```
!pip install pandas
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/pyt
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dis
Requirement already satisfied: numpy>=1.15.4 in /usr/local/lib/python3.7/di
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-pa
```

Double-click (or enter) to edit

Importing libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn import metrics
```

Data collection and processing

```
#loading csv to pandas data frame
gold_data = pd.read_csv('/content/gld_price_data.csv')
```

```
#print first five rows
gold_data.head()
```

	Date	SPX	GLD	USO	SLV	EUR/USD
0	1/2/2008	1447.160034	84.860001	78.470001	15.180	1.471692
1	1/3/2008	1447.160034	85.570000	78.370003	15.285	1.474491
2	1/4/2008	1411.630005	85.129997	77.309998	15.167	1.475492
3	1/7/2008	1416.180054	84.769997	75.500000	15.053	1.468299
4	1/8/2008	1390.189941	86.779999	76.059998	15.590	1.557099

```
#print last five data frame
gold_data.tail()
```

	Date	SPX	GLD	USO	SLV	EUR/USD

✓ 0s completed at 7:44 PM

```
#number of rows and columns
gold_data.shape
```

```
(2290, 6)
```

```
#getting some basic info about data
gold_data.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   object
 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: float64(5), object(1)
memory usage: 107.5+ KB
```

```
#checking number of missing values
gold_data.isnull().sum()
```

```
Date       0
SPX        0
GLD        0
USO        0
SLV        0
EUR/USD    0
dtype: int64
```

```
#getting the statistical measures of the data
gold_data.describe()
```

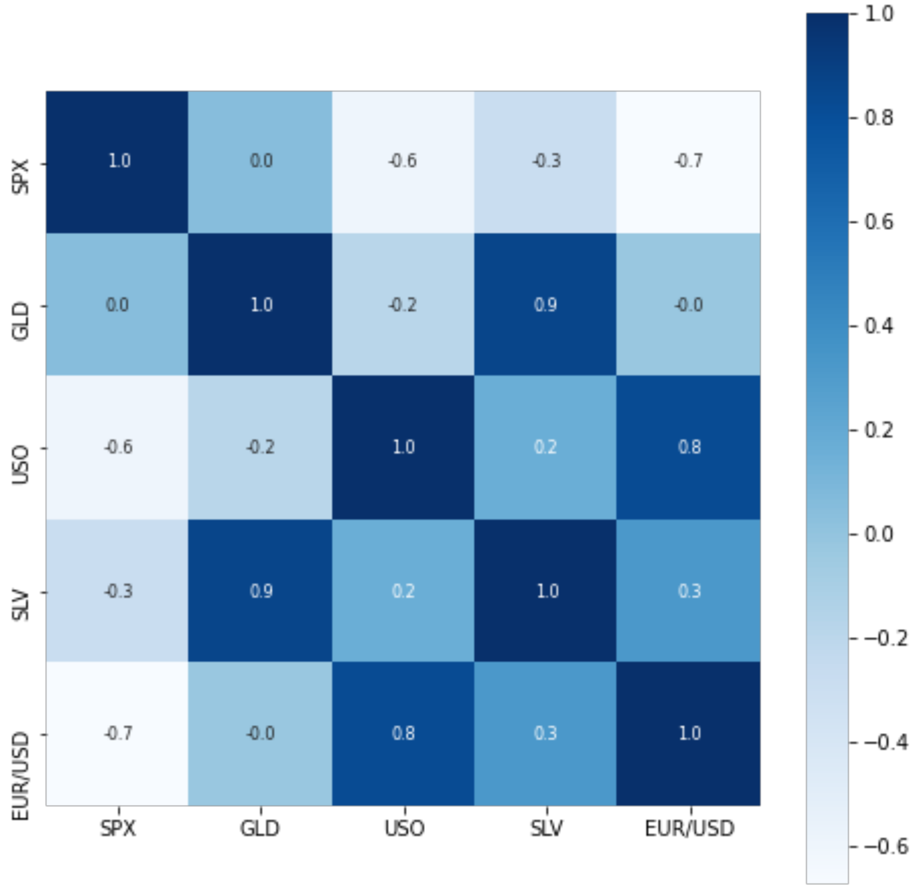
	SPX	GLD	USO	SLV	EUR/USD
count	2290.000000	2290.000000	2290.000000	2290.000000	2290.000000
mean	1654.315776	122.732875	31.842221	20.084997	1.283653
std	519.111540	23.283346	19.523517	7.092566	0.131547

correleation: 1.positive correlation 2.Negative correlation

```
correlation = gold_data.corr()
```

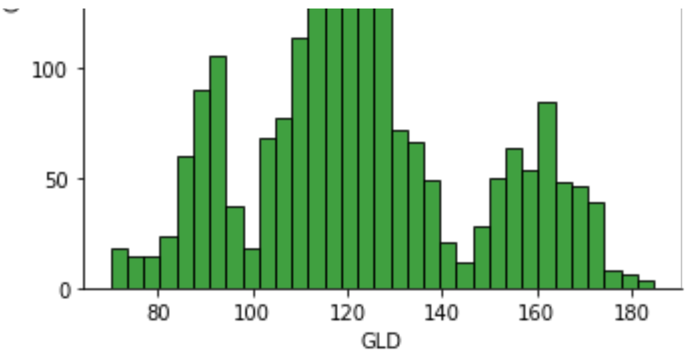
```
#constructing a heatmap to understand the correlation
plt.figure(figsize = (8,8))
sns.heatmap(correlation, cbar=True, square=True, fmt='.1f',annot=True, annot_kws
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc9717b6ed0>
```




```
#correlation values of GLD
print(correlation['GLD'])
```

```
SPX        0.049345
GLD        1.000000
```



splitting the features and target (gold and date)

```
X=gold_data.drop(['Date','GLD'],axis=1)
Y=gold_data['GLD']
```

```
print(X)
```

```
              SPX        USO      SLV   EUR/USD
0     1447.160034  78.470001  15.1800  1.471692
1     1447.160034  78.370003  15.2850  1.474491
2     1411.630005  77.309998  15.1670  1.475492
3     1416.180054  75.500000  15.0530  1.468299
4     1390.189941  76.059998  15.5900  1.557099
...           ...        ...      ...       ...
2285  2671.919922  14.060000  15.5100  1.186789
2286  2697.790039  14.370000  15.5300  1.184722
2287  2723.070068  14.410000  15.7400  1.191753
2288  2730.129883  14.380000  15.5600  1.193118
2289  2725.780029  14.405800  15.4542  1.182033

[2290 rows x 4 columns]
```

model Training: Random forest Regressor

```
regressor = RandomForestRegressor(n_estimators=100)
```

```
#training the model
regressor.fit(X_train,Y_train)
```

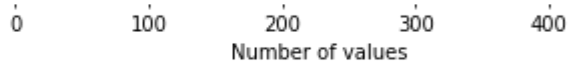
```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=None, max_features='auto', max_leaf_nodes=N
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=100, n_jobs=None, oob_score=False,
                      random_state=None, verbose=0, warm_start=False)
```

Model evaluation

```
#prediction on Test Data
test_data_prediction=regressor.predict(X_test)
```

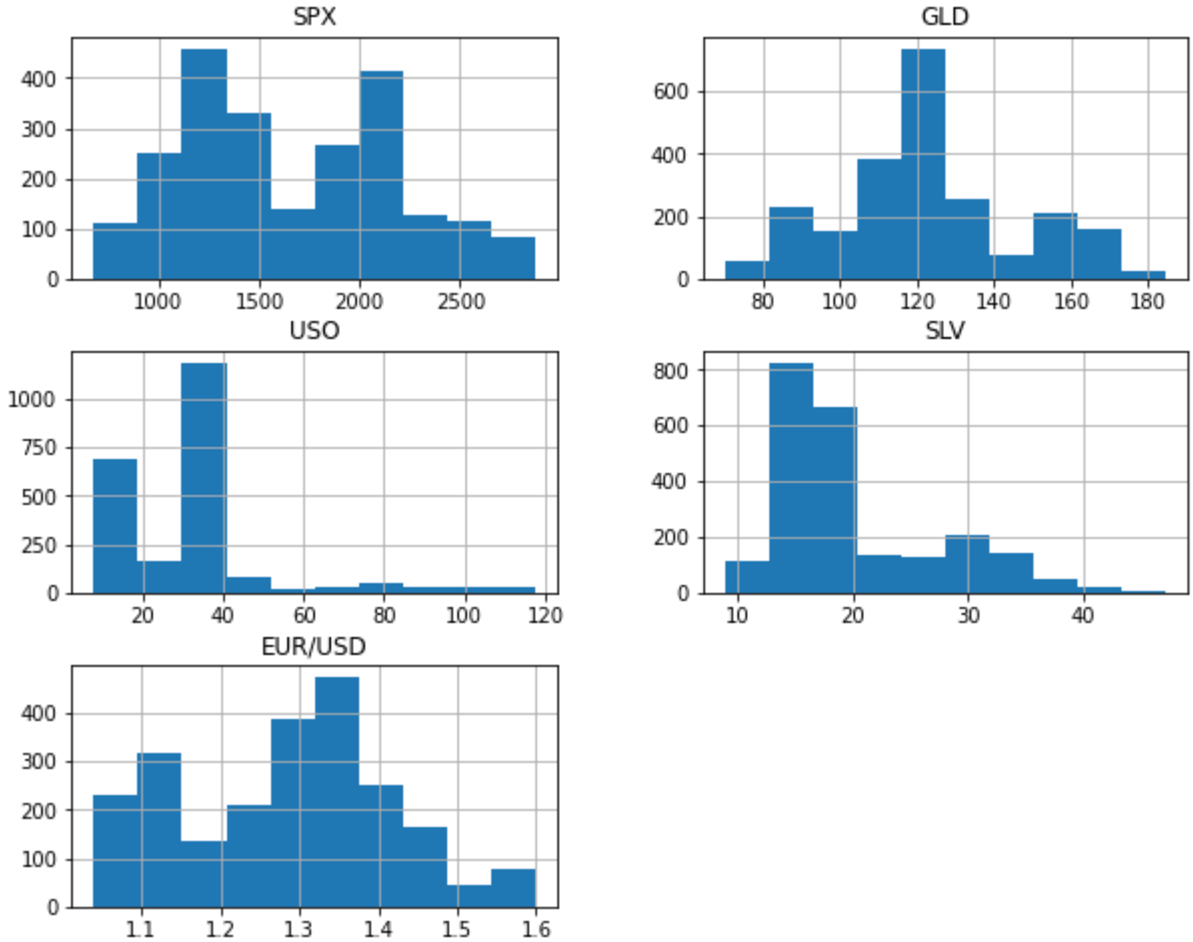
```
print(test_data_prediction)
```

```
 177.18359975 114.05389975 119.23850022  94.60920094 125.4568
 166.2639006  114.79080043 116.48660139  88.30239884 149.55870083
 120.31389971  89.32329984 112.22320006 117.52550016 118.75680106
  88.04179935  94.07690013 116.99959999 118.74670211 120.34000094
 126.72729835 121.83909961 147.83840026 164.82509981 118.55569972
 120.32310175 151.25430102 118.56999921 172.58109838 105.30009928
 105.00220059 150.11530103 113.54730116 124.77520086 147.95020051
 119.75630129 115.49720053 112.70910013 113.45340173 141.96240134
 117.7650977  102.95310047 115.81740104 103.94700204  98.56550042
 117.43340099  90.75380014  91.55400059 153.69229955 102.77949977
 154.72820079 114.28770122 138.83100151  90.13569812 115.60169971
 114.58449933 123.03300034 121.77270013 165.21610137  92.83459979
 135.127201   121.43629898 120.90700049 104.77760014 140.45060319
 121.83209901 116.64780041 113.58400063 127.01209765 122.82469942
 125.80669935 121.26220039  86.76309869 132.51220131 145.9791022
  92.6725994  159.18219961 159.50600246 126.35929908 164.97949922
 108.86529956 109.54000102 103.90329871  94.38880039 127.80910279
 107.09960039 161.64299963 121.75510044 132.20719989 130.2038021
 160.27409968  90.18799843 176.20870169 127.47270027 126.67839891
  86.36019928 124.4944992  150.60619729  89.55320025 106.85389964
 109.02779995  84.41309903 136.06449885 155.10220289 138.41960359
  74.34940021 152.71120143 126.10559995 126.66710051 127.4386992
 108.7787     156.05550057 114.64670114 116.89190147 125.13259945
 154.00800162 121.35999979 156.38809867  92.8839007  125.44870155
 125.80250019  87.78760045  92.11269925 126.14029967 128.2608033
```




```
gold_data.hist(figsize=(10,8))
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fc962c6fd10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fc962c29350>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fc962bde9d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fc962c09b90>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fc962b57710>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fc962b118d0>]],
      dtype=object)
```



std	NaN	319.111540	23.283346	19.523517	7.092566
min	NaN	676.530029	70.000000	7.960000	8.850000
25%	NaN	1239.874969	109.725000	14.380000	15.570000
50%	NaN	1551.434998	120.580002	33.869999	17.268500
75%	NaN	2073.010070	132.840004	37.827501	22.882499
max	NaN	2872.870117	184.589996	117.480003	47.259998