**《从零开始学Python数据分析与挖掘》—纠错**

**P57**：代码中的注释有误，第二段代码的最后一个注释，需改成下方：

# 第2至4行，2至5列

print(arr2[1:4,1:5])

**P58**：代码中的注释有误，第二段代码的最后一个注释，需改成下方：

# 第一行、最后一行和第二列、第三列、第四列构成的数组

print(arr2[np.ix\_([0,-1],[1,2,3])])

**P78**：5.1.2节的第一段中的最后一句存在错别字，需改成下方：

构造一个数据框可以应用如下方式：

**P80**：函数的参数解释中存在错别字（即解释中的倒数第二个参数）

**comment**：指定注释符，在读取数据时，如果碰到行首指定的注释符，则跳过该行；

P82:介绍pd.read\_excel()的参数列表中没有converters这一参数，需要在函数中的末尾加上converters参数；

P82:converters = {0:str}后面缺失英文逗号（,）；

~~P84:倒数第四行（连接桥梁（conn））~~

P91:正文的第12行，“两个问题的解决所使用的方法分布”（将分布改成分别）

P91：正文倒数第10行，“其思路就是按照邮箱中的@符风格”（将风格改成分割）

P94:最后一行，“该数据集存在4（改成5）条缺失观测，行号分别是4、5、7和9（改成2、4、5、7和9）

P95:第二段中“检测数据集是否存在重复观测（改成缺失观测）使用的是isnull方法”

P141:第11小点，“设值（改成置）水平值顺序”

**P154**：第二个公式有误，需改成下方：

~~P154:由于上式中的y’x和‘x’y均为常数，并且常数的转置就是其本身，因此y’x和‘x’y是相等的，感觉这句话有问题，二者不相等~~

**P155**：7.7.2节的第一段中公式有误，需改成下方：

P157：第二行删除State变量和California（改为New York）变量

~~P163:最后一行，pp图的思想是比对（对比）~~

**P165**：第一个公式有误，需改成下方：

**P167**：代码内一行有误，需改成下方：

# 模型修正

model3 = sm.formula.ols ('Profit ~ RD\_Spend + Marketing\_Spend', data = train).fit()

**P169**：第二段代码的一行有误，需改成下方：

# 计算异常值数量的比例

outliers\_ratio = sum(np.where((np.abs(profit\_outliers.resid\_stu)>2),1,0))/ profit\_outliers.shape[0]

**P169**：第三段代码有误，需改成下方：

model4 = sm.formula.ols ('Profit ~ RD\_Spend + Marketing\_Spend', data = none\_outliers).fit()

p175:不妨第三列是第二（改成一）列的两倍

p178:第一行，，由于折现（改成线）图中涉及回归模型系数

~~p179:第一个图的xlable为log（Lambda）~~

**P191**：最后一段内容，描述有误（即第二行），将其中一句话改成下方：

相反，当趋于负无穷大时，会趋于正无穷大，最终导致逼近于0；

**P199**：一整段代码包含几个小错误，需要整体替换为如下语句：

# 自定义绘制ks曲线的函数

def plot\_ks(y\_test, y\_score, positive\_flag):

# 对y\_test,y\_score重新设置索引

y\_test.index = np.arange(len(y\_test))

#y\_score.index = np.arange(len(y\_score))

# 构建目标数据集

target\_data = pd.DataFrame({'y\_test':y\_test, 'y\_score':y\_score})

# 按y\_score降序排列

target\_data.sort\_values(by = 'y\_score', ascending = False, inplace = True)

# 自定义分位点

cuts = np.arange(0.1,1,0.1)

# 计算各分位点对应的Score值

index = len(target\_data.y\_score)\*cuts

scores = target\_data.y\_score.iloc[index.astype('int')]

# 根据不同的Score值，计算Sensitivity和Specificity

Sensitivity = []

Specificity = []

for score in scores:

# 正例覆盖样本数量与实际正例样本量

positive\_recall = target\_data.loc[(target\_data.y\_test == positive\_flag) & (target\_data.y\_score>score),:].shape[0]

positive = sum(target\_data.y\_test == positive\_flag)

# 负例覆盖样本数量与实际负例样本量

negative\_recall = target\_data.loc[(target\_data.y\_test != positive\_flag) & (target\_data.y\_score<=score),:].shape[0]

negative = sum(target\_data.y\_test != positive\_flag)

Sensitivity.append(positive\_recall/positive)

Specificity.append(negative\_recall/negative)

# 构建绘图数据

plot\_data = pd.DataFrame({'cuts':cuts,'y1':1-np.array(Specificity),'y2':np.array(Sensitivity),

'ks':np.array(Sensitivity)-(1-np.array(Specificity))})

# 寻找Sensitivity和1-Specificity之差的最大值索引

max\_ks\_index = np.argmax(plot\_data.ks)

plt.plot([0]+cuts.tolist()+[1], [0]+plot\_data.y1.tolist()+[1], label = '1-Specificity')

plt.plot([0]+cuts.tolist()+[1], [0]+plot\_data.y2.tolist()+[1], label = 'Sensitivity')

# 添加参考线

plt.vlines(plot\_data.cuts[max\_ks\_index], ymin = plot\_data.y1[max\_ks\_index],

ymax = plot\_data.y2[max\_ks\_index], linestyles = '--')

# 添加文本信息

plt.text(x = plot\_data.cuts[max\_ks\_index]+0.01,

y = plot\_data.y1[max\_ks\_index]+plot\_data.ks[max\_ks\_index]/2,

s = 'KS= %.2f' %plot\_data.ks[max\_ks\_index])

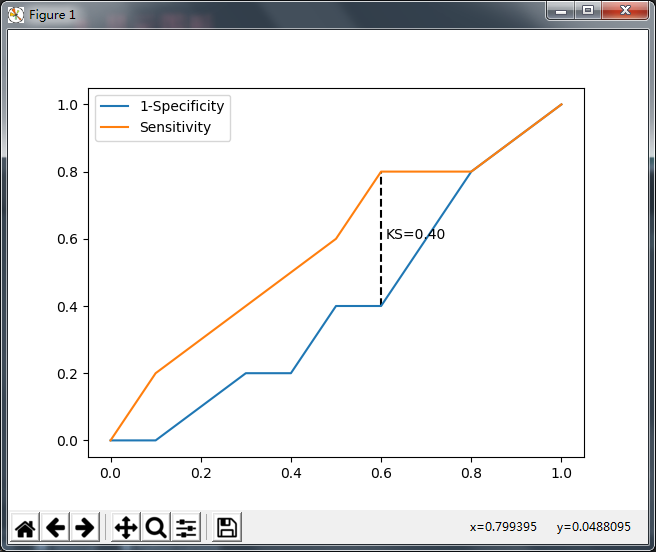
# 显示图例

plt.legend()

# 显示图形

plt.show()

p200:图94的k-s图，与上面的代码所绘制的不一样



**P208**：表格中存在错别字，需要将所有的“不够买”改成“不购买”

P214:第四行，解释C4.5(改成CART)

**P225**：第二段代码中的第一行参数传递错误，需将min\_samples\_leaf参数改成：

min\_samples\_leaf=4

p227:第二行，，根节点所选的变量为Sex\_male(改成Sex\_female)

**P333**：公式有误，需改成下方：

~~P335:第18行代码需要缩进~~