Non Linear Regression

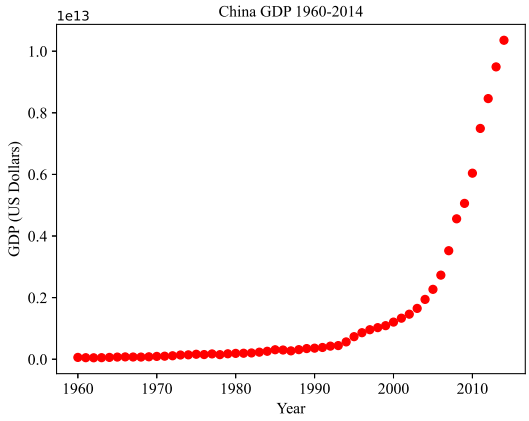
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import rc,font_manager

##Loading Data ##
ticks_font = font_manager.FontProperties(family='Times New Roman', style='normal',
    size=12, weight='normal', stretch='normal')
ax=plt.gca()

## Loading Data ##
df=pd.read_csv('D:\Python\edx\Machine Learning\china_gdp.csv')
with open('NonLinearReg.txt','a') as f:
    print(df.head(),file=f)
    print(df.describe(),file=f)

## Plotting the Dataset ##
x_data,y_data=(df['Year'].values,df['Value'].values)
plt.scatter(x_data,y_data,color='red')
plt.title('China GDP 1960-2014',fontname='Times New Roman',
          fontsize=12)
plt.xlabel('Year',fontname='Times New Roman',fontsize=12)
plt.ylabel('GDP (US Dollars)',fontname='Times New Roman',fontsize=12)
for label in ax.get_xticklabels():
    label.set_fontproperties(ticks_font)

for label in ax.get_yticklabels():
    label.set_fontproperties(ticks_font)
## Looking at the data plot, it resembles logistic graph ##

X_gdp=np.arange(-5,5,0.1)
Y_gdp=1/(1+np.exp(-X_gdp))

plt.figure()
plt.plot(X_gdp,Y_gdp,color='blue')
plt.title('Logistic Curve',fontname='Times New Roman',
          fontsize=12)
plt.ylabel('Dependent Variable',fontname='Times New Roman',fontsize=12)
plt.xlabel('Independent Variable',fontname='Times New Roman',fontsize=12)


## Function Sigmoid ##
```

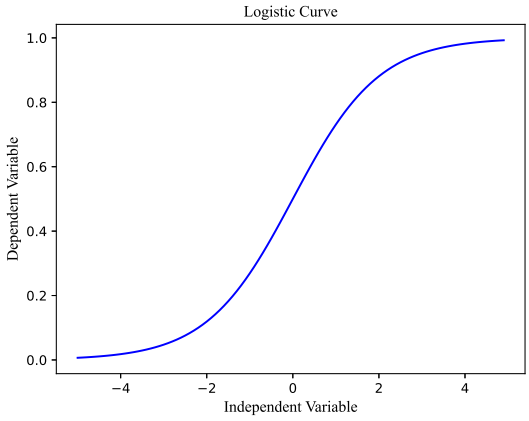
```
def sig(x,b1,b2):
    y=1/(1+np.exp(-b1*(x-b2)))
    return y

## Model building (Initial guess) ##

b1=0.1
b2=1990

y_prediction=sig(x_data,b1,b2)

fig, ax1 = plt.subplots()
plt.plot(x_data,y_prediction*1.5e13, label='Prediction')
plt.scatter(x_data,y_data,color='red', label='Actual Data')
plt.title('China GDP 1960-2014 Scatter and Prediction plot',fontname='Times New Roman',
          fontsize=12)
plt.xlabel('Year',fontname='Times New Roman',fontsize=12)
plt.ylabel('GDP (US Dollars)',fontname='Times New Roman',fontsize=12)
##leg = ax1.legend(loc='upper left', frameon=False)
leg=ax1.legend(fancybox=True, framealpha=1, shadow=True, borderpad=1)

## Optimized parameters for predicition ##
xdata=x_data/max(x_data)
ydata=y_data/max(y_data)

from scipy.optimize import curve_fit
popt,pcov=curve_fit(sig,xdata,ydata)
with open('NonLinearReg.txt','a') as f:
    print('beta1= %f, beta2= %f' %(popt[0],popt[1]),file=f)

## Using optimized parameters plot resulting regression model##

xopt=np.linspace(1960,2015,55)
xopt=xopt/max(xopt)
fig1, ax2 = plt.subplots()
yopt=sig(xopt,*popt)
plt.scatter(xdata*2015,ydata, label='Data')
plt.plot(xopt*2015,yopt,color='red', label='Fit')
plt.title('China GDP 1960-2014 Scatter and Prediction Optimized plot',fontname='Times Ne
          fontsize=12)
plt.xlabel('Year',fontname='Times New Roman',fontsize=12)
plt.ylabel('GDP (US Dollars)',fontname='Times New Roman',fontsize=12)
##leg = ax1.legend(loc='upper left', frameon=False)
leg2=ax2.legend(fancybox=True, framealpha=1, shadow=True, borderpad=1)
```

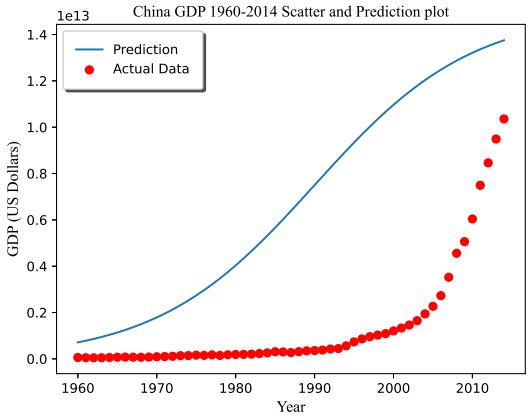
```
## Finding the accuracy of the model ##

#Split Test and Train Data #

msk=np.random.rand(len(df))<0.8
x_tr=xdata[msk]
x_ts=xdata[~msk]
y_tr=ydata[msk]
y_ts=ydata[~msk]

popt,pcov=curve_fit(sig,x_tr,y_tr) #Build model using training set

y_hat=sig(x_ts,*popt) #Predict using test set
from sklearn.metrics import r2_score
with open('NonLinearReg.txt','a') as f:
    print("Mean absolute error: %.2f" % np.mean(np.absolute(y_hat - y_ts)),file=f)
    print("Residual sum of squares (MSE): %.2f" % np.mean((y_hat- y_ts) ** 2),file=f)
    print("R2-score: %.2f" % r2_score(y_hat, y_ts),file=f )

## Display Plot
for label in ax.get_xticklabels():
    label.set_fontproperties(ticks_font)

for label in ax.get_yticklabels():
    label.set_fontproperties(ticks_font)
plt.show()
```

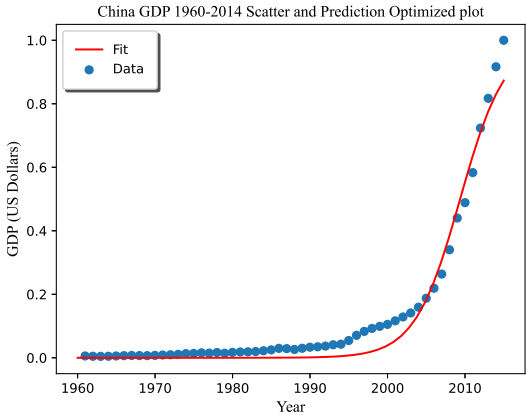
Solution:

```
   Year          Value
0  1960   5.918412e+10
1  1961   4.955705e+10
2  1962   4.668518e+10
3  1963   5.009730e+10
4  1964   5.906225e+10
              Year          Value
count     55.00000   5.500000e+01
mean    1987.00000   1.437042e+12
std       16.02082   2.500085e+12
min     1960.00000   4.668518e+10
25%     1973.50000   1.395123e+11
50%     1987.00000   3.074796e+11
75%     2000.50000   1.268748e+12
max     2014.00000   1.035483e+13
beta1= 690.451711, beta2= 0.997207
```










```
Mean absolute error: 0.02
Residual sum of squares (MSE): 0.00
R2-score: 0.95
```