## 加载一些用到的工具包，其中matplotlib为画图包

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

import sklearn.linear\_model

## 这部分的代码是定义一个处理数据的方程，不需要详细阅读

def prepare\_country\_stats(oecd\_bli, gdp\_per\_capita):

oecd\_bli = oecd\_bli[oecd\_bli["INEQUALITY"]=="TOT"]

oecd\_bli = oecd\_bli.pivot(index="Country", columns="Indicator", values="Value")

gdp\_per\_capita.rename(columns={"2015": "GDP per capita"}, inplace=True)

gdp\_per\_capita.set\_index("Country", inplace=True)

full\_country\_stats = pd.merge(left=oecd\_bli, right=gdp\_per\_capita,

left\_index=True, right\_index=True)

full\_country\_stats.sort\_values(by="GDP per capita", inplace=True)

remove\_indices = [0, 1, 6, 8, 33, 34, 35]

keep\_indices = list(set(range(36)) - set(remove\_indices))

return full\_country\_stats[["GDP per capita", 'Life satisfaction']].iloc[keep\_indices]

## 读取GDP和生活满意度数据，需要修改路径

datapath = "D:/E drive/SZTU/introduction of machine learning/handson-ml-master/datasets/lifesat/"

oecd\_bli = pd.read\_csv(datapath + "oecd\_bli\_2015.csv", thousands=',')

gdp\_per\_capita = pd.read\_csv(datapath + "gdp\_per\_capita.csv",thousands=',',delimiter='\t',

encoding='latin1', na\_values="n/a")

## 用刚才定义好的方程来整理数据

country\_stats = prepare\_country\_stats(oecd\_bli, gdp\_per\_capita)

## 令x变量为GDP，Y变量为生活满意度

X = np.c\_[country\_stats["GDP per capita"]]

y = np.c\_[country\_stats["Life satisfaction"]]

##画出x和y的散点图

country\_stats.plot(kind='scatter', x="GDP per capita", y='Life satisfaction')

plt.show()

## 拟合一个线性回归模型

model = sklearn.linear\_model.LinearRegression()

##将模型用在数据上，去训练数据

model.fit(X, y)

## 求出截距t0和斜率t1

t0, t1 = model.intercept\_[0], model.coef\_[0][0]

## 画出散点图和线性模型图

country\_stats.plot(kind='scatter', x="GDP per capita", y='Life satisfaction', figsize=(5,3))

plt.axis([0, 60000, 0, 10])

X=np.linspace(0, 60000, 1000)

plt.plot(X, t0 + t1\*X, "b")

plt.text(5000, 3.1, r"$\theta\_0 = 4.85$", fontsize=14, color="b")

plt.text(5000, 2.2, r"$\theta\_1 = 4.91 \times 10^{-5}$", fontsize=14, color="b")

plt.show()

## 对塞浦路斯的数据进行预测

# Make a prediction for Cyprus

X\_new = [[22587]] # Cyprus' GDP per capita

print(model.predict(X\_new)) # outputs [[ 5.96242338]]