

# 01.scraping

August 20, 2024

## 0.1 1. Import Necessary Libraries

```
[ ]: import timeit
import pandas as pd
from requests import get
from bs4 import BeautifulSoup

import warnings
warnings.filterwarnings('ignore')
```

### 0.1.1 2. Scraping the Properties

```
[ ]: start = 101
stop = 111

headers = {'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.11_
↳(KHTML, like Gecko) Chrome/23.0.1271.64 Safari/537.11',
          'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
↳q=0.8',
          'Accept-Charset': 'ISO-8859-1,utf-8;q=0.7,*;q=0.3',
          'Accept-Encoding': 'none',
          'Accept-Language': 'en-US,en;q=0.8',
          'Connection': 'keep-alive'}
```

### 2.1 Define a function for each Columns

```
[ ]: def property_name(soupy_object):    # return house or property name
    try:
        name = soupy_object.find('span', attrs = {'class': 'undefined'}).text
    except:
        name = None
    return name

def address_details(soupy_object):    # return address of property
    try:
        address = soupy_object.find('i', attrs = {'id': 'address'}).text
    except:
        address = None
    return address
```

```

def total_price(soupy_object):    # return total price of property
    try:
        price = soupy_object.find('span', attrs = {'id':"pdPrice2"}).text
    except:
        price = None
    return price

def rate_sqft(soupy_object):    # return total price of property
    try:
        rate = soupy_object.find('div', attrs = {'id':"pricePerUnitArea"}).text.
        ↪split(' ')[1]
    except:
        rate = None
    return rate

def area_type(soupy_object):    # return area parameters
    try:
        areatyp = soupy_object.find('div', attrs = {'id':'factArea'}).text
    except:
        areatyp = None
    return areatyp

def bedroom_count(soupy_object):    # return number of bedrooms
    try:
        bedroom = soupy_object.find('span', attrs={"id":"bedRoomNum"}).text.
        ↪split(' ')[0]
    except:
        bedroom = None
    return bedroom

def bathroom_count(soupy_object):    # return number of bathrooms
    try:
        bathroom = soupy_object.find('span', attrs= {'id':'bathroomNum'}).text.
        ↪split(' ')[0]
    except:
        bathroom = None
    return bathroom

def floor_num(soupy_object):    # return number of floor
    try:
        floornum = soupy_object.find('span', attrs = {'id':'floorNumLabel'}).
        ↪text.split(' ')[0]
    except:
        floornum = None
    return floornum

```

```

def property_age(soupy_object):    # return age of property
    try:
        age = soupy_object.find('span', attrs ={'id':'agePossessionLbl'}).text
    except:
        age = None
    return age

def availability(soupy_object):    # return area parameters
    try:
        avail = soupy_object.find('span', attrs = {'id':'Availability_Lbl'}).
↪text
    except:
        avail = None
    return avail

```

## 2.2 Return a Complete DataFrame into CSV File

```

[ ]: data_list = []
def get_all(start, stop):
    for pagenubmer in range(start, stop):
        url = f'https://www.99acres.com/
↪property-in-mumbai-ffid-page-{pagenubmer}'
        req = get(url, headers = headers)
        soup = BeautifulSoup(req.content, 'html.parser')
        links = soup.find_all('a', attrs= {"class":"body_med_
↪srpTuple__propertyName"})

        for k, item in enumerate(links):
            main_url = 'https://www.99acres.com'
            sub_url = item.get('href')
            data_url = main_url + sub_url
            request = get(data_url, headers=headers)
            soup_get = BeautifulSoup(request.content, 'html.parser')

            name = property_nane(soup_get)
            address = address_details(soup_get)
            price = total_price(soup_get)
            rate = rate_sqft(soup_get)
            areatyp = area_type(soup_get)
            bedroom = bedroom_count(soup_get)
            bathroom = bathroom_count(soup_get)
            floornum = floor_num(soup_get)
            age = property_age(soup_get)
            avail = availability(soup_get)

            data = {'Property_Name': name, 'Location': address, 'Price':price,
↪'Rate_SqFt':rate, 'Area_Tpye':areatyp,

```

```

        'Bedroom': bedroom, 'Bathroom':bathroom, 'Floor_No':
↪floornum, 'Property_Age':age, 'Availability':avail}
        data_list.append(data)

        timestart = timeit.default_timer()
        timestop = timeit.default_timer()
        print(f'You scraped page no : {pagenubmer}')
        print('Time :', timestop - timestart)

    return data_list

```

## 2.3 Define a DataFrame

```
[ ]: df1 = pd.DataFrame(get_all(start, stop))
```

```

-----
SSLCertVerificationError                                Traceback (most recent call last)
File d:\virtual_env\mlmodel\Lib\site-packages\urllib3\connectionpool.py:466, in
↪HTTPConnectionPool._make_request(self, conn, method, url, body, headers,
↪retries, timeout, chunked, response_conn, preload_content, decode_content,
↪enforce_content_length)
    465 try:
--> 466     self._validate_conn(conn)
    467 except (SocketTimeout, BaseSSLError) as e:

File d:\virtual_env\mlmodel\Lib\site-packages\urllib3\connectionpool.py:1095, in
↪HTTPSConnectionPool._validate_conn(self, conn)
    1094 if conn.is_closed:
-> 1095     conn.connect()
    1097 # TODO revise this, see https://github.com/urllib3/urllib3/issues/2791

File d:\virtual_env\mlmodel\Lib\site-packages\urllib3\connection.py:652, in
↪HTTPSConnection.connect(self)
    650 server_hostname_rm_dot = server_hostname.rstrip(".")
--> 652 sock_and_verified = _ssl_wrap_socket_and_match_hostname(
    653     sock=sock,
    654     cert_reqs=self.cert_reqs,
    655     ssl_version=self.ssl_version,
    656     ssl_minimum_version=self.ssl_minimum_version,
    657     ssl_maximum_version=self.ssl_maximum_version,
    658     ca_certs=self.ca_certs,
    659     ca_cert_dir=self.ca_cert_dir,
    660     ca_cert_data=self.ca_cert_data,
    661     cert_file=self.cert_file,
    662     key_file=self.key_file,
    663     key_password=self.key_password,
    664     server_hostname=server_hostname_rm_dot,
    665     ssl_context=self.ssl_context,

```

```

666     tls_in_tls=tls_in_tls,
667     assert_hostname=self.assert_hostname,
668     assert_fingerprint=self.assert_fingerprint,
669 )
670 self.sock = sock_and_verified.socket

```

```

File d:\virtual_env\mlmodel\Lib\site-packages\urllib3\connection.py:805, in
↳ _ssl_wrap_socket_and_match_hostname(sock, cert_reqs, ssl_version,
↳ ssl_minimum_version, ssl_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)
    803         server_hostname = normalized
--> 805 ssl_sock = _ssl_wrap_socket(
    806     sock=sock,
    807     keyfile=key_file,
    808     certfile=cert_file,
    809     key_password=key_password,
    810     ca_certs=ca_certs,
    811     ca_cert_dir=ca_cert_dir,
    812     ca_cert_data=ca_cert_data,
    813     server_hostname=server_hostname,
    814     ssl_context=context,
    815     tls_in_tls=tls_in_tls,
    816 )
    818 try:

```

```

File d:\virtual_env\mlmodel\Lib\site-packages\urllib3\util\ssl_.py:465, in
↳ _ssl_wrap_socket(sock, keyfile, certfile, cert_reqs, ca_certs, server_hostname,
↳ ssl_version, ciphers, ssl_context, ca_cert_dir, key_password, ca_cert_data,
↳ tls_in_tls)
    463     pass
--> 465 ssl_sock =
↳ _ssl_wrap_socket_impl(sock, context, tls_in_tls, server_hostname)
    466 return ssl_sock

```

```

File d:\virtual_env\mlmodel\Lib\site-packages\urllib3\util\ssl_.py:509, in
↳ _ssl_wrap_socket_impl(sock, ssl_context, tls_in_tls, server_hostname)
    507     return SSLTransport(sock, ssl_context, server_hostname)
--> 509 return ssl_context.wrap_socket(sock, server_hostname=server_hostname)

```

```

File ~\AppData\Local\Programs\Python\Python312\Lib\ssl.py:455, in SSLContext.
↳ wrap_socket(self, sock, server_side, do_handshake_on_connect,
↳ suppress_ragged_eofs, server_hostname, session)
    449 def wrap_socket(self, sock, server_side=False,
    450                 do_handshake_on_connect=True,
    451                 suppress_ragged_eofs=True,
    452                 server_hostname=None, session=None):
    453     # SSLSocket class handles server_hostname encoding before it calls
    454     # ctx._wrap_socket()

```

```

--> 455     return self.sslsocket_class._create(
456         sock=sock,
457         server_side=server_side,
458         do_handshake_on_connect=do_handshake_on_connect,
459         suppress_ragged_eofs=suppress_ragged_eofs,
460         server_hostname=server_hostname,
461         context=self,
462         session=session
463     )

```

File ~\AppData\Local\Programs\Python\Python312\Lib\ssl.py:1042, in SSLSocket.

```

↪ _create(cls, sock, server_side, do_handshake_on_connect, suppress_ragged_eofs,
↪ server_hostname, context, session)
1041         raise ValueError("do_handshake_on_connect should not be
↪ specified for non-blocking sockets")
-> 1042         self.do_handshake()
1043 except:

```

File ~\AppData\Local\Programs\Python\Python312\Lib\ssl.py:1320, in SSLSocket.

```

↪ do_handshake(self, block)
1319         self.settimeout(None)
-> 1320         self._sslobj.do_handshake()
1321 finally:

```

**SSLCertVerificationError**: [SSL: CERTIFICATE\_VERIFY\_FAILED] certificate verify  
↪ failed: self-signed certificate in certificate chain (\_ssl.c:1000)

During handling of the above exception, another exception occurred:

**SSL****Error** Traceback (most recent call last)

File d:\virtual\_env\mlmodel\Lib\site-packages\urllib3\connectionpool.py:789, in  
↪ HTTPConnectionPool.urlopen(self, method, url, body, headers, retries,  
↪ redirect, assert\_same\_host, timeout, pool\_timeout, release\_conn, chunked,  
↪ body\_pos, preload\_content, decode\_content, \*\*response\_kw)  
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!

```

```

File d:\virtual_env\mlmodel\Lib\site-packages\urllib3\connectionpool.py:490, in
↳ HTTPConnectionPool.make_request(self, conn, method, url, body, headers,
↳ retries, timeout, chunked, response_conn, preload_content, decode_content,
↳ enforce_content_length)
    489         new_e = _wrap_proxy_error(new_e, conn.proxy.scheme)
--> 490         raise new_e
    492 # conn.request() calls http.client.*.request, not the method in
    493 # urllib3.request. It also calls makefile (recv) on the socket.

```

```

SSLError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed:
↳ self-signed certificate in certificate chain (_ssl.c:1000)

```

The above exception was the direct cause of the following exception:

**MaxRetryError** Traceback (most recent call last)

```

File d:\virtual_env\mlmodel\Lib\site-packages\requests\adapters.py:667, in
↳ HTTPAdapter.send(self, request, stream, timeout, verify, cert, proxies)

```

```

    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:

```

```

File d:\virtual_env\mlmodel\Lib\site-packages\urllib3\connectionpool.py:843, in
↳ HTTPConnectionPool.urlopen(self, method, url, body, headers, retries,
↳ redirect, assert_same_host, timeout, pool_timeout, release_conn, chunked,
↳ body_pos, preload_content, decode_content, **response_kw)

```

```

    841         new_e = ProtocolError("Connection aborted.", new_e)
--> 843         retries = retries.increment(
    844             method, url, error=new_e, _pool=self, _stacktrace=sys.exc_info()[2]
    845         )
    846         retries.sleep()

```

```

File d:\virtual_env\mlmodel\Lib\site-packages\urllib3\util\retry.py:519, in
↳ Retry.increment(self, method, url, response, error, _pool, _stacktrace)

```

```

    518         reason = error or ResponseError(cause)

```

```
--> 519     raise MaxRetryError(_pool, url, reason) from reason # type:
↳ ignore[arg-type]
    521 log.debug("Incremented Retry for (url='%s'): %r", url, new_retry)

MaxRetryError: HTTPSConnectionPool(host='www.99acres.com', port=443): Max
↳ retries exceeded with url: /property-in-mumbai-ffid-page-101 (Caused by
↳ SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED]
↳ certificate verify failed: self-signed certificate in certificate chain (_ssl
↳ c:1000)')))
```

During handling of the above exception, another exception occurred:

```
SSLError                                Traceback (most recent call last)
Cell In[5], line 1
----> 1 df1 = pd.DataFrame(get_all(start, stop))
```

```
Cell In[4], line 5, in get_all(start, stop)
    3 for pagenummer in range(start, stop):
    4     url = f'https://www.99acres.com/
↳ property-in-mumbai-ffid-page-{pagenummer}'
----> 5     req = get(url, headers = headers)
    6     soup = BeautifulSoup(req.content, 'html.parser')
    7     links = soup.find_all('a', attrs= {"class": "body_med
↳ srpTuple__propertyName"})
```

```
File d:\virtual_env\mlmodel\Lib\site-packages\requests\api.py:73, in get(url,
↳ params, **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 d:\virtual_env\mlmodel\Lib\site-packages\requests\api.py:59, in
↳ request(method, 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 som
    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 d:\virtual_env\mlmodel\Lib\site-packages\requests\sessions.py:589, in
↳ Session.request(self, method, url, params, data, headers, cookies, files,
↳ auth, timeout, allow_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(prepare, **send_kwargs)
591 return resp

```

File d:\virtual\_env\mlmodel\Lib\site-packages\requests\sessions.py:703, in

```

    Session.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 d:\virtual\_env\mlmodel\Lib\site-packages\requests\adapters.py:698, in

```

    HTTPAdapter.send(self, request, stream, timeout, verify, cert, proxies)
    694         raise ProxyError(e, request=request)
    696     if isinstance(e.reason, _SSLError):
    697         # This branch is for urllib3 v1.22 and later.
--> 698         raise SSLError(e, request=request)
    700     raise ConnectionError(e, request=request)
    702 except ClosedPoolError as e:

```

```

SSLError: HTTPSConnectionPool(host='www.99acres.com', port=443): Max retries
exceeded with url: /property-in-mumbai-ffid-page-101 (Caused by
SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED]
certificate verify failed: self-signed certificate in certificate chain (_ssl
c:1000)'))))

```

## 2.4 Data Understanding

```
[ ]: df1.head()
```

```
[ ]: df1.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 175 entries, 0 to 174
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   Property_Name    160 non-null    object
 1   Location         160 non-null    object
 2   Price           160 non-null    object
 3   Rate_SqFt       160 non-null    object
 4   Area_Tpye       160 non-null    object
 5   Bedroom         159 non-null    object
 6   Bathroom        159 non-null    object
 7   Floor_No        159 non-null    object

```

```

      8   Property_Age   160 non-null   object
      9   Availability   160 non-null   object
dtypes: object(10)
memory usage: 13.8+ KB

```

```
[ ]: df1.duplicated().sum()
```

```
[ ]: 14
```

```
[ ]: df1.nunique()
```

```

[ ]: Property_Name      95
     Location          121
     Price             111
     Rate_SqFt         155
     Area_Tpye         144
     Bedroom            5
     Bathroom           5
     Floor_No           35
     Property_Age        5
     Availability        3
     dtype: int64

```

## 2.5 Create a CSV File

```
[ ]: df1.to_csv('Prop_101to110.csv', index_label = False)
```

```

[ ]: projectlist = pd.read_csv("Prop_101to110.csv")
     projectlist.head(16)

```

```

[ ]:
      Property_Name \
0      Yogsiddhi Sumukh Hills
1  Raunak Unnathi Woods Phase 6
2      Prasham Vishal 2
3      Toll Free 1800 41 99099
4      Vijay Park
5      Omkar Alta Monte
6      Haware Tiara
7      Toll Free 1800 41 99099
8      Sai Ashirwad
9      Advance Heights
10     Toll Free 1800 41 99099
11     Siddhivinayak Garima
12     Toll Free 1800 41 99099
13     Toll Free 1800 41 99099
14  Gajora Kalpavruksha Valley
15     Sai Kaveesha

```

```

      Location      Price Rate_SqFt \

```

0	703, Kandivali East, Mumbai Andheri-Dahisar, M...	1.65 Crore	26,612
1	Thane West, Thane, Mumbai	85 Lac	8,594
2	602, Borivali West, Mumbai Andheri-Dahisar, Mu...	3.05 Crore	33,738
3	Malad West, Mumbai Andheri-Dahisar, Mumbai	72 Lac	14,400
4	Kasar vadavali, Thane, Mumbai	48.5 Lac	8,083
5	Omkar Alta Monte Malad East, Malad East, Mumba...	3.18 Crore	25,645
6	Sector 13 Kharghar, Navi Mumbai, Mumbai	1.7 Crore	10,967
7	Kharghar, Navi Mumbai, Mumbai	92 Lac	7,666
8	Mira Road, Mira Road And Beyond, Mumbai	36.5 Lac	10,138
9	Kharghar, Navi Mumbai, Mumbai	1 Crore	8,849
10	Ghotkamp Koyana Vele, Navi Mumbai, Mumbai	45 Lac	7,500
11	Taloja, Navi Mumbai, Mumbai	32 Lac	8,000
12	Juhu, Mumbai South West, Mumbai	24 Crore	42,105
13	Juhu, Mumbai South West, Mumbai	4.5 Crore	30,000
14	Sector 23 Taloja, Navi Mumbai, Mumbai	42 Lac	84
15	Taloja, Navi Mumbai, Mumbai	36 Lac	8,571

	Area_Tpye	Bedroom	Bathroom \
0	Carpet area: 620 (57.6 sq.m.)	2.0	2.0
1	Super Built up area 989(91.88 sq.m.)Carpet are...	2.0	2.0
2	Carpet area: 904 (83.98 sq.m.)	3.0	3.0
3	Carpet area: 500 (46.45 sq.m.)	1.0	1.0
4	Super Built up area 600(55.74 sq.m.)Carpet are...	1.0	1.0
5	Super Built up area 1750(162.58 sq.m.)Built Up...	3.0	3.0
6	Built Up area: 1550 (144 sq.m.)Carpet area: 11...	3.0	3.0
7	Super Built up area 1200(111.48 sq.m.)Built Up...	2.0	2.0
8	Super Built up area 360(33.45 sq.m.)Carpet are...	1.0	1.0
9	Super Built up area 1130(104.98 sq.m.)Carpet a...	2.0	2.0
10	Carpet area: 600 (55.74 sq.m.)	2.0	2.0
11	Carpet area: 400 (37.16 sq.m.)	1.0	2.0
12	Super Built up area 5700(529.55 sq.m.)Built Up...	4.0	4.0
13	Super Built up area 1500(139.35 sq.m.)Built Up...	3.0	3.0
14	Carpet area: 49454 (4594.43 sq.m.)	1.0	2.0
15	Carpet area: 420 (39.02 sq.m.)	1.0	1.0

	Floor_No	Property_Age	Availability
0	7th	0 to 1 Year Old	Ready to move
1	5th	1 to 5 Year Old	Ready to move
2	6th	1 to 5 Year Old	Ready to move
3	5th	10+ Year Old	Ready to move
4	3rd	10+ Year Old	Ready to move
5	39th	0 to 1 Year Old	Ready to move
6	11st	5 to 10 Year Old	Ready to move
7	4th	10+ Year Old	Ready to move
8	Ground	10+ Year Old	Ready to move
9	13rd	0 to 1 Year Old	Ready to move
10	2nd	0 to 1 Year Old	Ready to move

11	2nd	1 to 5 Year Old	Ready to move
12	1st	10+ Year Old	Ready to move Property
13	5th	5 to 10 Year Old	Ready to move Property
14	1st	0 to 1 Year Old	Ready to move
15	10th	0 to 1 Year Old	Ready to move

### 0.1.2 3. Import the all Datasets and Concating

```
[ ]: df1 = pd.read_csv("Prop_001to050.csv")
df2 = pd.read_csv("Prop_051to100.csv")
df3 = pd.read_csv("Prop_101to150.csv")
```

```
[ ]: df = pd.concat([df1, df2, df3], ignore_index=False)
```

```
[ ]: df
```

```
[ ]:
Property_Name \
0          Omkar Alta Monte
1    T Bhimjyani Neelkanth Woods
2          Legend 1 Pramila Nagar
3          Toll Free 1800 41 99099
4          Toll Free 1800 41 99099
...
1184        Shagun White Woods
1185              Guru Anant
1186        Balaji Mayuresh Delta
1187        Balaji Mayuresh Delta
1188    Gurukrupa Tulsi Heights
```

	Location	Price	Rate_SqFt \
0	W.E.Highway, Malad East, Mumbai Andheri-Dahisa...	5 Crore	17,241
1	Manpada, Thane, Mumbai	2.4 Crore	12,631
2	Dahisar West, Mumbai Andheri-Dahisar, Mumbai	95 Lac	15,966
3	Vidyavihar West, Vidyavihar West, Central Mumb...	3.75 Crore	25,862
4	176 Cst Road, Kalina, Mumbai 400098, Santacruz...	3.5 Crore	39,954
...	...	...	...
1184	Sector 23 Ulwe, Navi Mumbai, Mumbai	1.22 Crore	10,338
1185	Sector 2 Ulwe, Navi Mumbai, Mumbai	88 Lac	8,073
1186	Ulwe, Navi Mumbai, Mumbai	1.37 Crore	10,579
1187	Ulwe, Navi Mumbai, Mumbai	1.71 Crore	9,243
1188	Ulwe, Navi Mumbai, Mumbai	95 Lac	8,636

	Area_Tpye	Bedroom	Bathroom \
0	Super Built up area 2900(269.42 sq.m.)Built Up...	3.0	4.0
1	Super Built up area 1900(176.52 sq.m.)Built Up...	3.0	3.0
2	Super Built up area 595(55.28 sq.m.)	1.0	2.0
3	Built Up area: 1450 (134.71 sq.m.)	3.0	3.0

4	Carpet area: 876 (81.38 sq.m.)	2.0	2.0
...	...	...	...
1184	Built Up area: 1180 (109.63 sq.m.)	2.0	2.0
1185	Built Up area: 1090 (101.26 sq.m.)	2.0	2.0
1186	Built Up area: 1295 (120.31 sq.m.)	2.0	2.0
1187	Built Up area: 1850 (171.87 sq.m.)	3.0	3.0
1188	Built Up area: 1100 (102.19 sq.m.)	2.0	2.0

	Floor_No	Property_Age	Availability
0	14th	0 to 1 Year Old	Ready to move
1	8th	1 to 5 Year Old	Ready to move Property
2	3rd	10+ Year Old	Ready to move
3	1st	5 to 10 Year Old	Ready to move
4	5th	5 to 10 Year Old	Ready to move
...	...	...	...
1184	2nd	1 to 5 Year Old	Ready to move
1185	11st	0 to 1 Year Old	Ready to move
1186	6th	1 to 5 Year Old	Ready to move
1187	6th	1 to 5 Year Old	Ready to move
1188	4th	0 to 1 Year Old	Ready to move

[2684 rows x 10 columns]

```
[ ]: df["Property_Name"] = df["Property_Name"].str.replace('Toll Free 1800 41_
↪99099','Unnamed Property')
```

```
[ ]: df.isna().sum()
```

```
[ ]: Property_Name    3
Location            5
Price              5
Rate_SqFt          3
Area_Tpye          3
Bedroom            82
Bathroom           82
Floor_No           72
Property_Age       24
Availability        11
dtype: int64
```

```
[ ]: df.dropna(inplace=True)
df.reset_index(drop=True, inplace=True)
```

```
[ ]: df.isna().sum()
```

```
[ ]: Property_Name    0
Location            0
```

```

Price          0
Rate_SqFt      0
Area_Tpye      0
Bedroom        0
Bathroom       0
Floor_No       0
Property_Age   0
Availability    0
dtype: int64

```

```
[ ]: df.duplicated().sum()
```

```
[ ]: 9
```

```
[ ]: df = df.drop_duplicates(ignore_index=True)
```

```
[ ]: df.to_csv('Raw_Property.csv', index=False)
```

```
[ ]: df = pd.read_csv('Raw_Property.csv')
```

```
[ ]: print('Shape of Data :', df.shape)
df
```

Shape of Data : (2581, 10)

```
[ ]:
      Property_Name \
0      Omkar Alta Monte
1  T Bhimjyani Neelkanth Woods
2      Legend 1 Pramila Nagar
3      Unnamed Property
4      Unnamed Property
...
2576      Shagun White Woods
2577      Guru Anant
2578      Balaji Mayuresh Delta
2579      Balaji Mayuresh Delta
2580      Gurukrupa Tulsi Heights

```

```

      Location      Price Rate_SqFt \
0  W.E.Highway, Malad East, Mumbai Andheri-Dahisa...  5 Crore  17,241
1      Manpada, Thane, Mumbai  2.4 Crore  12,631
2  Dahisar West, Mumbai Andheri-Dahisar, Mumbai  95 Lac  15,966
3  Vidyavihar West, Vidyavihar West, Central Mumb...  3.75 Crore  25,862
4  176 Cst Road, Kalina, Mumbai 400098, Santacruz...  3.5 Crore  39,954
...
2576      Sector 23 Ulwe, Navi Mumbai, Mumbai  1.22 Crore  10,338
2577      Sector 2 Ulwe, Navi Mumbai, Mumbai  88 Lac  8,073
2578      Ulwe, Navi Mumbai, Mumbai  1.37 Crore  10,579

```

2579	Ulwe, Navi Mumbai, Mumbai	1.71 Crore	9,243
2580	Ulwe, Navi Mumbai, Mumbai	95 Lac	8,636

	Area_Tpye	Bedroom	Bathroom	\
0	Super Built up area 2900(269.42 sq.m.)Built Up...	3.0	4.0	
1	Super Built up area 1900(176.52 sq.m.)Built Up...	3.0	3.0	
2	Super Built up area 595(55.28 sq.m.)	1.0	2.0	
3	Built Up area: 1450 (134.71 sq.m.)	3.0	3.0	
4	Carpet area: 876 (81.38 sq.m.)	2.0	2.0	
...	...	...	...	
2576	Built Up area: 1180 (109.63 sq.m.)	2.0	2.0	
2577	Built Up area: 1090 (101.26 sq.m.)	2.0	2.0	
2578	Built Up area: 1295 (120.31 sq.m.)	2.0	2.0	
2579	Built Up area: 1850 (171.87 sq.m.)	3.0	3.0	
2580	Built Up area: 1100 (102.19 sq.m.)	2.0	2.0	

	Floor_No	Property_Age	Availability
0	14th	0 to 1 Year Old	Ready to move
1	8th	1 to 5 Year Old	Ready to move Property
2	3rd	10+ Year Old	Ready to move
3	1st	5 to 10 Year Old	Ready to move
4	5th	5 to 10 Year Old	Ready to move
...	...	...	...
2576	2nd	1 to 5 Year Old	Ready to move
2577	11st	0 to 1 Year Old	Ready to move
2578	6th	1 to 5 Year Old	Ready to move
2579	6th	1 to 5 Year Old	Ready to move
2580	4th	0 to 1 Year Old	Ready to move

[2581 rows x 10 columns]

1 The End !!!