**Data Science Gold price Estimator:**

* Created a tool that estimates Gold price when they get a Gold.

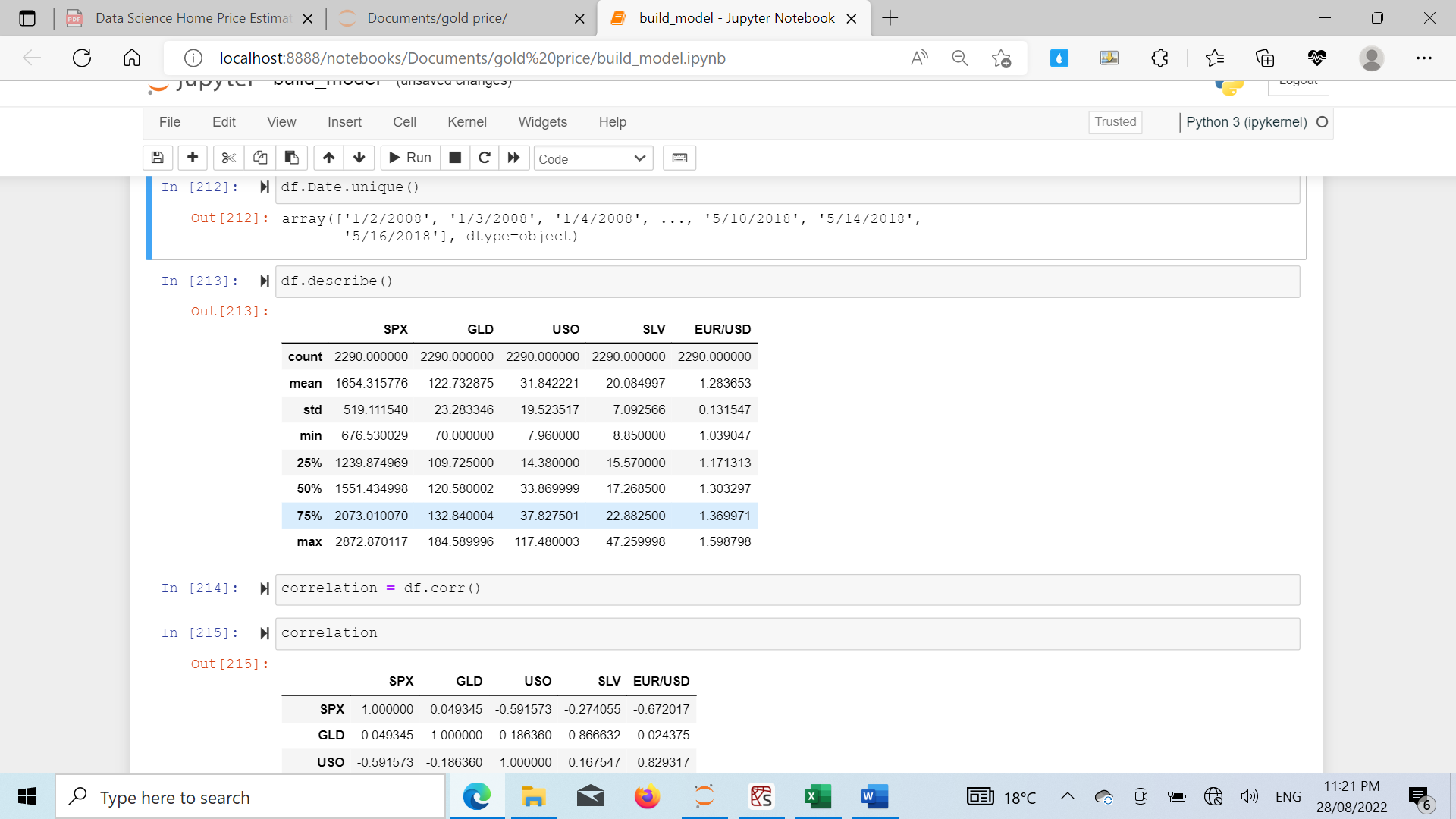
**Packages:** pandas, numpy, sklearn, matplotlib, seaborn.

**The solving mechanism**

* build machine learning model using python

**Describe the dataset**

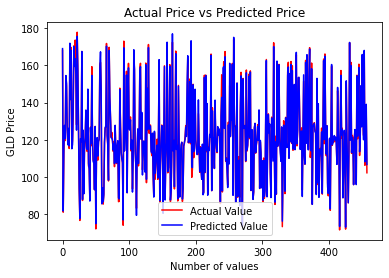
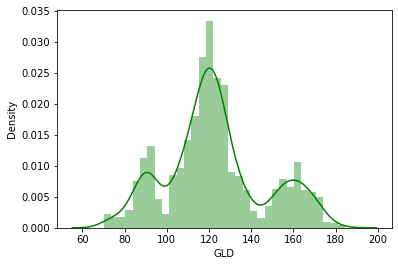
* Data source:
  + الرابط
* Data description
* I use pandas library to description dataset
  + df.info()
    - <class 'pandas.core.frame.DataFrame'>
    - RangeIndex: 2290 entries, 0 to 2289
    - Data columns (total 6 columns):
    - # Column Non-Null Count Dtype
    - --- ------ -------------- -----
    - 0 Date 2290 non-null datetime64[ns]
    - 1 SPX 2290 non-null float64
    - 2 GLD 2290 non-null float64
    - 3 USO 2290 non-null float64
    - 4 SLV 2290 non-null float64
    - 5 EUR/USD 2290 non-null float64
    - dtypes: datetime64[ns](1), float64(5)
    - memory usage: 107.5 KB
  + form output I know number of rows (2290) and num of columns (6)
  + name of columns and data type for each column
  + number of null values in columns
  + df.descibe()



* + I conclude from this table count , mean , min , median , max , standard deviation
  + From this information I know count of value in each column
  + Std mean standard deviation it help us to know the spread of values
  + Max , Min , mean , Median of each column

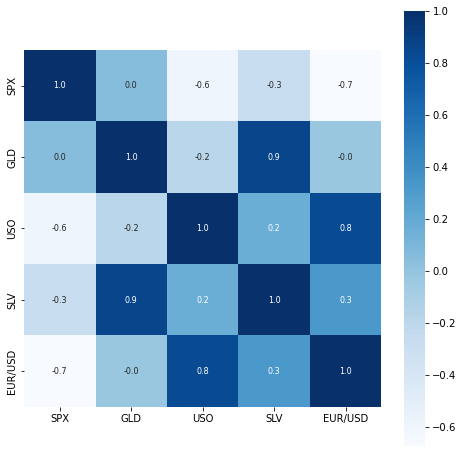
**descriptive statistics and data distribution charts**

I looked at the distributions of the data and the value counts for the various categorical variables. Below are a few highlights.

**Figure 1**

**Figure 2**



**Figure 3**

**التعليق**

* (figure 1) barplot in X name location Y count house in location
* ( figure 2 ) this chart dis correlation between [ total\_sqft , bath , price , bhk ]

**Model Building**

First, I split dataset to X an Y .

I also split the X and Y into train and tests sets with a test size of 20%.

I tried tow different models and I using GridsearchCV to reach the best model and best parameters.

I tried model:

* **Random Forest Regressor**– it can perform both regression and classification tasks. This will help us
* **Decision tree**– Because of the sparse data from the many categorical variables, I thought a normalized regression like lasso would be effective.

**Model performance**

The Random Forest Regressor model far outperformed the other approaches on the test and validation sets.

* **Random Forest Regressor**: = 99.45%
* **Decision tree**: = 99.40%