Name : Prasanna Thirukudanthai Raghavan

UID: 118287546

1. Download the data.

```
import pandas as pd
import matplotlib.pyplot as plt
import os
import numpy as np
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
%cd /content/drive/MyDrive/808W Assignments/HW2/Train/
```

```
    /content/drive/MyDrive/808W Assignments/HW2/Train
```

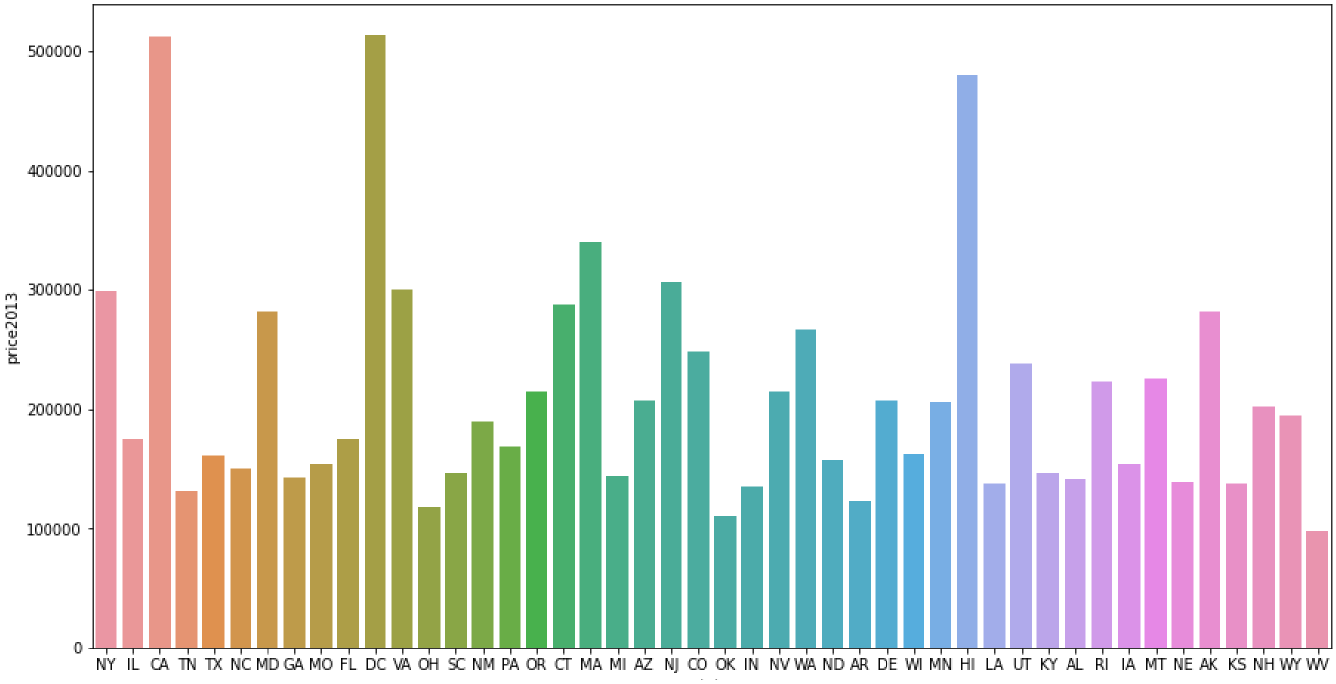
```
from pandas.io.parsers.readers import read_csv
df = read_csv("/content/drive/MyDrive/808W Assignments/HW2/Train/house_train.csv",encoding='l
```

```
df.head()
```

	id	zip	state	county	poverty	price2007	price2013
0	0	10467	NY	bronx	27.1	335200	294000
1	1	11226	NY	kings	21.9	471500	471600
2	2	60640	IL	cook	14.6	254600	174200
3	3	94109	CA	san francisco	10.6	707100	822600
4	4	11375	NY	queens	12.2	636400	681500

```
plt.figure(figsize=(15,8))
sns.barplot(data=df, x="state", y="price2013",ci=None)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa72b5b1ed0>
```



2. Predict 2013 home prices using state information only.

```
x1 = pd.get_dummies(data=df['state'])#, drop_first=True)
y = df['price2013']
x1.head()
```

	AK	AL	AR	AZ	CA	CO	CT	DC	DE	FL	...	RI	SC	TN	TX	UT	VA	WA	WI	WV	WY
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 46 columns

```
from sklearn.linear_model import LinearRegression

model = LinearRegression().fit(x1, y)
```

a. What is the intercept? What does it correspond to?

Answer: The intercept of the model is 281729 and it corresponds to the base price of home irrespective of which state it might be in.

b. How do you get this information from your regression?

Answer: We get this using the .intercept_ function from the sklearn library. In general line equation y = mx +c ; where c is the intercept. which is obtained by checking the y value when x is zero or more precisely when the coefficients do not affect the y value.

```
model.intercept_
```

```
    4.45331222942643e+18
```

c. Based on your regression coefficients, what states have the most and least expensive average homes?

```
print("The state with most average expensive homes is:",x1.columns[np.argmax(model.coef_)])
print("The state with least average expensive homes is:",x1.columns[np.argmin(model.coef_)])
```

```
    The state with most average expensive homes is: DC
    The state with least average expensive homes is: WV
```

d. How do you get this information from your regression?

We get this information by checking which coefficients have the highest and lowest value. The above gives us the extremeties in the coefficients which are taken from the array.

e. What is the average price of homes in those states?

Answer:
The average price of home in state with most average expensive homes is: 514288
The average price of home in state with least average expensive homes is: 98423

```
print("The average price of home in state with most average expensive homes is:",int(model.in
print("The average price of home in state with least average expensive homes is:",int(model.i
```

```
    The average price of home in state with most average expensive homes is: 514288
    The average price of home in state with least average expensive homes is: 98423
```

f. How do you get this information from your regression?

We get the price of the homes by adding the coefficients of the state with the intercept as seen

3. Predict 2013 home prices from state and county information

```
x2 = df[['state','county']]
x2 = pd.get_dummies(data=x2)#, drop_first=True)
x2.head()
```

	state_AK	state_AL	state_AR	state_AZ	state_CA	state_CO	state_CT	state_DC	state
0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	
3	0	0	0	0	1	0	0	0	
4	0	0	0	0	0	0	0	0	

5 rows × 674 columns

```
model2 = LinearRegression().fit(x2, y)#positive=True
```

```
model2.intercept_
```

```
1.998622919982483e+17
```

```
#model2.coef_
```

```
model2.predict(x2.iloc[[0]])
```

```
array([381600.])
```

a. What US counties have the highest and lowest regression coefficients? Why?

```
print("The ",x2.columns[45+np.argmax(model2.coef_[45:])],"has the highest coefficient which i
The  county_district of columbia has the highest coefficient which is: 3.35297075773475
```

The county with highest coefficient is due to the state DC,which has some of the highest prices compared to the other states, hence it is not a suprise that it also has the highest regression coefficient.

```
print("The ",x2.columns[45+np.argmin(model2.coef_[45:])],"has the lowest coefficient which is
```

```
The  county_east baton rouge has the lowest coefficient which is: -4.058350732086905e+1
```

The county with the lowest coefficient is due to the county having low housing prices in a state which generally has a higher housing price which correlates with the sign of the coefficient which indicates that houses in this county have significanctly lesser prices that other counties from the same state.

```
x3 = df[['county']]
x3 = pd.get_dummies(data=x3, drop_first=True)
x3.head()
```

	county_aiken	county_alachua	county_alamance	county_alameda	county_albany	county
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

5 rows × 627 columns

```
model3 = LinearRegression().fit(x3, y)
```

```
model3.intercept_
```

```
175427.27272727323
```

4. Build a regressor that best predicts average home values in this dataset.

```
df.head()
```

```
   id    zip  state         county  poverty  price2007  price2013
0   0  10467     NY          bronx     27.1     335200  2940[illegible]
1   1  11226     NY          kings     21.9     471500     471600
2   2  60640     IL           cook     14.6     254600     174200
3   3  94109     CA  san francisco     10.6     707100     822600
4   4  11375     NY         queens     12.2     636400     681500
```

```
%cd /content/drive/MyDrive/808W Assignments/HW2/Test/
```

```
/content/drive/MyDrive/808W Assignments/HW2/Test
```

```
test = read_csv("/content/drive/MyDrive/808W Assignments/HW2/Test/house_test.csv",encoding='l
test.head()
```

```
   id    zip  state       county  poverty  price2007
0   6  32162     FL       marion     13.0     265600
1  13  78572     TX      hidalgo     34.0      79900
2  20  11212     NY        kings     21.9     332000
3  30  37042     TN   montgomery     12.7      98700
4  37  85032     AZ     maricopa     12.9     266100
```

Using a combination of state and poverty data

a. Describe what you did to build the best predictor possible

---

I initially used a combination of state and poverty and then used state poverty and price 2007 to get the best predictor.

```
x4a = x1
x4a['poverty'] = df['poverty']
x4a.head()
```

	AK	AL	AR	AZ	CA	CO	CT	DC	DE	FL	...	TN	TX	UT	VA	WA	WI	WV	WY	price2007
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	335200
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	471500
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	254600
3	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	707100
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	636400

5 rows × 48 columns

```
model4a = LinearRegression(positive=True).fit(x4a, y)
model4a.intercept_
```

```
98423.07692316032
```

```
testpred = test['state']
testpred = pd.get_dummies(data=testpred)
```

Since the test data does not have some of the state values we have to check the ones it does not have and insert them so that we can predict the final values.

```
exstate = []
for i in pd.unique(df['state']):
  if i not in pd.unique(test['state']):exstate.append(i)
print(exstate)
testpred[exstate] = 0
testpred.head()
```

```
['WY', 'WV']
```

	AK	AL	AR	AZ	CA	CO	CT	DC	DE	FL	...	RI	SC	TN	TX	UT	VA	WA	WI	WY	WV
0	0	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	1	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 46 columns

```
testpred1 = testpred
testpred1['poverty'] = test['poverty']
testpred1.head()
```

	AK	AL	AR	AZ	CA	CO	CT	DC	DE	FL	...	TN	TX	UT	VA	WA	WI	WY	WV	price200
0	0	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	26560
1	0	0	0	0	0	0	0	0	0	0	...	0	1	0	0	0	0	0	0	7990
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	33200
3	0	0	0	0	0	0	0	0	0	0	...	1	0	0	0	0	0	0	0	9870
4	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	26610

5 rows × 48 columns

```
pricepred1 = model4a.predict(testpred)
print(pricepred1)
```

```
[174987.87878788 160655.1724138  298746.61764709 ... 281845.51971329
 281845.51971329 248585.45454547]
```

```
testsub1 = pd.DataFrame()
testsub1['id'] = test['id']
testsub1['prediction'] = pricepred1
testsub1.to_csv(r'/content/drive/MyDrive/808W Assignments/HW2/Subs/testsub1.csv')
```

Using a combination of state poverty and price 2007

```
x4b = x1
x4b['poverty'] = df['poverty']
x4b['price2007'] = df['price2007']
x4b.head()
```

```
model4b = LinearRegression(positive=True).fit(x4b, y)
model4b.intercept_
```

```
-166685.82647895574
```

```
testpred1['price2007'] = test['price2007']
testpred1.head()
```

	AK	AL	AR	AZ	CA	CO	CT	DC	DE	FL	...	TN	TX	UT	VA	WA	WI	WY	WV	price2007
0	0	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	26560
1	0	0	0	0	0	0	0	0	0	0	...	0	1	0	0	0	0	0	0	7990
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	33200
3	0	0	0	0	0	0	0	0	0	0	...	1	0	0	0	0	0	0	0	9870
4	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	26610

5 rows × 48 columns

```
pricepred2 = model4b.predict(testpred1)
print(pricepred2)
```

```
[3.57198353e+08 1.07487135e+08 4.46589059e+08 ... 4.52748601e+08
 1.03890996e+09 3.14797997e+08]
```

```
pricepred3 = regr_trans.predict(testpred1)
print(pricepred3)
```

```
[3816799.99999991 4709500.         4709500.         ... 2348700.00000004
 2856900.00000012 3816799.99999991]
```

```
testsub5 = pd.DataFrame()
testsub5['id'] = test['id']
testsub5['prediction'] = pricepred2
testsub5.to_csv(r'/content/drive/MyDrive/808W Assignments/HW2/Subs/testsub7.csv')
```

Kaggle!

Best Score : **58101.75723**

Username : **trprasanna**

Suppose you have 2 bags

Bag #1 has 1 black ball and 2 white balls.

Bag #2 has 1 black ball and 3 white balls.

Suppose you pick a bag at random, and select a ball from that bag.

What is the probability of selecting a white ball?

Answer:
Probability of picking white from bag 1 = $\frac{2}{3}$
Probability of picking white from bag 2 = $\frac{3}{4}$
P(white being from bag 1) = $\frac{1}{2}$x$\frac{2}{3}$ = $\frac{1}{3}$
P(white being from bag 2) = $\frac{1}{2}$x$\frac{3}{4}$ = $\frac{3}{8}$
The probability would be the sum of both as they are mutually independant i.e
P(white ball) = $\frac{17}{24}$

A soccer team wins 60% of its games when it scores the first goal, and 10% of its games when the opposing team scores first. If the team scores the first goal about 30% of the time, what fraction of the games does it win?

P(winning when first) = 0.6
P(winning when not first) = 0.1
P(scoring first) = 0.3
P(winning) = P(winning when first)*P(scoring first) + P(winning when not first)*P(scoring not first)
P(winning) = 0.6x0.3+0.1x0.7
P(winning) = 0.18x0.07
P(winning) = 0.25
The probability of winning is 0.25 or 25%

✓ 0s completed at 7:16 PM ● ×