Video Link :- Video link :- https://drive.google.com/file/d/1yVGuwqfPIA1iTOgnZeY5SkoxzBKnhaeM/view?usp=sharing

Github Link :- https://github.com/rishabh5197/Machine-Learning-Assignments/tree/main/Assessment-1

Assessment - 1 Feature Engineering

Team - 8

20MAI0075 BINAL MANOJ BARIYA

20MAI0076 MERAJ AHMED

20MAI0077 VIVEK DADHICH

20MAI0079 KHEMRAJ GUPTA

20MAI0080 SARANYA ROY

20MAI0081 MESHANK ADHIA

20MAI0082 RISHABH SHARMA

20MAI0083 ASTHA TEMBHRE

In [11]:
```
import numpy as np
import pandas as pd
```

In [18]:
```
read = pd.read_csv("Real_state_data.csv")
```

In [19]:
```
read.head()
```

Out[19]:

	Rooms_in_BHK	Number_of_buildings	Super_build_up_area	location	price_in_cr	Emi_availa
0	2	3.0	781	Bandra	5.75	
1	2	3.0	547	Kanjurmarg West	1.33	
2	1	3.0	510	Kalyan Complex	1.80	
3	3	9.0	1280	Chandivali, Powai	2.75	
4	3	3.0	740	Powai vihar	NaN	

5 rows × 22 columns

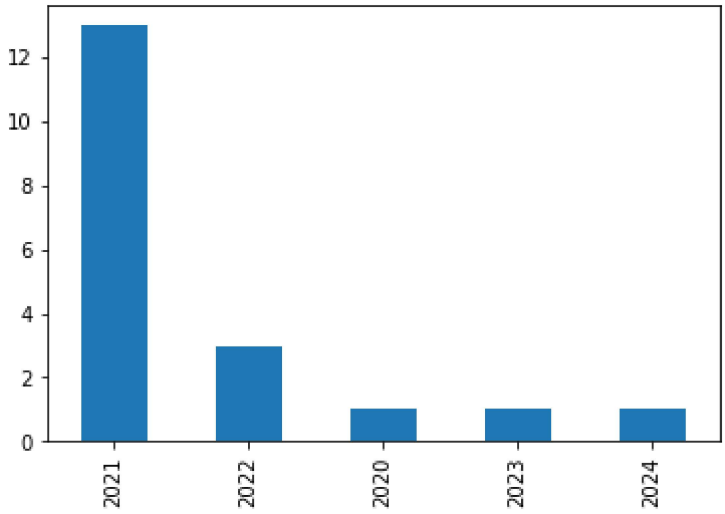
In [49]:
```
1 read.columns
```

Out[49]:
```
Index(['Rooms_in_BHK', 'Number_of_buildings', 'Super_build_up_area',
       'location', 'price_in_cr', 'Emi_available',
       'Emi_amount_in_lac_per_month', 'Furnished', 'Householder',
       'Car_parking_space', 'Outskirts', 'Floor_number', 'Total_floors',
       'Water_availability_24X7', 'Number_of_bathrooms', 'Balconies',
       'Constructed_or_Under_construction', 'Possession', 'Power_back_up',
       'Pet_allowed', 'Facing', 'Purchased', 'New', 'parking_balcony'],
      dtype='object')
```

Feature- 1

In [50]:
```
1 # Count of Possession Year
2
3 read["Possession"].value_counts().plot.bar()
```

Out[50]: <AxesSubplot:>



Feature- 2

In [43]:
```
1 a= read.groupby(by="Constructed_or_Under_construction",).sum()
```

In [51]:
```
1 a
```

Out[51]:

Constructed_or_Under_construction	Rooms_in_BHK	Number_of_buildings	Super_build_up_area	p
Constructed	36	22.0	19294	
Under Construction	8	24.0	3054	

Feature- 3

In [35]:
```
read['parking_balcony'] = ((read.Balconies >= 2) & (read.Car_parking_space == 'Yes'))
```

In [52]:
```
# Display percent of rows where two_and_two == 1
read[read['parking_balcony']==1].shape[0]/read.shape[0]
```

Out[52]: 0.2631578947368421

Feature- 4

In [27]:
```
# Are they New or not
# We have considered  Constructed_or_Under_construction which are in Under Constru
read["New"]= ((read["Constructed_or_Under_construction"] == "Under Construction") 
```

In [53]:
```
read.head()
```

Out[53]:

	Rooms_in_BHK	Number_of_buildings	Super_build_up_area	location	price_in_cr	Emi_availa
0	2	3.0	781	Bandra	5.75	
1	2	3.0	547	Kanjurmarg West	1.33	
2	1	3.0	510	Kalyan Complex	1.80	
3	3	9.0	1280	Chandivali, Powai	2.75	
4	3	3.0	740	Powai vihar	NaN	

5 rows × 24 columns

Feature- 5

In [48]:
```
#display the count of Landlords who owns furnished house
count = read[(read['Householder'] == 'Ownership') & (read['Furnished'] == 'Furnished'
print("The count of Householder who  like Furnished house are :- ",count)
```

The count of Householder who like Furnished house are :- 2

In [ ]:
```

```

In [ ]:
```
read["Parking_and_Balcony"] = ((read["Balconies"]==))
```

In [7]:
```
read["Possession"].dtype
```

Out[7]: dtype('O')

In [4]:
```
1 read['Possession']= pd.to_datetime(read['Possession'])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\anaconda3\lib\site-packages\pandas\core\arrays\datetimes.py in objects_to_datetim
e64ns(data, dayfirst, yearfirst, utc, errors, require_iso8601, allow_object)
   2084         try:
-> 2085             values, tz_parsed = conversion.datetime_to_datetime64(data)
   2086             # If tzaware, these values represent unix timestamps, so we

pandas\_libs\tslibs\conversion.pyx in pandas._libs.tslibs.conversion.datetime_to_datetime
64()

TypeError: Unrecognized value type: <class 'str'>

During handling of the above exception, another exception occurred:

OutOfBoundsDatetime                       Traceback (most recent call last)
<ipython-input-4-e1d48295cba6> in <module>
----> 1 read['Possession']= pd.to_datetime(read['Possession'])

~\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py in to_datetime(arg, err
ors, dayfirst, yearfirst, utc, format, exact, unit, infer_datetime_format, origin, cach
e)
    803             result = arg.map(cache_array)
    804         else:
--> 805             values = convert_listlike(arg._values, format)
    806             result = arg._constructor(values, index=arg.index, name=arg.name)
    807     elif isinstance(arg, (ABCDataFrame, abc.MutableMapping)):

~\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py in _convert_listlike_dat
etimes(arg, format, name, tz, unit, errors, infer_datetime_format, dayfirst, yearfirs
t, exact)
    463         assert format is None or infer_datetime_format
    464         utc = tz == "utc"
--> 465         result, tz_parsed = objects_to_datetime64ns(
    466             arg,
    467             dayfirst=dayfirst,

~\anaconda3\lib\site-packages\pandas\core\arrays\datetimes.py in objects_to_datetim
e64ns(data, dayfirst, yearfirst, utc, errors, require_iso8601, allow_object)
   2088             return values.view("i8"), tz_parsed
   2089         except (ValueError, TypeError):
-> 2090             raise e
   2091
   2092     if tz_parsed is not None:

~\anaconda3\lib\site-packages\pandas\core\arrays\datetimes.py in objects_to_datetim
e64ns(data, dayfirst, yearfirst, utc, errors, require_iso8601, allow_object)
   2073
   2074     try:
-> 2075         result, tz_parsed = tslib.array_to_datetime(
   2076             data,
   2077             errors=errors,

pandas\_libs\tslib.pyx in pandas._libs.tslib.array_to_datetime()
```

```
pandas\_libs\tslib.pyx in pandas._libs.tslib.array_to_datetime()

pandas\_libs\tslib.pyx in pandas._libs.tslib.array_to_datetime()

pandas\_libs\tslib.pyx in pandas._libs.tslib.array_to_datetime()

pandas\_libs\tslibs\conversion.pyx in pandas._libs.tslibs.conversion.convert_datetime_to_
tsobject()

pandas\_libs\tslibs\np_datetime.pyx in pandas._libs.tslibs.np_datetime.check_dts_bounds
()

OutOfBoundsDatetime: Out of bounds nanosecond timestamp: 1-06-23 00:00:00
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [30]:
```
1 read.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19 entries, 0 to 18
Data columns (total 22 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Rooms_in_BHK                  19 non-null     int64
 1   Number_of_buildings           12 non-null     float64
 2   Super_build_up_area           19 non-null     int64
 3   location                      19 non-null     object
 4   price_in_cr                   12 non-null     float64
 5   Emi_available                 19 non-null     object
 6   Emi_amount_in_lac_per_month   17 non-null     float64
 7   Furnished                     19 non-null     object
 8   Householder                   19 non-null     object
 9   Car_parking_space             19 non-null     object
 10  Outskirts                     19 non-null     object
 11  Floor_number                  18 non-null     float64
 12  Total_floors                  19 non-null     int64
 13  Water_availability_24X7       15 non-null     object
 14  Number_of_bathrooms           16 non-null     float64
 15  Balconies                     17 non-null     float64
 16  Constructed_or_Under_construction  19 non-null  object
 17  Possession                    18 non-null     object
 18  Power_back_up                 13 non-null     object
 19  Pet_allowed                   19 non-null     object
 20  Facing                        9 non-null      object
 21  Purchased                     19 non-null     object
dtypes: float64(6), int64(3), object(13)
memory usage: 3.4+ KB
```

In [29]:
```
1 read
```

Out[29]:

	Rooms_in_BHK	Number_of_buildings	Super_build_up_area	location	price_in_cr	Emi_avai
0	2	3.0	781	Bandra	5.75	
1	2	3.0	547	Kanjurmarg West	1.33	
2	1	3.0	510	Kalyan Complex	1.80	
3	3	9.0	1280	Chandivali, Powai	2.75	
4	3	3.0	740	Powai vihar	NaN	
5	1	7.0	700	Mira Road East	0.60	
6	4	NaN	2500	Thane	NaN	
7	4	2.0	2450	Goregaon West	4.15	
8	2	5.0	527	Thane	1.02	
9	1	1.0	450	Khar West	NaN	
10	1	4.0	755	Bandra east	NaN	
11	2	NaN	750	Borivali West	2.00	
12	3	NaN	1250	Virar, West	NaN	
13	3	NaN	2000	Worli	NaN	
14	1	1.0	550	Khar West	NaN	
15	2	NaN	895	Virar	0.37	
16	2	NaN	664	Virar	0.45	
17	4	5.0	3549	Agripada	32.00	
18	3	NaN	1450	Virar	0.68	

19 rows × 22 columns

In [3]:
```
1 read.head()
```

Out[3]:

s	Water_availability_24X7	Number_of_bathrooms	Balconies	Constructed_or_Under_construction	Po
5	Yes	NaN	NaN	Constructed	
4	Yes	2.0	NaN	Under Construction	
4	Yes	1.0	2.0	Constructed	
4	Yes	3.0	2.0	Under Construction	
3	Yes	2.0	1.0	Constructed	

In [4]:
```
1 read.shape
```

Out[4]: (19, 22)

In [5]:
```
1 read.isnull().sum()
```

Out[5]:
```
Rooms_in_BHK                    0
Number_of_buildings             7
Super_build_up_area             0
location                        0
price_in_cr                     7
Emi_available                   0
Emi_amount_in_lac_per_month     2
Furnished                       0
Householder                     0
Car_parking_space               0
Outskirts                       0
Floor_number                    1
Total_floors                    0
Water_availability_24X7         4
Number_of_bathrooms             3
Balconies                       2
Constructed_or_Under_construction    0
Possession                      1
Power_back_up                   6
Pet_allowed                     0
Facing                         10
Purchased                       0
dtype: int64
```

In [6]:
```
1 read.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19 entries, 0 to 18
Data columns (total 22 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Rooms_in_BHK                    19 non-null     int64  
 1   Number_of_buildings             12 non-null     float64
 2   Super_build_up_area             19 non-null     int64  
 3   location                        19 non-null     object 
 4   price_in_cr                     12 non-null     float64
 5   Emi_available                   19 non-null     object 
 6   Emi_amount_in_lac_per_month     17 non-null     float64
 7   Furnished                       19 non-null     object 
 8   Householder                     19 non-null     object 
 9   Car_parking_space               19 non-null     object 
 10  Outskirts                       19 non-null     object 
 11  Floor_number                    18 non-null     float64
 12  Total_floors                    19 non-null     int64  
 13  Water_availability_24X7         15 non-null     object 
 14  Number_of_bathrooms             16 non-null     float64
 15  Balconies                       17 non-null     float64
 16  Constructed_or_Under_construction  19 non-null  object 
 17  Possession                      18 non-null     object 
 18  Power_back_up                   13 non-null     object 
 19  Pet_allowed                     19 non-null     object 
 20  Facing                          9 non-null      object 
 21  Purchased                       19 non-null     object 
dtypes: float64(6), int64(3), object(13)
memory usage: 3.4+ KB
```

In [7]:
```
1 cat_columns = [c for c in read.columns if read[c].dtypes == "O"]
```

In [8]:
```
1 Num_columns = [c for c in read.columns if read[c].dtypes != "O"]
```

In [9]:
```
1 Num_columns
```

Out[9]:
```
['Rooms_in_BHK',
 'Number_of_buildings',
 'Super_build_up_area',
 'price_in_cr',
 'Emi_amount_in_lac_per_month',
 'Floor_number',
 'Total_floors',
 'Number_of_bathrooms',
 'Balconies']
```

In [10]:
```
1 cat_columns
```

Out[10]:
```
['location',
 'Emi_available',
 'Furnished',
 'Householder',
 'Car_parking_space',
 'Outskirts',
 'Water_availability_24X7',
 'Constructed_or_Under_construction',
 'Possession',
 'Power_back_up',
 'Pet_allowed',
 'Facing',
 'Purchased']
```

In [11]:
```
1 # read["Number_of_buildings"].fillna(read["Number_of_buildings"].mode())
2 # read["Number_of_buildings"].fillna(read["Number_of_buildings"],method="mode")
3 for i in Num_columns:
4 #     print(i)
5 #     print(read[i].mode())
6     read[i].fillna(read[i].mode(), inplace=True)
```

In [27]:
```
# read.Super_build_up_area
# read["Number_of_buildings"].fillna(value=read[i].mode(),inplace=True)
for i in Num_columns:
#     print(i)
#     print(read[i].mode())
    value = read[i].mode()
    print(value)
#     read[i].replace("NaN",  value)
```

```
0    2
dtype: int64
0    3.0
dtype: float64
0      450
1      510
2      527
3      547
4      550
5      664
6      700
7      740
8      750
9      755
10     781
11     895
12    1250
13    1280
14    1450
15    2000
16    2450
17    2500
18    3549
dtype: int64
0    1.02
1    1.80
2    4.15
3    5.75
dtype: float64
0    0.42
dtype: float64
0    7.0
dtype: float64
0     3
1    12
2    15
3    17
4    24
dtype: int64
0    2.0
dtype: float64
0    0.0
dtype: float64
```

In [ ]:
```
from sklearn.calibration
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [20]:
```
1 read["Number_of_buildings"]
```

Out[20]:
```
0     3.0
1     3.0
2     3.0
3     9.0
4     3.0
5     7.0
6     NaN
7     2.0
8     5.0
9     1.0
10    4.0
11    NaN
12    NaN
13    NaN
14    1.0
15    NaN
16    NaN
17    5.0
18    NaN
Name: Number_of_buildings, dtype: float64
```

In [ ]:
```
1
```

In [13]:
```
1 read.isnull().sum()
```

Out[13]:
```
Rooms_in_BHK                        0
Number_of_buildings                 7
Super_build_up_area                 0
location                            0
price_in_cr                         3
Emi_available                       0
Emi_amount_in_lac_per_month         2
Furnished                           0
Householder                         0
Car_parking_space                   0
Outskirts                           0
Floor_number                        0
Total_floors                        0
Water_availability_24X7             4
Number_of_bathrooms                 2
Balconies                           1
Constructed_or_Under_construction   0
Possession                          1
Power_back_up                       6
Pet_allowed                         0
Facing                             10
Purchased                           0
dtype: int64
```

In [ ]:
```
1
```

In [14]:
```
1 # read["Number_of_buildings"].mode()
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [15]:
```
1 # read.Number_of_buildings
```

In [ ]:
```
1
```

In [16]:
```
1 # for i in cat_columns:
2 #     print(i,":- ",read[i].unique,end=" ")
```