阿里音乐流行趋势预测-最终报告

团队成员：李懿，郑清卓，侯军

1 简介

阿里音乐拥有数百万的曲库资源，拥有数亿人次的用户试听、收藏等行为。希望以用户的历史播放数据为基础，通过对音乐平台上每个阶段艺人的试听量的预测，挖掘出即将成为潮流的艺人，从而实现对一个时间段内音乐流行趋势的准确把控。

我们分别从数据处理，数据分析，结果评价和结果展示四个部分入手，解决对于音乐趋势的预测问题。

2 问题陈述

我们需要解决的问题是，如何从过去 6 个月的历史用户行为中，对将来 2 个月的歌曲播放量进行预测。

比赛中提供了两个数据表：用户行为表和歌曲信息表。

表 1 用户行为表

name	data_type	describe	example
user_id	String	用户唯一标识	7063b3d0c075a4d276c5f06f4327cf4a
song_id	String	歌曲唯一标识	effb071415be51f11e845884e67c0f8c
gmt_create	String	用户播放时间（unix时间戳表示）精确到小时	1426406400
action_type	String	行为类型：1，播放；2，下载，3，收藏	1
Ds	String	记录收集日（分区）	20150315

表 2 歌曲信息表

name	data_type	describe	example
song_id	String	歌曲唯一标识	c81f89cf7edd24930641afa2e411b09c
artist_id	String	歌曲所属的艺人Id	03c6699ea836decbc5c8fc2dbae7bd3b
publish_time	String	歌曲发行时间，精确到天	20150325
song_init_plays	String	歌曲的初始播放数，表明该歌曲的初始热度	0
Language	String	数字表示1,2,3…	100
Gender	String	1,2,3	1

在天池平台上，我们需要提交一份对每个艺人在 9.1~10.30 的 60 天内每天的播放数据作为测评数据。文件中每一行包括艺人 ID，日期以及播放量。评价指标如下图所示。

设艺人j在第k天的实际播放数为$T_{j,k}$，参赛选手集合为U，艺人集合为W，参赛选手i的程序计算得到艺人j在第k天的播放数为$S_{i,j,k}$，

则参赛选手i对艺人j的播放预测和实际的方差归一化方差$\sigma_{i,j}$为：

$$\sigma_{i,j} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left((S_{i,j,k} - T_{j,k})/(T_{j,k})\right)^2}$$

而艺人j在的权重根据艺人的播放量平方根：

$$\phi_j = \sqrt{\sum_{k=1}^{N} T_{j,k}}$$

参赛选手i的预测为F_i

$$F_i = \sum_{j \in w}(1 - \sigma_{i,j}) * \phi_j$$

最终排名按照F值评判，F值越大，代表结果越优，排名越靠前。

图 1 测评标准

3 技术方案

3.1 数据预处理

由于原始数据是类似 log 信息的条目，为了避免后续重复统计，需要先统计每天的播放量。这里我们使用了两个维度的统计角度，从歌曲角度统计和歌手角度统计。

得到每日播放量的数据结构如下表所示。

表 3 每日播放量

name	data_type	describe	example
artist_id/song_id	string	歌手ID/歌曲ID	8fb3cef29f2c266af4c9ecef3b780e97
date	int	日期	20150301
play	int	播放量	102
download	int	下载量	23
like	int	收藏量	4

3.2 数据分析

因为最后需要提交的测评结果是每位艺人在 9.1~10.30，这 60 天的歌曲播放量，很自然地会先从歌手角度入手进行分析。选取 ID 为“5e2ef5473cbbdb335f6d51dc57845437”的歌手进行数据分析，首先绘制播放量、下载量和收藏量随日期变化的曲线，如图 2 所示。同时对下载量和播放量的关系，以及收藏量对播放量的关系，进行绘图，可以得到图 3 展示的散点图。

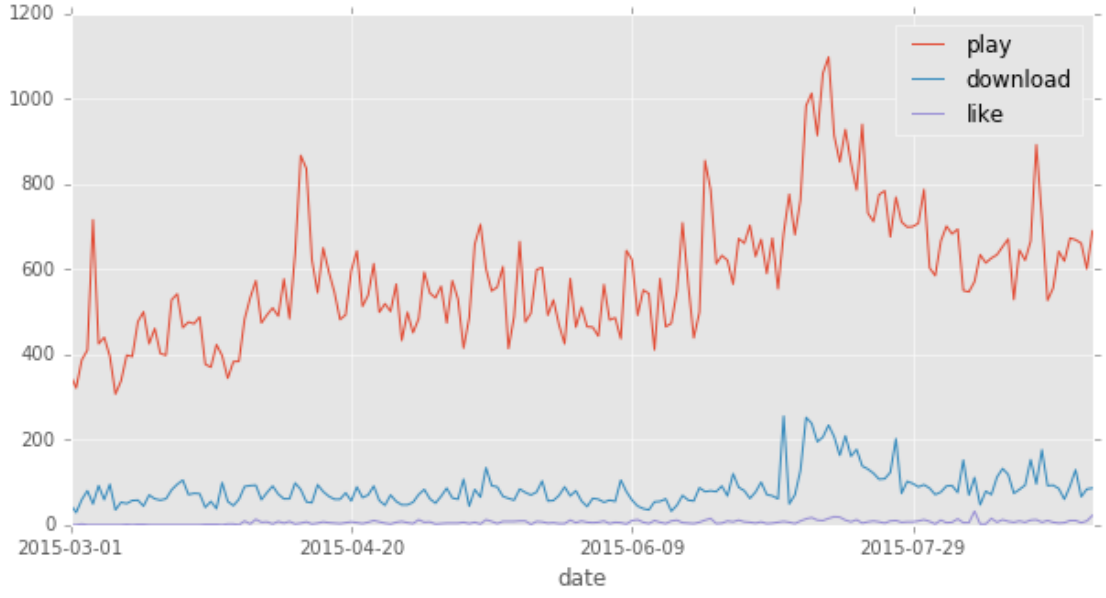

图 2 播放量、下载量和收藏量曲线

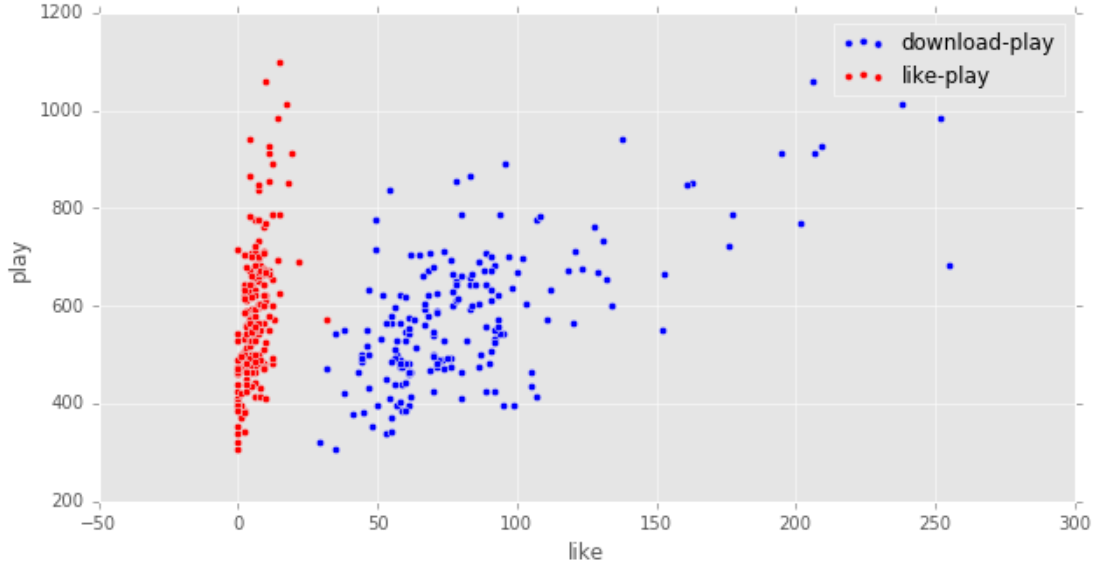

图 3 下载量、收藏量与播放量的关系

从图 2 可以看出，歌手的播放量大约为下载量的 5 倍，收藏量几乎可以忽略。同时播放量曲线的波动较大，呈现周期性。在图 3 的散点图中，蓝色点代表下载量与播放量的关系，红色点表示收藏量和播放量的关系。由于收藏量很小，而且相比于呈现正相关的下载播放量，体现不出明显的相关关系。所以我们接下来重点分析下载与播放之间的关系，而不再考虑收藏量对播放的影响。

3.3 均值模型

通过仔细的分析后，我们发现除了个别发布了新歌的歌手外，其他歌手的播放量基本呈现一个比较平稳的趋势。所以我们首先采用均值模型，对预处理后的每日播放数据进行预测。均值公式为:

$$X = \frac{1}{N}\left(\sum_{i=1}^{N} x_i\right)$$

使用朴素的均值方法可以得到稳定的播放趋势，基本符合未来歌曲趋势的预测。因为日期距离预测时间越近，越能拟合最终的结果，所以我们分别使用最近 6 天、7 天、8 天的数据，做均值分析。比较结果如图 4 所示，根据测评结果，最近 7 天的均值最接近真实值。

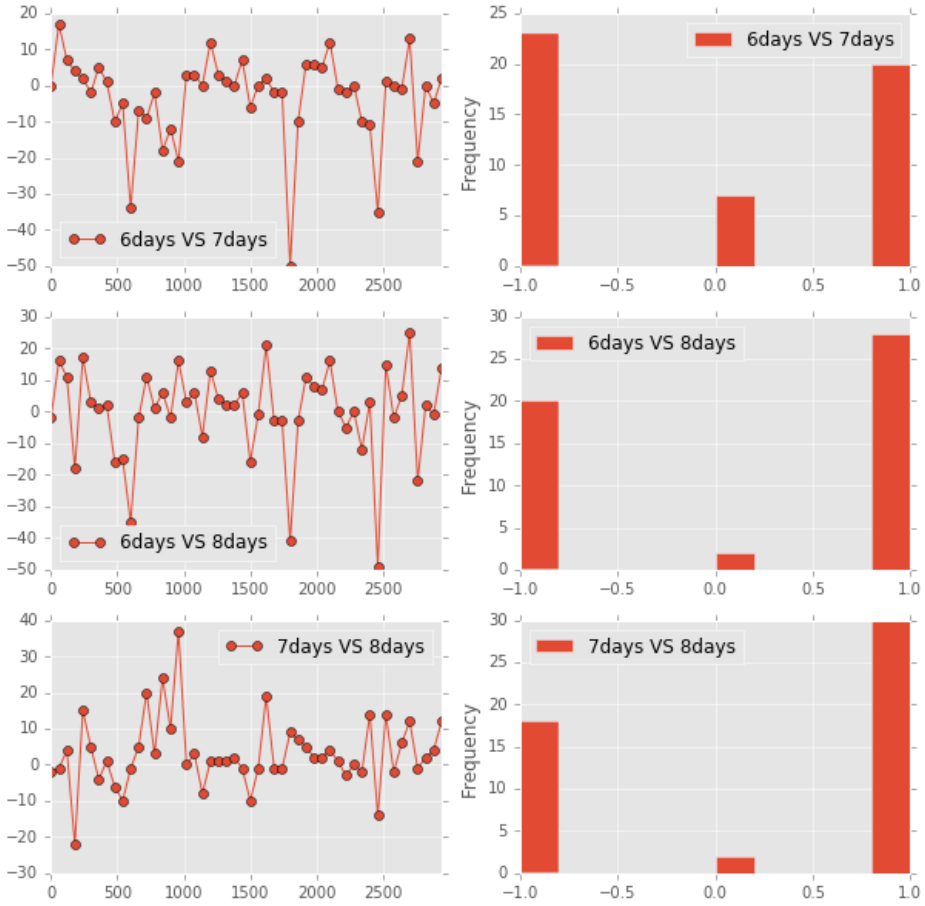

图 4 不同时间长度的均值对比

3.4 星期特征抽取

用户收听歌曲的频率和日期存在一定的关联，所以在播放曲线中体现出了周期性，在图 5 中可以看出周中比周末的收听量大。在加入星期的特征后，我们将每个星期中周一到周日的播放趋势，叠加到之前计算好的 7 天均值上，重复每周的播放趋势。这个方法得到了很好的效果，使得最终得分提升了 8.3%。

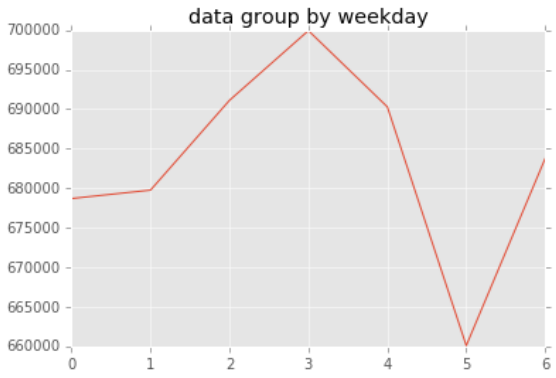

图 5 一周中每天的播放量趋势

3.5 歌曲特征抽取

如果将一位歌手的所有歌曲都单独进行播放曲线的绘制，我们可以从图 6 看到歌曲之间是有相似的走势的，所以我们对一个歌手的歌曲进行聚类，再分别对每个小类的歌曲趋势进行预测。

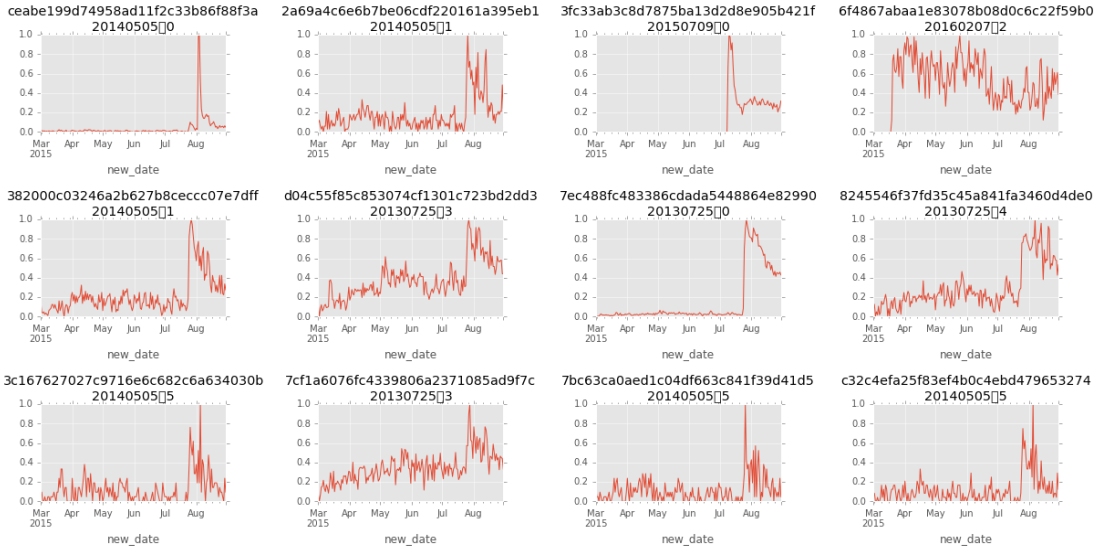

图 6 歌曲趋势

我们采用 DBSCAN 算法，对歌曲进行聚类。其中距离的度量方法是直接将两首歌曲的播放量，以日期为基准两两相减，再将绝对值类和得到。选择时间序列分析中的 ARIMA 模型，分别对每个类别进行趋势预测。最后得到的得分相比星期特征提取方法提升 11.6%。

4 实现与结果分析

代码实现部分基本上使用 Python 语言进行编程，结合 Jupyter Notebook 程序开发。由于这次的比赛只要求提交预测结果进行测评，我们没有建立完整的工程代码，主要使用 Notebook 提供的在线代码片段编辑功能进行数据处理、数据预测的工作。代码片段保存在 preprocess.ipynb、analyse.ipynb、result.ipynb 文件中。

在第一季结束时，我们的队伍排名 211 名。


我的成绩
第1赛季最优排名/成绩： 211 / 53049
第 2 赛季成绩
第 1 赛季成绩
时间
评分
当天排名
2016-06-11 21:47:54
15058.54704294
91
2016-06-10 09:48:24
15027.04284615
95
2016-06-09 09:43:31
15020.47489427
95
2016-06-07 22:05:11
14963.06159504
108
2016-06-05 23:13:29
6466.48772256
80

图 7 第一季排名