(https://skills.network/?
utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id
SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDA0101ENSkillsNetwork971-2022-
01-01)

House Sales in King County, USA

This dataset contains house sale prices for King County, which includes Seattle. It includes homes sold between May 2014 and May 2015.

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_basement	Square footage of the basement
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)

In [1]:

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

In [2]:

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

You will require the following libraries:

In [3]:

```
import piplite
await piplite.install(['pandas','matplotlib','scikit-learn','seaborn', 'numpy'])
```

```
---------------------------------------------------------------------------
-
ModuleNotFoundError                       Traceback (most recent call las
t)
~\AppData\Local\Temp/ipykernel_9648/977448495.py in <module>
----> 1 import piplite
      2 await piplite.install(['pandas','matplotlib','scikit-learn','seabo
rn', 'numpy'])

ModuleNotFoundError: No module named 'piplite'
```

In [ ]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler,PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
%matplotlib inline
```

Module 1: Importing Data Sets

The functions below will download the dataset into your browser:

In [ ]:

```
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())
```

In [ ]:

```
file_name='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDevelope
```

You will need to download the dataset; if you are running locally, please comment out the following code:

In [ ]:

```
await download(file_name, "kc_house_data_NaN.csv")
file_name="kc_house_data_NaN.csv"
```

Use the Pandas method **read_csv()** to load the data from the web address.

In [ ]:

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

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

In [ ]:

```
df.head()
```

Question 1

Display the data types of each column using the function dtypes, then take a screenshot and submit it, include your code in the image.

In [ ]:

```
df.dtypes
```

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

In [ ]:

```
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. Take a screenshot and submit it, make sure the `inplace` parameter is set to `True`

In [ ]:

```
df.drop(['id','Unnamed: 0'], axis=1, inplace=True)
df.describe(include='all')
```

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

In [ ]:

```
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`

In [ ]:

```
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`

In [ ]:

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

In [ ]:

```
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 dataframe.

In [ ]:

```
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.

In [ ]:

```
sns.boxplot(x='waterfront', y='price', data=df, linewidth=2, fliersize=5)
plt.show()
```

Question 5

Use the function `regplot` in the seaborn library to determine if the feature `sqft_above` is negatively or positively correlated with price.

In [ ]:

```
sns.regplot(x='sqft_above', y='price', data=df, color='red',scatter_kws={'alpha': 0.5,'s'
plt.show()
```

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

In [ ]:

```
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.

In [ ]:

```
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.

In [ ]:

```
x=df['sqft_living'].values.reshape(-1, 1)
y=df['price']
model= LinearRegression()
model.fit(x,y)
y_pred=model.predict(x)
r2=r2_score(y,y_pred)
print(r2)
```

Question 7

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

In [ ]:

```
features =["floors", "waterfront","lat" ,"bedrooms" ,"sqft_basement" ,"view" ,"bathrooms"
```

Then calculate the R^2. Take a screenshot of your code.

In [ ]:

```
z=df[features]
y=df['price']
model= LinearRegression()
model.fit(z, y)
model.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()`

In [ ]:

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

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.

In [ ]:

```
model= Pipeline(Input)
model.fit(df[features], df['price'])
y_pred= model.predict(df[features])
r2= r2_score(df['price'], y_pred)
print(r2)
```

Module 5: Model Evaluation and Refinement

Import the necessary modules:

In [ ]:

```
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:

In [ ]:

```
features =["floors", "waterfront","lat" ,"bedrooms" ,"sqft_basement" ,"view" ,"bathrooms"
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.

In [ ]:

```
from sklearn.linear_model import Ridge
```

In [ ]:

```
#perform a second order polynomial transform on the training data
model= Ridge(alpha=0.1)
# fit the Ridge regression object to the training data
model.fit(x_train, y_train)
# make predictions on the test data
y_pred= model.predict(x_test)
# calculate the R^2 score
r2= r2_score(y_test, y_pred)
print(r2)
```

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.

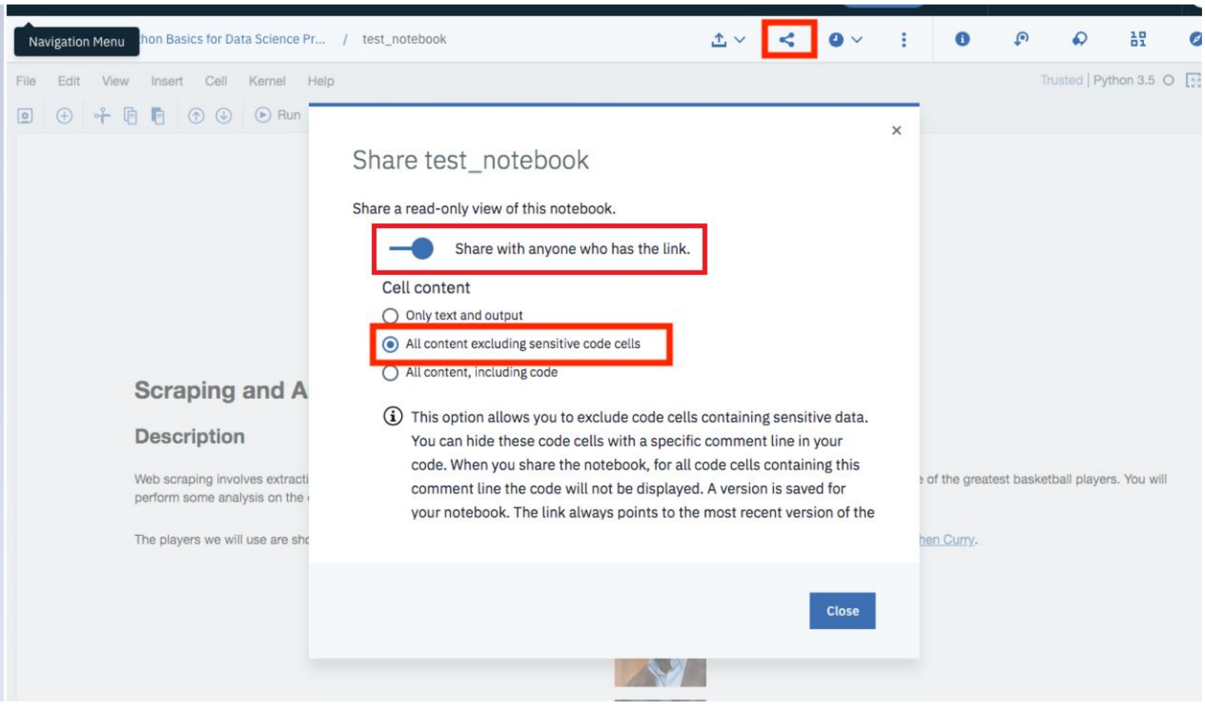
In [ ]:

```
#perform a second order polynomial transform on the training data
poly= PolynomialFeatures(degree=2)
x_train_poly= poly.fit_transform(x_train)
#perform a second order polynomial transform on the test data
x_test_poly= poly.transform(x_test)
```

In [ ]:

```
model= Ridge(alpha=0.1)
model.fit(x_train_poly, y_train)
y_pred= model.predict(x_test_poly)
r2= r2_score(y_test, y_pred)
print(r2)
```

Once you complete your notebook you will have to share it. Select the icon on the top right a marked in red in the image below, a dialogue box should open, and select the option all content excluding sensitive code cells.



Navigation Menu
hon Basics for Data Science Pr... / test_notebook
File Edit View Insert Cell Kernel Help
Run
Trusted | Python 3.5
Scraping and A
Description
Web scraping involves extracti ... of the greatest basketball players. You will perform some analysis on the
The players we will use are sho ... hen Curry.

Share test_notebook

Share a read-only view of this notebook.

Share with anyone who has the link.

Cell content

○ Only text and output

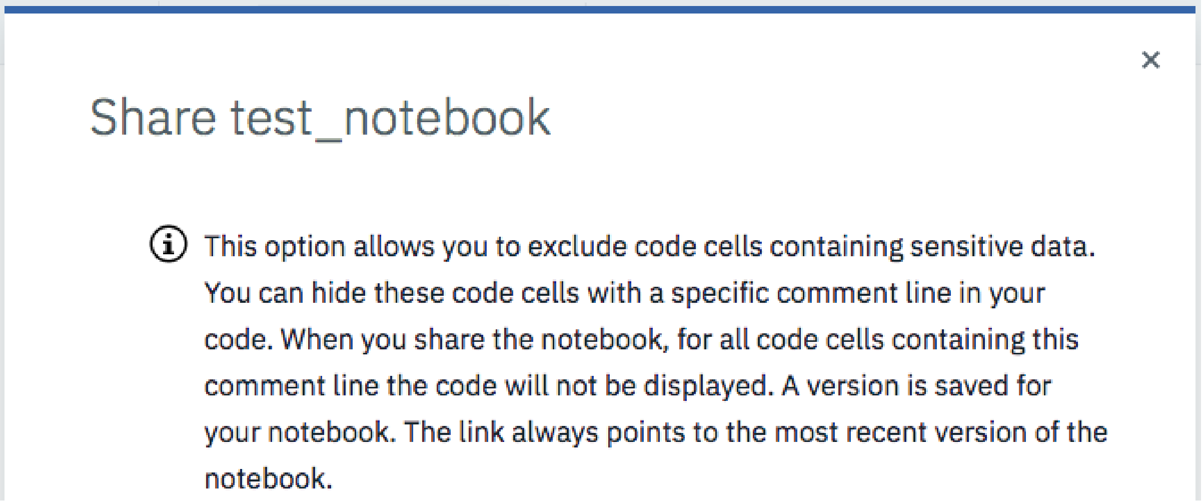
● All content excluding sensitive code cells

○ All content, including code

This option allows you to exclude code cells containing sensitive data. You can hide these code cells with a specific comment line in your code. When you share the notebook, for all code cells containing this comment line the code will not be displayed. A version is saved for your notebook. The link always points to the most recent version of the

Close

You can then share the notebook via a URL by scrolling down as shown in the following image:



Share test_notebook

This option allows you to exclude code cells containing sensitive data. You can hide these code cells with a specific comment line in your code. When you share the notebook, for all code cells containing this comment line the code will not be displayed. A version is saved for your notebook. The link always points to the most recent version of the notebook.

About the Authors:

[Joseph Santarcangelo (https://www.linkedin.com/in/joseph-s-50398b136/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDA0101ENSkillsNetwork20235326-2022-01-01)] 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 (https://www.linkedin.com/in/michelleccarey/?
utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id
SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDA0101ENSkillsNetwork20235326-
2022-01-01), Mavis Zhou (https://www.linkedin.com/in/jiahui-mavis-zhou-a4537814a?
utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id
SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDA0101ENSkillsNetwork20235326-
2022-01-01)

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

In [ ]: