260205	24-3090287	2024-11-04 09:27:27	Theft Crime	under investigation	T
260206	24-3090308	2024-11-04 09:40:00	Theft Crime	under investigation	T

120625 rows × 36 columns

In [54]:
```
import matplotlib.pyplot as plt
import pandas as pd



# Group by incident type and count unique case numbers
df_graph1 = df_non_duplicates.groupby('incident_type_primary')['case_number'].nunique(

# Set a threshold to group smaller slices together into 'Other'
threshold = 0.03 * df_graph1.sum()  # 3% threshold

# Grouping the smaller categories as "Other"
df_graph1_grouped = df_graph1[df_graph1 >= threshold]
df_graph1_grouped['Other'] = df_graph1[df_graph1 < threshold].sum()

# Increase the figure size for better readability
plt.figure(figsize=(12, 8))

# Create the pie chart with a reduced explode value
wedges, texts, autotexts = plt.pie(
    df_graph1_grouped,                      # Data for the pie chart
    labels=df_graph1_grouped.index,         # Labels for the slices
    autopct='%1.1f%%',                      # Show percentages on slices
    startangle=90,                          # Start the chart at 90 degrees
    explode=[0.04] + [0.01] * (len(df_graph1_grouped) - 1),  # Reduced explode values
    pctdistance=0.85,                       # Distance of percentage labels from the c
    labeldistance=1.2,                      # Move labels outside the chart for better
    textprops={'fontsize': 12},             # Set font size for better readability
    wedgeprops={'linewidth': 1, 'edgecolor': 'black'},  # Add borders to the wedges
    colors=plt.cm.Set3.colors               # Use the 'Set3' colormap for better color
)

# Set properties for better readability of percentage text
for autotext in autotexts:
    autotext.set_color('black')
    autotext.set_fontsize(10)

# Add a title
plt.title('Distribution of Unique Cases by Incident Type', fontsize=15)

# Ensure equal aspect ratio so that the pie is drawn as a circle
plt.gca().set_aspect('equal')

# Add a legend instead of labels for better visibility
plt.legend(wedges, df_graph1_grouped.index, title="Incident Types", loc="center left",

# Show the plot
plt.show()
```



In [57]:
```
df_non_duplicates
```

Out[57]:

	case_number	incident_datetime	incident_type_primary	incident_description	parent_incident_t
1	09-1210237	2009-01-01 00:00:00	Theft Crime	under investigation	T
2	09-3010604	2009-01-01 00:00:00	Sexual Crime	under investigation	Sexual Ass
4	09-1060143	2009-01-01 00:00:00	Theft Crime	under investigation	T
6	09-0770421	2009-01-01 00:00:00	Theft Crime	under investigation	T
11	09-0010819	2009-01-01 00:00:00	Theft Crime	under investigation	T
...	...	...	...	...	
260202	24-3090156	2024-11-04 07:03:51	Assault Crime	under investigation	Ass
260203	24-3090236	2024-11-04 08:33:26	Theft Crime	under investigation	T
260204	24-3090255	2024-11-04 08:58:45	Theft Crime	under investigation	T
260205	24-3090287	2024-11-04 09:27:27	Theft Crime	under investigation	T
260206	24-3090308	2024-11-04 09:40:00	Theft Crime	under investigation	T

120625 rows × 38 columns

PHASE II:

QUESTION

How to approach the analysis of temporal crime patterns, focusing on variations in crime across different days and times?

ANSWER

Kmeans clustering

Part 1: Algorithms/Visualizations

For this project, I applied KMeans Clustering, which is a machine learning algorithm discussed in class, to analyze the patterns of crime occurrences based on the day of the week and hour of the day. Below is an outline of the algorithm choices and visualizations.

Algorithms Used

1. KMeans Clustering (Unsupervised Machine Learning):
   - Purpose: KMeans clustering was used to categorize crime incidents based on temporal features—day of the week and time of day. This allows us to see clusters in crime activity patterns, identifying when certain types of crime are more prevalent throughout the week and day.
   - Tuning: I chose 4 clusters after evaluating different cluster counts to optimize interpretability and align with expected crime patterns. The clusters were then color-coded for visual distinction.
2. StandardScaler (Preprocessing):
   - Purpose: To ensure fair clustering by standardizing features (day of the week and hour of the day) so that the KMeans algorithm can process both variables on a comparable scale. This is crucial because raw hour and day values differ in range.

Visualizations

1. Crime Type Clustering Plot:
   - The scatter plot visualizes crime incidents clustered based on day and time. Days of the week are mapped on the x-axis, and times of the day are mapped on the y-axis.
   - Each cluster represents a distinct crime activity pattern with colors representing different clusters. This allows us to intuitively see when certain crime types peak across the week.

The visualization makes it clear which time periods have higher clustering of crimes, providing valuable insights into potential high-risk times.

Part 2: Explanation and Analysis

Justification for Algorithm Choice

I chose KMeans Clustering due to its suitability for grouping data points without requiring labeled data. In this context, we want to observe natural groupings in crime occurrences to identify patterns, rather than predicting specific outcomes. KMeans allows us to uncover hidden structures in the data, which can aid in forming insights into crime patterns across different days and times.

Model Training and Tuning

To train the model, I preprocessed the data using Label Encoding for categorical values (day of the week) and Standard Scaling for numeric values to normalize the range. For clustering, after experimenting with different numbers of clusters, I found that 4 clusters provided a balance between interpretability and complexity, aligning with typical crime pattern expectations (e.g., morning, afternoon, evening, and late evening clusters).

Effectiveness and Insights Gained

The KMeans clustering algorithm effectively highlighted patterns in crime data by categorizing crime occurrences into 4 clusters. Some observations include:

- Early Morning and Midnight: High clustering of specific crime types, possibly indicating thefts or other crimes that occur late at night.
- Afternoon and Evening: Patterns show higher clusters potentially related to public activities when more people are active outside.

This clustering provides actionable insights, suggesting that law enforcement could allocate resources more effectively based on identified high-risk time periods.

NOTE

Theft Crime is in lightgreen colour, Assault Crime in yellow, Vehicle Crime in purple, other in blue "Other": "red"

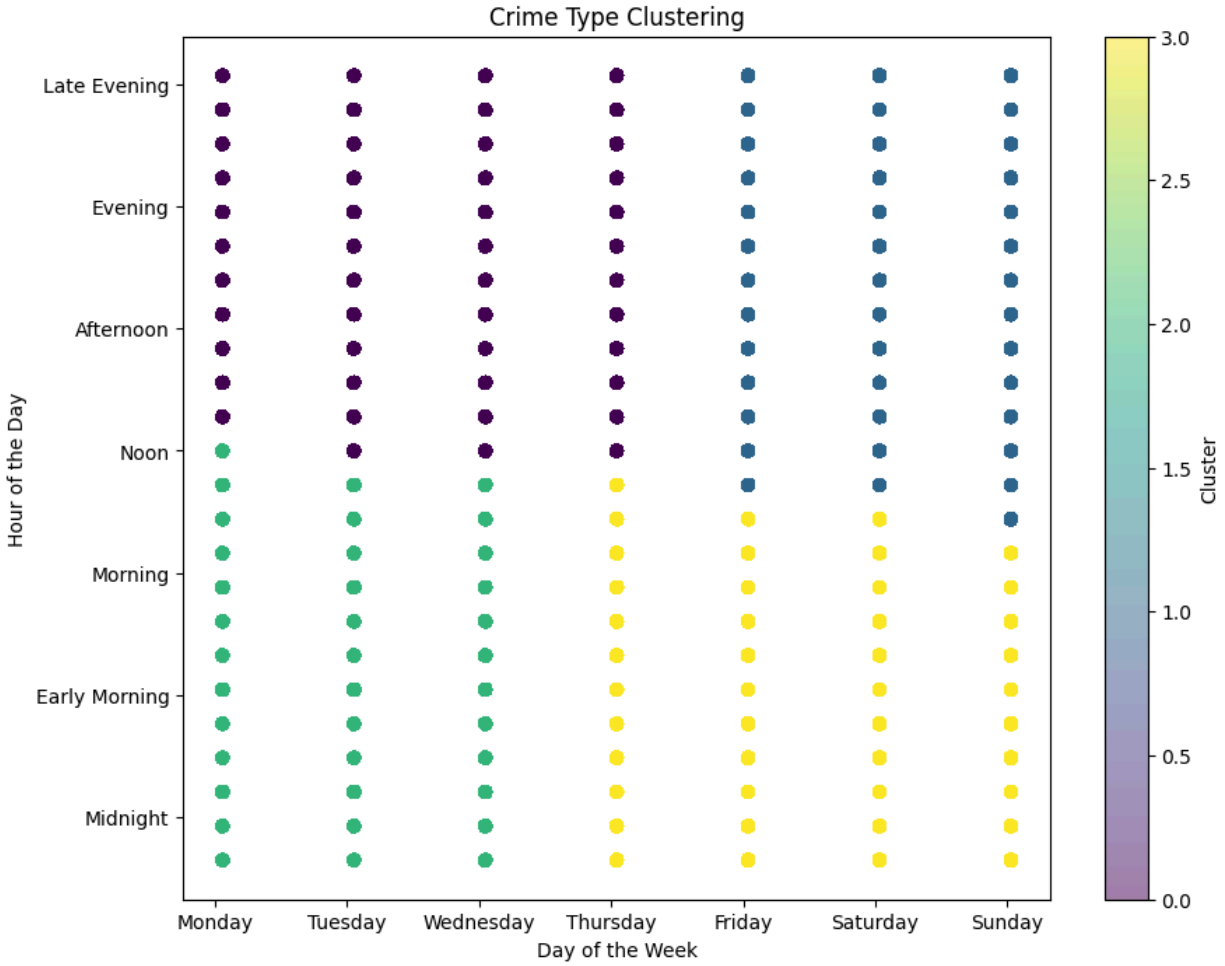
***CITATIONS**

Class notes
https://www.geeksforgeeks.org/k-means-clustering-introduction/

In [58]:
```
#every day analysis of crimes
#"Theft Crime": "lightgreen",..
# "Assault Crime": "yellow",..
# "Vehicle Crime": "purple",
# "Other": "red"
import pandas as pd
import numpy as np
```

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler, LabelEncoder
import matplotlib.pyplot as plt

# Encode the day_of_week column (categorical) using LabelEncoder
le_day_of_week = LabelEncoder()
df_non_duplicates['day_of_week_encoded'] = le_day_of_week.fit_transform(df_non_duplica

# Features for clustering (using encoded values)
X = df_non_duplicates[['day_of_week_encoded', 'hour_of_day']]

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Apply KMeans clustering with 4 clusters
kmeans = KMeans(n_clusters=4, random_state=42)
clusters = kmeans.fit_predict(X_scaled)
df_non_duplicates['cluster'] = clusters

# Define custom labels for x-axis (days of the week) and y-axis (time of day)
x_labels = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunda
y_labels = ['Midnight', 'Early Morning', 'Morning', 'Noon', 'Afternoon', 'Evening', 'L

# Create a figure for visualization
plt.figure(figsize=(10, 8))

# Scatter plot with color based on clusters
plt.scatter(X_scaled[:, 0], X_scaled[:, 1], c=clusters, cmap='viridis', alpha=0.5)

# Set custom x-axis labels for days of the week
plt.xticks(ticks=np.linspace(-1.5, 1.5, len(x_labels)), labels=x_labels)

# Set custom y-axis labels for time of day
plt.yticks(ticks=np.linspace(-1.5, 1.5, len(y_labels)), labels=y_labels)

# Set titles and labels
plt.title('Crime Type Clustering')
plt.xlabel('Day of the Week')
plt.ylabel('Hour of the Day')

# Add colorbar for clusters
plt.colorbar(label='Cluster')

plt.show()
```

```
/var/folders/89/hbkg9qpn41q375z67mzxbytc0000gn/T/ipykernel_23067/3577799079.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_non_duplicates['day_of_week_encoded'] = le_day_of_week.fit_transform(df_non_duplicates['day_of_week'])
/var/folders/89/hbkg9qpn41q375z67mzxbytc0000gn/T/ipykernel_23067/3577799079.py:26: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_non_duplicates['cluster'] = clusters
```



Question 2:

Question: How to classify crime types based on temporal and spatial features and how different types of crime based on time and location details?

Answer:

Algorithm Choice and Problem Statement

For this phase of the project, I applied the Support Vector Machine (SVM) algorithm with an RBF (Radial Basis Function) kernel to classify types of crimes based on features such as the hour of day, latitude, longitude, and month. The goal was to identify if certain types of crimes could be predicted based on spatial and temporal factors.

The SVM algorithm is suitable for this task due to its effectiveness in classification problems with complex decision boundaries. The RBF kernel is particularly powerful in non-linear cases, where the boundaries between classes are not easily separable in a linear way.

Data Preprocessing and Feature Selection

1. Feature Selection: I selected hour_of_day, latitude, longitude, and month as the main features for the model. These features were chosen because they are relevant in understanding crime patterns across different times and locations.
2. Encoding and Scaling:
- The target variable, incident_type, was encoded into numerical labels.
- To ensure that each feature had an equal impact on the model, I used Standard Scaling to standardize the features, as SVM is sensitive to the scale of input data. This preprocessing step was essential for the model to work effectively.

Model Training and Evaluation

The dataset was split into training and testing sets (80% training, 20% testing). The SVM model was then trained on the scaled training data, and predictions were made on the test data.

Performance Metrics:

- The model's performance was evaluated using a classification report and a confusion matrix.
- The classification report revealed an accuracy of 69%, but further analysis showed that the model performed well primarily for class 3 (likely a majority class), while the performance for other classes was very low, with precision, recall, and F1-scores close to zero for classes 0, 1, 2, and 4.
- The confusion matrix illustrated that most predictions fell into the majority class, indicating the model struggled with minority classes.

Analysis of Results and Challenges

1. Class Imbalance Issue: A significant class imbalance is evident in the dataset, with class 3 dominating the data. This imbalance caused the model to favor the majority class, resulting in poor precision and recall for the less frequent classes. This imbalance skewed the accuracy metric, making it appear as though the model was

performing well overall, while it actually struggled to generalize across different crime types.
2. Feature Importance: Since an RBF kernel does not provide direct feature importance, it was challenging to interpret which features were most influential. However, the success in predicting the majority class suggests that certain features, such as hour_of_day and latitude, may be closely associated with that class.

Insights and Potential Improvements

1. Alternative Algorithms: Given the poor performance on minority classes, other algorithms like Random Forest and logistic regression could be explored. These ensemble models can often handle imbalanced data more effectively by capturing complex patterns within the minority classes.But after going through those models it can seen that there is the same thing and we can see that the accuracy is low as shown below.

CITATIONS: https://www.geeksforgeeks.org/support-vector-machine-algorithm/
https://scikit-learn.org/1.5/modules/svm.html

In [63]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming df_non_duplicates is already loaded and preprocessed

# Select features for the model
features = ['hour_of_day', 'latitude', 'longitude', 'month']
X = df_non_duplicates[features]
y = df_non_duplicates['incident_type_encoded']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=

# Scale the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train the SVM model
svm_model = SVC(kernel='rbf', random_state=42)
svm_model.fit(X_train_scaled, y_train)

# Make predictions
y_pred = svm_model.predict(X_test_scaled)

# Print the classification report
```

```
print(classification_report(y_test, y_pred))


# Feature importance (for linear SVM)
if svm_model.kernel == 'linear':
    feature_importance = abs(svm_model.coef_[0])
    feature_names = X.columns
    feature_importance = pd.DataFrame({'feature': feature_names, 'importance': feature
    feature_importance = feature_importance.sort_values('importance', ascending=False)
    print(feature_importance)
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      4977
           1       0.00      0.00      0.00        80
           2       0.00      0.00      0.00       413
           3       0.69      1.00      0.82     16674
           4       0.00      0.00      0.00      1981

    accuracy                           0.69     24125
   macro avg       0.14      0.20      0.16     24125
weighted avg       0.48      0.69      0.56     24125
```

```
/Users/sanhitha/Library/Python/3.9/lib/python/site-packages/sklearn/metrics/_classifi
cation.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to control this be
havior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/Users/sanhitha/Library/Python/3.9/lib/python/site-packages/sklearn/metrics/_classifi
cation.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to control this be
havior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/Users/sanhitha/Library/Python/3.9/lib/python/site-packages/sklearn/metrics/_classifi
cation.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to control this be
havior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

In [62]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Assuming df_non_duplicates is your DataFrame

# Check for missing values and handle them if necessary
print(df_non_duplicates.isnull().sum())

# Fill missing values if necessary (example: fill with mode)
df_non_duplicates['day_of_week'].fillna(df_non_duplicates['day_of_week'].mode()[0], ir

# Encoding categorical variables (incident_type_primary and neighborhood)
le_incident = LabelEncoder()
le_neighborhood = LabelEncoder()
le_day_of_week = LabelEncoder()
```

```
df_non_duplicates['incident_type_encoded'] = le_incident.fit_transform(df_non_duplicat
df_non_duplicates['neighborhood_encoded'] = le_neighborhood.fit_transform(df_non_dupli
df_non_duplicates['day_of_week_encoded'] = le_day_of_week.fit_transform(df_non_duplica

# Selecting relevant features (day_of_week_encoded, hour_of_day, neighborhood_encoded)
features = ['day_of_week_encoded', 'hour_of_day', 'neighborhood_encoded']
X = df_non_duplicates[features]
y = df_non_duplicates['incident_type_encoded']

# Splitting the dataset into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=

### Logistic Regression Model ###

# Initialize Logistic Regression model
logistic_model = LogisticRegression(solver='liblinear')

# Train the model
logistic_model.fit(X_train, y_train)

# Make predictions using Logistic Regression
logistic_predictions = logistic_model.predict(X_test)

# Evaluate Logistic Regression model performance
print("Logistic Regression Results:")
print(f"Accuracy: {accuracy_score(y_test, logistic_predictions)}")
print("Confusion Matrix:")
print(confusion_matrix(y_test, logistic_predictions))
print("Classification Report:")
print(classification_report(y_test, logistic_predictions))

### Random Forest Classifier Model ###

# Initialize Random Forest Classifier
random_forest_model = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the model
random_forest_model.fit(X_train, y_train)

# Make predictions using Random Forest Classifier
rf_predictions = random_forest_model.predict(X_test)

# Evaluate Random Forest model performance
print("\nRandom Forest Classifier Results:")
print(f"Accuracy: {accuracy_score(y_test, rf_predictions)}")
print("Confusion Matrix:")
print(confusion_matrix(y_test, rf_predictions))
print("Classification Report:")
print(classification_report(y_test, rf_predictions))
```

```
case_number              0
incident_datetime        0
incident_type_primary    0
incident_description     0
parent_incident_type     0
                        ..
month                    0
incident_type_encoded    0
neighborhood_encoded     0
day_of_week_encoded      0
cluster                  0
Length: 40, dtype: int64
```

```
/var/folders/89/hbkg9qpn41q375z67mzxbytc0000gn/T/ipykernel_23067/2926733180.py:15: Fu
tureWarning: A value is trying to be set on a copy of a DataFrame or Series through c
hained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because t
he intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method
({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform
the operation inplace on the original object.


  df_non_duplicates['day_of_week'].fillna(df_non_duplicates['day_of_week'].mode()[0],
inplace=True)
/var/folders/89/hbkg9qpn41q375z67mzxbytc0000gn/T/ipykernel_23067/2926733180.py:15: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df_non_duplicates['day_of_week'].fillna(df_non_duplicates['day_of_week'].mode()[0],
inplace=True)
/var/folders/89/hbkg9qpn41q375z67mzxbytc0000gn/T/ipykernel_23067/2926733180.py:22: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df_non_duplicates['incident_type_encoded'] = le_incident.fit_transform(df_non_dupli
cates['incident_type_primary'])
/var/folders/89/hbkg9qpn41q375z67mzxbytc0000gn/T/ipykernel_23067/2926733180.py:23: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df_non_duplicates['neighborhood_encoded'] = le_neighborhood.fit_transform(df_non_du
plicates['neighborhood'])
/var/folders/89/hbkg9qpn41q375z67mzxbytc0000gn/T/ipykernel_23067/2926733180.py:24: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df_non_duplicates['day_of_week_encoded'] = le_day_of_week.fit_transform(df_non_dupl
icates['day_of_week'])
```

```
Logistic Regression Results:
Accuracy: 0.6911502590673575
Confusion Matrix:
[[    0     0     0  4977     0]
 [    0     0     0    80     0]
 [    0     0     0   413     0]
 [    0     0     0 16674     0]
 [    0     0     0  1981     0]]
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      4977
           1       0.00      0.00      0.00        80
           2       0.00      0.00      0.00       413
           3       0.69      1.00      0.82     16674
           4       0.00      0.00      0.00      1981

    accuracy                           0.69     24125
   macro avg       0.14      0.20      0.16     24125
weighted avg       0.48      0.69      0.56     24125

/Users/sanhitha/Library/Python/3.9/lib/python/site-packages/sklearn/metrics/_classifi
cation.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to control this be
havior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/Users/sanhitha/Library/Python/3.9/lib/python/site-packages/sklearn/metrics/_classifi
cation.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to control this be
havior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/Users/sanhitha/Library/Python/3.9/lib/python/site-packages/sklearn/metrics/_classifi
cation.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to control this be
havior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
Random Forest Classifier Results:
Accuracy: 0.67539896373057
Confusion Matrix:
[[  443     0     1  4519    14]
 [    7     0     0    73     0]
 [   33     0     0   379     1]
 [  763     0     1 15849    61]
 [  126     0     0  1853     2]]
Classification Report:
              precision    recall  f1-score   support

           0       0.32      0.09      0.14      4977
           1       0.00      0.00      0.00        80
           2       0.00      0.00      0.00       413
           3       0.70      0.95      0.81     16674
           4       0.03      0.00      0.00      1981

    accuracy                           0.68     24125
   macro avg       0.21      0.21      0.19     24125
weighted avg       0.55      0.68      0.59     24125
```

```
/Users/sanhitha/Library/Python/3.9/lib/python/site-packages/sklearn/metrics/_classifi
cation.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to control this be
havior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/Users/sanhitha/Library/Python/3.9/lib/python/site-packages/sklearn/metrics/_classifi
cation.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to control this be
havior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/Users/sanhitha/Library/Python/3.9/lib/python/site-packages/sklearn/metrics/_classifi
cation.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to control this be
havior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

In [ ]: