import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

from sklearn.linear\_model import LinearRegression

# 解决图标题中文乱码问题

import matplotlib as mpl

mpl.rcParams['font.sans-serif'] = ['SimHei'] # 指定默认字体

mpl.rcParams['axes.unicode\_minus'] = False # 解决保存图像是负号'-'显示为方块的问题

df = pd.read\_excel('.\\数据.xlsx')

x = np.array(df['总人口数'])

y = np.array(df['新能源汽车保有量'])

from cmath import exp

import numpy as np

import matplotlib.pyplot as plt

from scipy import optimize as op

def regress\_exponential\_with\_offset(x, y):

# sort values

ind = np.argsort(x)

x = x[ind]

y = y[ind]

# decaying exponentials need special treatment

# since we can't take the log of negative numbers.

neg = -1 if y[0] > y[-1] else 1

dx = np.diff(x)

dy = np.diff(y)

dy\_dx = dy / dx

# filter any remaining negative numbers.

v = x[:-1]

u = neg \* dy\_dx

ind = np.where(u > 0)[0]

v = v[ind]

u = u[ind]

# perform regression

u = np.log(u)

s, t = np.polyfit(v, u, 1)

a = s

b = neg \* np.exp(t) / a

yy = np.exp(a \* x) \* b

c = np.median(y - yy)

return a, b, c

# 得到返回的A，B值

t = regress\_exponential\_with\_offset(x, y)

a = t[0]

b = t[1]

c = t[2]

# 数据点与原先的进行画图比较

plt.scatter(x, y, marker='o', label='real')

x = np.arange(13.2, 14.2, 0.077)

Y = b \* np.exp(a \* x) + c

plt.plot(x, Y, color='red', label='curve\_fit')

plt.xlabel('Total population/billion')

plt.ylabel('Total number of new energy vehicles / ten thousand')

plt.legend()

plt.show()

print('指数方程为：{} \* exp({} \* x) + {}'.format(b,a,c))

import numpy as np

from sklearn import metrics

# MSE

print('MSE:',metrics.mean\_squared\_error(y, Y))

# RMSE

print('RMSE:',np.sqrt(metrics.mean\_squared\_error(y, Y)))

# MAE

print('MAE:',metrics.mean\_absolute\_error(y, Y))