HOUSE PRICE PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

IMPORTING LIBRARIES

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

LOADING DATASET

```
df=pd.read_csv('/content/house price data.csv')
df
```

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated	street	city	st
0	2014-05-02 00:00:00	3.130000e+05	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	2005	18810 Densmore Ave N	Shoreline	
1	2014-05-02 00:00:00	2.384000e+06	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	0	709 W Blaine St	Seattle	

0s completed at 10:38

HOUSE PRICE PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

Next steps: Generate code with df View recommended plots

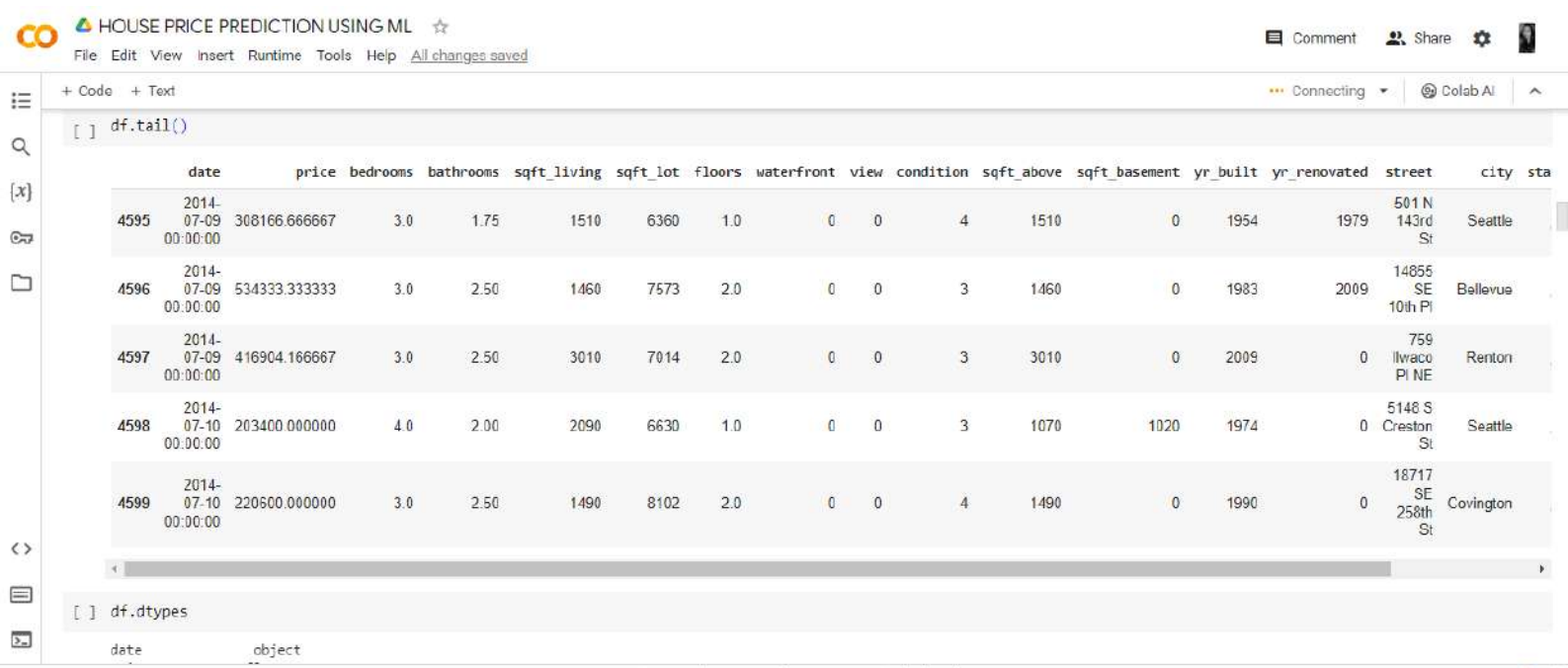
+ Code + Text

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

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated	street	city	statezip
0	2014-05-02 00:00:00	313000.0	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	2005	18810 Densmore Ave N	Shoreline	WA 98133
1	2014-05-02 00:00:00	2384000.0	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	0	709 W Blaine St	Seattle	WA 98119
2	2014-05-02 00:00:00	342000.0	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966	0	26206-26214 143rd Ave SE	Kent	WA 98042
3	2014-05-02 00:00:00	420000.0	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963	0	857 170th Pl NE	Bellevue	WA 98008
4	2014-05-02 00:00:00	550000.0	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976	1992	9105 170th Ave NE	Redmond	WA 98052

Next steps: Generate code with df View recommended plots

0s completed at 10:39


HOUSE PRICE PREDICTION USING ML
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
Connecting
Colab AI
[ ] df.tail()
date price bedrooms bathrooms sqft_living sqft_lot floors waterfront view condition sqft_above sqft_basement yr_built yr_renovated street city sta
4595 2014-07-09 00:00:00 308166.666667 3.0 1.75 1510 6360 1.0 0 0 4 1510 0 1954 1979 501 N 143rd St Seattle
4596 2014-07-09 00:00:00 534333.333333 3.0 2.50 1460 7573 2.0 0 0 3 1460 0 1983 2009 14855 SE 10th Pl Bellevue
4597 2014-07-09 00:00:00 416904.166667 3.0 2.50 3010 7014 2.0 0 0 3 3010 0 2009 0 759 Ilwaco Pl NE Renton
4598 2014-07-10 00:00:00 203400.000000 4.0 2.00 2090 6630 1.0 0 0 3 1070 1020 1974 0 5148 S Creston St Seattle
4599 2014-07-10 00:00:00 220600.000000 3.0 2.50 1490 8102 2.0 0 0 4 1490 0 1990 0 18717 SE 258th St Covington
[ ] df.dtypes
date object
Connecting to Python 3 Google Compute Engine backend

HOUSE PRICE PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk 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
```

FINDING MISSING VALUES

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

Connected to Python 3 Google Compute Engine backend

HOUSE PRICE PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

FINDING MISSING VALUES

```
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
```

DROPPING UNWANTED COLUMNS

Connected to Python 3 Google Compute Engine backend

HOUSE PRICE PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

DROPPING UNWANTED COLUMNS

```
df1=df.drop(['street','city','country','date','statezip','sqft_above'],axis=1)
df1
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_basement	yr_built	yr_renovated
0	3.130000e+05	3.0	1.50	1340	7912	1.5	0	0	3	0	1955	2005
1	2.384000e+06	5.0	2.50	3650	9050	2.0	0	4	5	200	1921	0
2	3.420000e+05	3.0	2.00	1930	11947	1.0	0	0	4	0	1966	0
3	4.200000e+05	3.0	2.25	2000	8030	1.0	0	0	4	1000	1963	0
4	5.500000e+05	4.0	2.50	1940	10500	1.0	0	0	4	800	1976	1992
...	...	...	...	...	...	...	...	...	...	...	...	...
4595	3.081667e+05	3.0	1.75	1510	6360	1.0	0	0	4	0	1954	1979
4596	5.343333e+05	3.0	2.50	1460	7573	2.0	0	0	3	0	1983	2009
4597	4.169042e+05	3.0	2.50	3010	7014	2.0	0	0	3	0	2009	0
4598	2.034000e+05	4.0	2.00	2090	6630	1.0	0	0	3	1020	1974	0
4599	2.206000e+05	3.0	2.50	1490	8102	2.0	0	0	4	0	1990	0

Connected to Python 3 Google Compute Engine backend

HOUSE PRICE PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

```
df1.describe()
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_basement	yr_built	yr_renovated
count	4.600000e+03	4600.000000	4600.000000	4600.000000	4.600000e+03	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000
mean	5.519630e+05	3.400870	2.160815	2139.346957	1.485252e+04	1.512065	0.007174	0.240652	3.451739	312.081522	1970.786304	808.608261
std	5.638347e+05	0.908848	0.783781	963.206916	3.588444e+04	0.538288	0.084404	0.778405	0.677230	464.137228	29.731848	979.414536
min	0.000000e+00	0.000000	0.000000	370.000000	6.380000e+02	1.000000	0.000000	0.000000	1.000000	0.000000	1900.000000	0.000000
25%	3.228750e+05	3.000000	1.750000	1460.000000	5.000750e+03	1.000000	0.000000	0.000000	3.000000	0.000000	1951.000000	0.000000
50%	4.609435e+05	3.000000	2.250000	1980.000000	7.683000e+03	1.500000	0.000000	0.000000	3.000000	0.000000	1976.000000	0.000000
75%	6.549625e+05	4.000000	2.500000	2620.000000	1.100125e+04	2.000000	0.000000	0.000000	4.000000	610.000000	1997.000000	1999.000000
max	2.659000e+07	9.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	4.000000	5.000000	4820.000000	2014.000000	2014.000000

```
[9] sns.heatmap(df1.corr())
```

```
<Axes: >
```

Connected to Python 3 Google Compute Engine backend

HOUSE PRICE PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

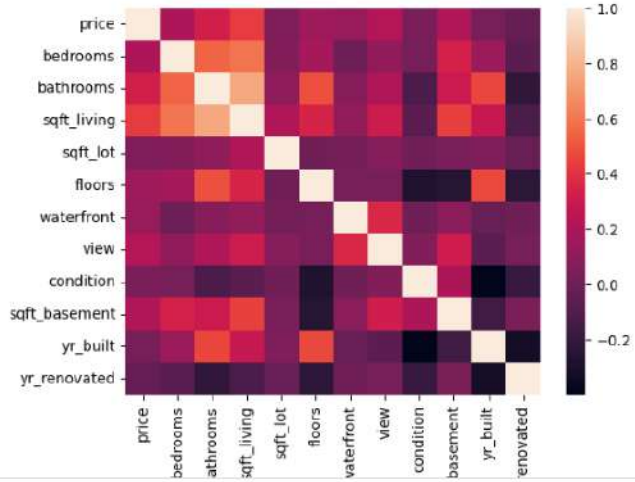
Comment Share

+ Code + Text

RAM Disk Colab AI

```
sns.heatmap(df1.corr())
```

```
<Axes: >
```



Connected to Python 3 Google Compute Engine backend

HOUSE PRICE PREDICTION USING ML

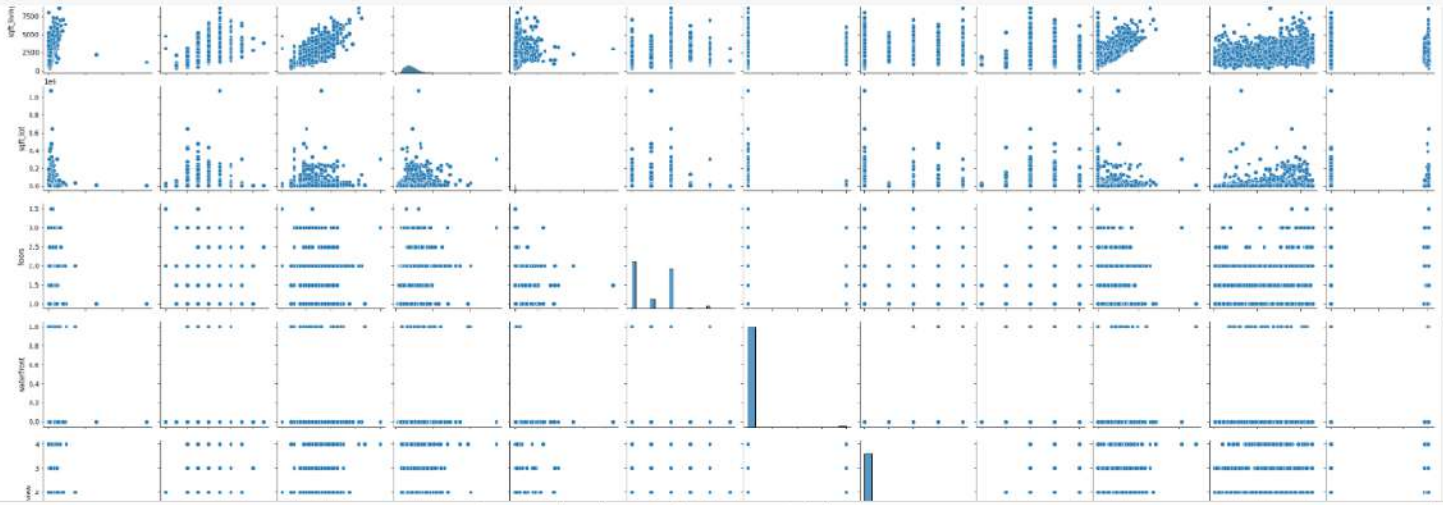
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

```
sns.pairplot(df1)
```



Connected to Python 3 Google Compute Engine backend

HOUSE PRICE PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

SPLITTING DATASET INTO X AND Y VARIABLES

```
[11] x=df1.drop("price",axis=1,inplace=False)
     y=df1.loc[:,"price"]
```

IMPORTING TRAIN TEST SPLIT

```
[12] 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=42)
     x_train
```

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_basement	yr_built	yr_renovated
2194	6.0	1.50	1930	8400	1.0	0	0	3	900	1971	0
958	2.0	2.25	1230	1280	2.0	0	0	3	270	2012	1912
1533	3.0	1.50	1470	14821	1.0	0	0	4	0	1958	1972
654	4.0	2.50	2700	4004	2.0	0	0	3	0	2004	2003
1497	2.0	1.00	740	9003	1.0	0	0	3	0	1949	1998

Connected to Python 3 Google Compute Engine backend

HOUSE PRICE PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

Focus the last run cell

```
[13] x_test
```

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_basement	yr_built	yr_renovated
3683	3.0	2.50	1460	1613	2.0	0	0	3	280	2007	0
4411	5.0	2.25	2000	7900	1.0	0	0	4	700	1986	0
2584	3.0	3.25	2940	5432	3.0	0	3	4	500	1978	2000
69	3.0	2.50	2200	7350	1.0	0	0	5	630	1988	0
1844	3.0	2.50	1720	8755	1.0	0	0	3	720	1983	2009
...	...	...	...	...	...	...	...	...	...	...	...
3437	6.0	3.50	3600	6875	2.0	0	0	3	860	2004	2003
3340	4.0	2.50	3920	12415	2.0	0	0	3	0	1997	0
1289	3.0	2.00	1490	8371	1.5	0	0	3	0	1984	0
449	3.0	1.75	1780	120661	1.0	0	0	4	0	1979	0
3774	4.0	3.00	2130	6325	1.0	0	0	5	690	1948	1985

1380 rows × 11 columns

Next steps: Generate code with x_test View recommended plots

Connected to Python 3 Google Compute Engine backend

HOUSE PRICE PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

```
y_train
```

```
2194    175000.000000
958     387000.000000
1533    264000.000000
654     723000.000000
1497    157500.000000
            ...
4426    282508.888889
466     875000.000000
3092    290000.000000
3772    600000.000000
860     475000.000000
Name: price, Length: 3220, dtype: float64
```

```
[15] y_test
```

```
3683     544000.0
4411          0.0
2584    1712500.0
69       365000.0
1844     275000.0
            ...
3437     620000.0
3340     770000.0
1289     255000.0
449      336900.0
3774     620000.0
Name: price, Length: 1380, dtype: float64
```

Connected to Python 3 Google Compute Engine backend

HOUSE PRICE PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

NORMALISATION USING STANDARD SCALER

```
[16] from sklearn.preprocessing import StandardScaler
     scaler=StandardScaler()
     x_train=scaler.fit_transform(x_train)
     x_test=scaler.fit_transform(x_test)
```

MODEL CREATION

LINEAR REGRESSION MODEL

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

```
array([ 305533.65307581,  322764.73100011, 1062656.22404855, ...,
        324554.37106862,  352705.58761827,  594789.76729055])
```

Connected to Python 3 Google Compute Engine backend

HOUSE PRICE PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

GRADIENT BOOSTING REGRESSOR MODEL

```
[18] from sklearn import ensemble
     gbr_model=ensemble.GradientBoostingRegressor()
     gbr_model.fit(x_train,y_train)
     y_pred=gbr_model.predict(x_test)
     y_pred
```

```
array([ 417794.69797032,  315020.67587048, 1010716.2945737 , ...,
        323599.11909502,  391084.70460385,  540441.23609539])
```

DECISION TREE REGRESSOR MODEL

```
from sklearn.tree import DecisionTreeRegressor
dt_model=LinearRegression()
dt_model.fit(x_train,y_train)
y_pred=dt_model.predict(x_test)
y_pred
```

```
array([ 305533.65307581,  322764.73100011, 1062656.22404855, ...,
        324554.37106862,  352705.58761827,  594789.76729055])
```

Connected to Python 3 Google Compute Engine backend

HOUSE PRICE PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

RANDOM FOREST REGRESSOR MODEL

```
[20] from sklearn.ensemble import RandomForestRegressor
     rf_model=RandomForestRegressor()
     rf_model.fit(x_train,y_train)
     y_pred=rf_model.predict(x_test)
     y_pred
```

```
array([439023.26      , 300300.        , 858642.27333332, ...,
       323979.5       , 395509.77777777, 608712.06      ])
```

PERFORMANCE EVALUATION

MEAN ABSOLUTE ERROR

```
from sklearn.metrics import mean_absolute_error
print("MEAN ABSOLUTE ERROR IS", mean_absolute_error(y_test,y_pred))
```

```
MEAN ABSOLUTE ERROR IS 196744.7409562903
```

Connected to Python 3 Google Compute Engine backend

HOUSE PRICE PREDICTION USING ML

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

MEAN ABSOLUTE PERCENTAGE ERROR

```
[25] from sklearn.metrics import mean_absolute_percentage_error
     print("ERROR PERCENTAGE IS", mean_absolute_percentage_error(y_test,y_pred))
```

```
ERROR PERCENTAGE IS 3.0345690625556115e+19
```

MEAN SQUARED ERROR

```
[26] from sklearn.metrics import mean_squared_error
     print("MEAN SQUARED ERROR IS",mean_squared_error(y_test,y_pred))
```

```
MEAN SQUARED ERROR IS 689521953555.1638
```

ROOT MEAN SQUARED ERROR

```
[27] from sklearn.metrics import mean_squared_error
     root=mean_squared_error(y_test,y_pred)
     squ=np.sqrt(root)
```

Connected to Python 3 Google Compute Engine backend

HOUSE PRICE PREDICTION USING ML ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

```
[26] from sklearn.metrics import mean_squared_error
     print("MEAN SQUARED ERROR IS",mean_squared_error(y_test,y_pred))
```

```
MEAN SQUARED ERROR IS 689521953555.1638
```

ROOT MEAN SQUARED ERROR

```
[27] from sklearn.metrics import mean_squared_error
     root=mean_squared_error(y_test,y_pred)
     squ=np.sqrt(root)
     print(squ)
```

```
830374.5862893227
```

r2_SCORE

```
from sklearn.metrics import r2_score
print("r2_score is",r2_score(y_test,y_pred))
```

```
r2_score is 0.04558092563961291
```

Connected to Python 3 Google Compute Engine backend