importing libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
# importing dataset

ds=pd.read_csv('position_salaries_dataset.csv')

x=ds.iloc[:,1:-1].values
y=ds.iloc[:,-1].values
```

In [3]:
```
x
```

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

In [4]:
```
y
```

Out[4]:
```
array([  45000,   50000,   60000,   80000,  110000,  150000,  200000,
        300000,  500000, 1000000])
```

In [5]:
```
# training the linear regression

from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(x,y)
```

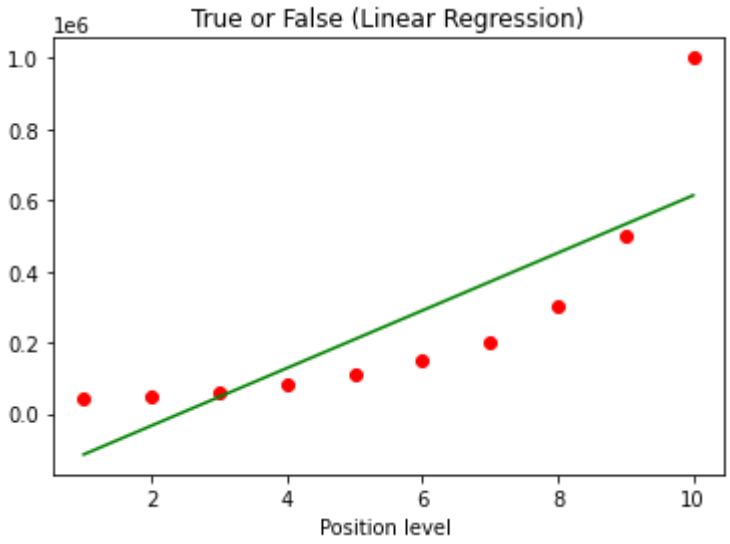
Out[5]:
```
▾ LinearRegression

LinearRegression()
```

In [10]:
```
# training the ploynominal regression model on the dataset

from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree = 2)
x_poly = poly_reg.fit_transform(x)
lin_reg_2 = LinearRegression()
lin_reg_2.fit(x_poly,y)
```

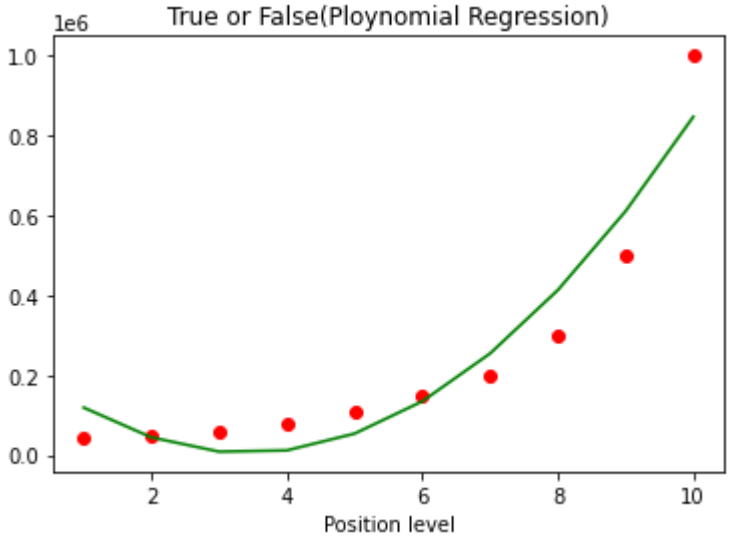
Out[10]:
```
▾ LinearRegression

LinearRegression()
```

In [15]:
```
#visualizing linear regression
```

```
plt.scatter(x, y, color ='red' )
plt.plot(x, lin_reg.predict(x), color = 'green')
plt.title('True or False (Linear Regression)')
plt.xlabel('Position level ')
plt.show('salary')
plt.show()
```



In [16]:
```
#Visualizing the polynomial Regression results

plt.scatter(x ,y,color = 'red' )
plt.plot(x ,lin_reg_2.predict(poly_reg.fit_transform(x)),color = 'green')
plt.title('True or False(Ploynomial Regression)')
plt.xlabel('Position level')
plt.show('salary')
plt.show()
```



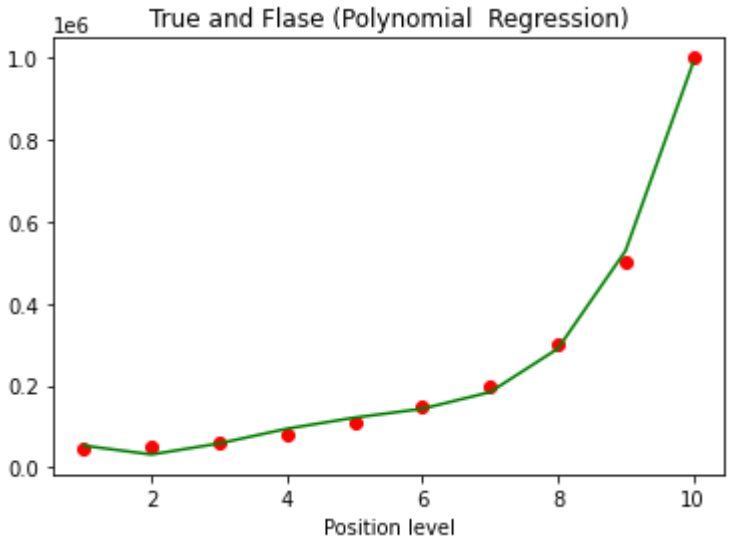
In [17]:
```
#Lets increase the degree

poly_reg = PolynomialFeatures(degree = 4 )
x_poly = poly_reg.fit_transform(x)
lin_reg_2 = LinearRegression()
lin_reg_2.fit(x_poly,y)
```

Out[17]:

```
LinearRegression()
```

In [18]:

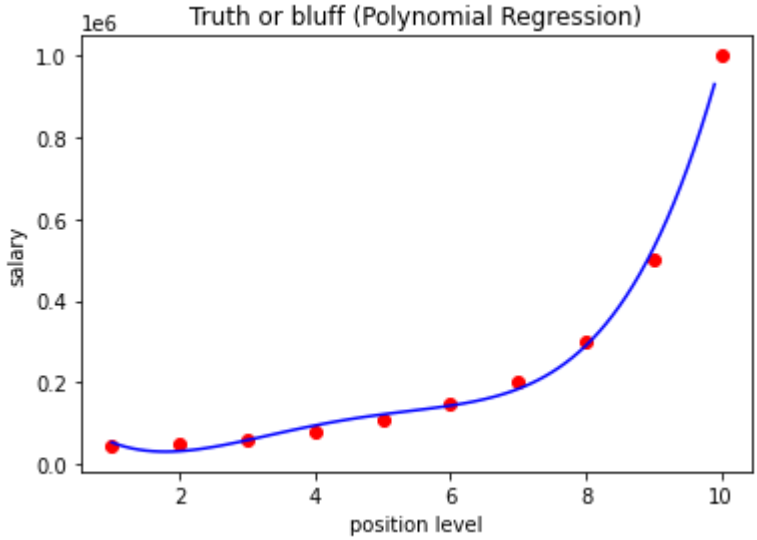
```
#Visualsing the polynomial regression

plt.scatter(x,y, color ='red' )
plt.plot(x, lin_reg_2.predict(poly_reg.fit_transform(x)), color = 'green')
plt.title('True and Flase (Polynomial  Regression)')
plt.xlabel('Position level')
plt.show('salary')
plt.show()
```



Visualzing the polynomial Regression results (for higher resolution and smoother curve)

In [22]:

```
x_grid = np.arange(min(x) ,max(x), 0.1 )
x_grid = x_grid.reshape(len(x_grid),1)
plt.scatter (x , y , color  = 'red')
plt.plot (x_grid , lin_reg_2.predict(poly_reg.fit_transform(x_grid)), color = 'blue
plt.title('Truth or bluff (Polynomial Regression)')
plt.xlabel('position level')
plt.ylabel('salary')
plt.show ()
```



In [23]:
```
#Predicting a new result with plonnomial Regression
lin_reg.predict([[6.5]])
```

Out[23]:
```
array([330378.78787879])
```

In [25]:
```
#predicting  a new results with ploymial Regression

lin_reg_2.predict(poly_reg.fit_transform([[6.5]]))
```

Out[25]:
```
array([158862.4526515])
```

In [ ]: