电影评分模型

摘要

本文主要研究各大平台的评分模型，根据各平台用户的评分数据，建立了基于数据分析的电影评分模型，分析模型并得出最合理的电影评分模型。

在问题一中，我们需要分析多家平台的评分模型。我们首先限定范围，将此问题的研究限于豆瓣、猫眼、IMDB 这三个平台。其次查阅资料获取豆瓣、猫眼、IMDB 这三家的电影评分模型，分析得出这三个平台对恶意刷分产生的影响能进行有效规避的是 IMDB 平台的贝叶斯统计模型，故我们团队认为此模型比较合理。一部电影的得分不仅考虑观众的评分，还得考虑到观众的数目，IMDB 平台模型下，评分人数超过 1250 则可以进入榜单，而我们则将其改为超过评论的平均数。

在问题二中，我们首先获取豆瓣和 IMDB 平台的 top50 的数据，发现差异不大，故将范围扩充到 top250。根据这些数据按国家、电影类型进行分类，画出相应的直方图，通过比较得出，影响两者排名不同的具体因素是文化差异、国家背景的不同、电影类型。

在问题三中，排除疫情因素的影响等，我们选取了两个影响电影票房的主要因素：投资和流行度，并假设票房只受这两个因素影响。首先通过可靠平台 kaggle 获取了相关数据，然后建立线性模型 $y=ax_1+bx_2+c$，其中 x_1 为投资变量，x_2 为流行度变量。为了解决运算过程数据爆炸问题，将投资和票房单位转成十万。建立平方损失函数，通过批量梯度下降算法，求得模型为 $y=2.7919x_1+22.6903x_2-4.6510$。通过该模型预测 Nolan 的《信条 Tenet》和娄烨的《兰心大剧院》的票房分别为 572330000 和 139590000。

在问题四中，我们对问题三中得到的票房预测模型，分别对 x_1 和 x_2 求导，得 $\frac{\partial y}{\partial x_1}=2.7919$、$\frac{\partial y}{\partial x_2}=22.6903$，故 x_2 对该模型的影响更大，即流行度是影响电影票房的最大因素。

关键词：电影票房预测、电影评分分析

一、问题重述

1.1 问题的提出

各大网站对电影都有不同的评分模型，电影评分对观众选择电影具有重要的参考意义。分析各网站的电影评分模型，得出最为合理的电影评分模型有很大的实际意义。

1.2 问题的重述

电影评分网站繁多，每个网站都有自己不同的评分模型和规则。有些电影为了提高评分，会雇佣水军进行恶意刷分，采用直接求平均分的模型会因此存在评分不合理的现象。故通过分析比较热门的网站：豆瓣、猫眼、IMDB，得出其电影评分模型，在此基础上分析计算出合理的电影评分模型并验证。然后基于此，分析豆瓣 top50 和 IMDBtop50 排名的影响因素，并通过先前得出的合理的模型，计算出该模型下的 top50 电影。在前面的基础上，建立电影票房预测模型，并且预

测 Nolan 的《信 条 Tenet》和娄烨的《兰心大剧院》的票房。分析影响电影票房的最重要的因素。

二、问题的分析

2.1 问题一的分析

题目要求分析多家评分网站，故我们选取了三家较为热门的评分网站：豆瓣、猫眼、IMDB。通过查找得到这三家网站各自的评分模型，并对这三个模型进行分析评价，讨论其对恶意刷分现象处理的有效性，得出较合理的模型。

2.2 问题二的分析

要分析影响豆瓣和 IMDB 两个平台 top50 电影的影响因素，故首先需要获取豆瓣和 IMDB 两个平台的 top50 的电影数据，然后将电影按类型、国内外电影等分类，画出直方图，比较两者的数值，得出相应的结论。考虑到两平台用户评分一般只限于这两个平台其中一个，很少会在两个平台都评价，故假设两个平台的数据没有交叉性，将豆瓣和 IMBD 和数据一同合并起来放进问题一中所得的合理模型进行评价，得出该模型下的 top50 电影。

2.3 问题三的分析

因为影响电影票房的因素很多，我们通过查阅资料分析发现影响票房的主要因素为投资、流行度。因此假设票房的影响因素为投资、流行度。通过相关权威平台收集相关的数据。选取线性模型 $y=ax_1+bx_2+c$，并随机初始化参数。建立平方损失函数$L=\frac{1}{10}\sum_{i=1}^{10}(y_i-\hat{y})^2$。通过批量梯度下降算法优化模型，即每次选取 10 个票房数据，计算损失函数 L，选取学习率 η，优化参数$\mathrm{a}=\mathrm{a}-\eta\frac{\partial L}{\partial a}$、$\mathrm{b}=\mathrm{b}-\eta\frac{\partial L}{\partial b}$、$\mathrm{c}=\mathrm{c}-\eta\frac{\partial L}{\partial c}$。循环优化 n 个周期，n 根据现有数据量确定。通过多次调整学习率 η 和循环周期数，确定最优模型，最终求得模型。再查找《信条 Tenet》和《兰心大剧院》的投资和流行度，带入计算得到预测票房值。

2.4 问题四的分析

由问题三中得到的票房预测模型，分析影响票房的最大因素。问题三中求得的票房预测模型为线性模型，故直接将其对变量求偏导，比较$\frac{\partial y}{\partial x_1}$和$\frac{\partial y}{\partial x_1}$的值，偏导数越大则对函数值的影响越大，故偏导值较大的则对票房影响最大。

三、模型假设

1.假设豆瓣和 IMDB 两平台的用户没有交叉性。
2.假设所获取的评分数据、评分人数具有真实性，没有平台操控因素。
3.假设电影票房的影响因素为投资、流行度。

四、模型的建立和求解

4.1 问题一求解

4.1.1 豆瓣的评分模型

评分对象：豆瓣注册老用户且用户账号为“非正常打分”账号。

规则：一人一票，五星制评分，最后转换成 10 分制。

算法：加权平均，权重为打分用户数的百分比。

假设一星至五星的人数分别为：A，B，C，D，E，则

豆瓣分=(A÷(A+B+C+D+E))×2+(B÷(A+B+C+D+E))×4+(C÷(A+B+C+D+E))×6+(D÷(A+B+C+D+E))×8+(E÷(A+B+C+D+E))×10

优点：算法简单易懂，易操作，直观。

缺点：水军恶意刷分，评分不真实；有观众根据心情评分，不具真实性。

4.1.2 猫眼评分模型

评分种类：点映评分：电影在正式上映前，观众购买点映场次电影票，并观看完电影后给出的评分；观众评分：电影正式上映后，观众购买电影票，并观看完后给出的评分；专业评分：由猫眼特邀的 81 位专业评委（截止 2017 年 2 月 16 日），在观影后对影片进行技术性评分。这些专业评委来自权威影评人、资深媒体人、电影学者组成。

规则：一人一票，评分 10 分制。

算法：加权平均。

优点：评分用户均为看过电影的用户；区分了专业和大众，有更多维度的参考；

缺点：水军恶意刷分，评分不真实；有观众根据心情评分，不具真实性。

4.1.3 IMDB 评分模型

评分对象：注册用户

规则：一人一票，10 分制。

算法：贝叶斯统计算法。

假设 R 为普通方法计算出的平局分，v 为投票人数，m 为进入 top250 的最小票数，C 为目前所有电影的平均分。

IMDB 分数=(v ÷ (v+m)) × R + (m ÷ (v+m)) × C

优点：IMDb 上的评分完全来自于网民的评价，凭的是参与评价的网民的自身喜好，参与评分的网民越多，IMDb 的评分越有可靠性。该算法复杂，可无效数据对整体的评分影响降低。

缺点：水军恶意刷分，评分不真实；有观众根据心情评分，不具真实性。

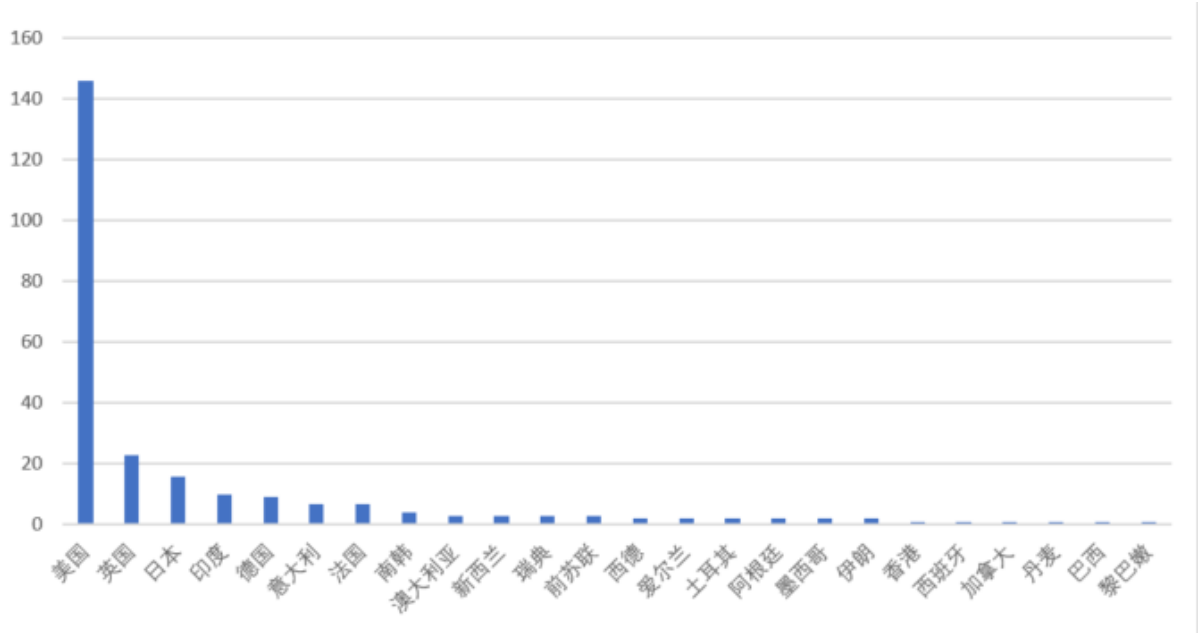
4.1.4 合理模型

综上可得，IMDB 评分模型可有效降低无效数据的干扰，较少恶意评分的影响，相较于其他两个更具权威性。但 IMDB 进入榜单的要求是评分人数超过 1250，我们为了充分考虑评分人数，将进入榜单的要求更改为超过评分人数的平均数。

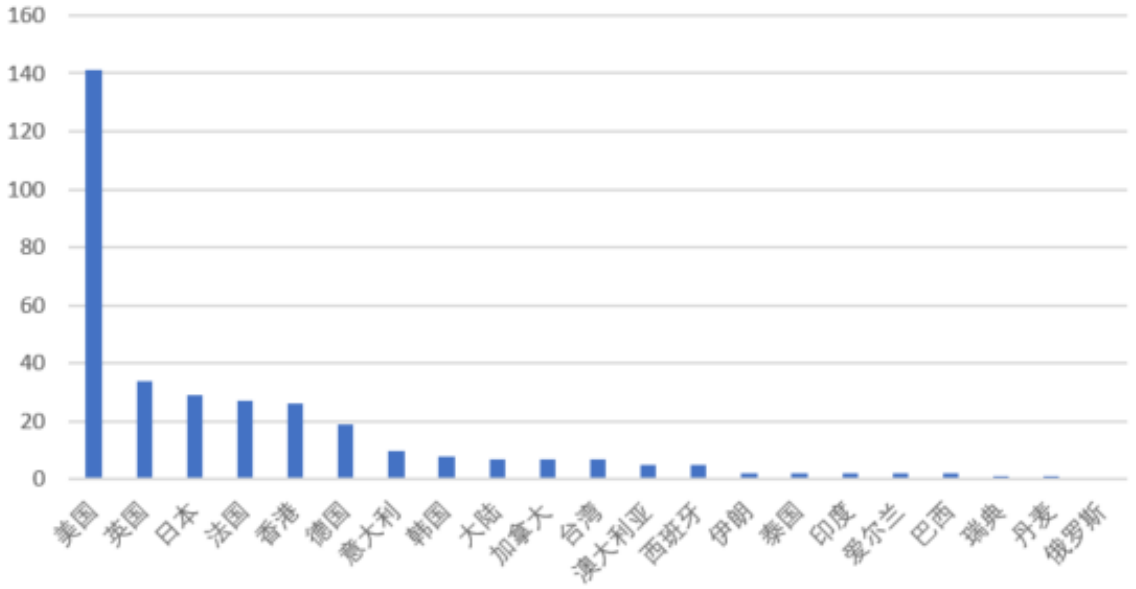
4.2 问题二求解

4.2.1 影响豆瓣和 IMDBTOP50 具体因素的分析

因为我们将 top50 的数据按电影类型、国家分类之后观察到的差异并不大，不利于判断，故我们将数据选取范围扩充到 top250。将数据整理得到相应的直方图，对比直方图进行分析。



IMDB 平台 top250 国家分类

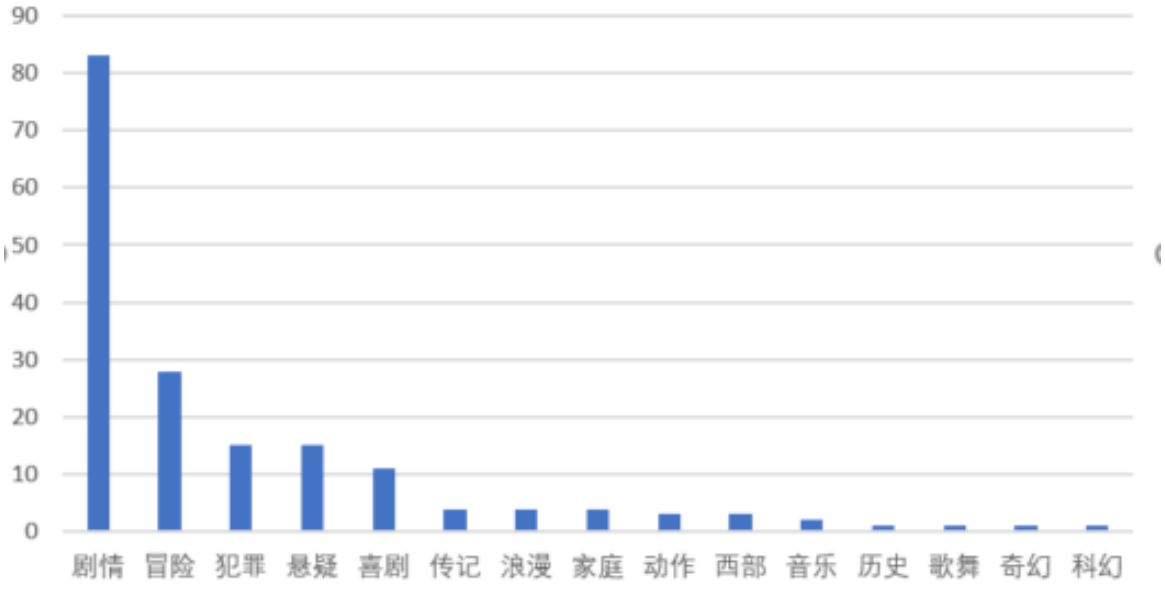


豆瓣平台 top250 按国家分类

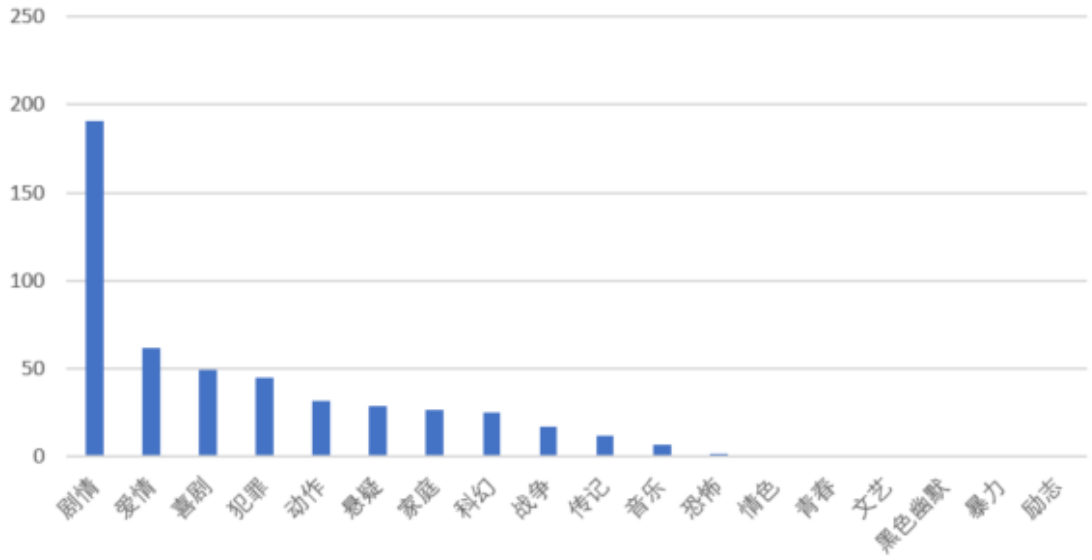
由按国家分类两个平台的直方图可知，两个平台用户都比较喜欢美国电影，IMDB 平台上大陆的电影几乎绝迹，与之相反，豆瓣 top50 中就至少包含了《霸

王别姬》、《大话西游之大圣娶亲》、《我不是药神这三部电影》。IMDB 用户主要为国外用户，而豆瓣用户主要为中国用户。可见，因为用户的差异、文化的因素、国家背景的不同，从而各平台的用户打分出现不同，导致两平台的 top50 榜单出现差异。

按电影类型分类直方图：



IMDB 平台 top250 按类型分类



豆瓣平台 top250 按类型分类

由按国家电影类型两个平台的直方图可知，两平台 top250 榜单的相同点：两平台的用户都比较喜欢剧情类型的电影；不同点：IMDB 平台的用户比较喜欢

冒险、犯罪、嫌疑类型的电影，而豆瓣平台爱情、喜剧、动作、家庭的电影得得分比较高，可见电影类型是影响两平台榜单的重要因素。

综上，影响豆瓣和 IMDB 排名不同的因素有文化差异、国家背景、电影类型。

4.2.2 问题一所得模型下的 TOP50 电影

为了增加评分的可靠性，故用尽可能多的数据得出该模型下的排名。考虑到两平台用户评分一般只限于这两个平台其中一个，很少会在两个平台都评价，故假设两个平台的数据没有交叉性，将豆瓣和 IMBD 和数据一同合并起来放进问题一中所得的合理模型进行评价，得出该模型下的 top50 电影如下：

1	肖申克的救赎
2	霸王别姬
3	阿甘正传
4	美丽人生
5	辛德勒的名单
6	这个杀手不太冷
7	泰坦尼克号
8	千与千寻
9	忠犬八公的故事
10	盗梦空间
11	海上钢琴师
12	楚门的世界
13	机器人总动员
14	放牛班的春天
15	星际穿越
16	熔炉
17	教父
18	三傻大闹宝莱坞
19	大话西游之大圣娶亲
20	疯狂动物城
21	无间道
22	龙猫
23	触不可及
24	蝙蝠侠: 黑暗骑士
25	活着
26	末代皇帝
27	指环王 3: 王者无敌
28	当幸福来敲门
29	怦然心动
30	寻梦环游记
31	何以为家
32	少年派的奇幻漂流
33	天空之城

34	哈尔的移动城堡
35	闻香识女人
36	摔跤吧！爸爸
37	飞屋环游记
38	大话西游之月光宝盒
39	罗马假日
40	哈利·波特与魔法石
41	我不是药神
42	搏击俱乐部
43	狮子王
44	猫鼠游戏
45	指环王 1：魔戒再现
46	美丽心灵
47	黑客帝国
48	本杰明·巴顿奇事
49	V 字仇杀队
50	西西里的美丽传说

4.3 问题三模型建立与求解

4.3.1 模型建立

设电影的投资为 x1，电影的流行度为 x2，票房为 y，建立模型：

$$y = ax_1 + bx_2 + c$$

假设电影的真实票房为$\hat{y}$,确定每次选取的数据量为 10，建立平方损失函数：

$$L = \frac{1}{10}\sum_{i=1}^{10}(y_i - \hat{y})^2$$

确定循环周期数为 50，设学习率 η，建立优化函数：

$$\mathrm{a} = \mathrm{a} - \eta\frac{\partial L}{\partial a}$$

$$\mathrm{b} = \mathrm{b} - \eta\frac{\partial L}{\partial b}$$

$$\mathrm{c} = \mathrm{c} - \eta\frac{\partial L}{\partial c}$$

为了防止运算过程数据爆炸，将投资和票房单位转成十万，学习率选取为 0.000003。

4.3.2 模型求解

将模型 a、b 按期望为 0，标准差为 0.01 的正态分布随机初始化，c 初始化为 0。 η=0.000003。求解步骤如下：

1．随机选取 10 个数据，分别计算函数值 y。

2．求解平方损失函数值 L。

3. 计算优化函数，更新模型参数。
4. 返回第一步。照此执行 50 个周期。
5. 衡量该模型下模型的合理性，否则调整学习率 η，重复步骤 1 至 4。

通过 python 求解 a=2.7919，b=22.6903，c=-4.6510，故模型为：

$$y=2.7919x+22.6903x-4.6510$$

《信条 Tenet》投资约为：205,000,000、流行度约为：5.5；《兰心大剧院》投资约为：50000000、流行度约为：9.0。将投资单位转成 10 万，再带入计算得到的票房（单位：十万），再将单位转成 1。预测票房如下：

《信条 Tenet》：572330000

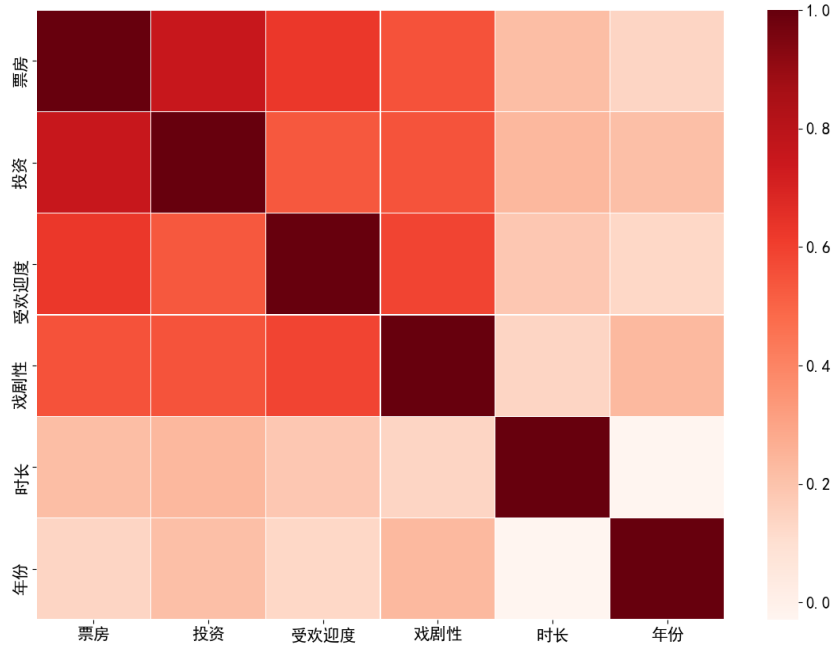
《兰心大剧院》：139590000

4.4 问题四求解

4.4.1 问题求解

问题三求得模型为：y=2.7919x+22.6903x-4.6510，分别计算对 x_1 和 x_2 的偏导：$\frac{\partial y}{\partial x_1}=2.7919$、$\frac{\partial y}{\partial x_2}=22.6903$。因为$\frac{\partial y}{\partial x_1}<\frac{\partial y}{\partial x_2}$，所以 x_2 对函数值 y 的影响更大。故影响票房的最大因素为流行度。

4.4.2 模型验证



电影各影响因素的对比热图

上图是通过数据统计得出的电影各影响因素的对比热图。观察此图可得，影响票房的最大因素为投资。

五、模型评价与改进

5.1 模型优点

1. 考虑了影响电影票房的主要因素投资和流行度，并且给出了具体的计算方法。
2. 采用了小批量梯度下降算法，减少了单次运算量，充分利用数据，模型拟合度较高。
3. 可以通过调整学习率，从而对模型进行调整。

5.2 模型缺点

1. 模型考虑因素较少，只考虑了投资和流行度，具有局限性。
2. 电影投资数据难以获取，投资数据也不够准确。
3. 该模型下，用现有的数据进行测试，误差较大，不够精确。
4. 模型与各变量都是简单的线性关系，难以更好的拟合数据。

5.3 模型改进

1. 考虑影响电影的更多因素，比如出品方、上映时间等。
2. 更改模型，使各变量与模型不只是简单的线性关系，提高拟合度。

参考文献

[1] 姜启源,谢金星.数学模型[J].北京:高等教育出版社,2011.
[2] 阿斯顿·章,李牧.动手学深度学习.人民邮电出版社,2019.

附录 A 问题三求解 Python 代码

```
import torch
import pandas as pd
import torch.utils.data as Data
import torch.nn as nn
from torch.nn import init
import torch.optim as optim

num_inputs = 2   #特征数个数

#准备数据
train_data = pd.read_csv("data/prediction/train.csv")
train_data.fillna(value=0)

features = torch.tensor(train_data[['budget','popularity']].values, dtype=torch.float64)
labels = torch.tensor(train_data['revenue'].values/1000000,dtype=torch.float64)
features[:,0] = features[:,0]/ 1000000

#  读取数据
batch_size = 10
dataset = Data.TensorDataset(features, labels) #  将训练数据的特征和标签组合
data_iter = Data.DataLoader(dataset, batch_size, shuffle=True) #  随机读取小批量

#  定义模型
class LinearNet(nn.Module):
    def __init__(self, n_feature):
        super(LinearNet, self).__init__()
        self.linear = nn.Linear(n_feature, 1)

    def forward(self, x):
        y = self.linear(x)
        return y

net = LinearNet(num_inputs).double()

#  初始化模型参数
init.normal_(net.linear.weight, mean=0, std=0.01)
init.constant_(net.linear.bias, val=0)   #  也可以直接修改 bias 的 data: net[0].bias.data.fill_(0)

#  定义损失函数
loss = nn.MSELoss()
```

```
# 定义优化算法
optimizer = optim.SGD(net.parameters(), lr=0.00003)

# 调整学习率
for param_group in optimizer.param_groups:
    param_group['lr'] *= 0.1 # 学习率为之前的0.1倍

# 训练模型
num_epochs = 50
for epoch in range(1, num_epochs + 1):
    for X, y in data_iter:
        output = net(X)
        l = loss(output, y.view(-1, 1))
        # print(l)
        # print(optimizer)
        l.backward()
        optimizer.step()
        optimizer.zero_grad()  # 梯度清零，等价于net.zero_grad()
    print('epoch %d, loss: %f' % (epoch, l.item()))

dense = net.linear
print(dense.weight)
print(dense.bias)

r_w = dense.weight.reshape([2,1])
#预测tenet 票房：投资：205,000,000；流行度：5.5
X = torch.tensor([[2050,5.5]],dtype=torch.float64)
revenue = torch.mm(X,r_w) + dense.bias
print("tenet 票房：")
print(revenue * 100000)

#预测兰心大剧院票房：投资：250，000，000；流行度：9.0
X = torch.tensor([[500,9.0]],dtype=torch.float64)
revenue = torch.mm(X,r_w) + dense.bias
print("《兰心大剧院》票房：")
print(revenue * 100000)
```