```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
df=pd.read_excel("fake_reg.xlsx")
df.head()
```

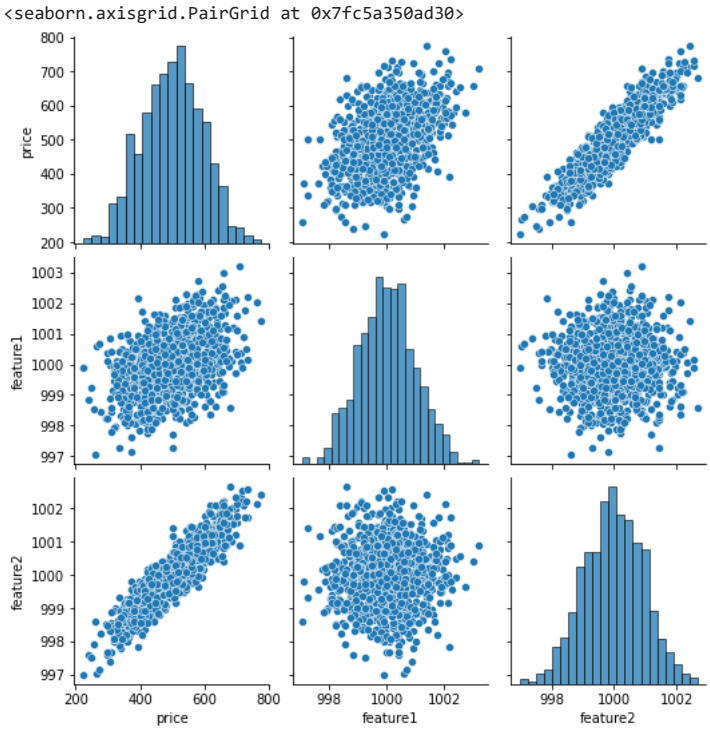
	price	feature1	feature2
0	461.527929	999.787558	999.766096
1	548.130011	998.861615	1001.042403
2	410.297162	1000.070267	998.844015
3	540.382220	999.952251	1000.440940
4	546.024553	1000.446011	1000.338531

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   price     1000 non-null   float64
 1   feature1  1000 non-null   float64
 2   feature2  1000 non-null   float64
dtypes: float64(3)
memory usage: 23.6 KB
```

```
sns.pairplot(data=df)
```

```
<seaborn.axisgrid.PairGrid at 0x7fc5a350ad30>
```




```
df.corr()
```

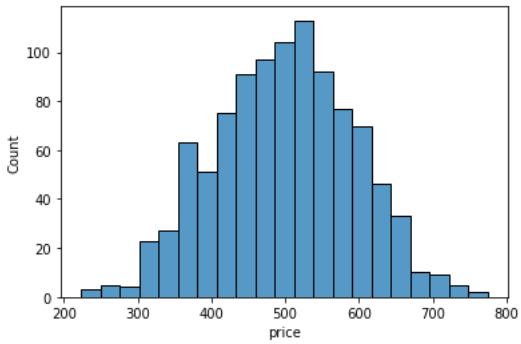
price	feature1	feature2

```
df.describe()
```

	price	feature1	feature2
count	1000.000000	1000.000000	1000.000000
mean	498.673029	1000.014171	999.979847
std	93.785431	0.974018	0.948330
min	223.346793	997.058347	996.995651
25%	433.025732	999.332068	999.316106
50%	502.382117	1000.009915	1000.002243
75%	564.921588	1000.637580	1000.645380
max	774.407854	1003.207934	1002.666308

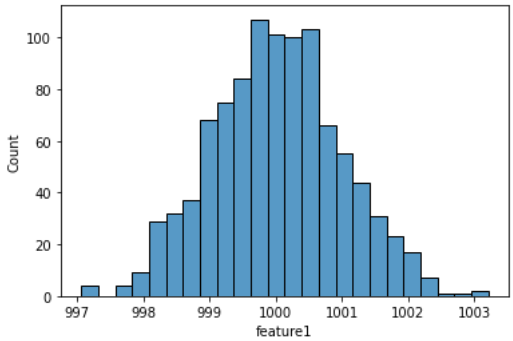
```
sns.histplot(df["price"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc5a05e2790>
```




```
sns.histplot(df["feature1"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc5a30b3610>
```




```
#sep X and Y
```

```
df.head()
```

	price	feature1	feature2
0	461.527929	999.787558	999.766096
1	548.130011	998.861615	1001.042403
2	410.297162	1000.070267	998.844015
3	540.382220	999.952251	1000.440940
4	546.024553	1000.446011	1000.338531

```
x=df.iloc[:,1:].values
x
```

```
array([[ 999.78755752,  999.7660962 ],
       [ 998.86161491, 1001.04240315],
       [1000.07026691,  998.84401463],
       ...,
       [1001.45164617,  998.84760554],
       [1000.77102275,  998.56285086],
       [ 999.2322436 , 1001.45140713]])
```

```
y=df["price"].values
y
```

```
344.445388  , 603.44292484, 494.19769248, 623.10445015,
615.5627227 , 501.09520221, 460.4308638 , 511.19375671,
619.10478486, 425.20422156, 439.38667763, 609.7226948 ,
520.97315983, 564.91022293, 574.57307473, 273.43263798,
416.57967214, 414.00400534, 602.54914671, 501.81396346,
657.25888889, 545.58520276, 363.11433585, 587.10134867,
535.98312568, 357.044414  , 543.91850962, 487.46952396,
491.03773416, 618.12341696, 442.93357949, 528.44015453,
492.87327105, 497.04387645, 606.29736199, 368.90204456,
463.83315596, 411.35636173, 670.21237633, 408.52236362,
484.43638366, 492.6419263 , 381.85298995, 613.95995262,
327.36024129, 491.79240663, 585.44412785, 380.3945669 ,
554.7626666 , 493.85668075, 526.2107462 , 385.39655988,
484.46159148, 332.22764352, 564.67207171, 649.44006825,
424.72778986, 618.08695111, 511.89276394, 331.12168828,
558.16409431, 493.56964948, 396.05573683, 631.37140169,
478.44486619, 628.54499456, 499.75474684, 506.8234736 ,
258.64403747, 478.35823792, 460.54250824, 552.62665953,
519.60816751, 456.98761551, 562.62597526, 560.98721013,
588.24268147, 579.43065472, 490.68891134, 372.77743562,
677.27628686, 420.76621323, 478.08636807, 608.79918051,
498.40036414, 351.09008918, 604.7965235 , 584.29415078,
552.36859639, 514.85636404, 324.75278473, 551.65009963,
646.42137115, 412.01627785, 542.05581314, 376.82027575,
485.65542199, 530.00104432, 531.48237221, 502.90170221,
320.22783441, 410.17228689, 307.49262471, 414.32154328,
429.47259522, 611.37280691, 579.49800824, 645.84815363,
528.21987259, 487.29933081, 530.99314592, 374.19760282,
453.82582684, 535.04562659, 381.36832678, 657.71868008,
403.06256371, 502.40592148, 612.48716221, 473.14373027,
648.49853653, 477.59301587, 478.85719007, 469.11310605,
558.87741587, 560.67805186, 484.56010504, 602.9701103 ,
501.88831717, 296.98984675, 409.95257743, 429.21452014,
529.71971585, 503.21614081, 656.65474461, 378.81430465,
627.64016416, 420.35676968, 347.19521436, 559.96108847,
437.90401436, 382.48198573, 446.61250992, 578.94348317,
427.85700338, 563.58363749, 603.55869587, 445.94046658,
577.31665049, 453.9716661 , 367.16096642, 591.50358214,
404.90316218, 403.90327413, 425.48260511, 657.80019814,
664.50877632, 503.05044876, 518.66756145, 544.49878325,
409.67675815, 338.5727094 , 427.48699506, 349.71799435,
434.38654316, 549.93636627, 524.5896561 , 502.90947332,
342.05906592, 622.03205011, 654.83912076, 434.69375374,
470.5546375 , 380.47707861, 560.3817655 , 549.2573511 ,
494.25163915, 450.42484415, 594.84047708, 412.90687671,
433.42313493, 512.04979227, 478.04811801, 413.29973991,
415.84564074, 511.58583419, 472.89142434, 657.93173622,
560.66409232, 587.74308283, 472.06766427, 390.86267833,
329.06703513, 526.72673086, 452.10238608, 413.87226234,
666.44170815, 487.7692307 , 560.63785511, 449.81047626,
499.94830987, 429.26772106, 508.91176483, 546.20393322,
621.59703226, 413.84178036, 449.02606966, 447.05319043,
595.48648065, 480.87733296, 667.70856487, 594.60958875,
600.04329091, 508.19978205, 359.19507992, 365.88328071,
576.73842525, 515.32519058, 423.65453319, 543.49227764,
577.29292367, 421.05894642, 625.08018462, 434.27528283,
657.73425067, 518.35614426, 481.86280607, 476.52607826,
457.31318609, 456.72099249, 403.31557562, 599.36709348])
```

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x=sc.fit_transform(x)
```

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.30,random_state=1)
```

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
```

```
from tensorflow.keras.layers import Dense
```

```
from scipy.stats.morestats import optimize
ann=Sequential()
ann.add(Dense(units=4,activation="relu"))
ann.add(Dense(units=4,activation="relu"))

ann.add(Dense(units=1))

ann.compile(optimizer="adam", loss="mse")
ann.fit(xtrain,ytrain,epochs=250)
```

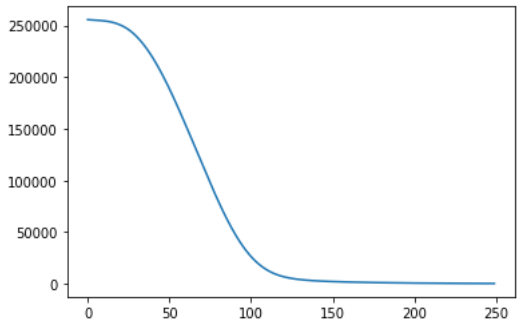
```
22/22 [==============================] - 0s 2ms/step - loss: 558.4442
Epoch 223/250
22/22 [==============================] - 0s 1ms/step - loss: 549.6254
Epoch 224/250
22/22 [==============================] - 0s 2ms/step - loss: 541.2476
Epoch 225/250
22/22 [==============================] - 0s 1ms/step - loss: 532.5423
Epoch 226/250
22/22 [==============================] - 0s 1ms/step - loss: 524.6273
Epoch 227/250
22/22 [==============================] - 0s 2ms/step - loss: 516.1660
Epoch 228/250
22/22 [==============================] - 0s 2ms/step - loss: 508.4138
Epoch 229/250
22/22 [==============================] - 0s 1ms/step - loss: 500.9248
Epoch 230/250
22/22 [==============================] - 0s 1ms/step - loss: 493.6517
Epoch 231/250
22/22 [==============================] - 0s 1ms/step - loss: 485.7346
Epoch 232/250
22/22 [==============================] - 0s 1ms/step - loss: 478.6998
Epoch 233/250
22/22 [==============================] - 0s 1ms/step - loss: 471.9727
Epoch 234/250
22/22 [==============================] - 0s 2ms/step - loss: 464.8781
Epoch 235/250
22/22 [==============================] - 0s 1ms/step - loss: 458.1343
Epoch 236/250
22/22 [==============================] - 0s 1ms/step - loss: 451.3167
Epoch 237/250
22/22 [==============================] - 0s 2ms/step - loss: 444.7928
Epoch 238/250
22/22 [==============================] - 0s 2ms/step - loss: 438.4914
Epoch 239/250
22/22 [==============================] - 0s 2ms/step - loss: 432.0179
Epoch 240/250
22/22 [==============================] - 0s 2ms/step - loss: 425.7774
Epoch 241/250
22/22 [==============================] - 0s 2ms/step - loss: 420.0637
Epoch 242/250
22/22 [==============================] - 0s 2ms/step - loss: 413.6709
Epoch 243/250
22/22 [==============================] - 0s 2ms/step - loss: 407.9890
Epoch 244/250
22/22 [==============================] - 0s 2ms/step - loss: 402.7051
Epoch 245/250
22/22 [==============================] - 0s 1ms/step - loss: 396.5181
Epoch 246/250
22/22 [==============================] - 0s 2ms/step - loss: 390.9955
Epoch 247/250
22/22 [==============================] - 0s 1ms/step - loss: 385.6749
Epoch 248/250
22/22 [==============================] - 0s 2ms/step - loss: 380.2315
Epoch 249/250
22/22 [==============================] - 0s 1ms/step - loss: 374.9934
Epoch 250/250
22/22 [==============================] - 0s 1ms/step - loss: 370.3470
<keras.callbacks.History at 0x7fc53e66a490>
```

```
loss=ann.history.history
```

```
loss_df=pd.DataFrame(loss)
```

```
plt.plot(loss_df['loss'])
```

```
[<matplotlib.lines.Line2D at 0x7fc53e669640>]
```




```
ypred=ann.predict(xtest)
```

```
10/10 [==============================] - 0s 2ms/step
```

```
ypred
```

```
pd.DataFrame({"Actual value":ytest,"predicted value":ypred.flatten()})
```

	Actual value	predicted value
0	489.057552	482.515289
1	526.210746	526.215759
2	518.360395	494.238770
3	552.778935	534.680481
4	649.395917	662.959229
...	...	...
295	530.276259	539.250732
296	481.645813	473.905182
297	460.694781	459.814667
298	518.263124	517.037842
299	512.595702	506.432861

300 rows × 2 columns

```
from sklearn.metrics import r2_score
r2_score(ytest,ypred)
```

```
0.9732481854406265
```

```
newdata=[[900,950]]
nd=sc.fit_transform(newdata)
ann.predict(nd)
```

```
1/1 [==============================] - 0s 103ms/step
array([[495.19467]], dtype=float32)
```

✓ 0s completed at 3:44 PM

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.