Name-Yash Gandhi TE IT Batch A 2017140014

FCI EXP4

Aim: Experiment on Measuring fit and error parameters for a model.

Theory:

Human brains are built to recognize patterns in the world around us. For example, we observe that if we practice our programming every day, our related skills grow. But how do we precisely describe this relationship to other people? How can we describe how strong this relationship is? we can describe relationships between phenomena, such as practice and skill, in terms of formal mathematical estimations called regressions.

Regressions are one of the most used tools in a data scientist's kit. We can plug our data back into our regression equation to see if the predicted output matches corresponding observed value seen in the data.

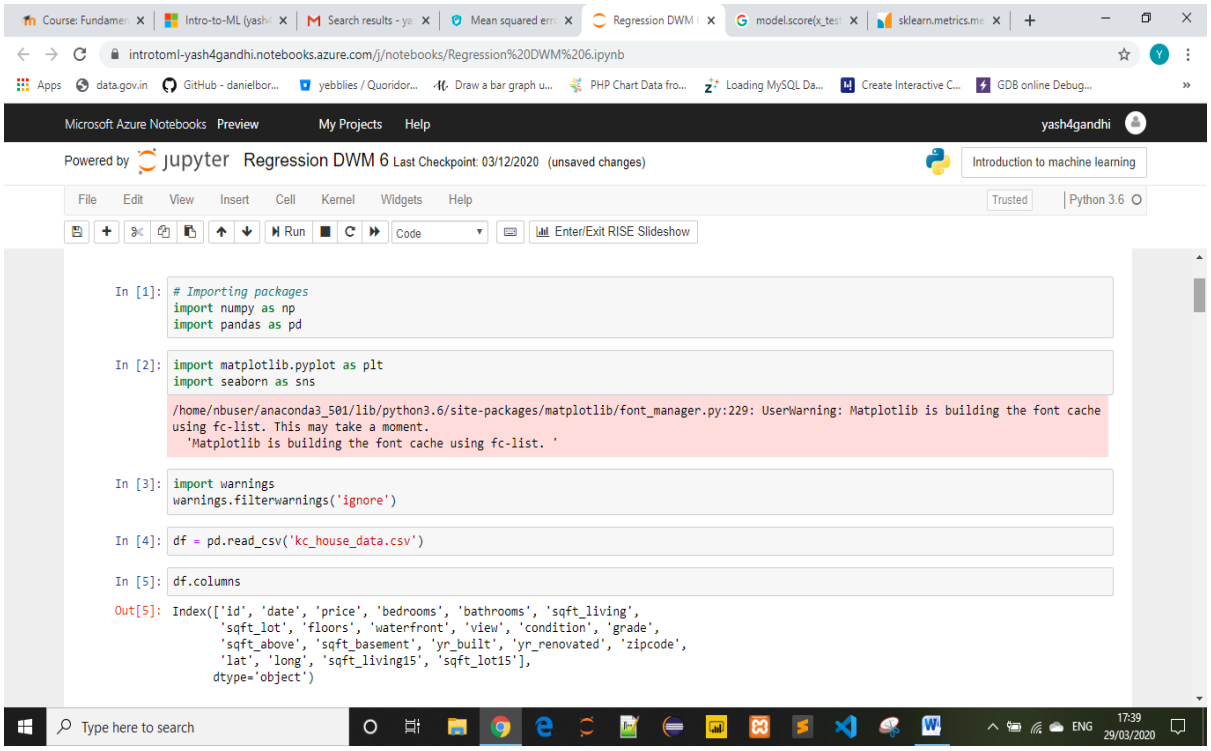
The quality of a regression model is how well its predictions match up against actual values, but how do we evaluate quality? Smart statisticians have developed error metrics to judge the quality of a model and enable us to compare regressions against other regressions with different parameters. These metrics are short and useful summaries of the quality of our data. There are many types of regression, but we will focus exclusively on metrics related to the linear regression.

The linear regression is the most used model in research and business and is the simplest to understand, so it makes sense to start developing your intuition on how they are assessed.

General steps to calculate the mean squared error from a set of X and Y values:

1) Find the regression line.
2) Insert your X values into the linear regression equation to find the new Y values (Y').
3) Subtract the new Y value from the original to get the error.
4) Square the errors.
5) Add up the errors.
6) Find the mean.

Using Simple linear regression to find house price



Microsoft Azure Notebooks Preview My Projects Help yash4gandhi

Powered by jupyter Regression DWM 6 Last Checkpoint: 03/12/2020 (unsaved changes)

Introduction to machine learning

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3.6

Run Code Enter/Exit RISE Slideshow

In [1]:
```
# Importing packages
import numpy as np
import pandas as pd
```

In [2]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

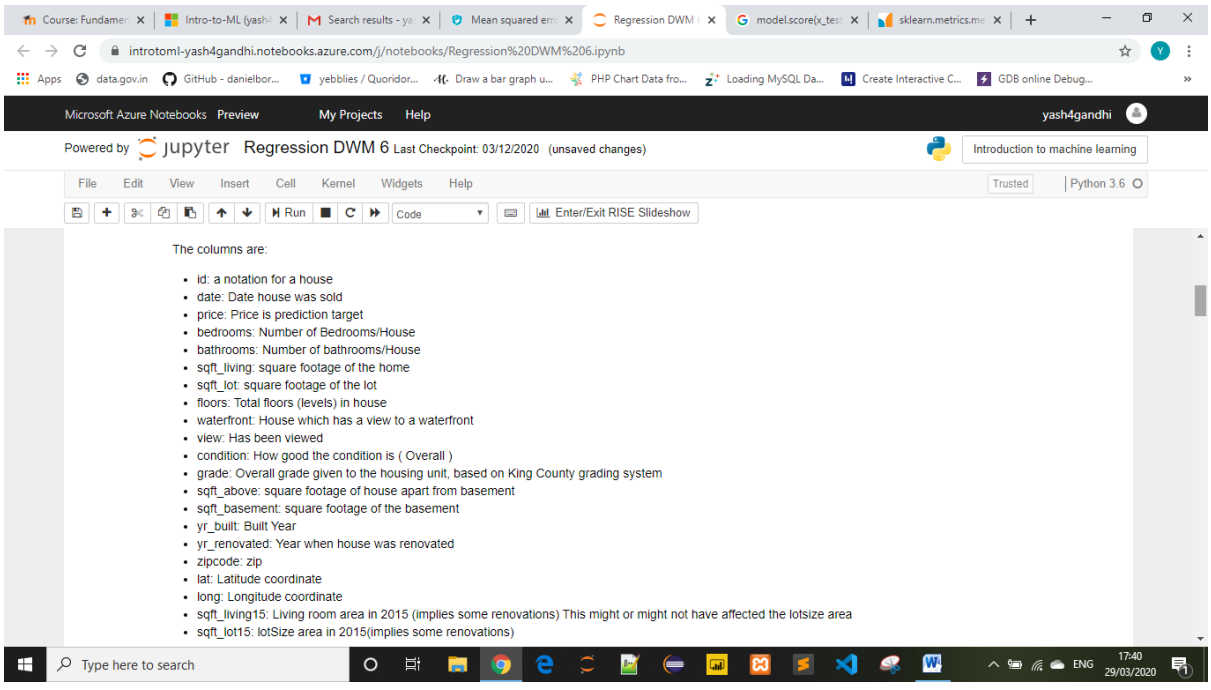
```
/home/nbuser/anaconda3_501/lib/python3.6/site-packages/matplotlib/font_manager.py:229: UserWarning: Matplotlib is building the font cache
using fc-list. This may take a moment.
  'Matplotlib is building the font cache using fc-list. '
```

In [3]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
df = pd.read_csv('kc_house_data.csv')
```

In [5]:
```
df.columns
```

Out[5]:
```
Index(['id', 'date', 'price', 'bedrooms', 'bathrooms', 'sqft_living',
       'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade',
       'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode',
       'lat', 'long', 'sqft_living15', 'sqft_lot15'],
      dtype='object')
```



Microsoft Azure Notebooks Preview My Projects Help yash4gandhi

Powered by jupyter Regression DWM 6 Last Checkpoint: 03/12/2020 (unsaved changes)

Introduction to machine learning

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3.6

Run Code Enter/Exit RISE Slideshow

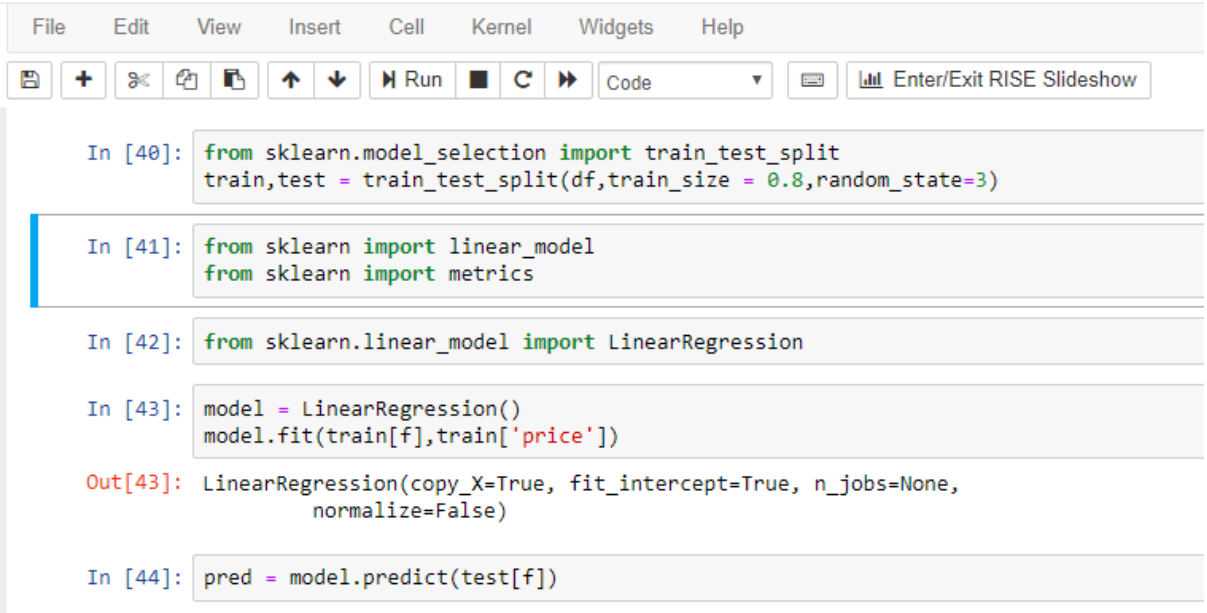
The columns are:

- id: a notation for a house
- date: Date house was sold
- price: Price is prediction target
- bedrooms: Number of Bedrooms/House
- bathrooms: Number of bathrooms/House
- 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
- 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)

We find the attribute that has the highest correlation with the prediction attribute price

In [50]:
```
f=['sqft_living']
```

Perform regression using the sklearn LinearRegression and predict the price



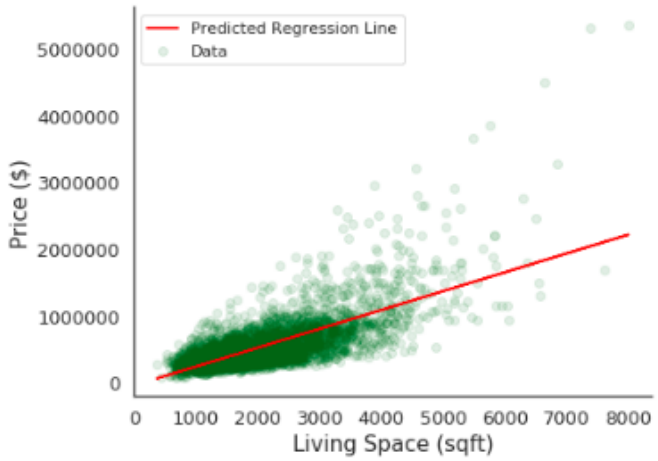
Output : General steps to calculate the mean squared error from a set of X and Y values:

1) Find the regression line.
2) Insert your X values into the linear regression equation to find the new Y values (Y').
3) Subtract the new Y value from the original to get the error.
4) Square the errors.
5) Add up the errors.
6) Find the mean.

Plotting the linear regression graph for price vs sqft_living

In [59]:
```
sns.set(style = "white", font_scale = 1)
plt.figure(figsize = (6.5,5))
plt.scatter(test[f], test['price'], color = 'darkgreen', label = "Data", alpha = 0.1)
plt.plot(test[f], model.predict(test[f]),color = "red", label = "Predicted Regression Line")
plt.xlabel("Living Space (sqft)", fontsize = 15)
plt.ylabel("Price ($)", fontsize = 15)
plt.xticks(fontsize = 13)
plt.yticks(fontsize = 13)
plt.legend()

plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)
```



File Edit View Insert Cell Kernel Widgets Help

Run Code Enter/Exit RISE Slideshow

In [47]:
```
actual = []
for i in test['price']:
    actual.append(i)

actual=np.array(actual)
Error=0
count=0
for i in range(len(pred)):
    se=pred[i]-actual[i]
    Error=Error+(se*se)
    #print(Error)
    count=count+1

MeanError=Error/count

print(MeanError)
print(count)
```

```
37517361292.793205
4323
```

In [48]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(test['price'],pred)
```

Out[48]:
```
37517361292.7932
```

Conclusion:

We have written the code for finding Mean Squared Error and also find it using the inbuilt sklearnmetric library and find that both are same