UC San Diego Extension
Cloud Services for Machine Learning

Summer 2020
Homework#2

Date Given: July 6, 2020 Due Date: July 12, 2020

==========

Analyze the data source in 'kc-house-data.csv' file. This data source is a part of databases available in the public domain. This file contains 21,613 observations of real-estate properties of King county in Washington state. The data for the following 21 variables are provided.

1. id
2. date
3. price
4. bedrooms
5. bathrooms
6. sqft_living
7. sqft_lot
8. floors
9. waterfront
10. view
11. condition
12. grade
13. sqft_above
14. sqft_basement
15. yr_built
16. yr_renovated
17. zipcode
18. latitude
19. longitude
20. sqft_living15
21. sqft_lot15

Read the raw data source file 'kc-house-data.csv'. Build a linear regression model (as described in Problem#1 and Problem#2 on the next page) using the following variables.

Response Variable:

- price (numerical)

Predictor Variables:

- sqft_living (numerical)
- bedrooms (numerical)
- waterfront (categorical):
  - Levels of waterfront: 0 = no waterfront, 1 = waterfront
- condition(categorical)
  - Levels of condition: 1,2,3,4,5

Problem#1

Build a regression model using the following characteristics.

- Programming Language: Python
- Cloud Platform: Colab
- Package: Scikit-Learn

Verify that your regression equation is as follows.

$$price = 66{,}581.53 + 305.72 * sqftliving - 52{,}704.36 * bedrooms + 783{,}090.68 * waterfront$$
$$-25{,}698.33 * condition2 - 8{,}811.75 * condition3 + 28{,}198.78 * condition4 + 100{,}565.72 * condition5$$

Predict the price of a home with following characteristics:

- sqft_living = 3,000
- bedrooms = 4
- waterfront = No (0)
- condition = 4

Verify that the $Predicted\ price = \$801{,}112.20$

==========

Problem#2

Build a regression model using the following characteristics.

- Programming Language: None
- Cloud Platform: AutoML GCP

The procedure to build a regression model on GCP is as follows.

1. GCP/Storage
    a. Create a Bucket in GCP
    b. Region: us-central1(Iowa)
    c. Upload Data file in that bucket
2. GCP/Table/Dataset
    a. Import data in a GCP Dataset from the bucket: Takes time
3. GCP/Table/Model
    a. Train the Model
        i. Select Target Variable + Budget: Takes time
    b. Evaluate the Model
    c. Test & Use: Deploy the Model: Takes time
        i. Prediction

Predict the price of a home with following characteristics:

- sqft_living = 3,000
- bedrooms = 4
- waterfront = No (0)
- condition = 4

The predicted value of the 'price' variable using GCP should be approximately equal to $801,112.20. Also compute the 95% prediction interval of the response variable 'price'.

Building a regression model on GCP will cost a certain amount. Please check the GCP charges on your account before and after you complete this assignment. Make sure you do not deplete the $300 credit you have on your account.

How to handle 'condition" categorical variable in Scikit-Learn:

The 'condition' variable is categorical with 5 levels. The values of this variable are 1,2,3,4,5. This does NOT mean that 5 > 4 > 3 > 2 > 1. Since there are 5 levels of this variable, we need to replace the 'condition' variable with 4 (k-1) dummy (indicator) variables when performing the regression in Scikit-Learn.

When regression is performed in GCP/AutoML, we need to declare the 'condition' variable as categorical during the 'train' stage of model building. The GCP/AutoML does not need one-hot-encoded data as input. GCP/AutoML will perform the one-hot-encoding internally.

We must convert the 'condition' variable into 4 separate dummy variables using one-hot-encoding. The logic used for prediction is shown in the table below. The 'condition=1" will be our base condition. All values will be computed relative to 'condition=1'.

	Var: **condition2**	Var: **condition3**	Var: **condition4**	Var: **condition5**
Condition1 (Base)	0	0	0	0
Condition2	1	0	0	0
Condition3	0	1	0	0
Condition4	0	0	1	0
Condition5	0	0	0	1

Regression equation is as follows.

$$price = 66{,}581.53 + 305.72 * sqftliving - 52{,}704.36 * bedrooms + 783{,}090.68 * waterfront$$
$$-25{,}698.33 * condition2 - 8{,}811.75 * condition3 + 28{,}198.78 * condition4 + 100{,}565.72 * condition5$$

- This means that the price of a house with '**condition=2**' is $25,698.33 **less** compared with the house with condition=0.

- This means that the price of a house with '**condition=5**' is $100,565.72 **more** compared with the house with condition=0.

Now let us predict the price of the house using different value of the 'condition' categorical variable.

==========

Predict the price of a home with following characteristics:

- sqft_living = 3,000
- bedrooms = 4
- waterfront = No (0)
- **condition = 1**

  $price = 66{,}581.53 + 305.72 * sqftliving(3{,}000) - 52{,}704.36 * bedrooms(4) + 783{,}090.68 * waterfront(0)$

  $price = 772{,}913.4$

==========

Predict the price of a home with following characteristics:

- sqft_living = 3,000
- bedrooms = 4
- waterfront = No (0)
- **condition = 2**

  $price = 66{,}581.53 + 305.72 * sqftliving(3{,}000) - 52{,}704.36 * bedrooms(4) + 783{,}090.68 * waterfront(0) - 25{,}698.33 * condition2(1)$

  $price = 747{,}215.1$

==========

Predict the price of a home with following characteristics:

- sqft_living = 3,000
- bedrooms = 4
- waterfront = No (0)
- **condition = 3**

  $price = 66{,}581.53 + 305.72 * sqftliving(3{,}000) - 52{,}704.36 * bedrooms(4) + 783{,}090.68 * waterfront(0) - 8{,}811.75 * condition3(1)$

  $price = 764{,}101.7$

==========

Predict the price of a home with following characteristics:

- sqft_living = 3,000
- bedrooms = 4
- waterfront = No (0)
- **condition = 5**
- $price = 66{,}581.53 + 305.72 * sqftliving(3{,}000) - 52{,}704.36 * bedrooms(4) + 783{,}090.68 * waterfront(0) + 100{,}565.72 * condition5(1)$
- $price = 873{,}479.2$

==========

Problem#1

Python Code

Housing_Data_Colab.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[1] import pandas as pd
    import numpy as np
    from sklearn import linear_model
```

```
[2] from google.colab import files
    uploaded = files.upload()
```

Choose Files 00 kc_house_data.csv

- **00 kc_house_data.csv**(application/vnd.ms-excel) - 2515206 bytes, last modified: 5/19/2016 - 100% done

```
Saving 00 kc_house_data.csv to 00 kc_house_data.csv
```

```
[ ] import io
    #data = pd.read_csv(io.StringIO(uploaded['00 kc_house_data.csv'].decode('utf-8')))
    data = pd.read_csv(io.BytesIO(uploaded['00 kc_house_data.csv']))
```

```
[3] data = pd.read_csv("00 kc_house_data.csv")

    print(data.head())
    print(data.tail())
    data.shape
```

```
           id             date     price  ...     long  sqft_living15  sqft_lot15
0  7129300520  20141013T000000  221900.0  ... -122.257           1340        5650
1  6414100192  20141209T000000  538000.0  ... -122.319           1690        7639
2  5631500400  20150225T000000  180000.0  ... -122.233           2720        8062
3  2487200875  20141209T000000  604000.0  ... -122.393           1360        5000
4  1954400510  20150218T000000  510000.0  ... -122.045           1800        7503

[5 rows x 21 columns]
               id             date  ...  sqft_living15  sqft_lot15
21608   263000018  20140521T000000  ...           1530        1509
```

```
[5]  predictedData = data[['sqft_living','bedrooms','waterfront','condition']]
```

```
[6]  dummyWaterfront = pd.get_dummies(predictedData.waterfront,prefix='wf')
     print(dummyWaterfront.head())

     dummyCondition = pd.get_dummies(predictedData.condition,prefix='condition')
     print(dummyCondition.head())
```

```
   wf_0  wf_1
0     1     0
1     1     0
2     1     0
3     1     0
4     1     0
   condition_1  condition_2  condition_3  condition_4  condition_5
0            0            0            1            0            0
1            0            0            1            0            0
2            0            0            1            0            0
3            0            0            0            0            1
4            0            0            1            0            0
```

```
[7]  merge=pd.concat([predictedData,dummyWaterfront,dummyCondition],axis='columns')
     merge.head()
```

	sqft_living	bedrooms	waterfront	condition	wf_0	wf_1	condition_1	condition_2	condition_3	condition_4	condition_5
0	1180	3	0	3	1	0	0	0	1	0	0
1	2570	3	0	3	1	0	0	0	1	0	0
2	770	2	0	3	1	0	0	0	1	0	0
3	1960	4	0	5	1	0	0	0	0	0	1
4	1680	3	0	3	1	0	0	0	1	0	0

```
finalData = merge.drop(['waterfront','condition','wf_0','condition_1'],axis='columns')
finalData.head()
```

	sqft_living	bedrooms	wf_1	condition_2	condition_3	condition_4	condition_5
0	1180	3	0	0	1	0	0
1	2570	3	0	0	1	0	0
2	770	2	0	0	1	0	0
3	1960	4	0	0	0	0	1
4	1680	3	0	0	1	0	0

```
[9]  X = finalData
     print (type(X))
     print (X.shape)

     y = data['price']
     print (type(y))
     print (y.shape)
```

```
<class 'pandas.core.frame.DataFrame'>
(21613, 7)
<class 'pandas.core.series.Series'>
(21613,)
```

```
[10] linreg = linear_model.LinearRegression()
     linreg.fit(X, y)
     print (linreg.intercept_)
     print (linreg.coef_)
```

```
66581.52860950812
[ 3.05716451e+02 -5.27043605e+04  7.83090677e+05 -2.56983253e+04
 -8.81175295e+03  2.81987756e+04  1.00565722e+05]
```

```
linreg.predict([[3000,4,0,0,0,1,0]])
```

```
array([801112.21594909])
```

$$price = 66{,}581.53 + 305.72 * sqftliving - 52{,}704.36 * bedrooms + 783{,}090.68 * waterfront$$
$$-25{,}698.33 * condition2 - 8{,}811.75 * condition3 + 28{,}198.78 * condition4 + 100{,}565.72 * condition5$$

Predict the price of a home with following characteristics:

- sqft_living = 3,000
- bedrooms = 4
- waterfront = No (0)
- condition = 4

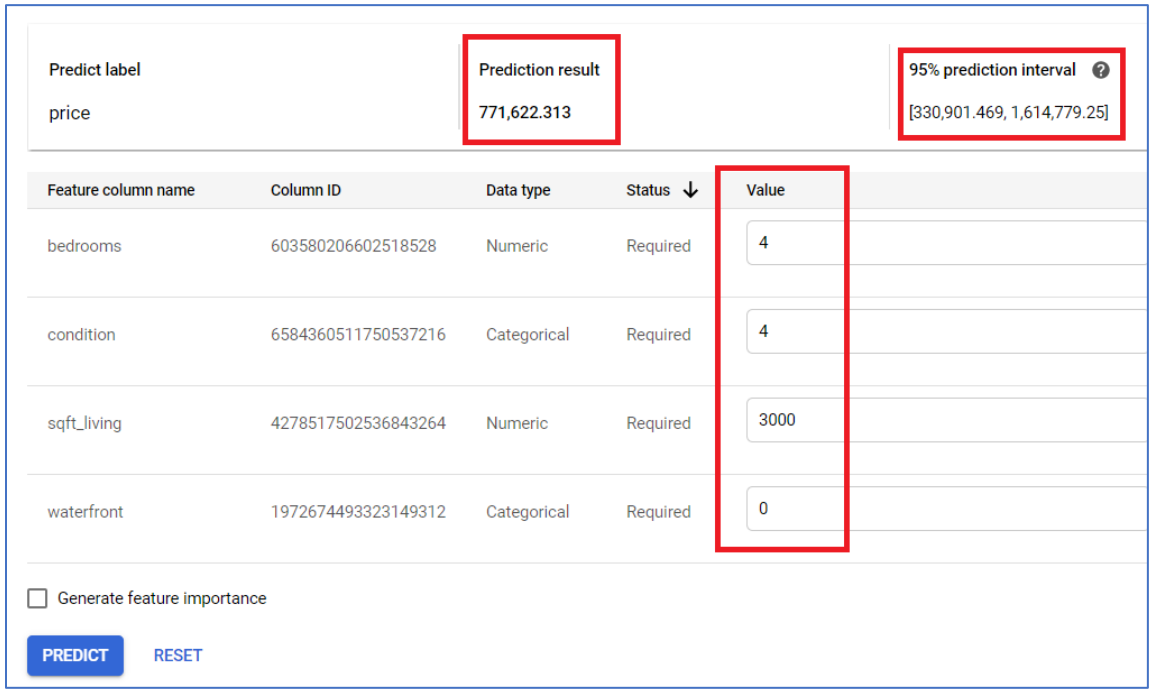
$Predicted\ price = 801,112.2$

Problem#2

Build a regression model using the following characteristics.

- Programming Language: None
- Cloud Platform: AutoML GCP

4. GCP/Storage
    a. Create a Bucket in GCP
    b. Region: us-central1(Iowa)
    c. Upload Data file in that bucket
5. GCP/Table/Dataset
    a. Import data in a GCP Dataset from the bucket: Takes time
6. GCP/Table/Model
    a. Train the Model
        i. Select Target Variable + Budget: Takes time
    b. Evaluate the Model
    c. Test & Use: Deploy the Model: Takes time
        i. Prediction



Predicted price = 771,622.313

95% prediction Interval: $330,901.47 - $1,614,779.25