口红销量预测

周 伟

（北京邮电大学数字媒体与设计艺术学院，数字媒体技术，2017110949）

摘要：从网站爬取口红销售数据，分析影响销售数据的重要因素以及根据销售因素建模预测其销售量。本文先将数据进行预处理得到实验数据，然后着重分析朴素贝叶斯判别分析算法、AdaBoost 算法以及随机森林算法在口红销量预测中的效果，并在随机森林算法中进行模型优化。通过实验结果表明总评价数、价格和描述分这三个因素对销售量的影响较大，对三个算法对比分析得出随机森林算法预测错误率最低，有较好的预测效果。

关键词： 口红销量；朴素贝叶斯；AdaBoost；随机森林

1 问题描述

美容界曾有过一个著名的调查:“假如你只能拥有一个化妆品，你想要的是什么产品？”95%以上的亚洲女性都选择了口红。口红是女性必备的美容化妆品之一，包括唇膏、唇棒、唇彩、唇釉等，能让唇部红润有光泽，达到滋润、保护嘴唇，增加面部美感及修正嘴唇轮廓的作用。口红的基本成分离不开凡士林和蜡质，不过现在也有不含蜡质的新配方，而维生素 A，E 等抗氧化成分，以及 SPF 防晒功能，都是时下热门唇膏的卖点。

为了让商家生产出更符合消费者的口红，提高其销售量，同时为了消费者能够买到更理想的口红，本文在京东网站上爬取 1600 多条销售数据。在这诸多影响口红销量的影响因素中，分析哪些因素对口红销量至关重要，根据这些因素预测商家的销售量。

2 数据描述

2.1 数据来源

从京东网站爬取口红销量数据 1695 条，如图 1 所示，一共还有 18 个字段。

图 1 京东网站口红销量片段

2.2 数据预处理

2. 2. 1 数据清洗

通过对图 1 分析发现有如下一些特点：

① 价格有“￥”，需要处理，字段存在缺失值；

② 字段“适合肤质”、“保质期”、“规格类型”和“适合人群”差异性不大，可以删去；

③ 销量的差异性大，为连续值，可划分等级处理；

本文对价格的做法是去掉“￥”，对于带有“～”的价格取均值,如图 2 所示。

17	¥45.08	17	45.08
18	¥26.82	18	26.82
19	¥19.00	19	19.00
20	¥19.96	20	19.96
21	¥17.50	21	17.50
22	¥19.80~¥22.00	22	20.9
23	¥55.00	23	55.00
24	¥18.90	24	18.90
25	¥19.00	25	19.00
26	¥19.90	26	19.90
27	¥29.79	27	29.79
28	¥45.00	28	45.00
29	¥256.36	29	256.36

图 2 价格平均处理图

同时，将文本数据转换格式，文本数字 character 转换为 numeric，文本字符 character 转换为 factor，同时删去字段“适合肤质”、“保质期”、“规格类型”和“适合人群”。

2.2.2 缺失值处理

加载 mice 包，调用 md.pattern()函数，可以查看缺失值，如下图 3 所示。

	row.names	店名	描述分	价格分	质量分	服务分	价格	总评价数	总销量	国家	防晒	是否进口	颜色	功效	
1	1540	1	1	1	1	1	1	1	1	1	1	1	1	1	0
2	41	1	1	1	1	1	1	1	1	1	1	1	0	1	1
3	89	1	1	1	1	1	1	1	1	1	1	1	1	0	1
4	1	1	1	1	1	1	1	1	1	1	1	1	0	0	2
5	3	1	1	1	1	1	1	1	1	1	0	0	0	1	2
6	21	1	1	1	1	1	1	1	1	0	0	0	1	0	4
7		0	0	0	0	0	0	0	0	21	24	24	42	111	222

图 3 数据缺失值详情

通过观看，可以发现除了店名，其余非数值的字段，都存在缺失的现象，记录共缺失 155 行，其中字段“功效”缺失的最多。为便于下文模型的分析，将缺失的字段全部删除，剩余非空数据的行数为 1540 行。

2.3 颜色字段词云

对字段“颜色”进行查看，可以看出颜色种类繁多，查看数据难以统计最受欢迎的颜色有哪几种，可通过分词工具进行分词处理，然后通过词云可形象化展示出来，从而在生产的时候可着重配齐受欢迎颜色的种类，“颜色”原始数据如图 4 所示。

1	Candy Yum Yum Girl About Town Saint Germain Lady Danger Ruby Woo Freckletone CHILI小辣椒
2	44＃歌剧名伶 37＃纵情 93＃兴奋 99＃海盗 96＃古灵精怪 94＃着迷 91＃吸引 90＃活泼 43＃亲爱 136＃悠扬 138＃激昂 102＃惬纺 42＃星眼 46＃ 152＃隐约 ＃154 ＃50
3	801樱花粉 802桃粉色 803橘色 804裸色 805经典红 806慕斯红 807高贵红 808玫紫红 809甜脂红
4	车厘子红 魅惑大红 浓橙红色 气质裸粉 心动玫粉 清新暖橘
5	蜜桃粉 蔷薇粉 慕斯红 珊瑚红
6	ZR08＃刘雯豆沙色 ZR07＃YSL限量梅子色 ZR04＃YSL大热新男色 SR07＃复古红色 SR08＃经典豆沙色 SR02＃大热姨妈色 SR06＃百搭玫瑰粉 SR05＃尹恩惠想你色 06＃宿醉紫红色（又名秀智色） ZR02
7	03#橘红色 04#玫瑰红 05#复古大红 08#偏粉玫红 09#偏红玫红 11#宿醉紫红 18#裸粉豆沙 19#大衣姐豆沙红 20#玫瑰粉 22#橘粉调豆沙色 24#玫瑰灰紫 27#紫调姨妈红 28#姨妈红 29#得鲜M06阿色
8	阳春白雪 夕阳赛欧 梅花三弄
9	MGH6932401397653 CCH6932401397653 MCJ6932401397653 FGH6932401397653 SMJ6932401397653 BBF6932401397653 XGH6932401397653 NHF6932401397653 SHF6932401397653 TGF6932
10	新版815露衣草色 新版806慕斯红 新版816复古红 新版811粉紫色 新版803橘色 新版805经典红 新版807高贵红 新版813玫瑰红 新版801樱花粉 新版810浪漫粉 新版802桃粉色 新版809甜脂红 新版804裸色
11	01#复古大红 02#橘红色 03#淡裸粉色 04#深红浆果色 06#甜蜜水粉色
12	升级版RD301 真我红 升级版RD302 吸血鬼 升级版PK001 梅子红 升级版PK002 李子红 升级版OR202 橘子红 升级版OR203 西柚红 升级版OR204 樱桃红 升级版OR201 金橙红 RD303 辣椒酱红 BR401 无
13	01＃ 02＃ 03＃ 04＃ 05＃ 06＃ 07＃ 08＃
14	白金 黑盒
15	01烈焰红 02茉莉粉 04芭比粉 06金珊瑚粉 07金玛莎红
16	1＃ 樱桃红 2＃ 苹果红 3＃ 橘子红
17	01＃ 02＃ 03＃ 04＃ 05＃ 06＃ 07＃ 08＃ 09＃ 10＃
18	1＃樱桃红（哑光） 2＃想你色（哑光） 3＃樱花粉（哑光） 4＃橘红色（哑光） 5＃复古红（哑光） 6＃豆沙色（哑光）
19	01＃姨妈色 02＃桃运粉 03＃美颜橘 04＃性感玫红 05＃樱花粉 06＃大红色 新款＃01温存（酒红色） 新款＃02想你（草莓红） 新款＃03情愫（樱桃红） 新款＃04暮婚（橘子红） 新款＃05余温（玫瑰

图 4 颜色原始数据

	seg_color	Freq
1	粉	763
2	红	745
3	1	672
4	2	587
5	3	482
6	4	392
7	01	380
8	02	368
9	PK0	331
10	5	321
11	03	301
12	6	283
13	珊瑚	267
14	RD0	263
15	04	253

图 5 销量最好的 15 种颜色及颜色词云

2.4 功效字段词云

对字段“功效”进行查看，可以看出功效的类型多，查看数据难以统计最受欢迎的功效有哪几种，可通过分词工具进行分词处理，然后通过词云可形象化展示出来，从而在生产或购买的时候，着重注意前几名的功效，“功效”原始数据如图 6 所示。

1	易上色,易卸妆,保湿,滋润
2	易上色,易卸妆,保湿,滋润
3	易上色
4	不沾杯,易上色,防脱色,防水锁色，不脱妆,丝绒雾面哑光
5	补水,防脱色,易上色,易卸妆,持久
6	提亮肤色,易上色
7	遮瑕,美白,易卸妆,滋润,持久
8	持久,咬唇妆,双色口红,不掉色
9	补水,保湿,滋润,防脱色,持久
10	防脱妆
11	防脱色,补水,均匀肤色,滋润,温和卸妆
12	保湿,滋润,防脱色,易上色,易卸妆
13	保湿,滋润,提亮肤色,易上色,持久
14	保湿,滋润,持久,不脱色,不掉色
15	易上色,防脱色,保湿,滋润,持久
16	保湿,防脱色,易上色,易卸妆,持久
17	易上色,防脱色,哑光裸色,保湿,滋润
18	滋润,防脱色,易上色,易卸妆,持久
19	防脱色,易上色,持久,水润娇唇
20	滋润,防晒,防脱色,易上色,持久

图 6 功效原始数据

	effect_seg	Freq
1	易上色	1056
2	滋润	963
3	持久	952
4	保湿	742
5	防脱色	626
6	易卸妆	385
7	补水	196
8	提亮肤色	137
9	温和卸妆	88
10	其他功效	57
11	均匀肤色	44
12	口红	35
13	不掉色	34
14	咬唇妆	34
15	不脱色	29
16	防水	26
17	唇膏	19
18	不沾杯	18
19	唇纹	17
20	淡化	17

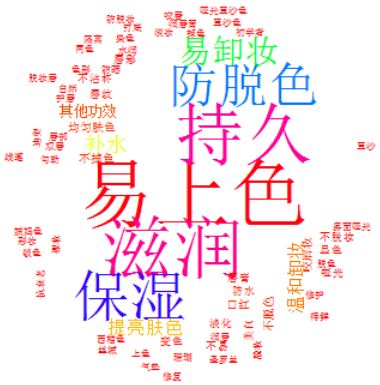


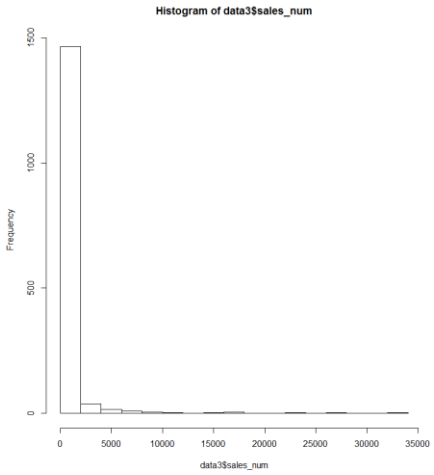
图 7 功效前 20 种的类型及功效词云

2.5 数据变换

2.5.1 对销售总量字段进行处理

通过绘制销售总量的统计直方图，可以发现分布不均匀，为了便于分析，将销售总量按

量分段，分成 A、B 和 C 等 3 个标准段，即销量的高、中和低三个层次。划分后，销售总量分类如表 1 所示。



1	32192
2	27027
3	27027
4	23496
5	17747
6	17270
7	17270
8	15523
9	15523
10	11992
11	10799
12	9923
13	9203
14	9203
15	8757
16	8757
17	7109
18	6905
19	6905
20	6902

图 8 销售总量统计直方图及销售总量前 20 图

表 1 销售总量等级划分

A	B	C
467	566	507

2.5.2 对功效字段进行处理

由于功效字段是文本，为了便于后文模型的处理，对其字段进行分词处理，并按照功效的重要性，进行权值计算。

1	易上色,易卸妆,保湿,滋润
2	易上色,易卸妆,保湿,滋润
3	易上色
4	不沾杯,易上色,防脱色,防水锁色，不脱妆,丝绒雾面哑光
5	补水,防脱色,易上色,易卸妆,持久
6	提亮肤色,易上色
7	遮瑕,美白,易卸妆,滋润,持久
8	持久,咬唇妆,双色口红,不掉色
9	补水,保湿,滋润,防脱色,持久
10	防脱妆
11	防脱色,补水,均匀肤色,滋润,温和卸妆
12	保湿,滋润,防脱色,易上色,易卸妆
13	保湿,滋润,提亮肤色,易上色,持久
14	保湿,滋润,持久,不脱色,不掉色
15	易上色,防脱色,保湿,滋润,持久
16	保湿,防脱色,易上色,易卸妆,持久
17	易上色,防脱色,哑光裸色,保湿,滋润
18	滋润,防脱色,易上色,易卸妆,持久
19	防脱色,易上色,持久,水润娇唇
20	滋润,防晒,防脱色,易上色,持久

	effect_seg	Freq
1	易上色	1056
2	滋润	963
3	持久	952
4	保湿	742
5	防脱色	626
6	易卸妆	385
7	补水	196
8	提亮肤色	137
9	温和卸妆	88
10	其他功效	57
11	均匀肤色	44
12	口红	35
13	不掉色	34
14	咬唇妆	34
15	不脱色	29
16	防水	26
17	唇膏	19
18	不沾杯	18
19	唇纹	17
20	淡化	17

1	18
2	18
3	4
4	17
5	21
6	9
7	13
8	11
9	23
10	0
11	23
12	22
13	23
14	20
15	22
16	21
17	20
18	21
19	12
20	20

图 9　功效字段原始数据、功效前 20 图、功效加权前 20 图

表 2 功效权值表

易上色	滋润	持久	保湿	防脱色	易卸妆	补水	温和卸妆	均匀肤色
4	5	4	5	4	4	5	5	4
不掉色	防水	水润	哑光	不沾杯	咬唇妆	不脱色	提亮肤色	其他功能
3	4	5	2	3	4	3	5	1

所采用的权值为 5 分制，按照功能的重要性进行标注权值分数。通过加权之后，对应的功能权重如图 9 所示，功效越多，相应的权值越重要。

2.5.3 对数值字段进行处理

由于数值各字段大小差异性较大，故采用“最大最小值”的方法进行数据归一化处理。

处理后的数据，如下图 10 所示。

	name	describe_score	price_score	quality_score	service_score	price	evaluate_num	sales_num	effect	sunScreen	country	imported
1	碧黛美妆专营店	0.8867925	0.8070175	0.8813559	0.71212121	0.148428390	0.0118206403	A	0.38297872	是	加拿大	是
2	碧黛美妆专营店	0.8867925	0.8070175	0.8813559	0.71212121	0.258589968	0.0089200035	A	0.38297872	否	法国	是
3	韩熙贞官方旗舰店	0.7547170	0.6315789	0.7288136	0.60606061	0.031442287	0.6896100497	A	0.08510638	否	韩国	否
4	ILISYA化妆品旗舰店	0.8113208	0.7017544	0.7796610	0.66666667	0.061546926	0.0546759138	A	0.36170213	否	中国内地	否
5	黄婉婷的美妆店	0.6603774	0.5263158	0.6271186	0.53030303	0.031745150	0.0000000000	C	0.44680851	否	中国内地	否
6	安妮珍选	0.6226415	0.5964912	0.5762712	0.51515152	0.010645655	0.0330628980	A	0.19148936	否	中国内地	否
7	尚佳人	0.6981132	0.6315789	0.6949153	0.57575758	0.011816727	0.0849472215	A	0.27659574	是	中国内地	否
8	美康粉黛旗舰店	0.6415094	0.5438596	0.6271186	0.54545455	0.031038469	0.0836822821	A	0.23404255	否	中国内地	否
9	伊诗兰顿官方旗舰店	0.7358491	0.6491228	0.6949153	0.60606061	0.011574436	0.0116461659	A	0.48936170	否	中国内地	否
10	韩熙贞官方旗舰店	0.7547170	0.6315789	0.7288136	0.60606061	0.030553887	0.6204527611	A	0.00000000	否	韩国	否
11	尚佳人	0.6981132	0.6315789	0.6949153	0.57575758	0.019731562	0.0009814185	A	0.48936170	否	中国内地	否
12	蜜糖美妆	0.8301887	0.7368421	0.8135593	0.69696970	0.019731562	0.3793727645	A	0.46808511	否	韩国	是
13	黛莉美妆	0.8490566	0.7719298	0.8135593	0.69696970	0.009535155	0.0187559976	A	0.48936170	否	中国内地	否
14	伊诗兰顿官方旗舰店	0.7358491	0.6491228	0.6949153	0.60606061	0.011574436	0.0870845328	A	0.42553191	否	中国内地	否
15	薇雅美妆	0.6415094	0.5789474	0.6101695	0.54545455	0.015693381	0.0288537032	A	0.46808511	否	中国内地	否
16	蜜糖美妆	0.8301887	0.7368421	0.8135593	0.69696970	0.011140332	0.0209369275	A	0.44680851	是	韩国	是
17	棉花糖美妆	0.7735849	0.6491228	0.7288136	0.63636364	0.036974595	0.0520806072	A	0.42553191	否	韩国	是
18	古采堂美妆店	0.6792453	0.5789474	0.6440678	0.57575758	0.018540299	0.3511515310	A	0.44680851	否	中国内地	否

Showing 1 to 19 of 1,540 entries

图 10 数值字段数据归一化处理

3 模型构建

3.1 数据抽样处理

由于模型的因变量“销售总量”已分段处理，故在数据抽样的时候，将按照分层抽样的规则进行，并对训练集和测试集按照 3:1 的原则进行。调用包 sampling 的 strata()函数进行分层抽样。其中，训练集 1154 行，测试集 386 行。

3.2 朴素贝叶斯分类

3.2.1 朴素贝叶斯算法原理

设每个数据样本用一个 n 维特征向量来描述 n 个属性的值，即：$\mathrm{X}=\{\mathrm{x}_1,\mathrm{x}_2,\cdots,\mathrm{x}_\mathrm{n}\}$，假定有 m 个类，分别用 $\mathrm{C}_1,\mathrm{C}_2,\cdots,\mathrm{C}_m$ 表示。给定一个未知的数据样本 X（即没有类标号），若朴素贝叶斯分类法将未知的样本 X 分配给类 C_i，则一定是

$$\mathrm{P}(\mathrm{C}_i\mid\mathrm{X})>\mathrm{P}(\mathrm{C}_j\mid\mathrm{X}) \qquad 1\le \mathrm{j}\le \mathrm{m},\ \mathrm{j}\ne \mathrm{i} \tag{1}$$

根据贝叶斯定理，由于 P(X)对于所有类为常数，最大化后验概率 $\mathrm{P}(\mathrm{C}_i\mid\mathrm{X})$ 可转化为最大化先验概率 $\mathrm{P}(\mathrm{X}\mid\mathrm{C}_i)\mathrm{P}(\mathrm{C}_i)$。如果训练数据集有许多属性和元组，计算 $\mathrm{P}(\mathrm{X}\mid\mathrm{C}_i)$ 的开销可能非常大，为此，通常假设各属性的取值互相独立，这样先验概率 $\mathrm{P}(\mathrm{X}_1\mid\mathrm{C}_i)\mathrm{P}(\mathrm{C}_i)$，$\mathrm{P}(\mathrm{X}_2\mid\mathrm{C}_i)\mathrm{P}(\mathrm{C}_i),\cdots,\mathrm{P}(\mathrm{X}_n\mid\mathrm{C}_i)\mathrm{P}(\mathrm{C}_i)$ 可以从训练数据集求得[1]。

根据此方法，对一个未知类别的样本 x，可以先分别计算出 X 属于每一个类别 C_i 的概率 $\mathrm{P}(\mathrm{X}\mid\mathrm{C}_i)\mathrm{P}(\mathrm{C}_i)$，然后选择其中概率最大的类别作为其类别。朴素贝叶斯算法成立的前提是各属性之间互相独立。当数据集满足这种独立性假设时，分类的准确度较高，否则可能较低。另外，该算法没有分类规则输出。

3.2.2 算法实现

```
library(klaR)
#  模型训练
#  对样本进行预处理，去掉  店名 name
data6<-data3[Train_data_num,-1]
#  生成判别规则
data6_Bayes<-NaiveBayes(sales_num~.,data6)
#  各类别下变量密度可视化
# 'sunScreen','country','imported'
```

```
# 对是否防晒各类别绘制密度图
plot(data6_Bayes,vars="describe_score",main="describe_score--密度图",n=20,lwd = 2,
col=c("DeepPink","#D55E00","DarkTurquoise","#0033FF","#000000","#009900"))
# 对价格各类别绘制密度图
plot(data6_Bayes,vars = "price",main="price--密度图",n=50,lwd = 2,col = c("DeepPink",
"#D55E00","DarkTurquoise","#0033FF","#000000","#009900"))
# 预测
data7<-data3[Test_data_num,-1]
View(data7)
pred_Bayes<-predict(data6_Bayes,data7)
# pred_Bayes
# 混淆矩阵
table(data7$sales_num,pred_Bayes$class)
# 计算贝叶斯判别预测错误概率
error_Bayes<-sum(as.numeric(as.numeric(pred_Bayes$class)
                                    !=as.numeric(data7$sales_num)))/nrow(data7)
error_Bayes
library(ggplot2)
library(corrplot)
cor_Bayes<-cor(data3[,c(2:7,9)])
corrplot(cor_Bayes,method="number")
corrplot(cor_Bayes,method="pie")
```

3.2.3 计算结果

通过函数 NaiveBayes(sales_num～.,data6)生成判别规则，并对各类别下变量密度进行可视化，绘制了描述分和价格分的密度图，如下图 11 所示。

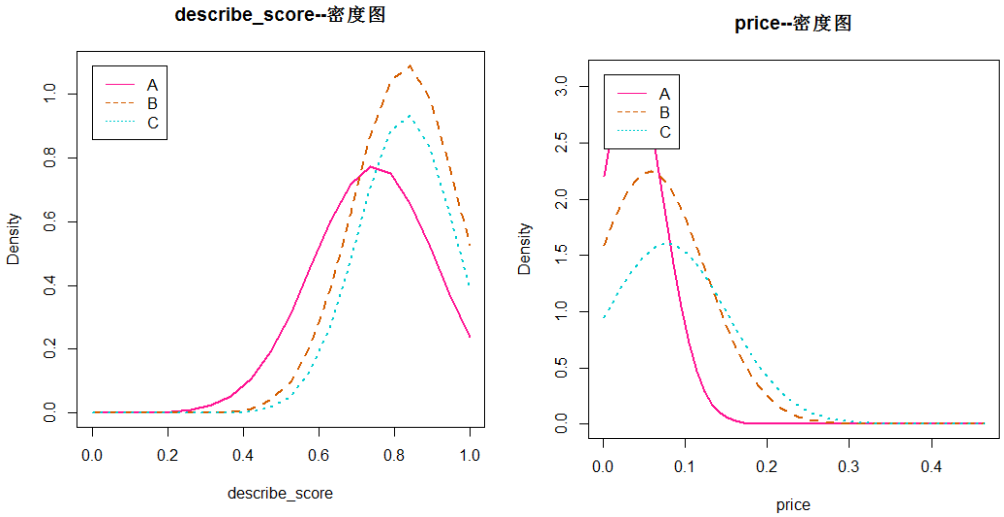


图 11 描述分和价格分的密度图

从图 11 中可以看出，在字段描述分密度图中，销售总量类型 B 和 C 主要集中在 0.82 左右，而类型 A 主要集中在 0.73 左右；在字段价格分密度图中，类型 A 的数量比其他 2 个类型要小。

经过预测后，混淆矩阵如下表 3 所示。经过计算可知，错误率为 53.11%，通过绘制各

变量之间的相关性图 12 可知，不满足各变量之间独立的条件，因此造成的错误率较高。因此，本文将进行其他算法的建模。

表 3 朴素贝叶斯预测混淆矩阵

	A	B	C
A	109	83	36
B	20	40	44
C	10	12	32

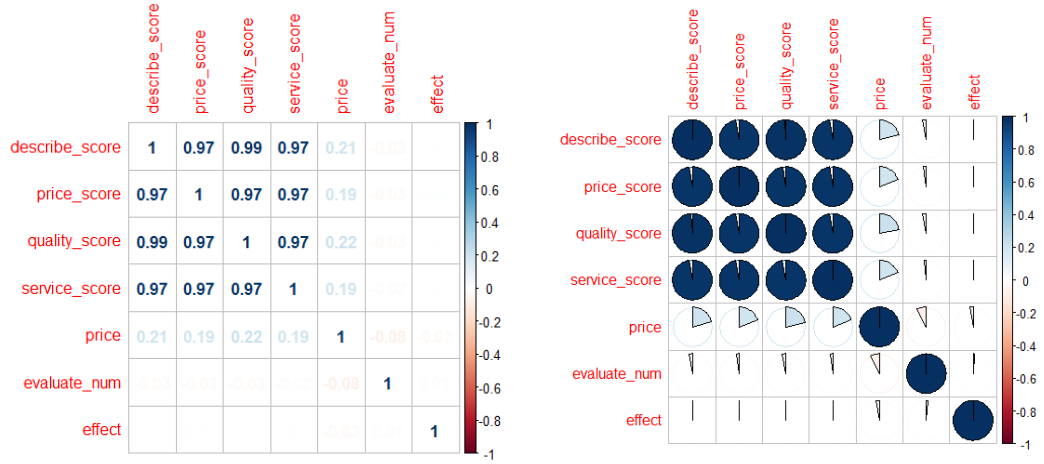


图 12 各变量相关性图

3.3 集成学习

3.3.1 AdaBoost 算法原理

AdaBoost 算法的工作机制是首先从训练集用初始权重训练出一个弱学习器 1，根据弱学习的学习误差率表现来更新训练样本的权重，使得之前弱学习器 1 学习误差率高的训练样本点的权重变高，使得这些误差率高的点在后面的弱学习器 2 中得到更多的重视。然后基于调整权重后的训练集来训练弱学习器 2。如此重复进行，直到弱学习器数达到事先指定的数目 T，最终将这 T 个弱学习器通过集合策略进行整合，得到最终的强学习器[2]。

对于分类问题的预测，通常使用的是投票法。假设我们的预测类别是 $\{c_1, c_2, ...c_k\}$，对于任意一个预测样本 x，模型的 T 个弱学习器的预测结果分别是 $(h_1(x), h_2(x)...h_T(x))$。最简单的投票法是相对多数投票法，也就是我们常说的少数服从多数，也就是 T 个弱学习器的对样本 x 的预测结果中，数量最多的类别 c_i 为最终的分类类别。如果不止一个类别获得最高票，则随机选择一个做最终类别[1]。

稍微复杂的投票法是绝对多数投票法，也就是常说的要票过半数。在相对多数投票法的基础上，不光要求获得最高票，还要求票过半数。否则会拒绝预测。更加复杂的是加权投票法，和加权平均法一样，每个弱学习器的分类票数要乘以一个权重，最终将各个类别的加权票数求和，最大的值对应的类别为最终类别。

3.3.2 算法实现

```
library(adabag)
library(rpart)
# 模型训练
# 对样本进行预处理,训练集
data4<-data3[Train_data_num,-1]
View(data4)
```

```
# 模型构建
boost=boosting(sales_num~.,data4,boos = TRUE,mfinal = 500)
# print(boost)
# 查看 boost 所生成的输出项名称
names(boost)
# 查看树的构成
boost$trees[20]
# 投票情况
boost$votes[200:215,]
# 预测类别
boost$class[200:215]
# 模型 boost 各输入变量的相对重要性
sort(boost$importance,decreasing = TRUE)
# 通过 control 参数控制基分类树的复杂度
boost2=boosting(sales_num~.,data4,boos = TRUE,mfinal = 500,
                control = rpart.control(maxdepth = 7))
# 查看树的构成
boost2$trees[20]
# 预测
data5<-data3[Test_data_num,-1]
# View(data5)
# 预测
pred_boost=predict(boost2,data5)
# 查看测试集的混淆矩阵
pred_boost$confusion
# 查看测试集的错误率
pred_boost$error
# (p=sum(as.numeric(pre_data3!=Test_data$sales_num))/nrow(Test_data))
pred_boost$class<-as.factor(pred_boost$class)
error_boost<-sum(as.numeric(as.numeric(pred_boost$class)
                            !=as.numeric(data5$sales_num)))/nrow(data5)
error_boost
```

3.3.3 计算结果

调用函数 boosting(sales_num.～,data4,boos = TRUE, mfinal = 500)构建模型，参数 mfinal 代表算法的迭代次数，即基分类器的个数，本实验设置其大小为 500，参数 boos 为采用各观测样本的相应权值来抽取 boostrap 样本。

通过调用语句 sort(boost$importance,decreasing = TRUE)可以查看变量的重要性，其变量的重要性如下表 4 所示。

表 4 AdaBoost 算法的变量重要性

evaluate_num	price	describe_score	effect	price_score
60.13014995	13.44923611	5.65788338	5.23803415	4.91359794
country	quality_score	service_score	sunScreen	imported
3.71546100	3.38505195	2.81904101	0.59512376	0.09642075

通过调用函数 pred_boost=predict(boost2,data5)进行预测，pred_boost$confusion

函数生成混淆矩阵，如下表 5 所示。模型的预测错误率为 17.88%，进一步使用随机森林算法进行建模分析。

表 5　AdaBoost 预测混淆矩阵

	A	B	C
A	209	18	5
B	16	74	15
C	3	12	34

3.4 集成学习

3.4.1 随机森林算法原理

随机森林通过自助法（bootstrap）重采样技术，从原始训练样本集 N 中有放回地重复随机抽取 k 个样本生成新的训练样本集合，然后根据自助样本集生成 k 个分类树组成随机森林，新数据的分类结果按分类树投票多少形成的分数而定。其实质是对决策树算法的一种改进，将多个决策树合并在一起，每棵树的建立依赖于一个独立抽取的样品，森林中的每棵树具有相同的分布，分类误差取决于每一棵树的分类能力和它们之间的相关性。特征选择采用随机的方法去分裂每一个节点，然后比较不同情况下产生的误差。能够检测到的内在估计误差、分类能力和相关性决定选择特征的数目。单棵树的分类能力可能很小，但在随机产生大量的决策树后，一个测试样品可以通过每一棵树的分类结果经统计后选择最可能的分类[2]。

在建立每一棵决策树的过程中，有两点需要注意采样与完全分裂。首先是两个随机采样的过程，随机森林对输入的数据要进行行、列的采样。对于行采样，采用有放回的方式，也就是在采样得到的样本集合中，可能有重复的样本。假设输入样本为 N 个，那么采样的样本也为 N 个。这样使得在训练的时候，每一棵树的输入样本都不是全部的样本，使得相对不容易出现过拟合。然后进行列采样，从 M 个特征中，选择 m 个（m << M）。之后就是对采样之后的数据使用完全分裂的方式建立出决策树，这样决策树的某一个叶子节点要么是无法继续分裂的，要么里面的所有样本的都是指向的同一个分类。一般很多的决策树算法都一个重要的步骤——剪枝，但是这里不这样干，由于之前的两个随机采样的过程保证了随机性，所以就算不剪枝，也不会出现过拟合。

3.4.2 算法实现

```
library(randomForest)
set.seed(50)
num_sales=nrow(data3)
set.seed(111)
# 总销量数：sales_num
# 构建决策树为 500 棵的随机森林模型
data.rf=randomForest(sales_num~.-name,data=data3,ntree=500,
                     importance=TRUE,proximity=TRUE,subset=Train_data_num)
# 展示所构建的随机森林模型
print(data.rf)
# 提取随机森林模型中的重要值
importance(data.rf)
# 提取随机森林中以第一种度量标准得到的重要值
# importance(data.rf,type=1)
# 对重要值进行排序并取值
data.rf_importance<-data.frame(sort(importance(data.rf,type=1)[,1],decreasing = TRUE))
names(data.rf_importance)<-'importance_sorted'
```

```
data.rf_importance
# 调用 varlmPlot 函数绘制变量重要性曲线
varImpPlot(data.rf)
# 随机森林：模型优化
# 自变量的个数，出去店的名字
num_var=ncol(data3)-1
num_var
# 设置模型误判率向量的初始值
rate=1
# 依次逐个增加节点所选变量个数
for(i in 1:num_var){
  set.seed(200)
  # mtry:用来决定随机森林中决策树的每次分支时所选的变量个数
  model=randomForest(sales_num~.-name,data=data3,mtry=i,importance=TRUE,
                     ntree=1000,proximity=TRUE,subset=Train_data_num)
  # 计算基于 OOB 数据的模型误判率均值
  rate[i]=mean(model$err.rate)
  # 展示模型简要信息
  print(model)
}
# 展示所有模型误判率的均值
rate
# 选择节点数为的作为模型的优化
set.seed(222)
model=randomForest(sales_num~.-name,data=data3,mtry=5,importance=TRUE,ntree=1000,
                   proximity=TRUE,subset=Train_data_num)
# 绘制模型误差与决策树数量的关系图
plot(model,col=1:6)
# 为图像添加图例 topright
legend(800,0.180,"A",cex=0.9,bty="n")
legend(800,0.352,"B",cex=0.9,bty="n")
legend(800,0.394,"C",cex=0.9,bty="n")
# 当决策树大于之后，模型趋于稳定，选择决策树的数量为，进行优化
set.seed(230)
model2=randomForest(sales_num~.-name,data=data3,mtry=5,importance=TRUE,ntree=400,
                    proximity=TRUE,subset=Train_data_num)
# 展示模型的简要信息
print(model2)
# 绘制相应的柱状图，展示随机森林模型中每棵决策树的节点数
hist(treesize(model2))
# 随机森林模型的可视化,注意，要有一个临近矩阵
# 不同类别，使用的符号不一样
MDSplot(data.rf,data3$sales_num,palette = rep(1,6),pch=as.numeric(data3$sales_num))
varImpPlot(model2)
```

```
# 对测试集进行目标变量预测
pre_data3=predict(model2,Test_data,type="class")
# 显示预测结果
pre_data3
# 获取混淆矩阵
table(Test_data$sales_num,pre_data3)
# 计算错误
error_randomForest=sum(as.numeric(pre_data3!=Test_data$sales_num))/nrow(Test_data)
error_randomForest
```

3.4.3 计算结果

调用函数 randomForest(sales_num～.-name,data=data3,ntree=500,importance=TRUE,proximity=TRUE,subset=Train_data_num)构建随机森林的模型，其中决策树的数量为500 棵。其 OOB 估计错误率为 29.12%，调用函数 importance(data.rf)可以查看随机森林模型中变量的重要值，如下表 5 所示,随机森林模型中两种测算方式下的自变量重要程度对比如下图 13 所示。

表 6 随机森林算法的变量重要性

evaluate_num	price	describe_score	price_score	quality_score
137.767206	41.429671	34.570018	33.107264	31.563326
service_score	effect	country	imported	sunScreen
29.112551	22.218197	21.304685	13.531906	6.441043

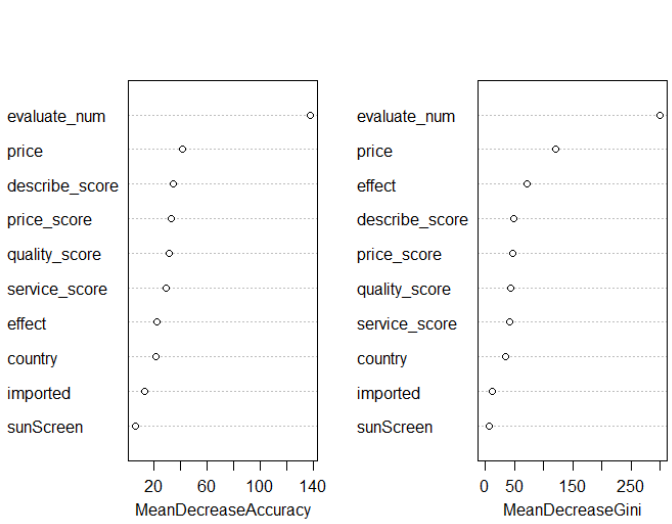


图 13 随机森林模型中两种测算方式下自变量重要对比

3.4.4 模型优化

影响随机森林模型的两个主要为：①决策树节点分支所选的变量个数；②随机森林模型中决策树的数量。使用 randomForest()其两个参数都设为默认值，但在实际使用过程中，该默认参数不一定是最优的参数值，因此本实验在构建模型时进一步确定优化参数[3]。其中对于决策树节点分支所选的变量个数的确定，采用逐一增加变量的方法进行建模，最后寻找到最优的参数。

（1）节点最优变量个数

通过参数 mtry 来改变节点变量的个数,即从 1 逐渐增加到 10(其中 10 为自变量的个数)，如下表 7 所示。从输出结果可以观察到，当决策树节点所选变量时为 5 时，模型的误判率均

值是最低的，因此将参数 mtry 的值设置为 5。

表 7 节点变量个数对应误判率

1	2	3	4	5
34.62%	29.06%	29.59%	29.09%	28.41%
6	7	8	9	10
29.68%	28.75%	28.98%	29.05%	28.54%

（2）决策树数量

在确定了模型中决策树的节点节点最优变量个数之后，还需要进一步确定模型中决策树的数量。在确定该参数时，将用到模型的可视化分析。通过调用函数 plot(model,col=1:6)，绘制模型误差与决策树数量的关系图 14，可以观察到当决策树的数量大于 400 之后，模型误差趋于稳定，因此将模型中的决策树数量大致确定为 400 左右，以此来达到最优模型。

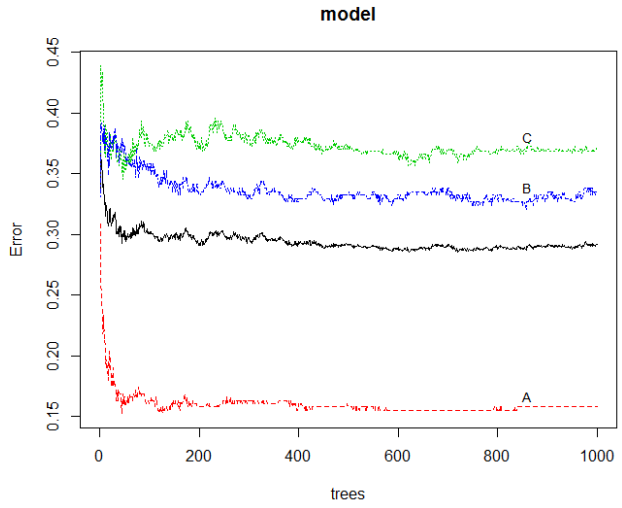


图 14 模型误差与决策树数量关系

在经过上述分析以后，本实验确定最优模型为决策树节点处变量个数为 5，模型中决策树数量为 400 的模型。再次进行建模，模型基于 OOB 数据的总体误判率为 29.03%；模型中决策树的节点数最少为 200 个，而节点数最多为 270 个，如图 15 所示；其中，评论总数、价格和价格分字段对模型的预测能力影响比较大；在图 16 中，3 个类别在中上角出现严重交叉现象，这也说明模型预测精度低的原因。

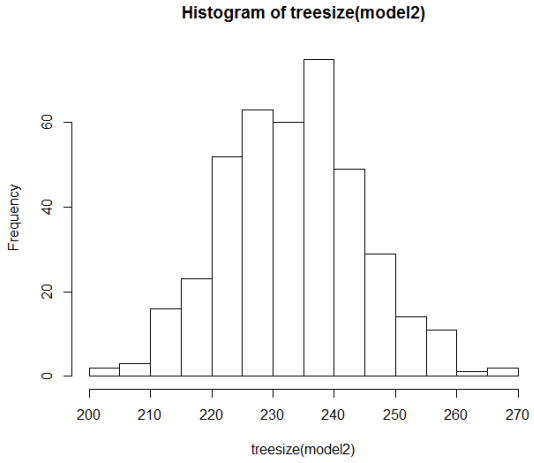


图 15 模型中决策树节点数柱状图

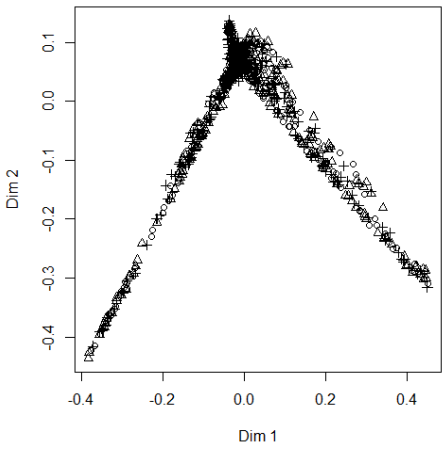


图 16 自变量在二维情况下各类别的分布情况

（3）模型预测

通过调用函数 predict(model2,Test_data,type="class")来对模型进行预测，其混淆矩阵为下表 8 所示，其错误率为 6.99%。

表 8　随机森林的预测混淆矩阵

	A	B	C
A	116	1	0
B	7	113	7
C	1	11	130

3.5 模型评价

将构建的模型，在测试集中运行，得到如下表 9 所示的错误率。可以看到，朴素贝叶斯分类错误率相差不大，已经超过 50%，AdaBoost 算法的错误率为 17.88%，而随机森林的错误率只有 6.99%，在这三种算法中，随机森林的算法最好。

表 9 三种算法错误率

算法	朴素贝叶斯分类	AdaBoost	随机森林
错误率	53.11%	17.88%	6.99%

由分析可知，①该数据集不符合朴素贝叶斯判别分析执行的前提条件——各变量条件独立，即参与建立判别分析规则的这些变量是有着显著的相关性的，很大程度上影响了预测结果的好坏；②非数值自变量在一定程度上不能归一化处理，在预测时有一定影响；③在随机森林中，3 个类别存在严重交叉，对模型的预测有一定影响，但模型的鲁棒性较强，预测的错误率在三个模型中最低。

4 主要结论

本文着重研究数据挖掘中朴素贝叶斯判别分析算法、AdaBoost 算法以及随机森林算法在口红销量预测中的效果。通过对常用算法的比较研究，总结了它们的长处和不足。在挖掘之前，对数据库进行了数据清理、数据转换、数据词云等数据预处理，处理了空缺数据、将连续值属性离散化，为进一步挖掘打好基础。 通过本文的研究，初步实现了数据挖掘技术在商品口红的应用，但是仍然存在着一些问题需进一步研究：(1)在数据预处理方面还不够完善，还需要依靠其它数据库工具人工完成；(2)文本数据处理不强，需要进一步提高。

5 课程建议

通过本课程的学习，首先是自学会了 R 语言(课下看了《R 语言编程艺术》、《R 语言实战》、《数据挖掘：R 语言实战》以及《R 语言数据分析与挖掘实战》等书籍)，然后是很喜欢老师上课的风格，可以边讨论边上课，这种氛围真的很喜欢。

想给出的几点建议是：①每个实验的时间能控制好，对于简单的模型可以交给学生做，不用花时间逐一讲解；②能在 github 上给出实验的代码，课下不用逐行敲，能把时间和精力集中在算法实现和代码的理解上；③可以发一下参考论文，课上有时间能讨论一下；④如果可以的话，也可以点评一下最近的数据挖掘比赛，给学生一些启发作用。

最后，感谢艾老师辛勤的付出，谢谢！

参考文献

[1] 李航. 统计学习方法[M]. 清华大学出版社, 2012.

[2] 周志华. 机器学习 : = Machine learning[M]. 清华大学出版社, 2016.

[3] 黄文, 王正林. 数据挖掘 : R 语言实战[M]. 电子工业出版社, 2014.