USA Housing Prices Data Cleaning

Import Libraries

In [1]:

```
# Importing libraries
import pandas as pd
import numpy as np
import re

from sklearn.preprocessing import LabelEncoder
```

Import DataSet

In [2]:

```
# importing  dataset
usa_housing_df = pd.read_csv("USA_Housing.csv")

pd.set_option("display.max_columns",None)
```

Data Cleaning

In [3]:

```
usa_housing_df
```

Out[3]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	79545.45857	5.682861	7.009188	4.09	23086.80050	1.059034e+06	208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.64245	6.002900	6.730821	3.09	40173.07217	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.06718	5.865890	8.512727	5.13	36882.15940	1.058988e+06	9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3	63345.24005	7.188236	5.586729	3.26	34310.24283	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.19723	5.040555	7.839388	4.23	26354.10947	6.309435e+05	USNS Raymond\nFPO AE 09386
...	...	...	...	...	...	...	...
4995	60567.94414	7.830362	6.137356	3.46	22837.36103	1.060194e+06	USNS Williams\nFPO AP 30153-7653
4996	78491.27543	6.999135	6.576763	4.02	25616.11549	1.482618e+06	PSC 9258, Box 8489\nAPO AA 42991-3352
4997	63390.68689	7.250591	4.805081	2.13	33266.14549	1.030730e+06	4215 Tracy Garden Suite 076\nJoshualand, VA 01...
4998	68001.33124	5.534388	7.130144	5.44	42625.62016	1.198657e+06	USS Wallace\nFPO AE

							73316
	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
4999	65510.58180	5.992305	6.792336	4.07	46501.28380	1.298950e+06	37778 George Ridges Apt. 509\nEast Holly, NV 2...

5000 rows × 7 columns

In [4]:

```
usa_housing_df.describe()
```

Out[4]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	9925.650114	3.531176e+05
min	17796.631190	2.644304	3.236194	2.000000	172.610686	1.593866e+04
25%	61480.562390	5.322283	6.299250	3.140000	29403.928700	9.975771e+05
50%	68804.286405	5.970429	7.002902	4.050000	36199.406690	1.232669e+06
75%	75783.338665	6.650808	7.665871	4.490000	42861.290770	1.471210e+06
max	107701.748400	9.519088	10.759588	6.500000	69621.713380	2.469066e+06

Change Column Names

In [5]:

```
# changing column names
usa_housing_df.rename(columns={"Avg. Area Income":"income",
                               "Avg. Area House Age":"house_area",
                               "Avg. Area Number of Rooms":"noof_rooms",
                               "Avg. Area Number of Bedrooms":"noof_bedrooms",
                               "Area Population":"population",
                               "Price":"price"}
    ,inplace=True)
```

In [6]:

```
# validating column names
usa_housing_df.columns
```

Out[6]:

```
Index(['income', 'house_area', 'noof_rooms', 'noof_bedrooms', 'population',
       'price', 'Address'],
      dtype='object')
```

In [7]:

```
usa_housing_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   income          5000 non-null   float64
 1   house_area      5000 non-null   float64
 2   noof_rooms      5000 non-null   float64
 3   noof_bedrooms   5000 non-null   float64
 4   population      5000 non-null   float64
 5   price           5000 non-null   float64
 6   Address         5000 non-null   object 
dtypes: float64(6), object(1)
```

```
memory usage: 273.6+ KB
```

In [8]:

```
usa_housing_df
```

Out[8]:

	income	house_area	noof_rooms	noof_bedrooms	population	price	Address
0	79545.45857	5.682861	7.009188	4.09	23086.80050	1.059034e+06	208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.64245	6.002900	6.730821	3.09	40173.07217	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.06718	5.865890	8.512727	5.13	36882.15940	1.058988e+06	9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3	63345.24005	7.188236	5.586729	3.26	34310.24283	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.19723	5.040555	7.839388	4.23	26354.10947	6.309435e+05	USNS Raymond\nFPO AE 09386
...	...	...	...	...	...	...	...
4995	60567.94414	7.830362	6.137356	3.46	22837.36103	1.060194e+06	USNS Williams\nFPO AP 30153-7653
4996	78491.27543	6.999135	6.576763	4.02	25616.11549	1.482618e+06	PSC 9258, Box 8489\nAPO AA 42991-3352
4997	63390.68689	7.250591	4.805081	2.13	33266.14549	1.030730e+06	4215 Tracy Garden Suite 076\nJoshualand, VA 01...
4998	68001.33124	5.534388	7.130144	5.44	42625.62016	1.198657e+06	USS Wallace\nFPO AE 73316
4999	65510.58180	5.992305	6.792336	4.07	46501.28380	1.298950e+06	37778 George Ridges Apt. 509\nEast Holly, NV 2...

5000 rows × 7 columns

Now Let's address the *'Address'* Column

In [9]:

```
address_col = usa_housing_df['Address']
```

In [10]:

```
address_col
```

Out[10]:

```
0       208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1       188 Johnson Views Suite 079\nLake Kathleen, CA...
2       9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3                          USS Barnett\nFPO AP 44820
4                         USNS Raymond\nFPO AE 09386
                              ...
4995                  USNS Williams\nFPO AP 30153-7653
4996               PSC 9258, Box 8489\nAPO AA 42991-3352
4997    4215 Tracy Garden Suite 076\nJoshualand, VA 01...
4998                          USS Wallace\nFPO AE 73316
4999    37778 George Ridges Apt. 509\nEast Holly, NV 2...
Name: Address, Length: 5000, dtype: object
```

In [11]:

```
var = address_col[14]
var
var2 = address_col[1]
var2
```

Out[11]:

```
'188 Johnson Views Suite 079\nLake Kathleen, CA 48958'
```

In [12]:

```
right_after_n = r"\n(.*)"
left_before_n = r"(.*)\n"

new_var = re.findall(right_after_n,var)
first_var = re.findall(left_before_n, var)

new_var1 = new_var[0]
new_var2 = first_var[0]

print(new_var2)
print("before last comma :",type(new_var2))

print(new_var1)
print("after last comma :",type(new_var1))
```

```
PSC 5330, Box 4420
before last comma : <class 'str'>
APO AP 08302
after last comma : <class 'str'>
```

In [13]:

```
# extracting the postal code and checking if there are any categorical values
address_col = list(address_col)
print(type(address_col))
```

```
<class 'list'>
```

Sorting the abbreviations of States, expanding it and revalidating the data

In [14]:

```
new_state_ls = []

right_after_n = r"\n(.*)"

states_pattern = r".*([A-Z]{2}).*"

for i , j in enumerate(address_col):
    # iterating through each addresses

    address_after_newline = re.findall(right_after_n , j)
    address_after_newline = str(address_after_newline[0])
    state_address = re.findall(states_pattern, address_after_newline)
    state_address = str(state_address[0])
    new_state_ls.append(state_address)

print(len(new_state_ls))
```

```
5000
```

In [15]:

```
new_state_ls[5]
```

Out[15]:

```
'KS'
```

In [16]:

```
new_state_ss = pd.Series(new_state_ls)
state_count= list(new_state_ss.unique())
print(len(state_count))
```

```
62
```

In [17]:

```
state_count
```

Out[17]:

```
['NE',
 'CA',
 'WI',
 'AP',
 'AE',
 'KS',
 'CO',
 'TN',
 'AA',
 'NM',
 'PW',
 'AR',
 'HI',
 'ME',
 'IN',
 'MI',
 'DE',
 'AZ',
 'MA',
 'MN',
 'AL',
 'NY',
 'NV',
 'VA',
 'ID',
 'OK',
 'NH',
 'MO',
 'WV',
 'WY',
 'MH',
 'UT',
 'SD',
 'CT',
 'AK',
 'WA',
 'RI',
 'NJ',
 'KY',
 'NC',
 'IA',
 'VT',
 'FM',
 'ND',
 'LA',
 'MP',
 'OR',
 'TX',
 'DC',
 'PR',
 'MT',
 'AS',
 'OH',
 'MS',
 'IL',
 'VI',
 'GA',
 'PA',
 'MD',
 'SC',
 'GU',
 'FL']
```

In [18]:

```
state_and_repititions = {"state_abbr":state_count,
```

```
                          "noOf_repeatitions":[]}
# new_state_ls.count('KS')
# state_and_repititions["states"][0]

for i in state_and_repititions["state_abbr"]:
    print(type(i))
    print(i)
    reps = new_state_ls.count(i)
    state_and_repititions["noOf_repeatitions"].append(reps)



state_and_repititions
```

```
<class 'str'>
NE
<class 'str'>
CA
<class 'str'>
WI
<class 'str'>
AP
<class 'str'>
AE
<class 'str'>
KS
<class 'str'>
CO
<class 'str'>
TN
<class 'str'>
AA
<class 'str'>
NM
<class 'str'>
PW
<class 'str'>
AR
<class 'str'>
HI
<class 'str'>
ME
<class 'str'>
IN
<class 'str'>
MI
<class 'str'>
DE
<class 'str'>
AZ
<class 'str'>
MA
<class 'str'>
MN
<class 'str'>
AL
<class 'str'>
NY
<class 'str'>
NV
<class 'str'>
VA
<class 'str'>
ID
<class 'str'>
OK
<class 'str'>
NH
<class 'str'>
MO
<class 'str'>
WV
```

```
<class 'str'>
WY
<class 'str'>
MH
<class 'str'>
UT
<class 'str'>
SD
<class 'str'>
CT
<class 'str'>
AK
<class 'str'>
WA
<class 'str'>
RI
<class 'str'>
NJ
<class 'str'>
KY
<class 'str'>
NC
<class 'str'>
IA
<class 'str'>
VT
<class 'str'>
FM
<class 'str'>
ND
<class 'str'>
LA
<class 'str'>
MP
<class 'str'>
OR
<class 'str'>
TX
<class 'str'>
DC
<class 'str'>
PR
<class 'str'>
MT
<class 'str'>
AS
<class 'str'>
OH
<class 'str'>
MS
<class 'str'>
IL
<class 'str'>
VI
<class 'str'>
GA
<class 'str'>
PA
<class 'str'>
MD
<class 'str'>
SC
<class 'str'>
GU
<class 'str'>
FL
```

Out[18]:

```
{'state_abbr': ['NE',
  'CA',
  'WI',
  'AP',
```

```
    'AE',
    'KS',
    'CO',
    'TN',
    'AA',
    'NM',
    'PW',
    'AR',
    'HI',
    'ME',
    'IN',
    'MI',
    'DE',
    'AZ',
    'MA',
    'MN',
    'AL',
    'NY',
    'NV',
    'VA',
    'ID',
    'OK',
    'NH',
    'MO',
    'WV',
    'WY',
    'MH',
    'UT',
    'SD',
    'CT',
    'AK',
    'WA',
    'RI',
    'NJ',
    'KY',
    'NC',
    'IA',
    'VT',
    'FM',
    'ND',
    'LA',
    'MP',
    'OR',
    'TX',
    'DC',
    'PR',
    'MT',
    'AS',
    'OH',
    'MS',
    'IL',
    'VI',
    'GA',
    'PA',
    'MD',
    'SC',
    'GU',
    'FL'],
 'noOf_repeatitions': [84,
    78,
    67,
    170,
    167,
    67,
    75,
    77,
    177,
    79,
    70,
    69,
    70,
    75,
```

```
 80,
 77,
 89,
 86,
 77,
 69,
 70,
 77,
 70,
 85,
 81,
 76,
 72,
 81,
 81,
 75,
 69,
 77,
 80,
 74,
 67,
 73,
 71,
 71,
 89,
 89,
 76,
 86,
 75,
 84,
 73,
 78,
 91,
 78,
 81,
 73,
 72,
 78,
 72,
 70,
 64,
 55,
 80,
 81,
 62,
 74,
 91,
 75]}
```

In [19]:

```
state_and_repititions = pd.DataFrame.from_dict(state_and_repititions)
state_and_repititions
```

Out[19]:

	state_abbr	noOf_repeatitions
0	NE	84
1	CA	78
2	WI	67
3	AP	170
4	AE	167
...	...	...
57	PA	81
58	MD	62
59	SC	74

	state_abbr	noOf_repeatitions
60	GU	91
61	FL	75

62 rows × 2 columns

In [20]:

```
state_and_repititions[state_and_repititions["state_abbr"]=='WY']
```

Out[20]:

	state_abbr	noOf_repeatitions
29	WY	75

We might need create a new dataset with the abbreviations of USA states and its abbreviation because as you can see there are `50` states in USA however our data here posses `12` states that are not from USA so let us figure it out

The below mentioned data is from ->website
`https://www.faa.gov/air_traffic/publications/atpubs/cnt_html/appendix_a.html`

`"Unit 9446"` is likely the unit number assigned to a specific `military installation or organization.` `"Box 0958"` refers to a specific `mailbox` within the `unit`. `"DPO"` stands for `"Duty Postal Office,"` which is a type of `post office` used by the `military` to process mail for service members and their families stationed overseas. `"AE"` is the postal code for the US military postal region in `Europe and Africa.` `"97025"` is likely the `Military Postal Code` for a specific location within the `AE region.`

In [21]:

```
data = { 'State':
        ['Unknown','Unknown', 'Unknown', 'Alabama', 'Alaska', 'Arizona', 'Arkansas',
         'American Samoa' , 'California', 'Colorado', 'Connecticut', 'Delaware',
         'District Of Columbia' , 'Florida', 'Georgia','Guam', 'Hawaii', 'Idaho',
         'Illinois', 'Indiana', 'Iowa', 'Kansas', 'Kentucky', 'Louisiana', 'Maine', 'Mar
yland',
         'Massachusetts', 'Michigan', 'Minnesota', 'Mississippi', 'Missouri', 'Montana',
'Nebraska',
         'Nevada', 'New Hampshire', 'New Jersey', 'New Mexico', 'New York', 'North Carol
ina',
         'North Dakota', 'Northern Mariana Islands' , 'Unknown','Ohio', 'Oklahoma', 'Ore
gon',
         'Pennsylvania','Puerto Rico','Unknown', 'Rhode Island', 'South Carolina', 'Sout
h Dakota',
         'Tennessee', 'Texas', 'Utah', 'Vermont', 'Virginia','Virgin Islands', 'Washingt
on',
         'West Virginia', 'Wisconsin', 'Wyoming','Unknown'],
        'Abbreviation':
        ['AA','AE', 'AP', 'AL', 'AK', 'AZ', 'AR', 'AS', 'CA', 'CO', 'CT', 'DE', 'DC' ,'F
L', 'GA',
         'GU', 'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA', 'MI', '
MN', 'MS',
         'MO', 'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY', 'NC', 'ND', 'MP' ,'MH', 'OH', '
OK', 'OR',
         'PA', 'PR', 'PW', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT', 'VA','VI', 'WA', 'W
V', 'WI',
         'WY','FM']
       }
data
```

Out[21]:

```
{'State': ['Unknown',
  'Unknown',
  'Unknown',
  'Alabama',
```

```
   'Alaska',
   'Arizona',
   'Arkansas',
   'American Samoa',
   'California',
   'Colorado',
   'Connecticut',
   'Delaware',
   'District Of Columbia',
   'Florida',
   'Georgia',
   'Guam',
   'Hawaii',
   'Idaho',
   'Illinois',
   'Indiana',
   'Iowa',
   'Kansas',
   'Kentucky',
   'Louisiana',
   'Maine',
   'Maryland',
   'Massachusetts',
   'Michigan',
   'Minnesota',
   'Mississippi',
   'Missouri',
   'Montana',
   'Nebraska',
   'Nevada',
   'New Hampshire',
   'New Jersey',
   'New Mexico',
   'New York',
   'North Carolina',
   'North Dakota',
   'Northern Mariana Islands',
   'Unknown',
   'Ohio',
   'Oklahoma',
   'Oregon',
   'Pennsylvania',
   'Puerto Rico',
   'Unknown',
   'Rhode Island',
   'South Carolina',
   'South Dakota',
   'Tennessee',
   'Texas',
   'Utah',
   'Vermont',
   'Virginia',
   'Virgin Islands',
   'Washington',
   'West Virginia',
   'Wisconsin',
   'Wyoming',
   'Unknown'],
 'Abbreviation': ['AA',
   'AE',
   'AP',
   'AL',
   'AK',
   'AZ',
   'AR',
   'AS',
   'CA',
   'CO',
   'CT',
   'DE',
   'DC',
   'FL',
```

```
  'GA',
  'GU',
  'HI',
  'ID',
  'IL',
  'IN',
  'IA',
  'KS',
  'KY',
  'LA',
  'ME',
  'MD',
  'MA',
  'MI',
  'MN',
  'MS',
  'MO',
  'MT',
  'NE',
  'NV',
  'NH',
  'NJ',
  'NM',
  'NY',
  'NC',
  'ND',
  'MP',
  'MH',
  'OH',
  'OK',
  'OR',
  'PA',
  'PR',
  'PW',
  'RI',
  'SC',
  'SD',
  'TN',
  'TX',
  'UT',
  'VT',
  'VA',
  'VI',
  'WA',
  'WV',
  'WI',
  'WY',
  'FM']}
```

In [22]:

```
# if inside US terriory or not
address_dict = {
    "state_abbr":new_state_ls,
    "state_name":[],
    "cnf_USA_state":[]
}


for i , j in enumerate(address_dict["state_abbr"]):
    if j in data['Abbreviation']:
        get_index = int(data['Abbreviation'].index(j))
        address_dict["state_name"].append(data['State'][get_index])
        if address_dict["state_name"][i] == "Unknown":
            address_dict['cnf_USA_state'].append(False)
        else :
            address_dict['cnf_USA_state'].append(True)
```

In [23]:

```
address_dict = pd.DataFrame.from_dict(address_dict)
```

```
address_dict
```

Out[23]:

	state_abbr	state_name	cnf_USA_state
0	NE	Nebraska	True
1	CA	California	True
2	WI	Wisconsin	True
3	AP	Unknown	False
4	AE	Unknown	False
...	...	...	...
4995	AP	Unknown	False
4996	AA	Unknown	False
4997	VA	Virginia	True
4998	AE	Unknown	False
4999	NV	Nevada	True

5000 rows × 3 columns

Separating the postal codes, and categorizing it

In [24]:

```
postal_pattern = r"\d+-\d+|\d+"
postal_code = []
for i in range(len(address_col)):
    var = address_col[i]
    matches = re.findall(postal_pattern, var)
    matches = matches[0]
    # print(matches)
    postal_code.append({"postal_code":matches})

address_dict["postal_code"] = pd.DataFrame(postal_code)
address_dict
```

Out[24]:

	state_abbr	state_name	cnf_USA_state	postal_code
0	NE	Nebraska	True	208
1	CA	California	True	188
2	WI	Wisconsin	True	9127
3	AP	Unknown	False	44820
4	AE	Unknown	False	09386
...	...	...	...	...
4995	AP	Unknown	False	30153-7653
4996	AA	Unknown	False	9258
4997	VA	Virginia	True	4215
4998	AE	Unknown	False	73316
4999	NV	Nevada	True	37778

5000 rows × 4 columns

In [25]:

```
var4 = address_col[4995]
```

```
postal_code = re.findall(postal_pattern, var4)

postal_code = postal_code[0]

postal_code
```

Out[25]:

```
'30153-7653'
```

In [26]:

```
address_dict['postal_code'].nunique()
```

Out[26]:

```
4078
```

In [27]:

```
address_dict['state_categorized'] = LabelEncoder().fit_transform(address_dict['state_abbr'])
```

In [28]:

```
address_dict['postal_code_categorized'] = LabelEncoder().fit_transform(address_dict['postal_code'])
```

In [29]:

```
address_dict
```

Out[29]:

	state_abbr	state_name	cnf_USA_state	postal_code	state_categorized	postal_code_categorized
0	NE	Nebraska	True	208	37	835
1	CA	California	True	188	8	746
2	WI	Wisconsin	True	9127	59	3694
3	AP	Unknown	False	44820	4	1814
4	AE	Unknown	False	09386	1	391
...	...	...	...	...	...	...
4995	AP	Unknown	False	30153-7653	4	1203
4996	AA	Unknown	False	9258	0	3748
4997	VA	Virginia	True	4215	55	1692
4998	AE	Unknown	False	73316	1	2946
4999	NV	Nevada	True	37778	41	1514

5000 rows × 6 columns

Filtering the City Name & Now sorting out the city name

In [30]:

```
city_name_pattern = r"\n(.*)[A-Z]{2}"

pattern4 = re.findall(city_name_pattern, address_col[499])

cn  = pattern4[0]

cn1 = cn.rstrip()

cn2 = cn1.replace(',','')
```

```
cn2
```

Out[30]:

```
'East Nicholashaven'
```

In [31]:

```
City_name = []

for i in range(len(address_col)):

    cN = re.findall(city_name_pattern, address_col[i])
    cN1 = cN[0]
    cN2 = cN1.rstrip()
    if ',' in cN2:
        cN2 = cN2.replace(',','')
    City_name.append(cN2)

address_dict["City_name"] = City_name
```

In [32]:

```
address_dict
```

Out[32]:

	state_abbr	state_name	cnf_USA_state	postal_code	state_categorized	postal_code_categorized	City_name
0	NE	Nebraska	True	208	37	835	Laurabury
1	CA	California	True	188	8	746	Lake Kathleen
2	WI	Wisconsin	True	9127	59	3694	Danieltown
3	AP	Unknown	False	44820	4	1814	FPO
4	AE	Unknown	False	09386	1	391	FPO
...	...	...	...	...	...	...	...
4995	AP	Unknown	False	30153-7653	4	1203	FPO
4996	AA	Unknown	False	9258	0	3748	APO
4997	VA	Virginia	True	4215	55	1692	Joshualand
4998	AE	Unknown	False	73316	1	2946	FPO
4999	NV	Nevada	True	37778	41	1514	East Holly

5000 rows × 7 columns

Filtering out the street address

Now lets filter out the part before `\n`

In [33]:

```
street_pattern = r"^(.*)\n"

pattern5 = re.findall(street_pattern, address_col[0])

pattern5
```

Out[33]:

```
['208 Michael Ferry Apt. 674']
```

In [34]:

```
street_address_ = []
```

```
for i in range(len(address_col)):
    street_address = re.findall(street_pattern, address_col[i])
    street_address = street_address[0]
    street_address_.append(street_address)


street_address_= pd.Series(street_address_)
address_dict["street_address"] = street_address_
# street_address_
```

In [35]:

```
address_dict
```

Out[35]:

	state_abbr	state_name	cnf_USA_state	postal_code	state_categorized	postal_code_categorized	City_name	street_ad
0	NE	Nebraska	True	208	37	835	Laurabury	208 Mi Ferry Apt
1	CA	California	True	188	8	746	Lake Kathleen	188 Joh Views
2	WI	Wisconsin	True	9127	59	3694	Danieltown	9127 Eliza Strav
3	AP	Unknown	False	44820	4	1814	FPO	USS Ba
4	AE	Unknown	False	09386	1	391	FPO	U Rayr
...	...	...	...	...	...	...	...	
4995	AP	Unknown	False	30153-7653	4	1203	FPO	USNS Will
4996	AA	Unknown	False	9258	0	3748	APO	PSC 9258
4997	VA	Virginia	True	4215	55	1692	Joshualand	4215 T Garden
4998	AE	Unknown	False	73316	1	2946	FPO	USS Wa
4999	NV	Nevada	True	37778	41	1514	East Holly	37778 Ge Ridges

5000 rows × 8 columns

In [36]:

```
address_dict
```

Out[36]:

	state_abbr	state_name	cnf_USA_state	postal_code	state_categorized	postal_code_categorized	City_name	street_ad
0	NE	Nebraska	True	208	37	835	Laurabury	208 Mi Ferry Apt
1	CA	California	True	188	8	746	Lake Kathleen	188 Joh Views
2	WI	Wisconsin	True	9127	59	3694	Danieltown	9127 Eliza Strav
3	AP	Unknown	False	44820	4	1814	FPO	USS Ba
4	AE	Unknown	False	09386	1	391	FPO	U Rayr
...	...	...	...	...	...	...	...	

4995	state_abbr	state_name	cnf_USA_state	postal_code	state_categorized	postal_code_categorized	City_name	street_addr
4996	AA	Unknown	False	9258	0	3748	APO	PSC 9258
4997	VA	Virginia	True	4215	55	1692	Joshualand	4215 T Garden
4998	AE	Unknown	False	73316	1	2946	FPO	USS Wa
4999	NV	Nevada	True	37778	41	1514	East Holly	37778 Ge Ridges

5000 rows × 8 columns

Assembling the columns together to usa_housing_df

In [37]:

```
usa_housing_df['street_address'] = address_dict['street_address']
```

In [38]:

```
usa_housing_df['city_names'] = address_dict['City_name']
```

In [39]:

```
usa_housing_df['state_name'] = address_dict['state_name']
```

In [40]:

```
usa_housing_df['state_abbr'] = address_dict['state_abbr']
```

In [41]:

```
usa_housing_df['postal_code'] = address_dict['postal_code']
```

In [42]:

```
usa_housing_df['state_wise category'] = address_dict['state_categorized']
```

In [43]:

```
usa_housing_df['categorized_postal_code'] = address_dict['postal_code_categorized']
```

In [44]:

```
usa_housing_df['cnf_USA_state'] = address_dict['cnf_USA_state']
```

In [45]:

```
usa_housing_df
```

Out[45]:

	income	house_area	noof_rooms	noof_bedrooms	population	price	Address	street_addre
0	79545.45857	5.682861	7.009188	4.09	23086.80050	1.059034e+06	208 Michael Ferry Apt. 674\nLaurabury, NE 3701...	208 Mich Ferry Apt. 6
1	79248.64245	6.002900	6.730821	3.09	40173.07217	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...	188 Johns Views Su 0
2	61287.06718	5.865890	8.512727	5.13	36882.15940	1.058988e+06	9127 Elizabeth Stravenue\nDanieltown,	9127 Elizab Straver

	income	house_area	noof_rooms	noof_bedrooms	population	price	WI 06482... Address	Travel street_addre
3	63345.24005	7.188236	5.586729	3.26	34310.24283	1.260617e+06	USS Barnett\nFPO AP 44820	USS Barn
4	59982.19723	5.040555	7.839388	4.23	26354.10947	6.309435e+05	USNS Raymond\nFPO AE 09386	US Raymo
...	...	...	...	...	...	...	...	
4995	60567.94414	7.830362	6.137356	3.46	22837.36103	1.060194e+06	USNS Williams\nFPO AP 30153-7653	USNS Willia
4996	78491.27543	6.999135	6.576763	4.02	25616.11549	1.482618e+06	PSC 9258, Box 8489\nAPO AA 42991-3352	PSC 9258, B 84
4997	63390.68689	7.250591	4.805081	2.13	33266.14549	1.030730e+06	4215 Tracy Garden Suite 076\nJoshualand, VA 01...	4215 Tra Garden Su
4998	68001.33124	5.534388	7.130144	5.44	42625.62016	1.198657e+06	USS Wallace\nFPO AE 73316	USS Walla
4999	65510.58180	5.992305	6.792336	4.07	46501.28380	1.298950e+06	37778 George Ridges Apt. 509\nEast Holly, NV 2...	37778 Geor Ridges A

5000 rows × 15 columns

In [46]:

```
usa_housing_df.to_csv('USA_housing_parsed_data.csv')
```