# Instructions

In this assignment, you are a Data Analyst working at a Real Estate Investment Trust. The Trust would like to start investing in Residential real estate. You are tasked with determining the market price of a house given a set of features. You will analyze and predict housing prices using attributes or features such as square footage, number of bedrooms, number of floors, and so on. This is a template notebook; your job is to complete the ten questions. Some hints to the questions are given.

As you are completing this notebook, take and save the **screenshots** of the final outputs of your solutions (e.g., final charts, tables, calculation results etc.). They will need to be shared in the following Peer Review section of the Final Project module.

# About the Dataset

This dataset contains house sale prices for King County, which includes Seattle. It includes homes sold between May 2014 and May 2015. It was taken from here. It was also slightly modified for the purposes of this course.

| Variable | Description |
|---|---|
| id | A notation for a house |
| date | Date house was sold |
| price | Price is prediction target |
| bedrooms | Number of bedrooms |
| bathrooms | Number of bathrooms |
| sqft_living | Square footage of the home |
| sqft_lot | Square footage of the lot |
| floors | Total floors (levels) in house |
| waterfront | House which has a view to a waterfront |
| view | Has been viewed |
| condition | How good the condition is overall |
| grade | overall grade given to the housing unit, based on King County grading system |
| sqft_above | Square footage of house apart from basement |
| sqft_ba | Square footage of the basement |

| Variable | Description |
|---|---|
| sement | |
| yr_built | Built Year |
| yr_renovated | Year when house was renovated |
| zipcode | Zip code |
| lat | Latitude coordinate |
| long | Longitude coordinate |
| sqft_living15 | Living room area in 2015(implies-- some renovations) This might or might not have affected the lotsize area |
| sqft_lot15 | LotSize area in 2015(implies-- some renovations) |

## Import the required libraries

```
# All Libraries required for this lab are listed below. The libraries pre-installed on Skills Network Labs are commented.
# !mamba install -qy pandas==1.3.4 numpy==1.21.4 seaborn==0.9.0 matplotlib==3.5.0 scikit-learn==0.20.1
# Note: If your environment doesn't support "!mamba install", use "!pip install"

# Surpress warnings:
def warn(*args, **kwargs):
    pass
import warnings
warnings.warn = warn

#!pip install -U scikit-learn

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
%pip install seaborn
import seaborn as sns
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler,PolynomialFeatures
from sklearn.linear_model import LinearRegression
%matplotlib inline
print('Done')
```

```
Done
```

# Module 1: Importing Data Sets

Download the dataset by running the cell below.

```
import piplite
await piplite.install('seaborn')

from pyodide.http import pyfetch

async def download(url, filename):
    response = await pyfetch(url)
    if response.status == 200:
        with open(filename, "wb") as f:
            f.write(await response.bytes())

filepath='https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DA0101EN-
SkillsNetwork/labs/FinalModule_Coursera/data/kc_house_data_NaN.csv'

await download(filepath, "housing.csv")
file_name="housing.csv"
```

Load the csv:

```
df = pd.read_csv(file_name)
```

> Note: This version of the lab is working on JupyterLite, which requires the dataset to be downloaded to the interface.While working on the downloaded version of this notebook on their local machines(Jupyter Anaconda), the learners can simply **skip the steps above,** and simply use the URL directly in the `pandas.read_csv()` function. You can uncomment and run the statements in the cell below.

```
#filepath='https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DA0101EN-
SkillsNetwork/labs/FinalModule_Coursera/data/kc_house_data_NaN.csv'
#df = pd.read_csv(filepath, header=None)
```

We use the method head to display the first 5 columns of the dataframe.

```
df.head()
```

## Question 1

Display the data types of each column using the function dtypes. Take a screenshot of your code and output. You will need to submit the screenshot for the final project.

```
#Enter Your Code, Execute and take the Screenshot
df.dtypes
```

We use the method describe to obtain a statistical summary of the dataframe.

```
df.describe()
```

# Module 2: Data Wrangling

## Question 2

Drop the columns "id" and "Unnamed: 0" from axis 1 using the method drop(), then use the method describe() to obtain a statistical summary of the data. Make sure the inplace parameter is set to True. Take a screenshot of your code and output. You will need to submit the screenshot for the final project.

```
#Enter Your Code, Execute and take the Screenshot
df.drop(['id', 'Unnamed: 0'], axis=1, inplace=True)

df.describe()
```

We can see we have missing values for the columns  bedrooms and  bathrooms

```
print("number of NaN values for the column bedrooms :",
df['bedrooms'].isnull().sum())
print("number of NaN values for the column bathrooms :",
df['bathrooms'].isnull().sum())
```

We can replace the missing values of the column 'bedrooms' with the mean of the column 'bedrooms'  using the method replace(). Don't forget to set the inplace parameter to True

```
mean=df['bedrooms'].mean()
df['bedrooms'].replace(np.nan,mean, inplace=True)
```

We also replace the missing values of the column 'bathrooms' with the mean of the column 'bathrooms'  using the method replace(). Don't forget to set the  inplace  parameter top  True

```
mean=df['bathrooms'].mean()
df['bathrooms'].replace(np.nan,mean, inplace=True)

print("number of NaN values for the column bedrooms :",
df['bedrooms'].isnull().sum())
print("number of NaN values for the column bathrooms :",
df['bathrooms'].isnull().sum())
```

# Module 3: Exploratory Data Analysis

## Question 3

Use the method value_counts to count the number of houses with unique floor values, use the method .to_frame() to convert it to a data frame. Take a screenshot of your code and output. You will need to submit the screenshot for the final project.

```
#Enter Your Code, Execute and take the Screenshot
df['floors'].value_counts()
df['floors'].value_counts().to_frame()
```

## Question 4

Use the function boxplot in the seaborn library to determine whether houses with a waterfront view or without a waterfront view have more price outliers. Take a screenshot of your code and boxplot. You will need to submit the screenshot for the final project.

```
sns.boxplot(x="waterfront", y="price", data=df)
```

## Question 5

Use the function regplot in the seaborn library to determine if the feature sqft_above is negatively or positively correlated with price. Take a screenshot of your code and scatterplot. You will need to submit the screenshot for the final project.

```
#Enter Your Code, Execute and take the Screenshot
sns.regplot(x="sqft_above", y="price", data=df)
```

We can use the Pandas method corr() to find the feature other than price that is most correlated with price.

```
df.corr()['price'].sort_values()
```

# Module 4: Model Development

We can Fit a linear regression model using the longitude feature 'long' and caculate the R^2.

```
X = df[['long']]
Y = df['price']
lm = LinearRegression()
lm.fit(X,Y)
lm.score(X, Y)
```

## Question 6

Fit a linear regression model to predict the 'price' using the feature 'sqft_living' then calculate the R^2. Take a screenshot of your code and the value of the R^2. You will need to submit it for the final project.

```python
#Enter Your Code, Execute and take the Screenshot
lm = LinearRegression()

X = df[['sqft_living']]
Y = df['price']

lm.fit(X, Y)
print('The R-square is: ', lm.score(X, Y))
```

## Question 7

Fit a linear regression model to predict the 'price' using the list of features:

```python
features = ['floors',
'waterfront','lat' ,'bedrooms' ,'sqft_basement' ,'view' ,'bathrooms','sqft_living15','sqft_above','grade','sqft_living']
```

Then calculate the R^2. Take a screenshot of your code and the value of the R^2. You will need to submit it for the final project.

```python
#Enter Your Code, Execute and take the Screenshot
Z = df[['floors',
'waterfront','lat' ,'bedrooms' ,'sqft_basement' ,'view' ,'bathrooms','sqft_living15','sqft_above','grade','sqft_living']]
Y = df['price']

lm = LinearRegression()

lm.fit(Z, Y)
print('The R-square is: ', lm.score(Z, Y))
```

## This will help with Question 8

Create a list of tuples, the first element in the tuple contains the name of the estimator:

'scale'

'polynomial'

'model'

The second element in the tuple contains the model constructor

StandardScaler()

PolynomialFeatures(include_bias=False)

LinearRegression()

```
Input=[('scale',StandardScaler()),('polynomial',
PolynomialFeatures(include_bias=False)),('model',LinearRegression())]
```

### Question 8

Use the list to create a pipeline object to predict the 'price', fit the object using the features in the list features, and calculate the R^2. Take a screenshot of your code and the value of the R^2. You will need to submit it for the final project.

```
#Enter Your Code, Execute and take the Screenshot
pipe=Pipeline(Input)
pipe

pipe.fit(Z,Y)

pipe.score(Z, Y)
```

# Module 5: Model Evaluation and Refinement

Import the necessary modules:

```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
print("done")
```

We will split the data into training and testing sets:

```
features =["floors",
"waterfront","lat" ,"bedrooms" ,"sqft_basement" ,"view" ,"bathrooms","
sqft_living15","sqft_above","grade","sqft_living"]
X = df[features]
Y = df['price']

x_train, x_test, y_train, y_test = train_test_split(X, Y,
test_size=0.15, random_state=1)


print("number of test samples:", x_test.shape[0])
print("number of training samples:",x_train.shape[0])
```

## Question 9

Create and fit a Ridge regression object using the training data, set the regularization parameter to 0.1, and calculate the R^2 using the test data. Take a screenshot of your code and the value of the R^2. You will need to submit it for the final project.

```
from sklearn.linear_model import Ridge

#Enter Your Code, Execute and take the Screenshot
RigeModel=Ridge(alpha=0.1)

RigeModel.fit(x_train, y_train)

RigeModel.score(x_test, y_test)
```

## Question 10

Perform a second order polynomial transform on both the training data and testing data. Create and fit a Ridge regression object using the training data, set the regularisation parameter to 0.1, and calculate the R^2 utilising the test data provided. Take a screenshot of your code and the R^2. You will need to submit it for the final project.

```
#Enter Your Code, Execute and take the Screenshot

pr=PolynomialFeatures(degree=2)
x_train_pr=pr.fit_transform(x_train[["floors",
"waterfront","lat" ,"bedrooms" ,"sqft_basement" ,"view" ,"bathrooms","
sqft_living15","sqft_above","grade","sqft_living"]])
x_test_pr=pr.fit_transform(x_test[["floors",
"waterfront","lat" ,"bedrooms" ,"sqft_basement" ,"view" ,"bathrooms","
sqft_living15","sqft_above","grade","sqft_living"]])

RigeModel=Ridge(alpha=0.1)

RigeModel.fit(x_train_pr, y_train)

RigeModel.score(x_test_pr, y_test)
```

Once you complete your notebook you will have to share it. You can download the notebook by navigating to "File" and clicking on "Download" button.   This will save the (.ipynb) file on your computer. Once saved, you can upload this file in the "My Submission" tab, of the "Peer-graded Assignment" section.

Joseph Santarcangelo has a PhD in Electrical Engineering, his research focused on using machine learning, signal processing, and computer vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Other contributors: Michelle Carey, Mavis Zhou

## Change Log

| Date (YYYY-MM-DD) | Version | Changed By | Change Description |
|---|---|---|---|
| 2020-12-01 | 2.2 | Aije Egwaikhide | Coverted Data describtion from text to table |
| 2020-10-06 | 2.1 | Lakshmi Holla | Changed markdown instruction of Question1 |
| 2020-08-27 | 2.0 | Malika Singla | Added lab to GitLab |
| 2022-06-13 | 2.3 | Svitlana Kramar | Updated Notebook sharing instructions |