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

In [2]:
```
dataset=pd.read_csv(r"D:\ML_Course\Works_on_python\Polynomial_Regression\Position_Salaries.csv")
```

In [3]:
```
dataset.head() # output is varrying exponentially
```

Out[3]:

	Position	Level	Salary
0	Business Analyst	1	45000
1	Junior Consultant	2	50000
2	Senior Consultant	3	60000
3	Manager	4	80000
4	Country Manager	5	110000

no need for position coz having level

In [4]:
```
dataset.isnull().any()
```

Out[4]:
```
Position    False
Level       False
Salary      False
dtype: bool
```

In [5]:
```
x=dataset.iloc[:,1:2].values
y=dataset.iloc[:,2:3].values
```

```
In [6]: x
```

```
Out[6]: array([[ 1],
               [ 2],
               [ 3],
               [ 4],
               [ 5],
               [ 6],
               [ 7],
               [ 8],
               [ 9],
               [10]], dtype=int64)
```

```
In [7]: y
```

```
Out[7]: array([[  45000],
               [  50000],
               [  60000],
               [  80000],
               [ 110000],
               [ 150000],
               [ 200000],
               [ 300000],
               [ 500000],
               [1000000]], dtype=int64)
```

```
In [8]: import matplotlib.pyplot as plt
        plt.scatter(dataset["Level"],dataset["Salary"])
```

```
Out[8]: <matplotlib.collections.PathCollection at 0x162dc242278>
```

```
In [9]: from sklearn.linear_model import LinearRegression
        lin_reg=LinearRegression()
        lin_reg.fit(x,y)
```

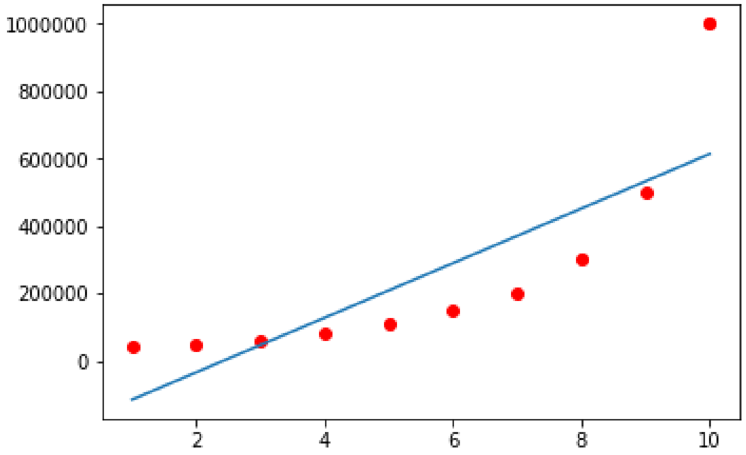
```
Out[9]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
                 normalize=False)
```

```
In [10]: y_pred=lin_reg.predict(x)
```

In [11]:
```
#plotting best fit line

plt.scatter(x,y,color="red")#x is ques y is key paper
plt.plot(x,y_pred)#y_pred is ans for key paper
```

Out[11]:
```
[<matplotlib.lines.Line2D at 0x162de814f98>]
```



In [12]:
```
y
```

Out[12]:
```
array([[  45000],
       [  50000],
       [  60000],
       [  80000],
       [ 110000],
       [ 150000],
       [ 200000],
       [ 300000],
       [ 500000],
       [1000000]], dtype=int64)
```

In [13]:

```
y_pred
```

Out[13]:

```
array([[-114454.54545455],
       [ -33575.75757576],
       [  47303.03030303],
       [ 128181.81818182],
       [ 209060.60606061],
       [ 289939.39393939],
       [ 370818.18181818],
       [ 451696.96969697],
       [ 532575.75757576],
       [ 613454.54545455]])
```

In [14]:

```
from sklearn.metrics import r2_score
accuracylinear=r2_score(y,y_pred)
```

In [15]:

```
accuracylinear # poor accuracy
```

Out[15]:

```
0.6690412331929895
```

```
must do model_selection that is train test split
here only smaller rows x=10 at In[6] thats why no needed but it is must
```

In [16]:

```
x
```

Out[16]:

```
array([[ 1],
       [ 2],
       [ 3],
       [ 4],
       [ 5],
       [ 6],
       [ 7],
       [ 8],
       [ 9],
       [10]], dtype=int64)
```

In [17]:

```
from sklearn.preprocessing import PolynomialFeatures
poly_reg=PolynomialFeatures(degree=4) # power 4, chnage degree and see accuracy
x_poly=poly_reg.fit_transform(x)
```

In [18]:
```
x_poly
```

Out[18]:
```
array([[1.000e+00, 1.000e+00, 1.000e+00, 1.000e+00, 1.000e+00],
       [1.000e+00, 2.000e+00, 4.000e+00, 8.000e+00, 1.600e+01],
       [1.000e+00, 3.000e+00, 9.000e+00, 2.700e+01, 8.100e+01],
       [1.000e+00, 4.000e+00, 1.600e+01, 6.400e+01, 2.560e+02],
       [1.000e+00, 5.000e+00, 2.500e+01, 1.250e+02, 6.250e+02],
       [1.000e+00, 6.000e+00, 3.600e+01, 2.160e+02, 1.296e+03],
       [1.000e+00, 7.000e+00, 4.900e+01, 3.430e+02, 2.401e+03],
       [1.000e+00, 8.000e+00, 6.400e+01, 5.120e+02, 4.096e+03],
       [1.000e+00, 9.000e+00, 8.100e+01, 7.290e+02, 6.561e+03],
       [1.000e+00, 1.000e+01, 1.000e+02, 1.000e+03, 1.000e+04]])
```

In [19]:
```
poly_reg.fit(x_poly,y) # making equation with values of x_poly
```

Out[19]:
```
PolynomialFeatures(degree=4, include_bias=True, interaction_only=False)
```

In [20]:
```
lin_reg_2=LinearRegression()
lin_reg_2.fit(x_poly,y)
```

Out[20]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [36]:
```
poly=lin_reg_2.predict(x_poly)
```

In [37]:
```
y
```

Out[37]:
```
array([[  45000],
       [  50000],
       [  60000],
       [  80000],
       [ 110000],
       [ 150000],
       [ 200000],
       [ 300000],
       [ 500000],
       [1000000]], dtype=int64)
```

In [45]:
```
poly
```

Out[45]:
```
array([[ 53356.64335683],
       [ 31759.90675992],
       [ 58642.19114218],
       [ 94632.86713289],
       [121724.941725  ],
       [143275.05827513],
       [184003.49650353],
       [289994.17249412],
       [528694.6386945 ],
       [988916.0839159 ]])
```

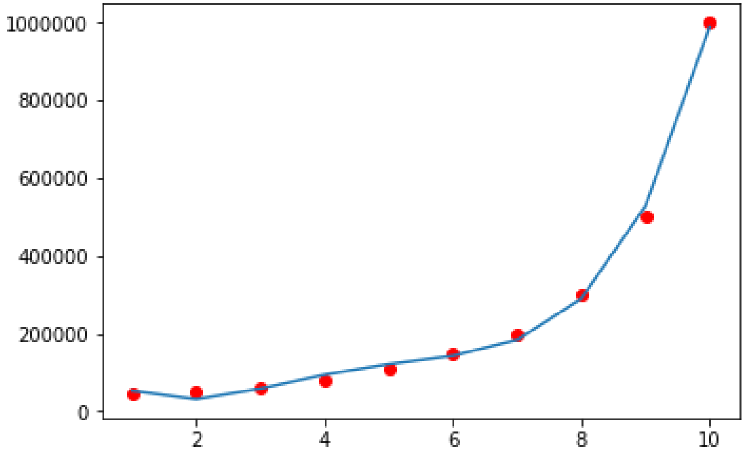
In [38]:
```
accuracy=r2_score(y,poly)
```

In [42]:
```
accuracy
```

Out[42]:
```
0.9973922891706614
```

In [39]:
```
plt.scatter(x,y,color="red")
plt.plot(x,poly)
```

Out[39]:
```
[<matplotlib.lines.Line2D at 0x1be7b0f1748>]
```



In [40]:
```
y=lin_reg_2.predict([[1,2,4,8,16]])
```

```
In [41]: y
```

```
Out[41]: array([[31759.90675992]])
```