

2018年京港大学生大数据建模大赛

The logo for KKbox is displayed in a large, light gray, sans-serif font. It is centered within a dark blue horizontal band that features a blurred background of silhouettes of people with their arms raised, suggesting a concert or festival atmosphere. The overall design is modern and tech-oriented.

KKbox

基于用户音乐数据的可视化分析与推荐系统初探

第二组

周世逸



一、背景信息介绍

传统音乐



需要下载

需有大量储存记忆量

比较单一乏味

串流音乐

- ▣ 只要连上网就可以即时享受音乐
- ▣ 不占手机记忆容量多功能（有推荐功能，看视频）
- ▣ 递送成本几乎为零，销售单曲不会增加成本
- ▣ 利用少量的金钱购买聆听大量绑售音乐的权利



串流音乐市场



- 国际唱片交流协会在2017公布全球音乐产业年缔造了173亿美元的营收
- 在173亿美元的营收中，有38%来自串流音乐（66亿美元）
- 串流音乐营收也第一次超越实体唱片销售
- 成为全球音乐产业的销售主力
- IFPI还估计全球已有1.76亿名付费串流用户

串流音乐大致分为两种

付款音乐平台利用用户大数据和月费方式赚取金钱

The logo for KKBOX, featuring the text "kkbox" in white lowercase letters on a blue rectangular background.

KKBOX



Spotify

免费音乐平台和广告商合作，赚取广告费

The logo for MixerBox, featuring the text "MixerBox" in a blue, stylized font with a musical note integrated into the letter 'M'.

Mixer Box



酷播音樂

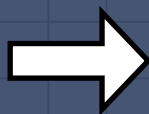
华语歌曲数量**最完整**的音乐串流平台

多达**4000多万**的歌曲

高达**117亿**的累积点播次数



透过大数据了解用户聆听行为



除了可以更加精准推荐音乐外，也可以帮助唱片公司在宣传上调整策略



「一起听」

有名人或乐团建立电台播歌给粉丝听，也可以自己当DJ，公开目前正在收听的歌曲，朋友便能同步聆听



KKBOX的特点和使用



「一点聆」

可以自订心情，时间，
类型产生播放清单

人工推荐歌单

依照类别如**Techno**，**British Rock**，
流行等等列出固定的推荐歌曲新闻或
专栏相关歌曲：

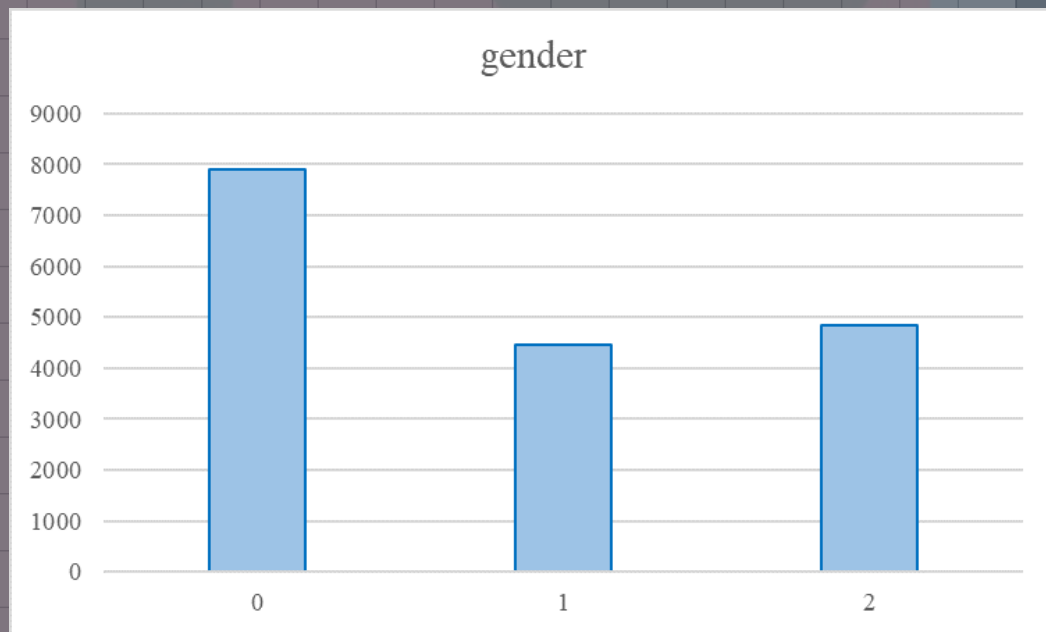
自新新闻内容或名人专栏，皆会在内
容中置放相关歌曲

KKBOX的特点和使用



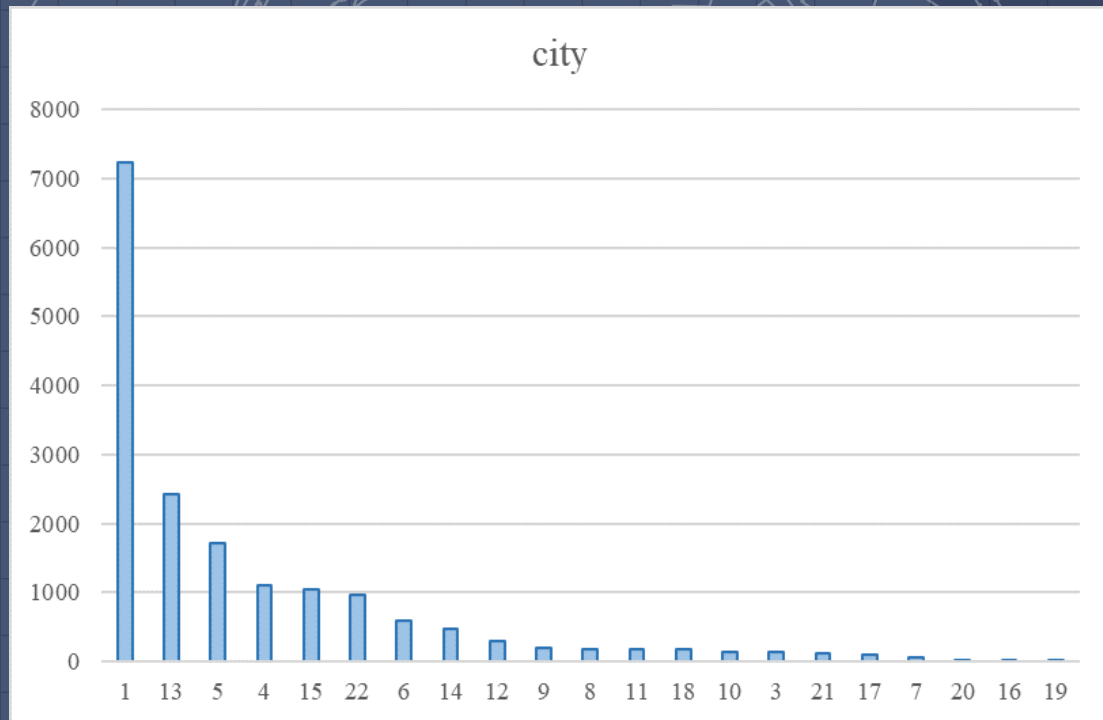
二、用户总体特征的描述

2.1 性别分布



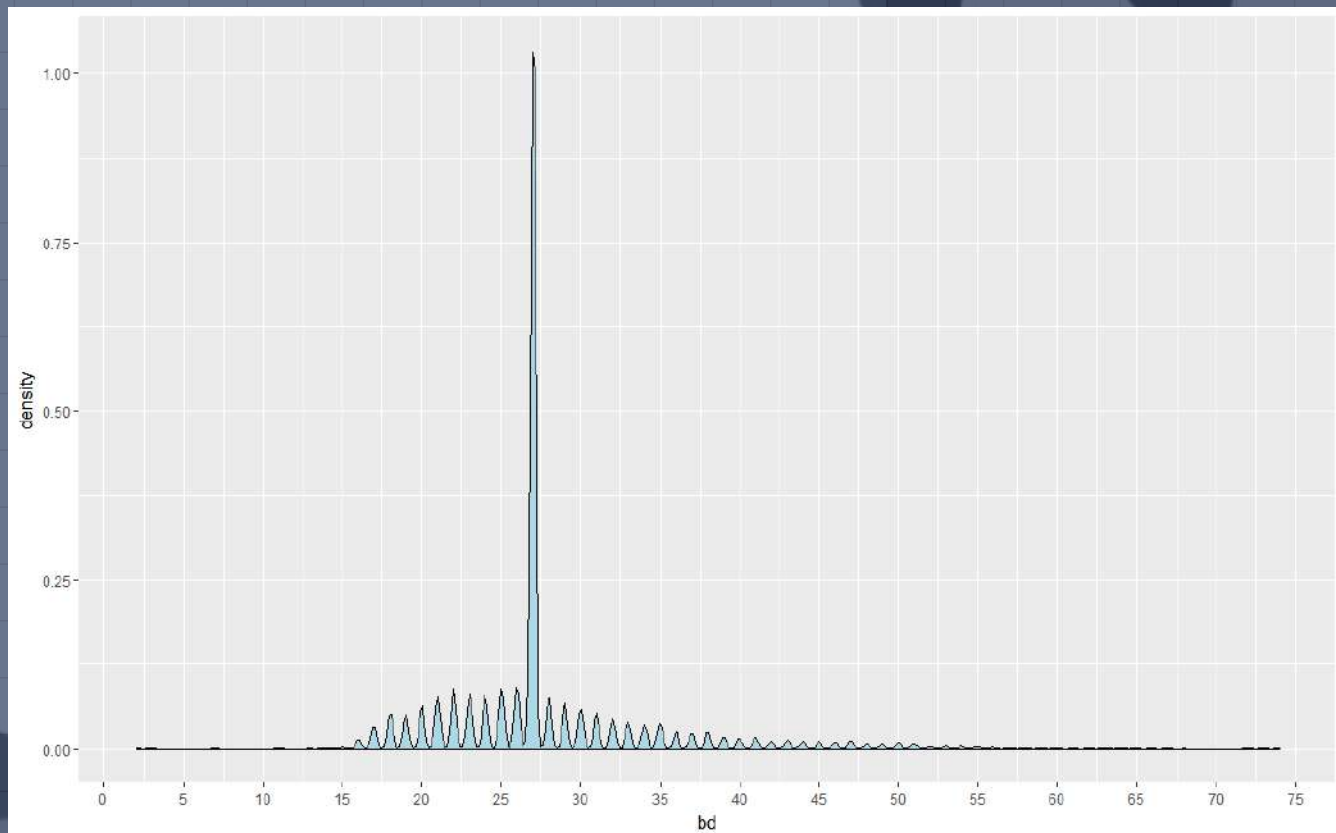
gender	frequency	percent
0	7893	45.93%
1	4453	25.91%
2	4840	28.16%
total	17186	100.00%

2.2 城市分布



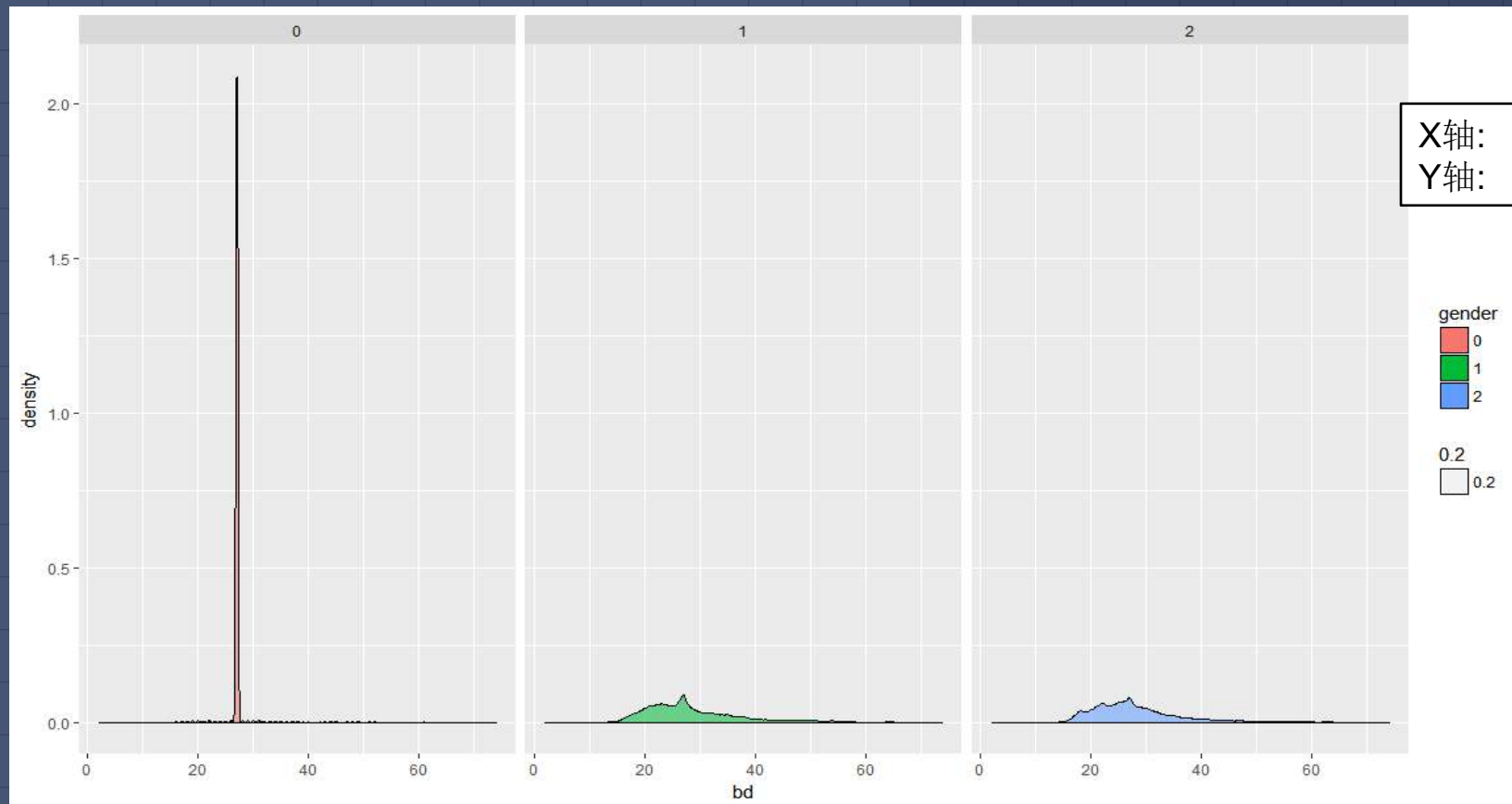
city	frequency	percent
1	7232	42.08%
13	2433	14.16%
5	1725	10.04%
4	1109	6.45%
15	1036	6.03%
22	958	5.57%
6	589	3.43%
14	475	2.76%
12	301	1.75%
9	204	1.19%
8	182	1.06%
11	173	1.01%
18	173	1.01%
10	141	0.82%
3	130	0.76%
21	125	0.73%
17	98	0.57%
7	58	0.34%
20	18	0.10%
16	18	0.10%
19	8	0.05%
total	17186	100.00%


2.3 年龄分布



X轴: 年龄
Y轴: 频率

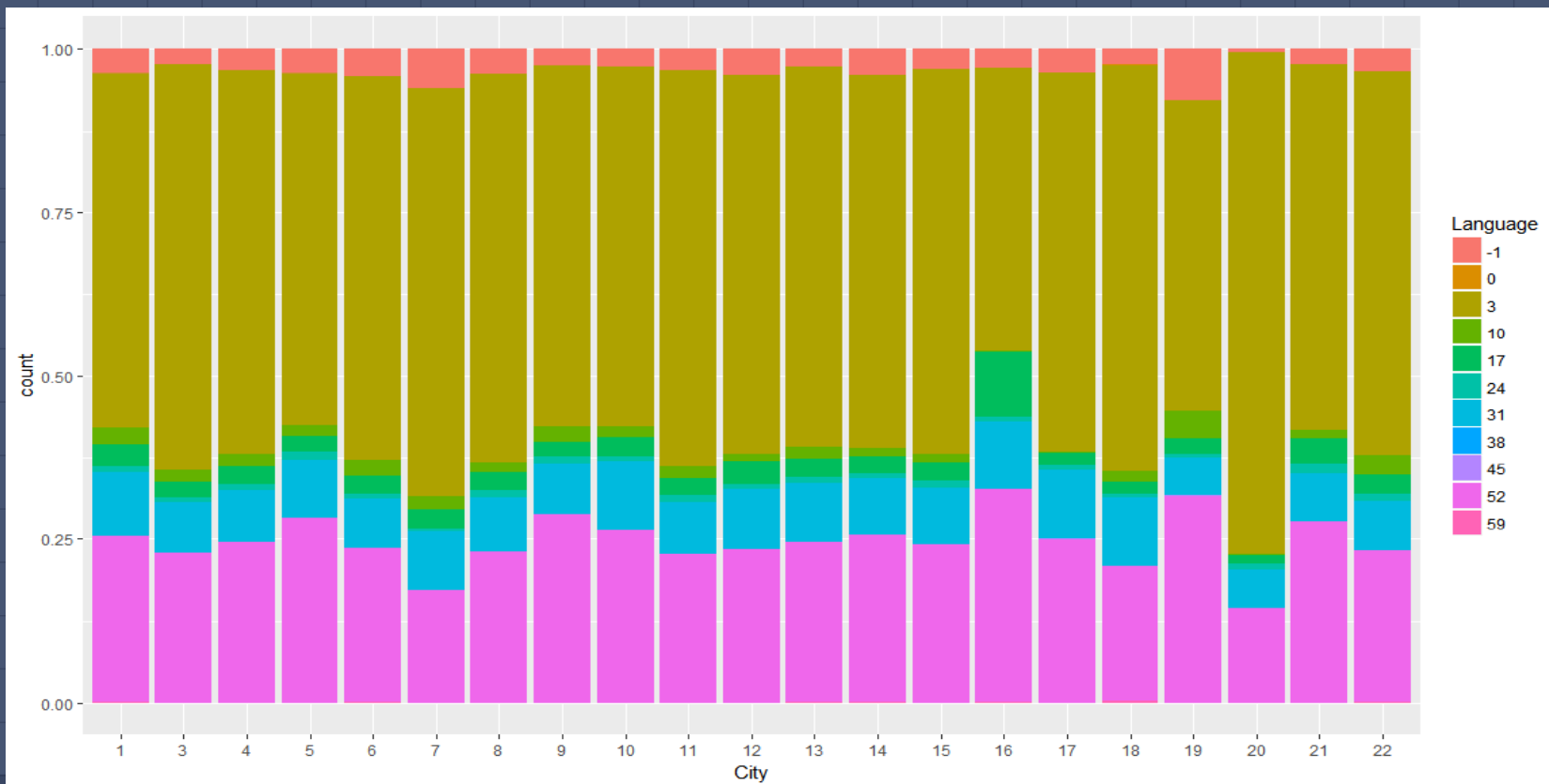
2.4 性别与年龄的关系



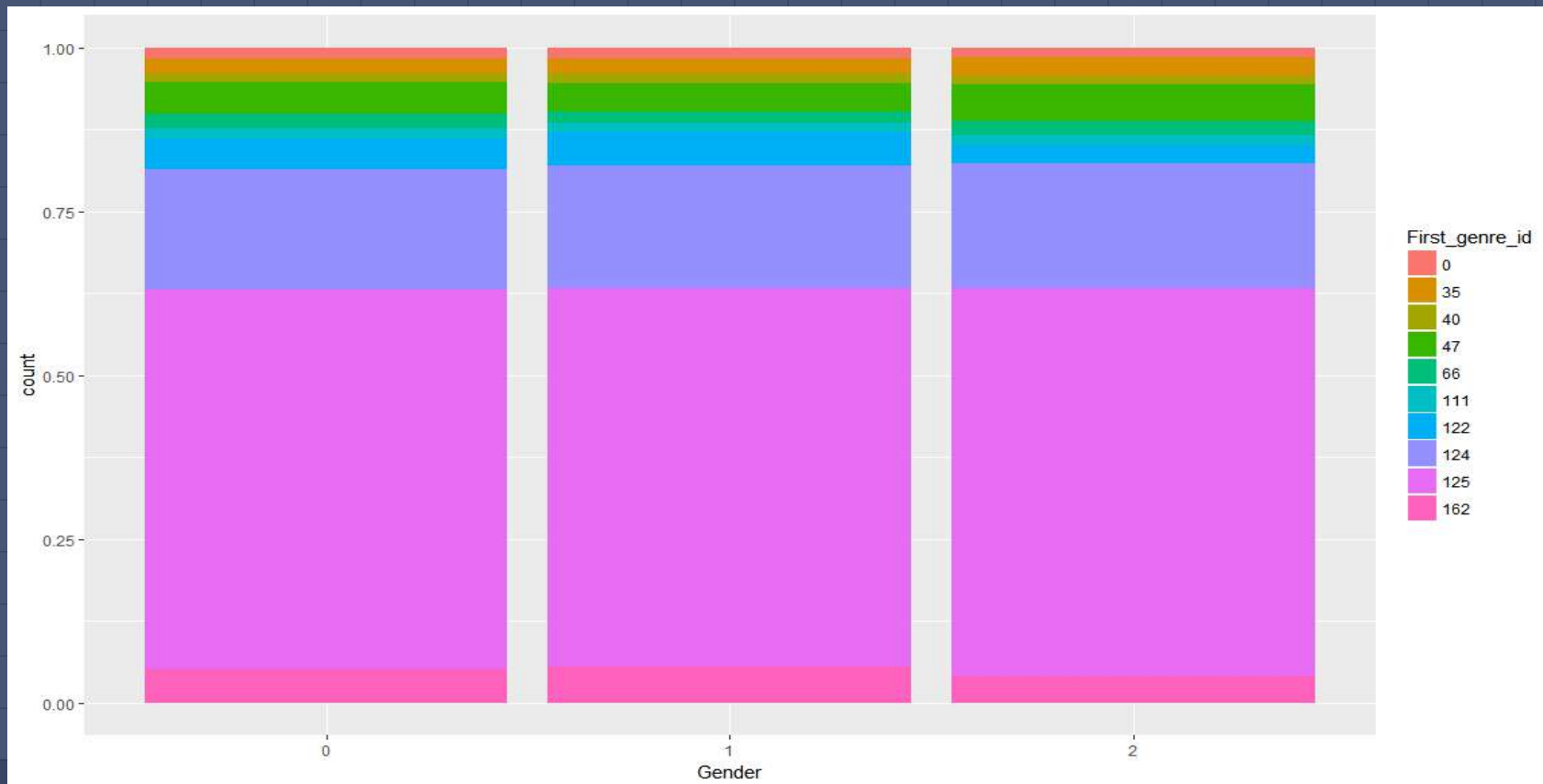
The image features a central black silhouette of a person wearing large headphones. The background is a light gray gradient, overlaid with a faint, multi-colored city skyline and a white audio waveform. The text is centered over the person's torso.

三、用户总体特征与歌曲信息间的关系探究

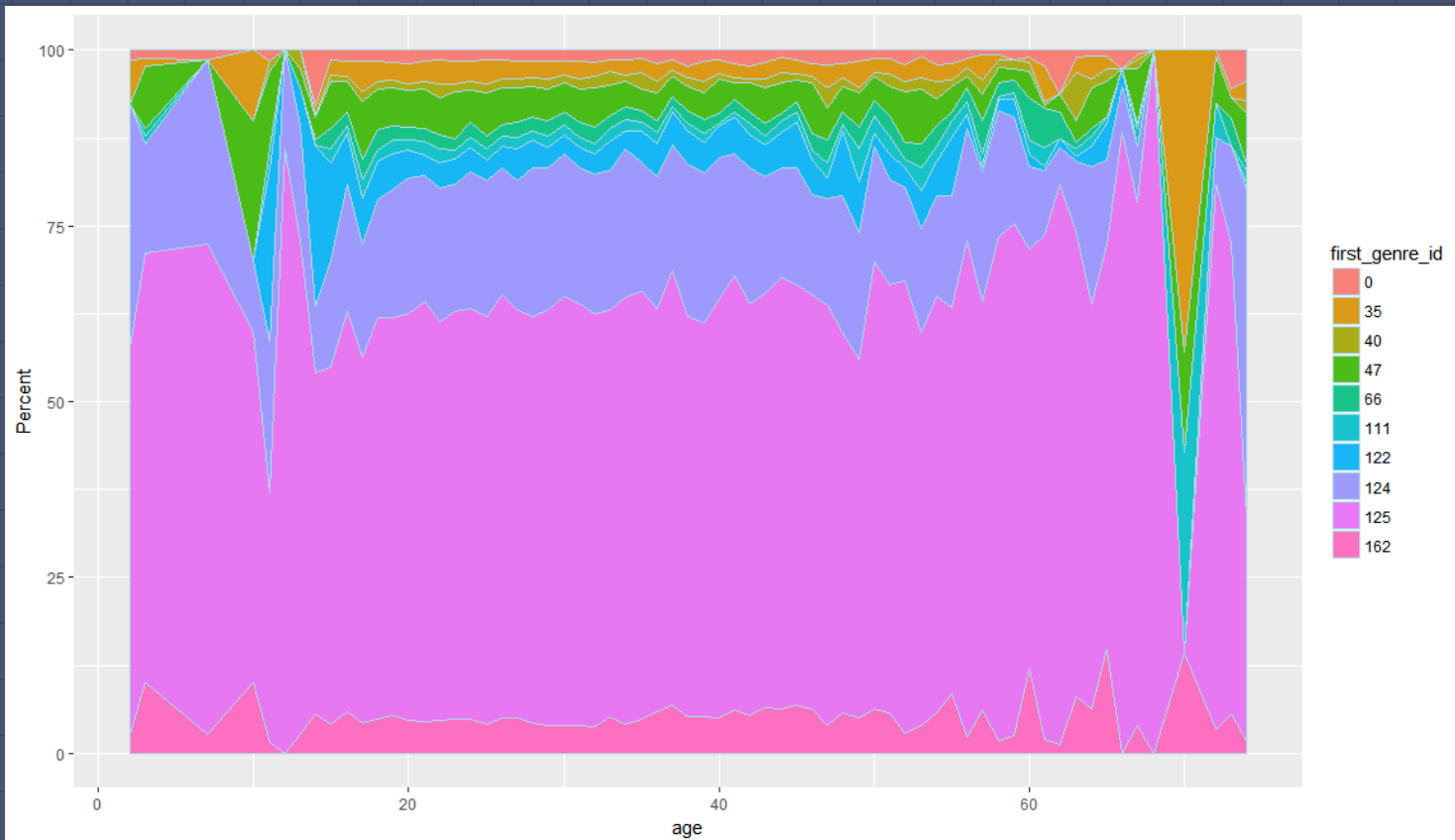
3.1 語言与城市



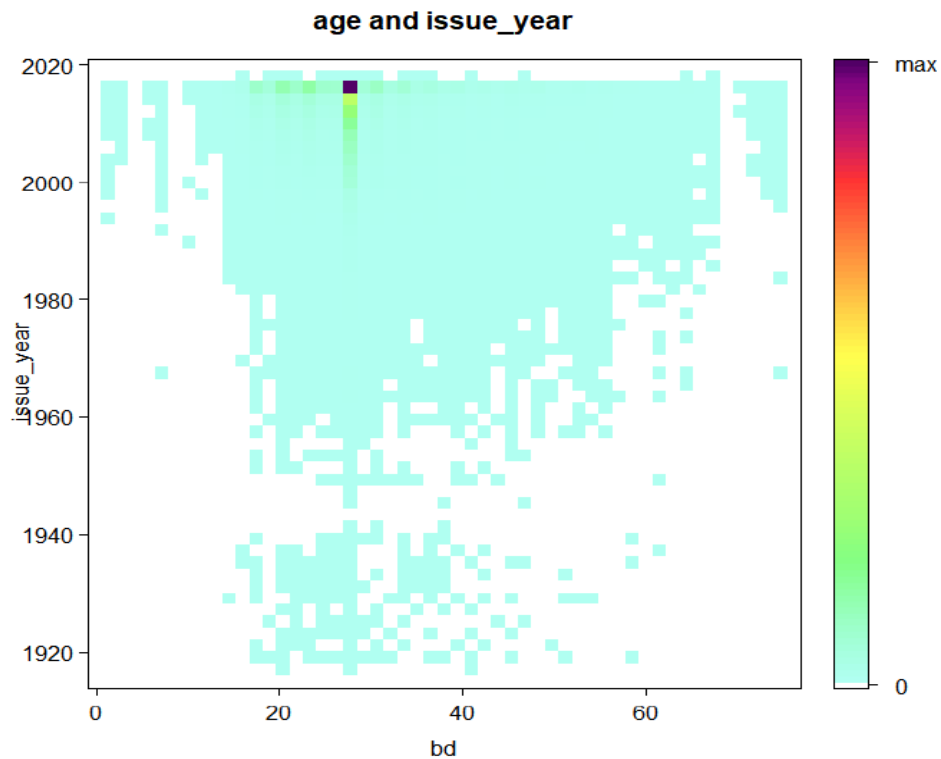
3.2 性别与风格



3.3 年龄与风格



3.4 年龄与发行年份



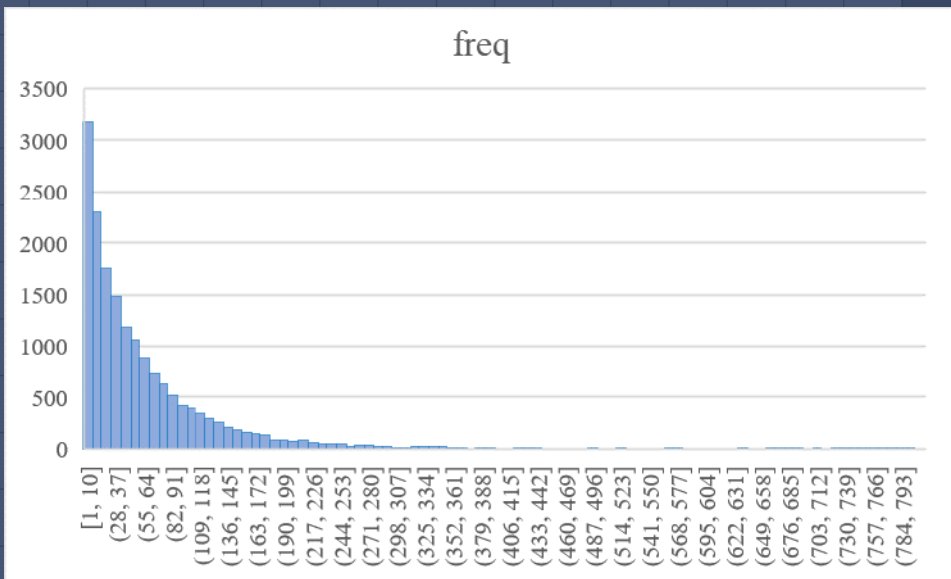
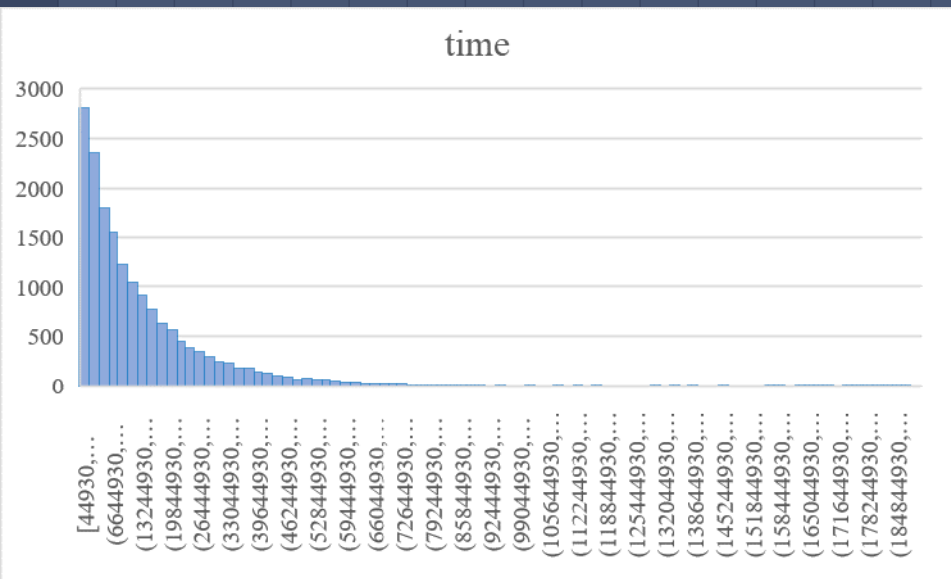


四 推荐系统初探

4.1 用户划分



4.1 用户划分

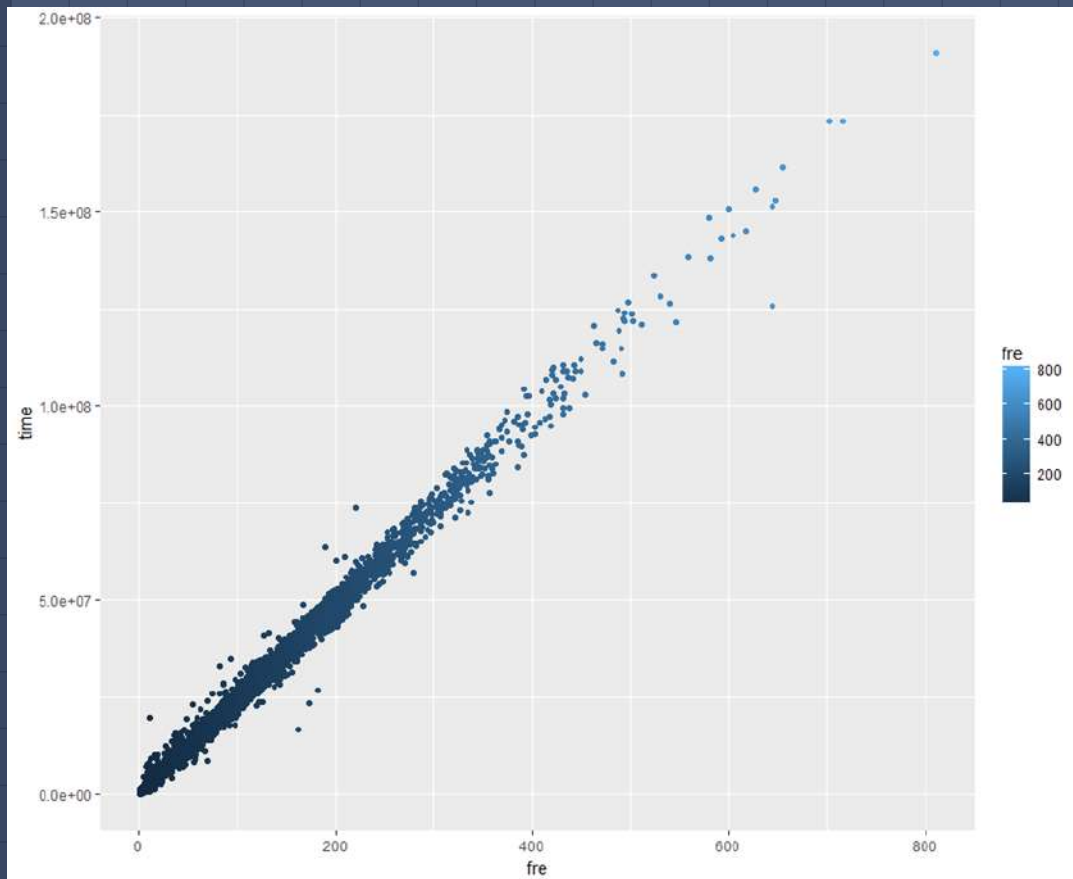


经过分析

不能确定数据为用户听某首歌曲的总时长

再观察不同用户的frequency的分布情况

4.1 用户划分



- Freq与time的分布状况具有一定的相似性
- 进一步作出二者之间的散点
- 图发现其具有明显的正相关关系
- 用freq来代替time作为衡量用户喜爱听歌程度的变量
- 以此为依据对用户进行划分。

4.1 用戶划分

- Rank 1 - 低頻使用者 (頻率少於20, 33.26%)
- Rank 2 - 中頻使用者 (頻率20-299, 65.5%)
- Rank 3 - 高頻使用者 (頻率300-400, 0.8%)
- Rank 4 - 非常高頻使用者 (頻率400以上, 0.4%)

4.2 用户等级与用户信息、 用户偏好的关系



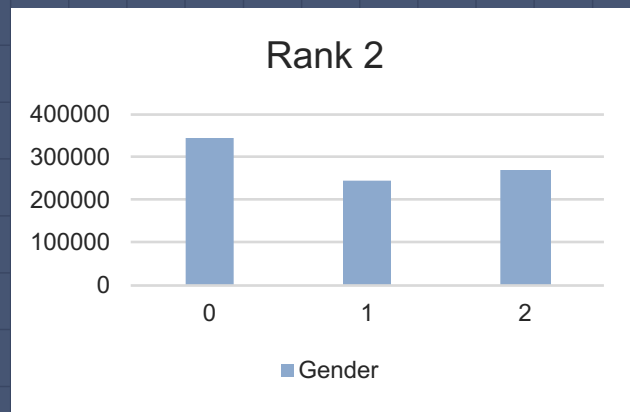
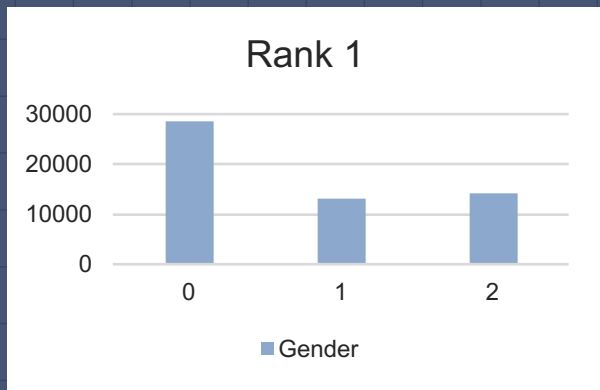
4.2 用户等级与用户信息、用户偏好的关系



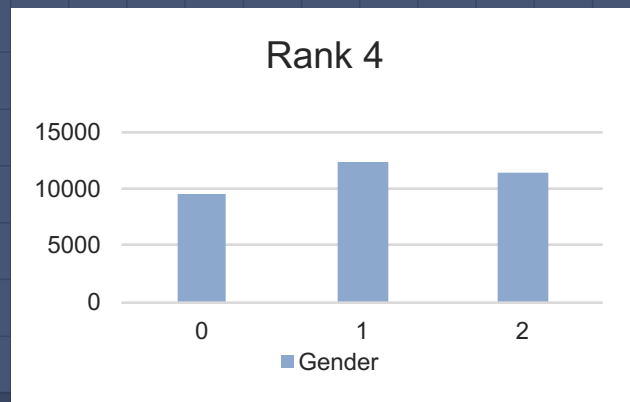
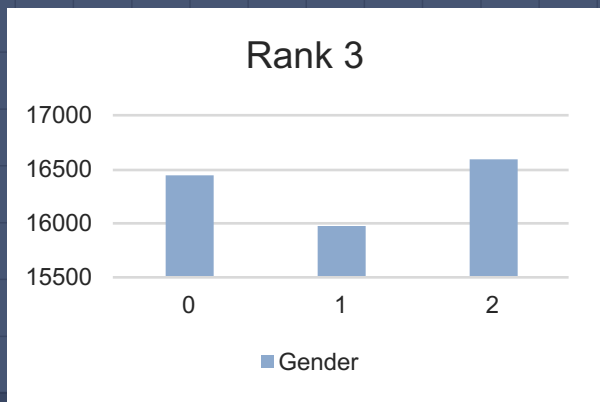
(1) rank与年龄

	年龄
Rank 1	2 - 73
Rank 2	2 - 74
Rank 3	17 -56
Rank 4	17 - 63

4.2 用户等级与用户信息、用户偏好的关系



(2) rank与性别



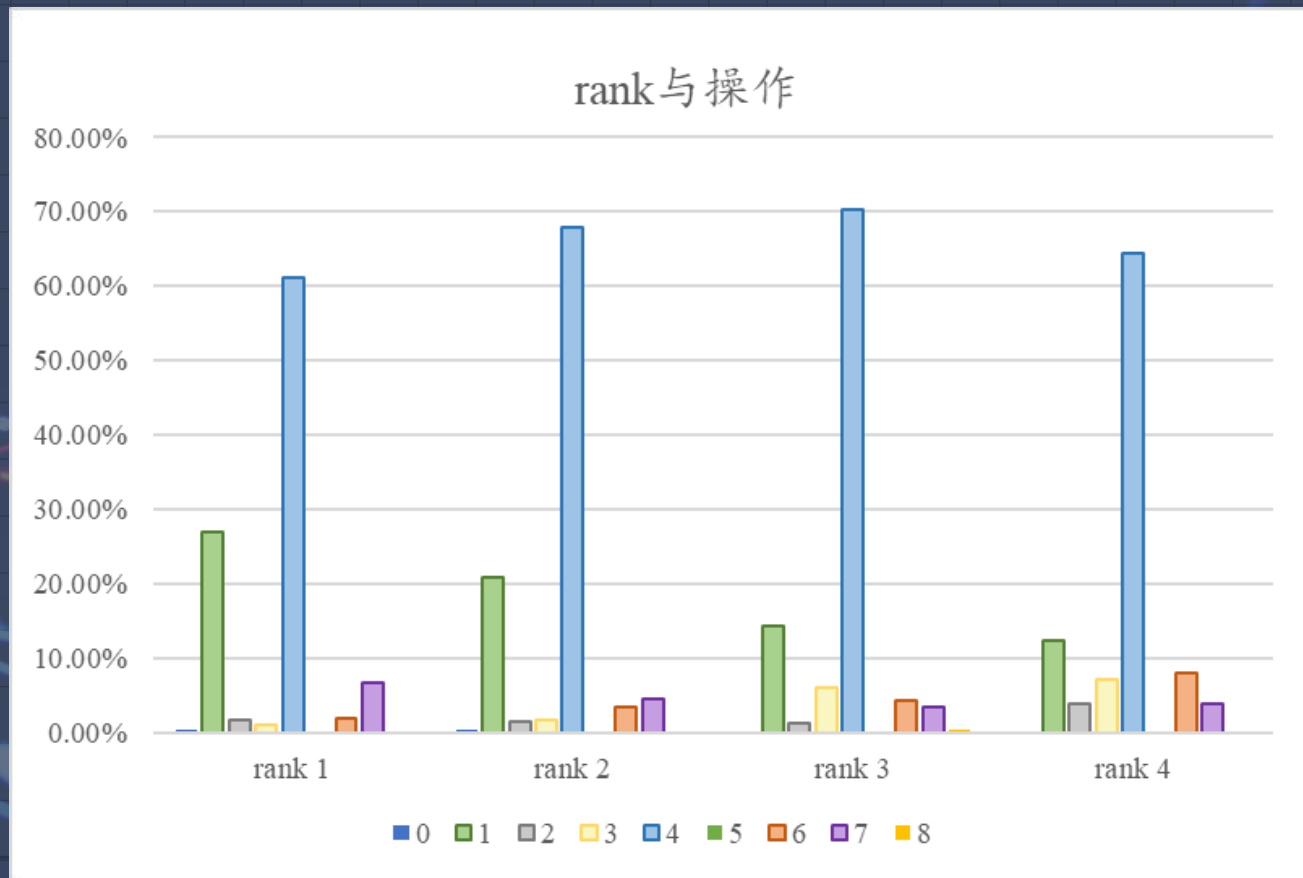
4.2 用户等级与用户信息、用户偏好的关系



(3) rank与风格

	风格
Top 5	125
Top 4	124
Top 3	47
Top 2	162
Top 1	35

4.2 用户等级与用户信息、用户偏好的关系



(4) rank与操作



4.3 推荐方式简述

4.3 推荐方式简述

新用户
以及
Rank 1

知道他们的信息基本上只有注册时填写的基本信息

而从第二部分，用户基本信息对其听歌偏好影响较小

有利于帮助用户探索其喜欢的风格

对此类用户推荐时，主要参考当前不同风格中较为流行的歌曲

4.3 推荐方式简述

Rank 2
与
Rank 3

主要的目标人群

具有升级为rank 4的潜力

操作信息可能不足以
提供丰富的信息

因此通过构建
模型的方式对
其进行推荐

4.3 推荐方式简述

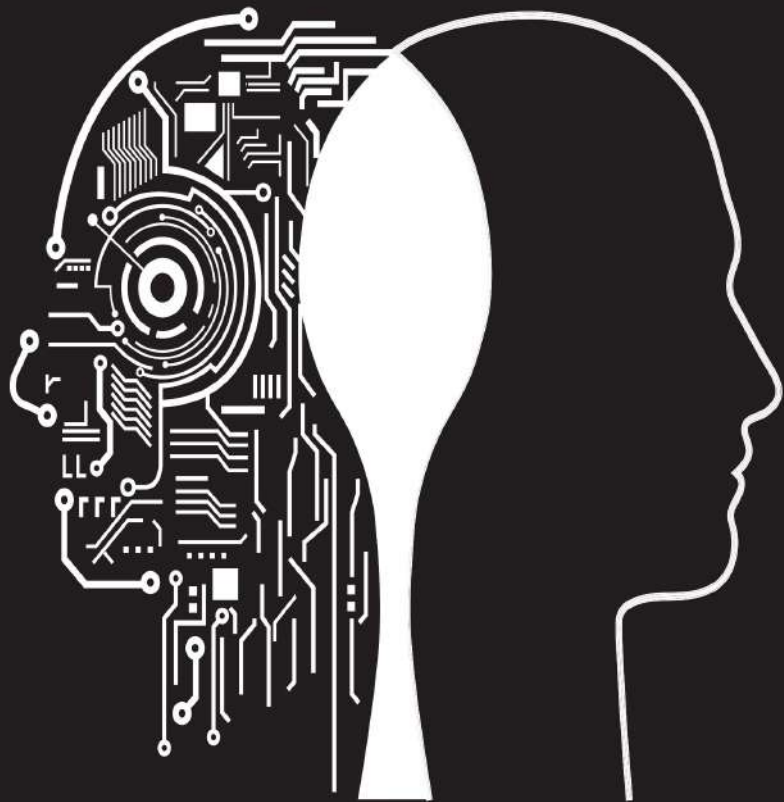
Rank 4

其非常喜爱用该app听歌

可以从中得到
其非常丰富的
历史信息

并且他们听歌
的种类与风格
已涉及很多

对于该类用户的推荐则
着重于对其未听过风格
的探索



4.4 模型构建

4.4 模型构建

Prefer = P (difference) * P (frequency)

- 用户对一类歌曲的兴趣相较于总体的评估
- 得出该用户在全体用户中对此类型歌曲的相对偏好程度

- 对用户自身相较于收听的其他歌曲在某一种类上收听频率的评估
- 得出用户对某一类软件的偏好程度

4.4 模型构建

对于P (difference)

先计算D, D=用户听一类型歌的频率-所有用户听一类型歌的平均频率

再将D带入 $p = \frac{1}{1+e^{-D}}$

对于P (frequency)

P (frequency) = 用户收听一类型歌的频率

(用户收听某一风格歌曲的次数/用户听歌总次数)

在实际操作时...

通过大型计算设备计算出所有用户对于某种类型的歌曲所得prefer的平均值, 记为P0

之后我们便可以计算出某一用户对于此一类型歌曲的prefer*, 与P0比较

若prefer* ≥ P0, 便推荐,
若prefer* < P0, 便不推荐

4.4 推荐模型构建

msno	first_genre_id	P1	P2	P
301	125	1.0000	0.9300	0.9300
28789	125	0.9957	0.8704	0.8667
4474	125	0.9885	0.8070	0.7977
29973	125	0.9984	0.6857	0.6846
30241	66	0.8241	0.1034	0.0853
33013	47	0.9956	0.1324	0.1318
28540	124	0.0000	0.0435	0.0000
7532	47	0.0712	0.0055	0.0004
1785	122	0.0489	0.0000	0.0000

附录

```
Data<-read.csv("Data.csv",header=TRUE,stringsAsFactors=FALSE)
data<-subset(Data,Frequency=="rank 2"
              &(first_genre_id==0|first_genre_id==35|
                first_genre_id==40|first_genre_id==47|
                first_genre_id==66|first_genre_id==111|
                first_genre_id==122|first_genre_id==124|
                first_genre_id==125|first_genre_id==162))

yonghu_fengge<-data[,c("msno","first_genre_id")]
yonghu_fengge<-as.data.frame(table(yonghu_fengge))
fengge<-data[,c("first_genre_id")]
fengge<-as.data.frame(table(fengge))
head(fengge)
fengge$Freq<-fengge$Freq/11236
colnames(fengge)<-c("first_genre_id","avg")

yonghu_fengge_avg<-merge(yonghu_fengge,fengge,by=c("first_genre_id","first_genre_id"))
yonghu_fengge_avg$p<-yonghu_fengge_avg$avg-yonghu_fengge_avg$Freq
yonghu_fengge_avg$P1<-1/(1+exp(yonghu_fengge_avg$p))

yonghu<-data[,c("msno")]
yonghu<-as.data.frame(table(yonghu))
colnames(yonghu)<-c("msno")
colnames(yonghu)<-c("msno","total")
yonghu_fengge_total<-merge(yonghu_fengge,yonghu,by=c("msno","msno"))
yonghu_fengge_total$P2<-yonghu_fengge_total$Freq/yonghu_fengge_total$total

yonghu_fengge_avg1<-yonghu_fengge_avg[,c("msno","first_genre_id","Freq","P1")]
yonghu_fengge_total1<-yonghu_fengge_total[,c("msno","first_genre_id","Freq","P2")]
yonghu_fengge_P<-merge(yonghu_fengge_avg1,yonghu_fengge_total1,by=c("msno","first_genre_id"))
yonghu_fengge_P$P<-(yonghu_fengge_P$P1)*(yonghu_fengge_P$P2)
head(yonghu_fengge_P)
```

 **BIG**
DATA,
 **BIG**
DEAL

