Name:Snehal S Patil

TrackCode:ML

Task2:Predict The Prices Of House

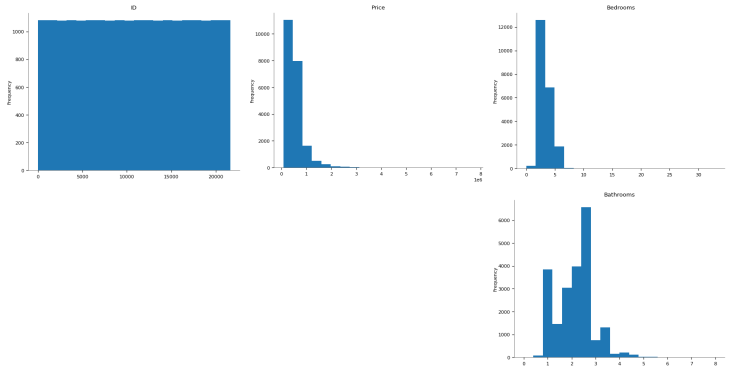
```
1 import pandas as pd
2 import numpy as np
```

```
1 df=pd.read_csv("https://github.com/YBI-Foundation/Dataset/raw/main/House%20Prices.csv")
2 df
```

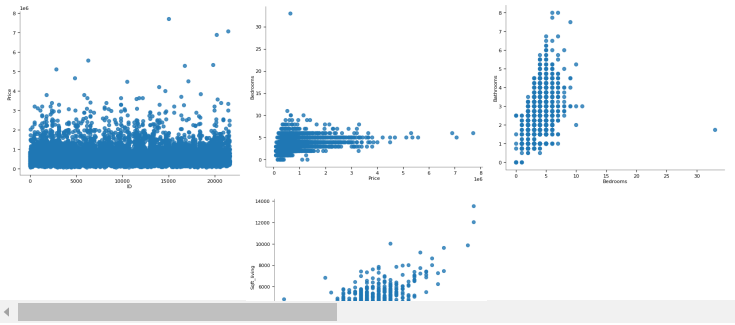
	ID	Date	Price	Bedrooms	Bathrooms	Sqft_living	Sqft_lot	F
0	1	20140916T000000	280000.0	6	3.00	2400	9373	
1	2	20150422T000000	300000.0	6	3.00	2400	9373	
2	3	20140508T000000	647500.0	4	1.75	2060	26036	
3	4	20140811T000000	400000.0	3	1.00	1460	43000	
4	5	20150401T000000	235000.0	3	1.00	1430	7599	
...	...	...	...	...	...	...	...	
21608	21609	20140725T000000	365000.0	5	2.00	1600	4168	
21609	21610	20150311T000000	380000.0	2	1.00	1040	7372	
21610	21611	20140624T000000	339000.0	3	1.00	1100	4128	
21611	21612	20140703T000000	399900.0	2	1.75	1410	1005	
21612	21613	20141030T000000	268950.0	3	1.00	1320	8100	

21613 rows × 21 columns

Distributions



2-d distributions



```
1 df.head()
```

	ID	Date	Price	Bedrooms	Bathrooms	Sqft_living	Sqft_lot	Floors	Waterfront	View	...	Grade	Sqft_above	Sqft_base
0	1	20140916T000000	280000.0	6	3.00	2400	9373	2.0	0	0	...	7	2400	
1	2	20150422T000000	300000.0	6	3.00	2400	9373	2.0	0	0	...	7	2400	

2	3	20140508T000000	647500.0	4	1.75	2060	26036	1.0	0	0	...	8	1160

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   ID             21613 non-null  int64  
 1   Date           21613 non-null  object 
 2   Price          21613 non-null  float64
 3   Bedrooms       21613 non-null  int64  
 4   Bathrooms      21613 non-null  float64
 5   Sqft_living    21613 non-null  int64  
 6   Sqft_lot       21613 non-null  int64  
 7   Floors         21613 non-null  float64
 8   Waterfront     21613 non-null  int64  
 9   View           21613 non-null  int64  
 10  Condition      21613 non-null  int64  
 11  Grade          21613 non-null  int64  
 12  Sqft_above     21613 non-null  int64  
 13  Sqft_basement  21613 non-null  int64  
 14  Yr_built       21613 non-null  int64  
 15  Yr_renovated   21613 non-null  int64  
 16  zipcode        21613 non-null  int64  
 17  Lat            21613 non-null  float64
 18  Long           21613 non-null  float64
 19  Sqft_living15  21613 non-null  int64  
 20  Sqft_lot15     21613 non-null  int64  
dtypes: float64(5), int64(15), object(1)
memory usage: 3.5+ MB
```

```
1 df.describe()
```

	ID	Price	Bedrooms	Bathrooms	Sqft_living	Sqft_lot	Floors	Waterfront	View	Condi
count	21613.00000	2.161300e+04	21613.000000	21613.000000	21613.000000	2.161300e+04	21613.000000	21613.000000	21613.000000	21613.00
mean	10807.00000	5.401822e+05	3.370842	2.114757	2079.899736	1.510697e+04	1.494309	0.007542	0.234303	3.40
std	6239.28002	3.673622e+05	0.930062	0.770163	918.440897	4.142051e+04	0.539989	0.086517	0.766318	0.65
min	1.00000	7.500000e+04	0.000000	0.000000	290.000000	5.200000e+02	1.000000	0.000000	0.000000	1.00
25%	5404.00000	3.219500e+05	3.000000	1.750000	1427.000000	5.040000e+03	1.000000	0.000000	0.000000	3.00
50%	10807.00000	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e+03	1.500000	0.000000	0.000000	3.00
75%	16210.00000	6.450000e+05	4.000000	2.500000	2550.000000	1.068800e+04	2.000000	0.000000	0.000000	4.00
max	21613.00000	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	3.500000	1.000000	4.000000	5.00

```
1 df[['Bedrooms']].value_counts()
2
```

```
Bedrooms
3           9824
4           6882
2           2760
5           1601
6            272
1            199
7             38
0             13
8             13
9              6
10             3
11             1
33             1
Name: count, dtype: int64
```

```
1 df[['Bathrooms']].value_counts()
```

```
Bathrooms
2.50         5380
1.00         3852
1.75         3048
2.25         2047
2.00         1930
```

```
1.50         1446
2.75         1185
3.00          753
3.50          731
3.25          589
3.75          155
4.00          136
4.50          100
4.25           79
0.75           72
4.75           23
5.00           21
5.25           13
5.50           10
0.00           10
1.25            9
6.00            6
0.50            4
5.75            4
6.25            2
6.50            2
6.75            2
8.00            2
7.50            1
7.75            1
Name: count, dtype: int64
```

```
df[['Sqft_living']].value_counts()
```

```
Sqft_living
1300           138
1400           135
1440           133
1660           129
1010           129
               ...
2456             1
2473             1
2478             1
2481             1
13540            1
Name: count, Length: 1038, dtype: int64
```

```
df.columns
```

```
Index(['ID', 'Date', 'Price', 'Bedrooms', 'Bathrooms', 'Sqft_living',
       'Sqft_lot', 'Floors', 'Waterfront', 'View', 'Condition', 'Grade',
       'Sqft_above', 'Sqft_basement', 'Yr_built', 'Yr_renovated', 'zipcode',
       'Lat', 'Long', 'Sqft_living15', 'Sqft_lot15'],
      dtype='object')
```

```
df.shape
```

```
(21613, 21)
```

```
y=df['Price']
```

```
y.shape
```

```
(21613,)
```

```
y
```

```
0        280000.0
1        300000.0
2        647500.0
3        400000.0
4        235000.0
           ...
21608    365000.0
21609    380000.0
21610    339000.0
21611    399900.0
21612    268950.0
Name: Price, Length: 21613, dtype: float64
```

```
1 x=df[['Bedrooms', 'Bathrooms', 'Sqft_living',
2        'Sqft_lot','Sqft_living15', 'Sqft_lot15']]
```

```
1 x.shape
2
```

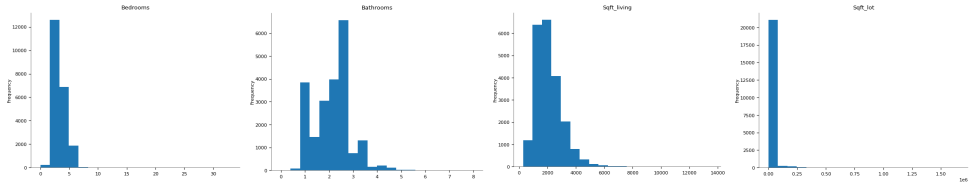
```
(21613, 6)
```

```
1 x
```

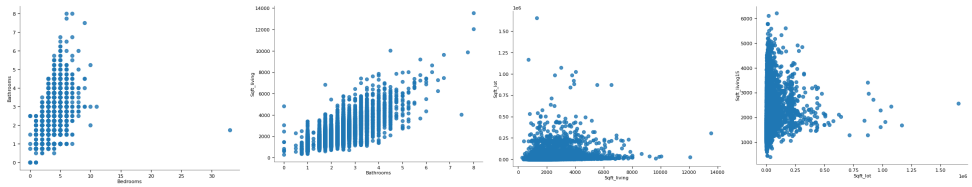
	Bedrooms	Bathrooms	Sqft_living	Sqft_lot	Sqft_living15	Sqft_lot15
0	6	3.00	2400	9373	2060	7316
1	6	3.00	2400	9373	2060	7316
2	4	1.75	2060	26036	2590	21891
3	3	1.00	1460	43000	2250	20023
4	3	1.00	1430	7599	1290	10320
...	...	...	...	...	...	...
21608	5	2.00	1600	4168	1190	4168
21609	2	1.00	1040	7372	1930	5150
21610	3	1.00	1100	4128	1510	4538
21611	2	1.75	1410	1005	1440	1188
21612	3	1.00	1320	8100	1000	8100

21613 rows × 6 columns

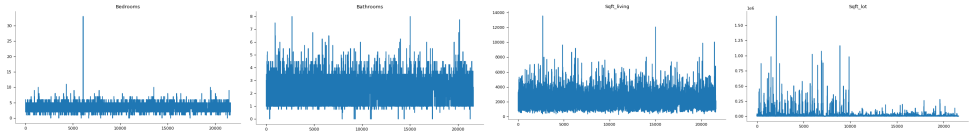
Distributions



2-d distributions



Values




```
1 from sklearn.model_selection import train_test_split
```

```
1 x_train,x_test,y_train,y_test=train_test_split(x,y, test_size=0.3,random_state=222529)
```

```
1 x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

```
((15129, 6), (6484, 6), (15129,), (6484,))
```

```
1 from sklearn.linear_model import LinearRegression
```

```
1 lr=LinearRegression()
```

```
1 lr.fit(x_train,y_train)
```

```
LinearRegression
LinearRegression()
```

```
y_pred=lr.predict(x_test)
```

```
y_pred.shape
```

```
(6484,)
```

```
y_pred
```

```
array([497844.51474271, 307193.5024687 , 440275.88332779, ...,
       501885.22737581, 281005.21986967, 742647.79487191])
```

```
from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score
```

```
mean_squared_error(y_test,y_pred)
```

```
69374029792.9394
```

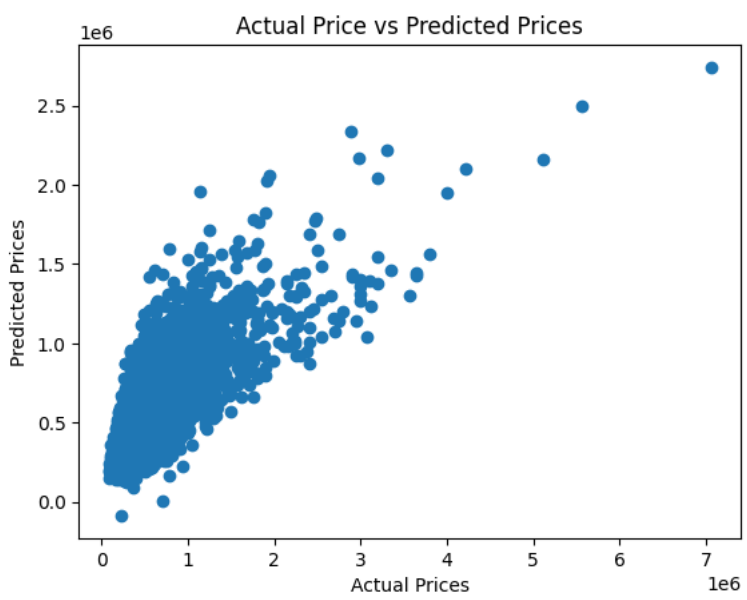
```
mean_absolute_error(y_test,y_pred)
```

```
170909.97763844364
```

```
r2_score(y_test,y_pred)
```

```
0.5214172896943516
```

```
import matplotlib.pyplot as plt
plt.scatter(y_test,y_pred)
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual Price vs Predicted Prices")
plt.show()
```




```
df_new=df.sample(1)
```

```
df_new
```

	ID	Date	Price	Bedrooms	Bathrooms	Sqft_living	Sqft_lot	Floors	Waterfront	View	...	Grade	Sqft_above	Sqft
7671	7672	20141119T000000	1010000.0	4	3.5	3350	3752	2.0	0	0	...	9	2550	

1 rows × 21 columns

```
df_new.shape
```

```
(1, 21)
```

```
x_new=df_new.drop([ 'Bedrooms', 'Bathrooms', 'Sqft_living',
       'Sqft_lot','Sqft_living15', 'Sqft_lot15','Price'],axis=1)
```