colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

house price prediction.ipynb

File Edit View Insert Runtime Tools Help Last saved at March 29

Comment Share

+ Code + Text Connect Colab AI

Importing dataset

```
[ ] import numpy as np
    import pandas as pd
    df=pd.read_csv('/content/data.csv')
    df
```

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

house price prediction.ipynb

File Edit View Insert Runtime Tools Help Last saved at March 29

Comment Share

+ Code + Text Connect Colab AI

```
import numpy as np
import pandas as pd
df=pd.read_csv('/content/data.csv')
df
```

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_re
0	2014-05-02 00:00:00	3.130000e+05	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	
1	2014-05-02 00:00:00	2.384000e+06	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	
2	2014-05-02 00:00:00	3.420000e+05	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966	
3	2014-05-02 00:00:00	4.200000e+05	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963	
4	2014-05-02 00:00:00	5.500000e+05	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976	

ENG 00:06

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

house price prediction.ipynb

File Edit View Insert Runtime Tools Help Last saved at March 29

+ Code + Text

Connect Colab AI

	00:00:00												
4	2014-05-02 00:00:00	5.500000e+05	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4595	2014-07-09 00:00:00	3.081667e+05	3.0	1.75	1510	6360	1.0	0	0	4	1510	0	1954
4596	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
4597	2014-07-09 00:00:00	4.168042e+05	3.0	2.50	3010	7014	2.0	0	0	3	3010	0	2008
4598	2014-07-10 00:00:00	2.034000e+05	4.0	2.00	2090	6630	1.0	0	0	3	1070	1020	1974
4599	2014-07-10 00:00:00	2.268000e+05	3.0	2.50	1490	8102	2.0	0	0	4	1490	0	1990

4600 rows × 18 columns

ENG 00:07

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

house price prediction.ipynb

File Edit View Insert Runtime Tools Help Last saved at March 29

Comment Share

+ Code + Text

Connect Colab AI

```
[ ] df.head()
```

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovate
0	2014-05-02 00:00:00	313000.0	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	200
1	2014-05-02 00:00:00	2384000.0	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	
2	2014-05-02 00:00:00	342000.0	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966	
3	2014-05-02 00:00:00	420000.0	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963	
4	2014-05-02 00:00:00	550000.0	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976	199

ENG 00:07

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

house price prediction.ipynb

File Edit View Insert Runtime Tools Help Last saved at March 29

+ Code + Text

Connect Colab AI

```
df.tail()
```

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_r
4595	2014-07-09 00:00:00	308166.666667	3.0	1.75	1510	6360	1.0	0	0	4	1510	0	1954	
4596	2014-07-09 00:00:00	534333.333333	3.0	2.50	1460	7573	2.0	0	0	3	1460	0	1983	
4597	2014-07-09 00:00:00	416904.166667	3.0	2.50	3010	7014	2.0	0	0	3	3010	0	2009	
4598	2014-07-10 00:00:00	203400.000000	4.0	2.00	2090	6630	1.0	0	0	3	1070	1020	1974	
4599	2014-07-10 00:00:00	220600.000000	3.0	2.50	1490	8102	2.0	0	0	4	1490	0	1990	

ENG 00:07

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

house price prediction.ipynb

File Edit View Insert Runtime Tools Help Last saved at March 29

Comment Share

+ Code + Text

Connect Colab AI

```
df.dtypes
```

```
date             object
price           float64
bedrooms        float64
bathrooms       float64
sqft_living       int64
sqft_lot          int64
floors          float64
waterfront        int64
view              int64
condition         int64
sqft_above        int64
sqft_basement     int64
yr_built          int64
yr_renovated      int64
street           object
city             object
statezip         object
country          object
dtype: object
```

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

```
date            0
```

ENG 00:07

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

house price prediction.ipynb

File Edit View Insert Runtime Tools Help Last saved at March 29

Comment Share

+ Code + Text

Connect Colab AI

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

```
date             0
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
view             0
condition        0
sqft_above       0
sqft_basement    0
yr_built         0
yr_renovated     0
street           0
city             0
statezip         0
country          0
dtype: int64
```

```
[ ] df.columns
```

```
Index(['date', 'price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot',
```

ENG 00:07

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

house price prediction.ipynb

File Edit View Insert Runtime Tools Help Last saved at March 29

Comment Share

+ Code + Text Connect Colab AI

```
[ ] df.columns
```

```
Index(['date', 'price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot',
       'floors', 'waterfront', 'view', 'condition', 'sqft_above',
       'sqft_basement', 'yr_built', 'yr_renovated', 'street', 'city',
       'statezip', 'country'],
      dtype='object')
```

preprocessing

```
[ ] df.loc[df.price==0,'price']=np.NaN
```

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

```
date             0
price           49
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
```

ENG 00:07

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```python
df.isna().sum()
```

```
date             0
price           49
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
view             0
condition        0
sqft_above       0
sqft_basement    0
yr_built         0
yr_renovated     0
street           0
city             0
statezip         0
country          0
dtype: int64
```

```python
df['price']=df['price'].fillna(df['price'].mean())
df.isna().sum()
```

```
date            0
price           0
bedrooms        0
bathrooms       0
```

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```python
from sklearn.preprocessing import LabelEncoder
lab=LabelEncoder()
df['street']=lab.fit_transform(df['street'])
df['city']=lab.fit_transform(df['city'])
df['statezip']=lab.fit_transform(df['statezip'])
df['country']=lab.fit_transform(df['country'])
df
```

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_re
0	2014-05-02 00:00:00	3.130000e+05	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	
1	2014-05-02 00:00:00	2.384000e+06	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	
2	2014-05-02 00:00:00	3.420000e+05	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966	
3	2014-05-02 00:00:00	4.200000e+05	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963	
4	2014-05-02 00:00:00	5.500000e+05	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976	

ENG 00:07

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

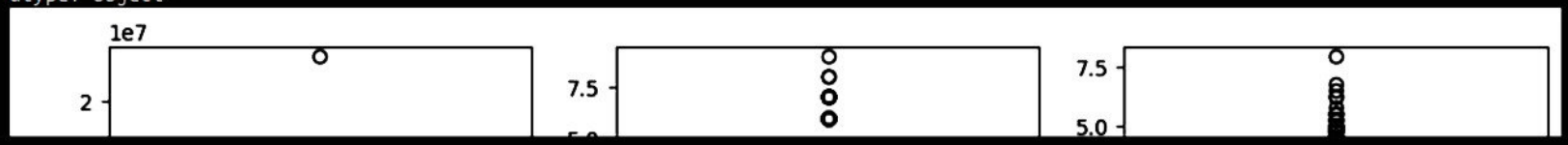
Connect Colab AI

```
df.drop(['date','waterfront','view','street'],axis=1,inplace=True)
```

Removing outliers

```
import matplotlib.pyplot as plt
import seaborn as sns
df.plot(kind='box',figsize=(12,12),layout=(6,3),sharex=False,subplots=True)
```

```
price                Axes(0.125,0.77;0.227941x0.11)
bedrooms           Axes(0.398529,0.77;0.227941x0.11)
bathrooms          Axes(0.672059,0.77;0.227941x0.11)
sqft_living          Axes(0.125,0.638;0.227941x0.11)
sqft_lot           Axes(0.398529,0.638;0.227941x0.11)
floors             Axes(0.672059,0.638;0.227941x0.11)
condition            Axes(0.125,0.506;0.227941x0.11)
sqft_above         Axes(0.398529,0.506;0.227941x0.11)
sqft_basement      Axes(0.672059,0.506;0.227941x0.11)
yr_built             Axes(0.125,0.374;0.227941x0.11)
yr_renovated       Axes(0.398529,0.374;0.227941x0.11)
city               Axes(0.672059,0.374;0.227941x0.11)
statezip             Axes(0.125,0.242;0.227941x0.11)
country            Axes(0.398529,0.242;0.227941x0.11)
dtype: object
```

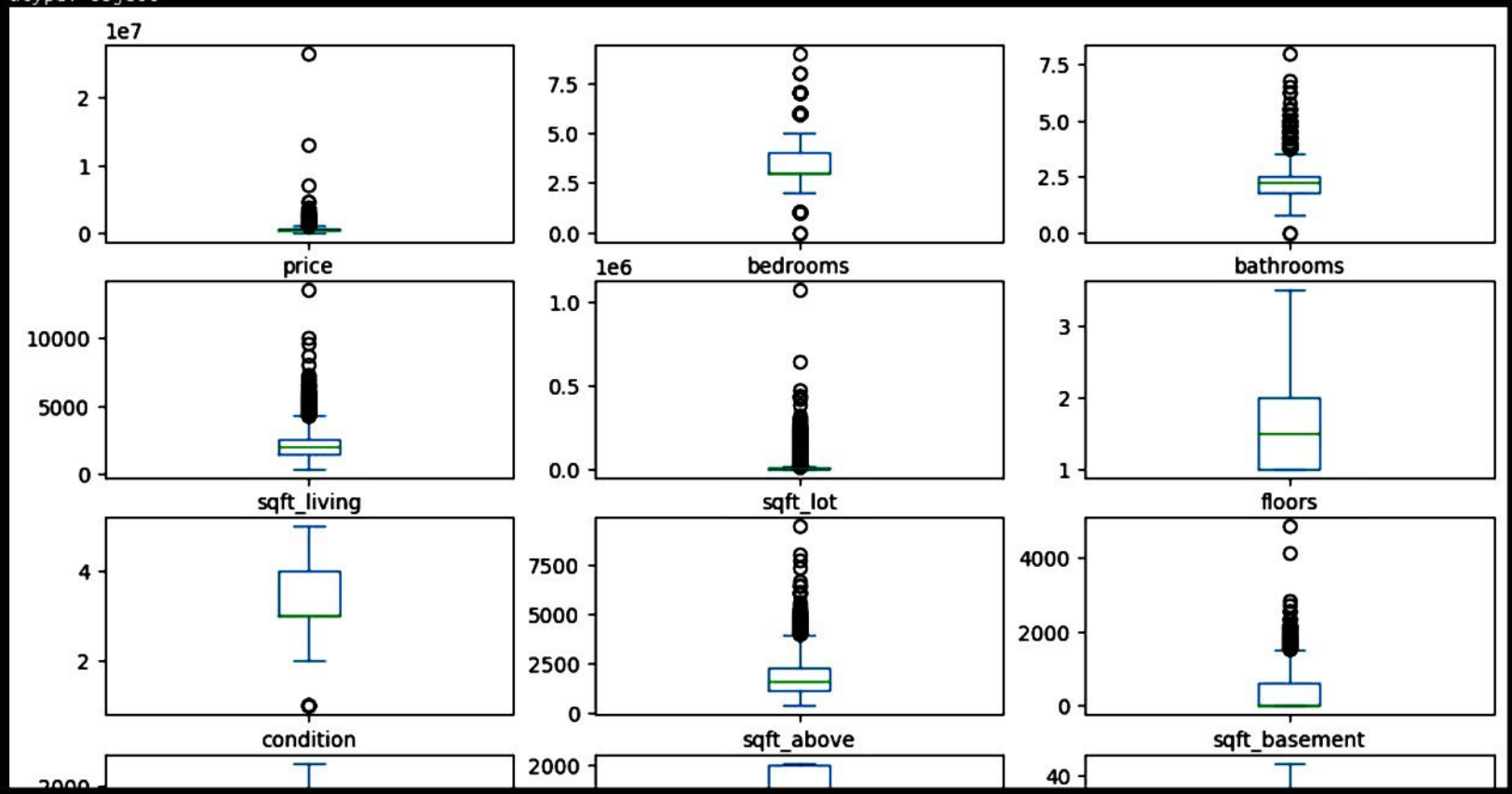


colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

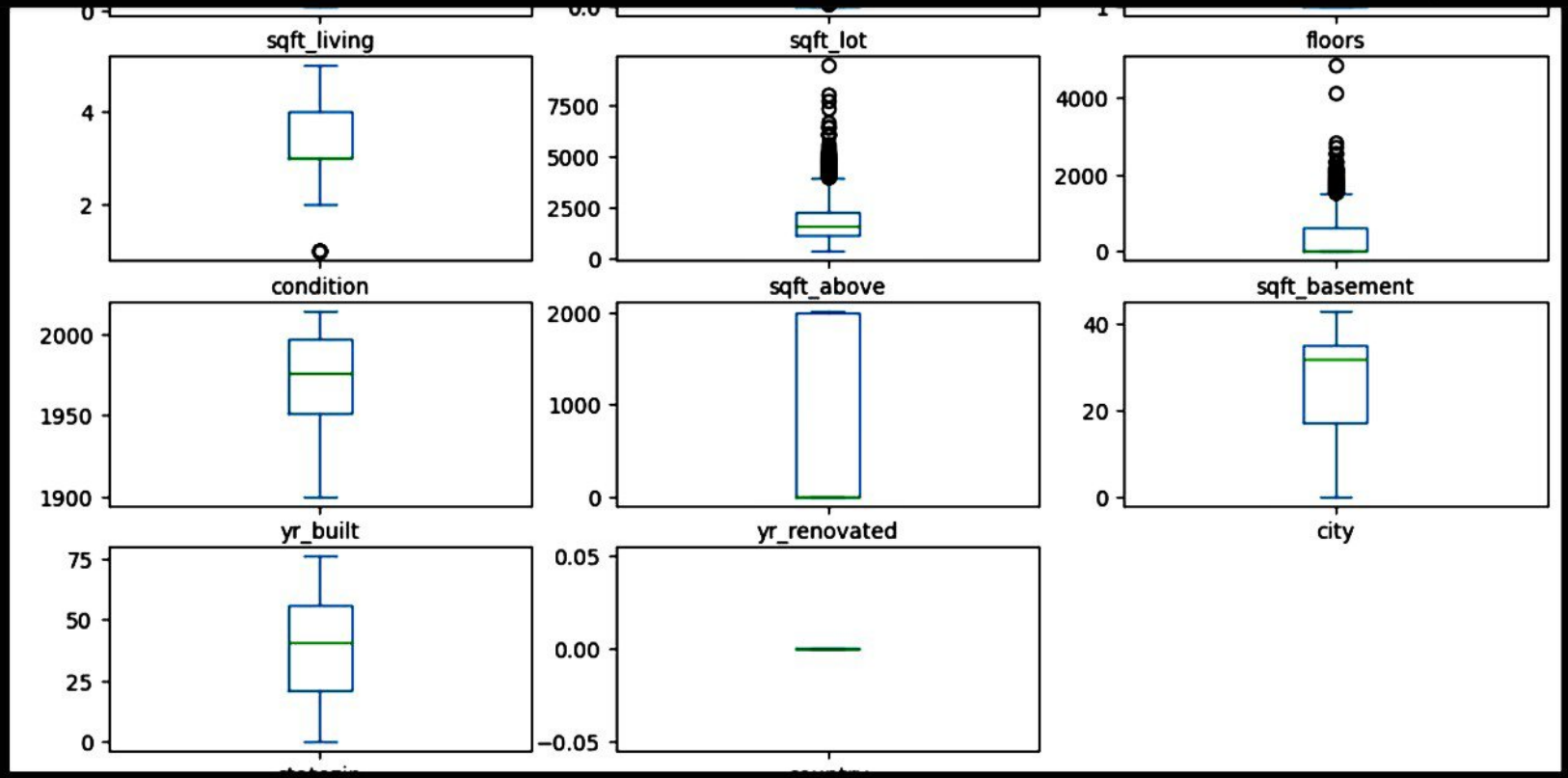
```
dtype: object
```



colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI




```
[ ] def outlierSupresser(column):
```

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```python
def outlierSupresser(column):
    print("----------------{}----------------".format(column.name))
    df = column
    Q1 = df.quantile(0.25)
    Q3 = df.quantile(0.75)
    IQR = Q3 - Q1

    bottomLimit = Q1 - 1.5 * IQR
    topLimit = Q3 + 1.5 * IQR

    print("Q1: {}".format(Q1))
    print("Q3: {}".format(Q3))
    print("bottomLimit: {}".format(bottomLimit))
    print("topLimit: {}".format(topLimit))

    outlierDf = (bottomLimit > df) | (topLimit < df)
    outlierCount = outlierDf.sum()
    print("There is {} outlier data in {} column.\n".format(outlierCount, column.name))

    return column.apply(lambda x: topLimit if topLimit < x else x)
```

```python
df["price"] = outlierSupresser(df["price"])
df["sqft_living"] = outlierSupresser(df["sqft_living"])
df["sqft_lot"] = outlierSupresser(df["sqft_lot"])
df["sqft_above"] = outlierSupresser(df["sqft_above"])
df["sqft_basement"] = outlierSupresser(df["sqft_basement"])
```

ENG 00:08

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
[ ] -----------------price-----------------
    Q1: 328158.9285715
    Q3: 654962.5
    bottomLimit: -162046.42857125
    topLimit: 1145167.85714275
    There is 249 outlier data in price column.

    -----------------sqft_living-----------------
    Q1: 1460.0
    Q3: 2620.0
    bottomLimit: -280.0
    topLimit: 4360.0
    There is 129 outlier data in sqft_living column.

    -----------------sqft_lot-----------------
    Q1: 5000.75
    Q3: 11001.25
    bottomLimit: -4000.0
    topLimit: 20002.0
    There is 541 outlier data in sqft_lot column.

    -----------------sqft_above-----------------
    Q1: 1190.0
    Q3: 2300.0
    bottomLimit: -475.0
    topLimit: 3965.0
    There is 116 outlier data in sqft_above column.

    -----------------sqft_basement-----------------
    Q1: 0.0
```

ENG 00:08

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
bottomLimit: -915.0
topLimit: 1525.0
There is 82 outlier data in sqft_basement column.
```

```
floatColumns = df.select_dtypes(include = ['float64']).columns
df[floatColumns] = df[floatColumns].astype(int)
df.head()
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	sqft_above	sqft_basement	yr_built	yr_renovated	city	statezip	country
0	313000	3	1	1340	7912	1	3	1340	0	1955	2005	36	62	0
1	1145167	5	2	3650	9050	2	5	3370	280	1921	0	35	58	0
2	342000	3	2	1930	11947	1	4	1930	0	1966	0	18	26	0
3	420000	3	2	2000	8030	1	4	1000	1000	1963	0	3	7	0
4	550000	4	2	1940	10500	1	4	1140	800	1976	1992	31	31	0

feature selection

```
x_discrete=df[['city','statezip','country']]
x_continuous=df[['bedrooms','sqft_living','sqft_lot','floors','condition','sqft_above','sqft_basement','yr_built','yr_renovated']]
y_copy=df['price']
```

ENG 00:08

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```python
x_discrete=df[['city','statezip','country']]
x_continuous=df[['bedrooms','sqft_living','sqft_lot','floors','condition','sqft_above','sqft_basement','yr_built','yr_renovated']]
y_copy=df['price']
```

```python
from sklearn.feature_selection import f_classif
score=f_classif(x_continuous,y_copy)
score
```

```
(array([1.30574079, 3.56075571, 1.04450729, 1.27339717, 0.94873394,
        2.56895355, 1.11484848, 1.09139209, 0.96136468]),
 array([3.98274938e-010, 6.28520646e-197, 1.59505141e-001, 1.33256042e-008,
        8.82559101e-001, 3.91313363e-109, 6.40373735e-003, 2.26939996e-002,
        8.12952464e-001]))
```

```python
[ ] f_value=pd.Series(score[0],index=x_continuous.columns)
    f_value.sort_values(ascending=False)
```

```
sqft_living      3.560756
sqft_above       2.568954
bedrooms         1.305741
floors           1.273397
sqft_basement    1.114848
yr_built         1.091392
sqft_lot         1.044507
yr_renovated     0.961365
condition        0.948734
dtype: float64
```

ENG 00:08

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
[ ] p_value=pd.Series(score[1],index=x_continuous.columns)
    p_value.sort_values(ascending=False)
```

```
condition         8.825591e-01
yr_renovated      8.129525e-01
sqft_lot          1.595051e-01
yr_built          2.269400e-02
sqft_basement     6.403737e-03
floors            1.332560e-08
bedrooms          3.982749e-10
sqft_above       3.913134e-109
sqft_living      6.285206e-197
dtype: float64
```

drop the column 'condition' and 'yr_renovated'

```
[ ] df.drop(['yr_renovated'],axis=1,inplace=True)
    df
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	sqft_above	sqft_basement	yr_built	city	statezip	country
0	313000	3	1	1340	7912	1	3	1340	0	1955	36	62	0
1	1145167	5	2	3650	9050	2	5	3370	280	1921	35	58	0
2	342000	3	2	1930	11947	1	4	1930	0	1966	18	26	0
3	420000	3	2	2000	8030	1	4	1000	1000	1963	3	7	0

ENG 00:08

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
[ ] df.drop(['yr_renovated'],axis=1,inplace=True)
    df
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	sqft_above	sqft_basement	yr_built	city	statezip	country
0	313000	3	1	1340	7912	1	3	1340	0	1955	36	62	0
1	1145167	5	2	3650	9050	2	5	3370	280	1921	35	58	0
2	342000	3	2	1930	11947	1	4	1930	0	1966	18	26	0
3	420000	3	2	2000	8030	1	4	1000	1000	1963	3	7	0
4	550000	4	2	1940	10500	1	4	1140	800	1976	31	31	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4595	308166	3	1	1510	6360	1	4	1510	0	1954	35	62	0
4596	534333	3	2	1460	7573	2	3	1460	0	1983	3	6	0
4597	416904	3	2	3010	7014	2	3	3010	0	2009	32	37	0
4598	203400	4	2	2090	6630	1	3	1070	1020	1974	35	71	0
4599	220600	3	2	1490	8102	2	4	1490	0	1990	9	26	0

4600 rows × 13 columns

Data visualization

ENG 00:08

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

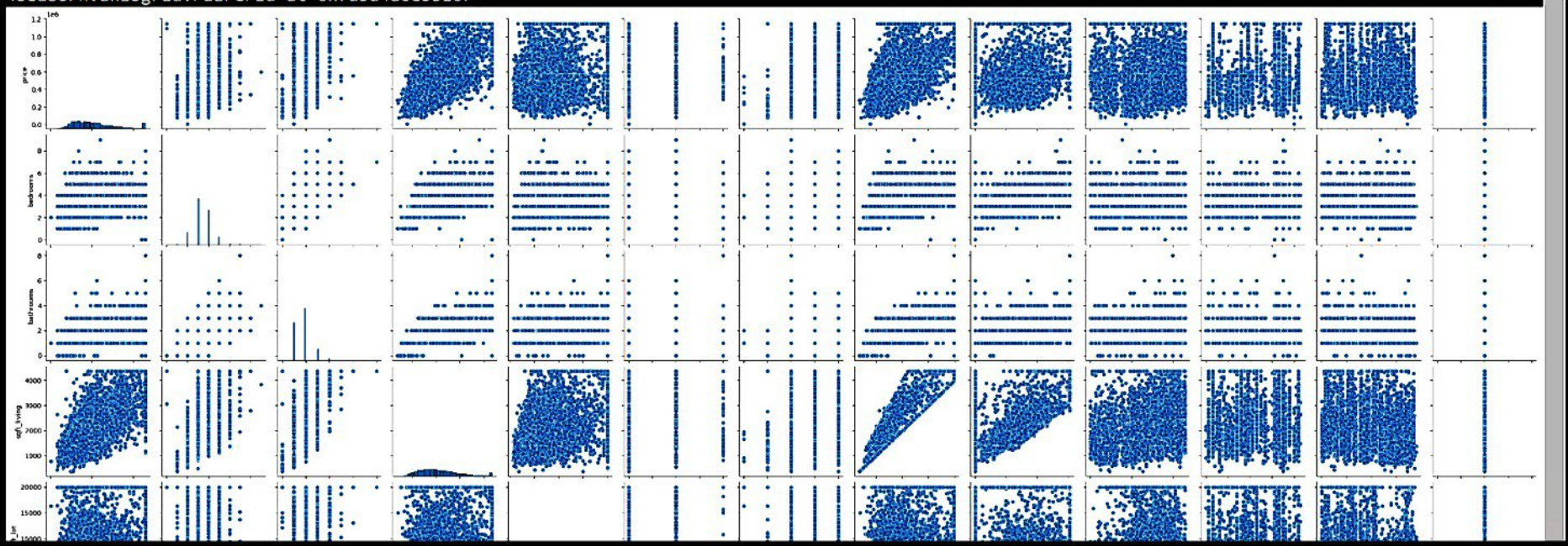
+ Code + Text

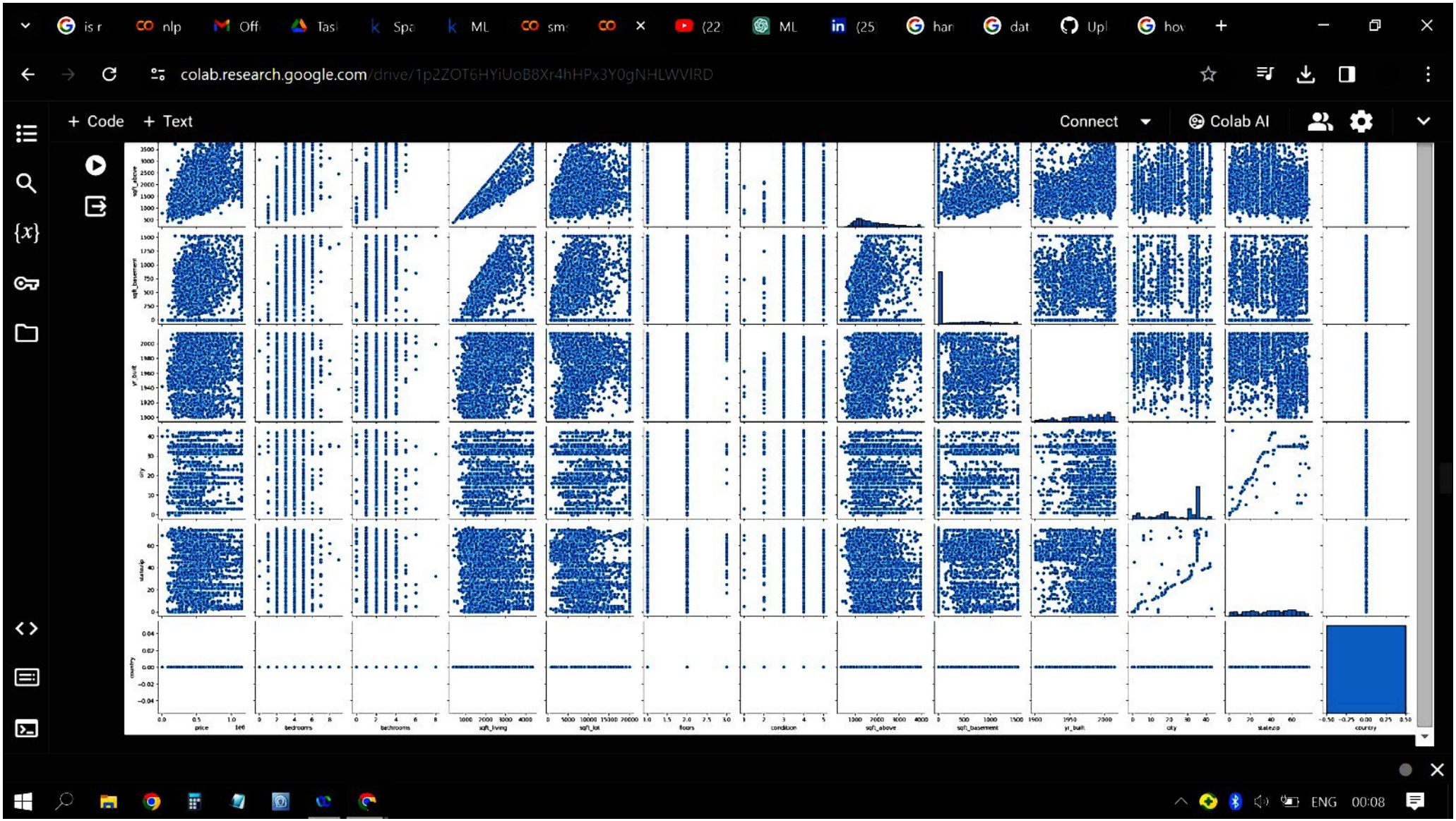
Connect Colab AI

Data visualization

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7d3d4a6e3310>
```




colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD
+ Code
+ Text
Connect
Colab AI
price
bedrooms
bathrooms
sqft_living
sqft_lot
floors
condition
sqft_above
sqft_basement
yr_built
city
statezip
country

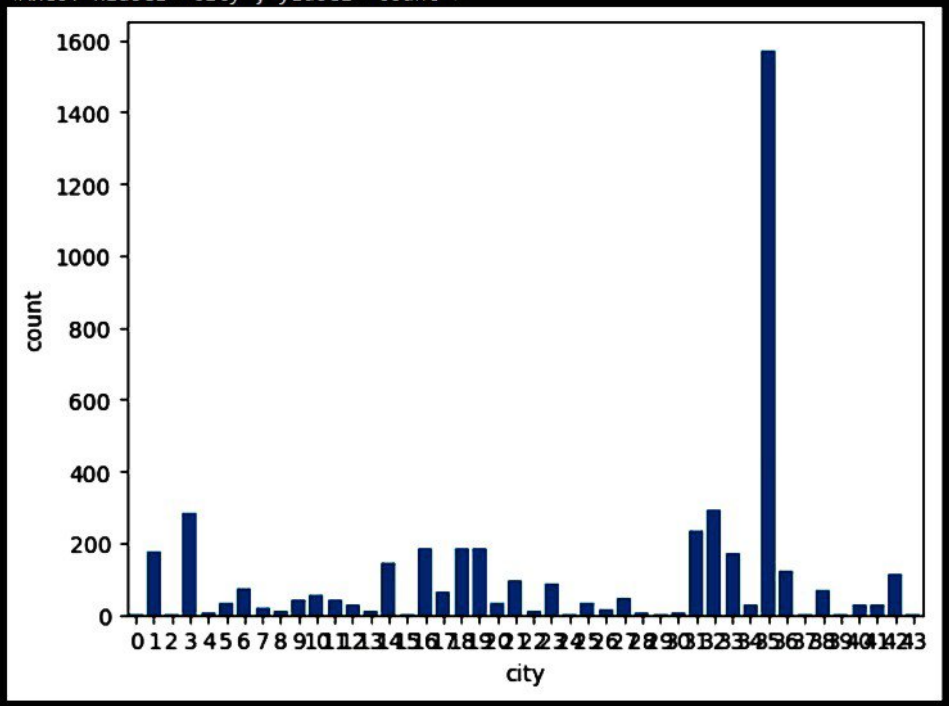
colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
sns.countplot(x='city',data=df)
```

```
<Axes: xlabel='city', ylabel='count'>
```




```
sns.countplot(x='statezip',data=df)
```

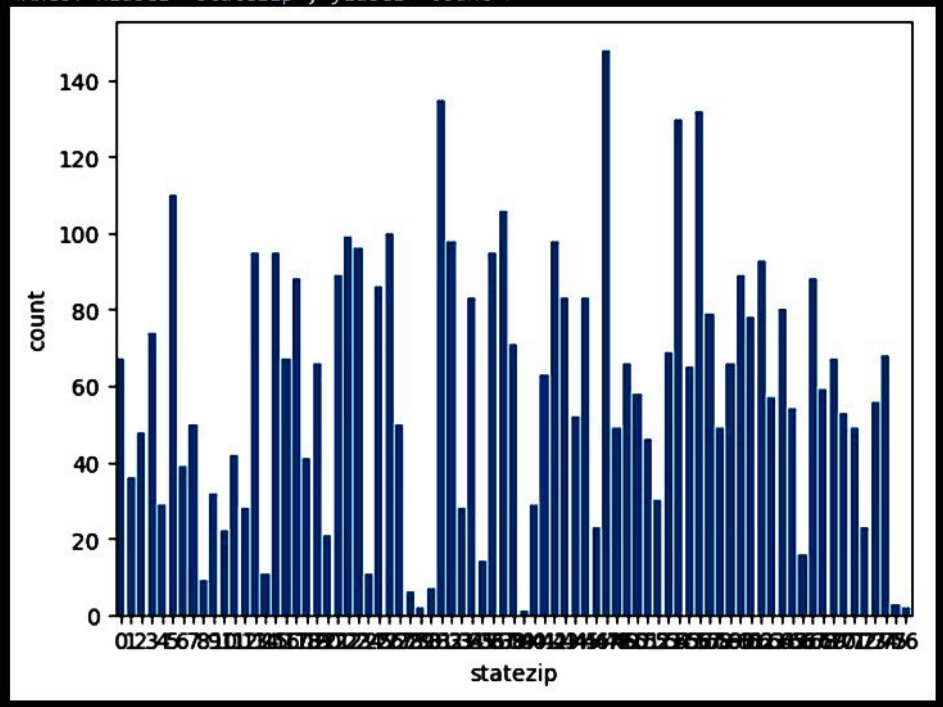
colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
sns.countplot(x='statezip',data=df)
```

```
<Axes: xlabel='statezip', ylabel='count'>
```




```
sns.countplot(x='country',data=df)
```

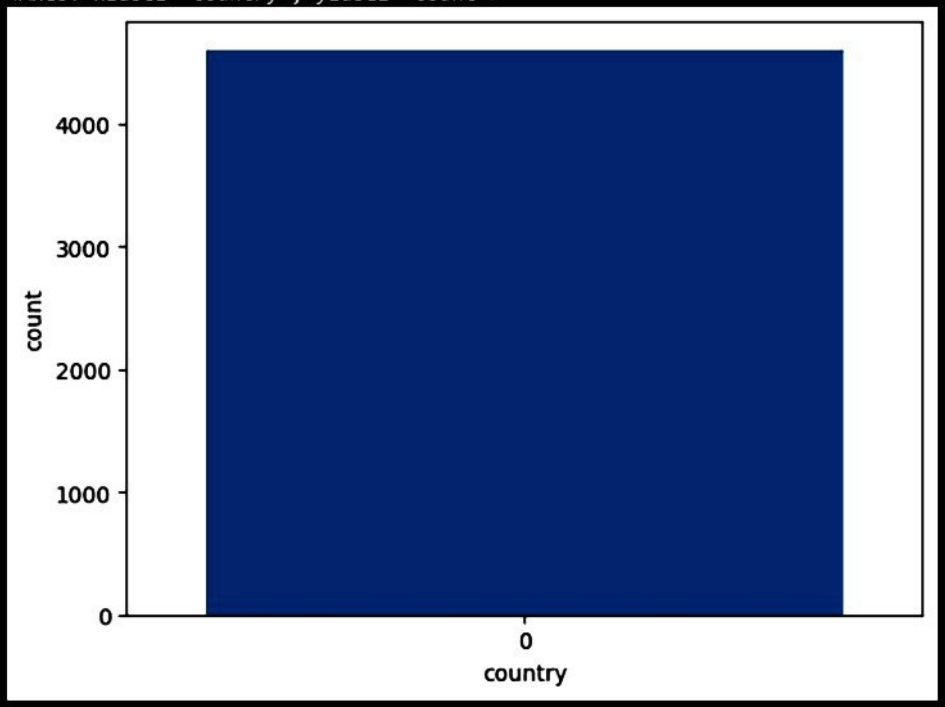
colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
sns.countplot(x='country',data=df)
```

```
<Axes: xlabel='country', ylabel='count'>
```



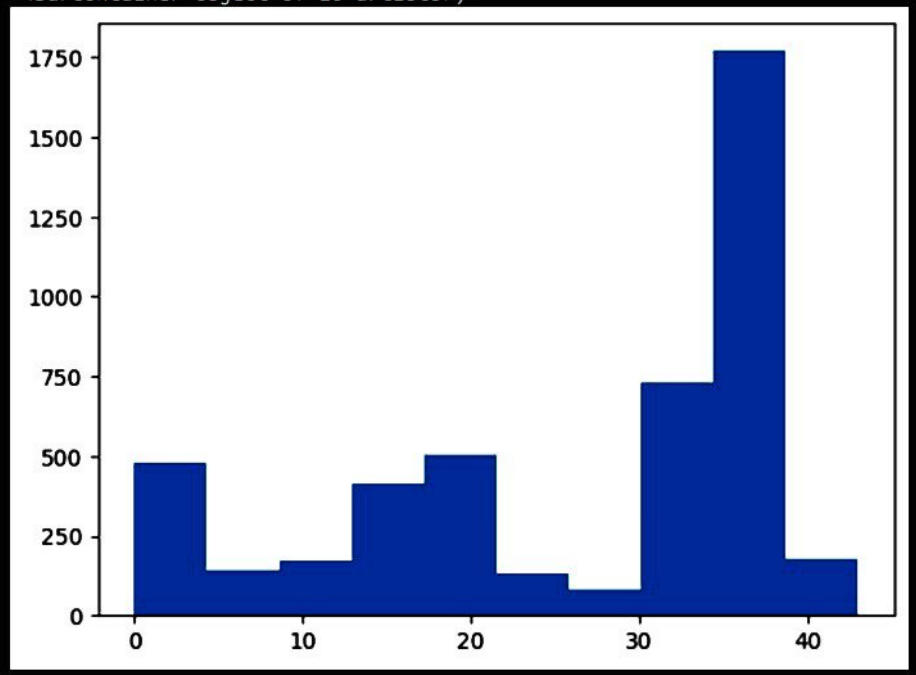
colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
plt.hist(df['city'])
```

```
(array([ 477.,  140.,  171.,  413.,  504.,  132.,   83.,  732., 1770.,
         178.]),
 array([ 0. ,  4.3,  8.6, 12.9, 17.2, 21.5, 25.8, 30.1, 34.4, 38.7, 43. ]),
 <BarContainer object of 10 artists>)
```



ENG 00:08

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
df.corr()
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	sqft_above	sqft_basement	yr_built	city	statezip	cour
price	1.000000	0.352934	0.497848	0.692196	0.167517	0.262519	0.057861	0.596150	0.293789	0.042165	0.071788	-0.051608	
bedrooms	0.352934	1.000000	0.498491	0.610687	0.205573	0.157433	0.025080	0.490990	0.326977	0.142461	-0.130447	-0.152773	
bathrooms	0.497848	0.498491	1.000000	0.691491	0.133373	0.463729	-0.118819	0.632485	0.243448	0.394516	-0.069532	-0.153825	
sqft_living	0.692196	0.610687	0.691491	1.000000	0.369314	0.352074	-0.061743	0.868918	0.401282	0.304907	-0.112065	-0.204127	
sqft_lot	0.167517	0.205573	0.133373	0.369314	1.000000	-0.136099	0.041942	0.351269	0.088008	0.066608	-0.253944	-0.301288	
floors	0.262519	0.157433	0.463729	0.352074	-0.136099	1.000000	-0.306464	0.534263	-0.267486	0.562218	0.035746	-0.087240	
condition	0.057861	0.025080	-0.118819	-0.061743	0.041942	-0.306464	1.000000	-0.183273	0.212909	-0.399698	-0.010592	0.027666	
sqft_above	0.596150	0.490990	0.632485	0.868918	0.351269	0.534263	-0.183273	1.000000	-0.089467	0.425725	-0.121701	-0.252845	
sqft_basement	0.293789	0.326977	0.243448	0.401282	0.088008	-0.267486	0.212909	-0.089467	1.000000	-0.175481	0.004218	0.058066	
yr_built	0.042165	0.142461	0.394516	0.304907	0.066608	0.562218	-0.399698	0.425725	-0.175481	1.000000	-0.211110	-0.339494	
city	0.071788	-0.130447	-0.069532	-0.112065	-0.253944	0.035746	-0.010592	-0.121701	0.004218	-0.211110	1.000000	0.683512	
statezip	-0.051608	-0.152773	-0.153825	-0.204127	-0.301288	-0.087240	0.027666	-0.252845	0.058066	-0.339494	0.683512	1.000000	
country	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

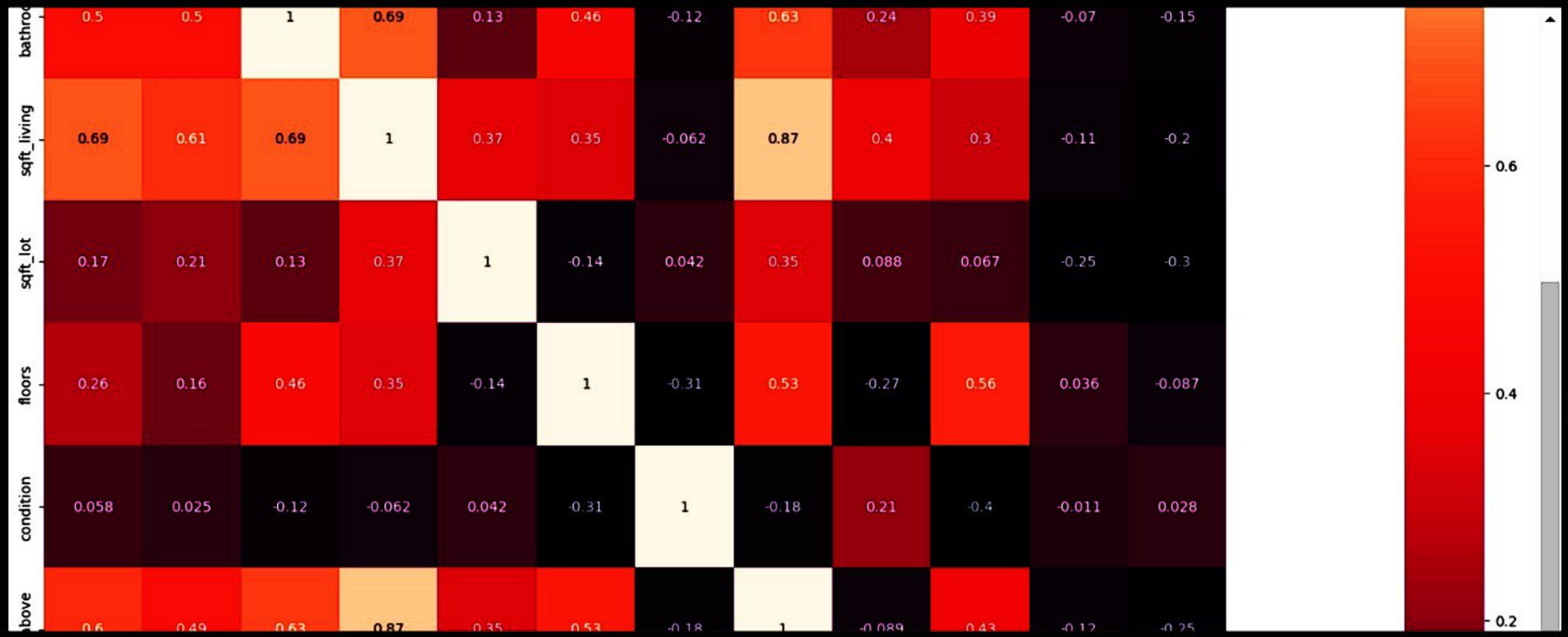
ENG 00:08

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
plt.figure(figsize=(20,20))
sns.heatmap(df.corr(),annot=True)
plt.show()
```



ENG 00:08

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
x=df.drop(['price'],axis=1)
x
```

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	sqft_above	sqft_basement	yr_built	city	statezip	country
0	3	1	1340	7912	1	3	1340	0	1955	36	62	0
1	5	2	3650	9050	2	5	3370	280	1921	35	58	0
2	3	2	1930	11947	1	4	1930	0	1966	18	26	0
3	3	2	2000	8030	1	4	1000	1000	1963	3	7	0
4	4	2	1940	10500	1	4	1140	800	1976	31	31	0
...	...	...	...	...	...	...	...	...	...	...	...	...
4595	3	1	1510	6360	1	4	1510	0	1954	35	62	0
4596	3	2	1460	7573	2	3	1460	0	1983	3	6	0
4597	3	2	3010	7014	2	3	3010	0	2009	32	37	0
4598	4	2	2090	6630	1	3	1070	1020	1974	35	71	0
4599	3	2	1490	8102	2	4	1490	0	1990	9	26	0

4600 rows × 12 columns

```
[ ] y=df['price']
```

ENG 00:08

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
y=df['price']
y
```

```
0        313000
1       1145167
2        342000
3        420000
4        550000
          ...
4595     308166
4596     534333
4597     416904
4598     203400
4599     220600
Name: price, Length: 4600, dtype: int64
```

splitting training and testing data

```
[ ] from sklearn.model_selection import train_test_split
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=1)
    x_train
```

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	sqft_above	sqft_basement	yr_built	city	statezip	country
3990	4	3	3990	9786	2	3	3965	0	2004	31	32	0
1507	4	2	2680	7178	2	3	2680	0	2004	38	38	0

ENG 00:08

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
[ ] from sklearn.model_selection import train_test_split
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=1)
    x_train
```

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	sqft_above	sqft_basement	yr_built	city	statezip	country
3990	4	3	3990	9786	2	3	3965	0	2004	31	32	0
1507	4	2	2680	7178	2	3	2680	0	2004	38	38	0
1652	3	2	2675	20002	2	3	2675	0	1984	33	43	0
2279	6	5	3440	4500	2	3	3280	160	2007	32	35	0
2106	4	1	2220	6500	2	4	2220	0	1964	6	68	0
...	...	...	...	...	...	...	...	...	...	...	...	...
2895	4	1	1220	9600	1	3	1220	0	1980	7	10	0
2763	3	1	1970	8200	1	5	1420	550	1963	36	70	0
905	3	2	1720	1916	2	3	1720	0	2005	32	34	0
3980	4	3	4100	20002	2	3	2500	1525	1981	42	41	0
235	3	2	1540	6760	1	5	1210	330	1980	19	22	0

3220 rows × 12 columns

```
[ ] y_train
```

ENG 00:08

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
y_train
```

```
3990    866000
1507    515000
1652    660000
2279    495000
2106    480000
         ...
2895    310000
2763    540000
905     280000
3980    775000
235     503000
Name: price, Length: 3220, dtype: int64
```

```
[ ] x_test
```

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	sqft_above	sqft_basement	yr_built	city	statezip	country
1351	5	2	2330	10750	1	4	1190	1140	1962	17	16	0
1687	3	2	2730	5832	2	3	2730	0	1998	3	4	0
1297	3	2	1220	1186	3	3	1220	0	2007	35	47	0
2101	4	1	2700	7875	1	4	2700	0	1968	32	36	0
3920	3	4	2080	2250	3	3	2080	0	1997	35	47	0

ENG 00:09

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

model creation

```
[ ] from sklearn.linear_model import LinearRegression
    model=LinearRegression()
    model.fit(x_train,y_train)
    y_pred=model.predict(x_test)
    y_pred
```

```
array([516099.8682461 , 623482.11533822, 366041.73606568, ...,
       447975.22364404, 961346.85621771, 635772.87508342])
```

```
[ ] df1 = pd.DataFrame({'Actual_value': y_test, 'predicted_value': y_pred, 'difference': y_test - y_pred})
```

```
[ ] print(df1)
```

```
      Actual_value  predicted_value     difference
1351        389000    516099.868246 -127099.868246
1687       1000000    623482.115338  376517.884662
1297        355000    366041.736066  -11041.736066
2101        390000    653574.193292 -263574.193292
3920        715000    635880.868351   79119.131649
...            ...              ...            ...
3490        440000    546648.690570 -106648.690570
3584        528000    301258.247921  226741.752079
855         488000    447975.223644   40024.776356
309        1145167    961346.856218  183820.143782
```

ENG 00:09

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
[ ] print(df1)
```

```
      Actual_value  predicted_value     difference
1351        389000    516099.868246 -127099.868246
1687       1000000    623482.115338  376517.884662
1297        355000    366041.736066  -11041.736066
2101        390000    653574.193292 -263574.193292
3920        715000    635880.868351   79119.131649
...            ...              ...            ...
3490        440000    546648.690570 -106648.690570
3584        528000    301258.247921  226741.752079
855         488000    447975.223644   40024.776356
309        1145167    961346.856218  183820.143782
1929        653000    635772.875083   17227.124917

[1380 rows x 3 columns]
```

```
[ ] from sklearn.metrics import mean_absolute_error,mean_absolute_percentage_error,mean_squared_error,r2_score
    print("MAE IS",mean_absolute_error(y_test,y_pred))
    print("MAPE IS",mean_absolute_percentage_error(y_test,y_pred))
    msq=mean_squared_error(y_test,y_pred)
    print("MSE IS",msq)
    sq=np.sqrt(msq)
    print("RMSE IS",sq)
    print("R2 SCORE IS",r2_score(y_test,y_pred))
```

```
MAE IS 132272.75476824856
MAPE IS 0.3095557509564483
MSE IS 29195366483.321426
```

ENG 00:09

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
[ ] from sklearn.metrics import mean_absolute_error,mean_absolute_percentage_error,mean_squared_error,r2_score
    print("MAE IS",mean_absolute_error(y_test,y_pred))
    print("MAPE IS",mean_absolute_percentage_error(y_test,y_pred))
    msq=mean_squared_error(y_test,y_pred)
    print("MSE IS",msq)
    sq=np.sqrt(msq)
    print("RMSE IS",sq)
    print("R2 SCORE IS",r2_score(y_test,y_pred))
```

```
MAE IS 132272.75476824856
MAPE IS 0.3095557509564483
MSE IS 29195366483.321426
RMSE IS 170866.51656577256
R2 SCORE IS 0.5523464513995682
```

```
[ ] from sklearn.tree import DecisionTreeClassifier
    from sklearn.ensemble import RandomForestClassifier
    d_model=DecisionTreeClassifier()
    r_model=RandomForestClassifier(n_estimators=100,max_depth=16,max_features=10)
    model=[d_model,r_model]
```

```
[ ] for i in model:
      print("model is ",i)
      i.fit(x_train,y_train)
      y_pred=i.predict(x_test)
      print("*"*1000)
```

ENG 00:09

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
[ ] from sklearn.tree import DecisionTreeClassifier
    from sklearn.ensemble import RandomForestClassifier
    d_model=DecisionTreeClassifier()
    r_model=RandomForestClassifier(n_estimators=100,max_depth=16,max_features=10)
    model=[d_model,r_model]
```

```
[ ] for i in model:
      print("model is ",i)
      i.fit(x_train,y_train)
      y_pred=i.predict(x_test)
      print("*"*1000)
      print("MAE IS",mean_absolute_error(y_test,y_pred))
      print("MAPE IS",mean_absolute_percentage_error(y_test,y_pred))
      msq=mean_squared_error(y_test,y_pred)
      print("MSE IS",msq)
      sq=np.sqrt(msq)
      print("R2_SCORE IS",r2_score(y_test,y_pred))
      print("*"*1000)
```

```
model is  DecisionTreeClassifier()
**********************************************************************************************************************************************
MAE IS 152982.12463768115
MAPE IS 0.33598627409748516
MSE IS 47659719666.32174
R2_SCORE IS 0.2692318952009428
**********************************************************************************************************************************************
model is  RandomForestClassifier(max_depth=16, max_features=10)
```

ENG 00:09

colab.research.google.com/drive/1p2ZOT6HYiUoB8Xr4hHPx3Y0gNHLWVlRD

+ Code + Text

Connect Colab AI

```
[ ] for i in model:
      print("model is ",i)
      i.fit(x_train,y_train)
      y_pred=i.predict(x_test)
      print("*"*1000)
      print("MAE IS",mean_absolute_error(y_test,y_pred))
      print("MAPE IS",mean_absolute_percentage_error(y_test,y_pred))
      msq=mean_squared_error(y_test,y_pred)
      print("MSE IS",msq)
      sq=np.sqrt(msq)
      print("R2_SCORE IS",r2_score(y_test,y_pred))
      print("*"*1000)
```

```
model is  DecisionTreeClassifier()
**********************************************************************************************************************
MAE IS 152982.12463768115
MAPE IS 0.33598627409748516
MSE IS 47659719666.32174
R2_SCORE IS 0.2692318952009428
**********************************************************************************************************************
model is  RandomForestClassifier(max_depth=16, max_features=10)
**********************************************************************************************************************
MAE IS 149472.7144927536
MAPE IS 0.33921404773743496
MSE IS 49693725584.10725
R2_SCORE IS 0.23804441319104497
**********************************************************************************************************************
```

ENG 00:09