# 概要：

推荐在我们的日常生活中被广泛使用。尤其是在电子商务领域，一个好的推荐系统可以帮助用户地段。在本文中，我们介绍了KKBOX的方法。音乐推荐挑战。在这次挑战中，我们要求建立一个能够预测用户将在用户的第一次在KKBOX中观察到的听力事件。我们的解决方案是主要基于系统的、广泛的特征工程以及一组简单的增强树分类算法，两者都可以很容易地用于工业。基于比赛已经结束，同时网络上已经给出了第一名的解决方案，本文的主要内容在于解析特征工程。

# 介绍：

在这次比赛中，组织者还提供了三个属性用户歌曲交互上下文，以及每个歌曲和用户。同时本次比赛是可以使用第三方的巩固数据；但是，在本解决方案中没有找到我们可以找到的公共数据集在挑战期内有效使用。第一次侦听没有明确的时间戳信息用户歌曲对的事件。但是，示例的顺序训练和测试数据集本身是按时间顺序的（显然，Kaggle组织者没有对数据集进行无序处理）。使用此时间顺序信息和新的注册信息，我们可以大约提取的第一个侦听事件的时间戳用户歌曲对。

特征工程是数据科学中的关键一步流程，正好在建模阶段之前。这是其中之一

预测中最重要和最耗时的任务分析项目。其目的是从原始数据设计，使模型更容易、更快地训练的功能，以及提高它的性能。实际上，几乎所有的赢家最近的比赛有广泛使用的特点工程设计，并投入大量的时间和精力来设计这些特征。例如，在极端情况下，赢家，他们花费了95%的特征工程时间，建模时间仅为5%。特征工程

包括深入的数据探索以了解它们的特性，如以及创造有意义和有帮助的领域知识

特征。

# 数据集描述：

在这个音乐推荐比赛中，有五个csv文件夹。数据集的结构非常简单，如下列表。train.csv和test.csv包含用户ID、歌曲ID和三个上下文属性。其他三个csv文件包括每个歌曲和用户的属性。我们可以合并所有这些属性在train.csv和test.csv中生成联合表。

Member.csv

|  |  |  |
| --- | --- | --- |
| 字段明 | 说明 | 备注 |
| Msno | ID | 唯一标示 |
| City | 城市 |  |
| Dd(age) | 生日 |  |
| gender | 性别 |  |
| Registered\_via | 注册级别 |  |
| Registered\_time | 注册时间 |  |
| Expiration\_date | 到期日期 |  |

列表一

songs.csv

|  |  |  |
| --- | --- | --- |
| 字段明 | 说明 | 备注 |
| Song\_id | ID | 唯一标示 |
| Song\_length | 歌曲长度 |  |
| Genre\_ids | 类组IDs | 一首歌可以含有多个类型 |
| Artist\_name | 歌手名称 | 备注 |
| Composer | 作曲家 |  |
| lyricist | 作词家 |  |
| language | 语言 | 中英文歌曲等 |

列表二

Song\_extra\_info.csv

|  |  |  |
| --- | --- | --- |
| 字段 | 说明 |  |
| Song\_id |  |  |
| Song\_name |  |  |
| isrc |  |  |

列表三

Train.Csv/test.csv

|  |  |  |
| --- | --- | --- |
| 字段 | 说明 | 备注 |
| Mson |  |  |
| Song\_id |  |  |
| Source\_system\_tab |  | 类别型字段 |
| Source\_screen\_name |  | 类别型字段 |
| Source\_type |  | 类别型字段 |
| Targe |  | Test.csc没有本目标字段 |

列表四

# 探索：

数据挖掘是数据挖掘项目的第一步，很大程度上依赖于数据集。重要的是

如何处理任务。在上表格中，我们可以看到我们的数据集的结构并不十分复杂。因此，我们

在执行数据之前，将所有表合并到一个联合表中勘探描述如下。

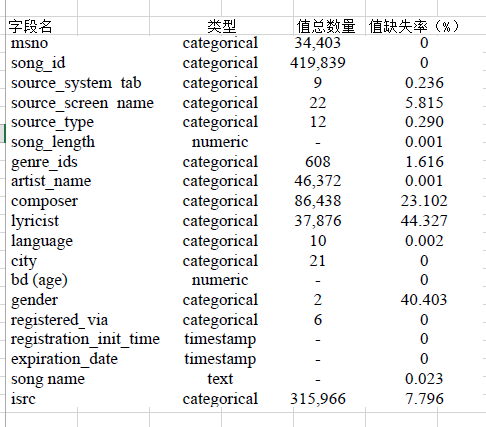
## 每个特征中的值计数

因为数据集中的大多数特性都是分类的特点是，我们计算每一个的不同值的数量

分类特征如表3所示。用户数，歌曲数，艺术家的名字，作曲家，抒情诗人都很大，所以很热门编码将不是合适的预处理方法，因为维度的诅咒和记忆的代价计算。在这种情况下，选择其他编码方法或者找到这些特性的一些表示将有助于模型。

## 缺失值

理解和处理缺失的值也很重要影响预测模型的性能。我们做一个简短的（列表五）中观察到的缺失值率汇总。作曲家、作词家和性别的缺失价值率高于其他特征。因为在这个挑战中，竞争对手可以使用外部数据，这些特性将是很好的候选搜索外部数据。虽然我们没有使用任何外部数据，因为缺乏时间，它将有助于丰富，通过外部数据，高价值缺失率的特点。此外，在不同的特征还使我们为歌手的名字比作曲者和作词者的名字还要多。



列表五

## 训练集与测试集

我们也希望看到训练和测试之间的区别，因为大多数数据挖掘算法假设训练和测试集必须相同分布（相同分布）。训练集和测试集中分别有7377418和2556790个例子，这几乎是2.88比1。针对培训中的目标集合，正示例数与负的例子大约是1，所以我们有一个平衡的二分类任务。检查训练集合测试集在不同特征上的分布，例如，我们比较上下文功能中每个值的计数（Source\_screen\_name，source\_system\_tab，source\_type）,在训练和测试集之间。

在图2中，我们可以看到训练和测试集确实是不同的；例如源系统选项卡中“my library”值的百分比训练集比测试集大10%。这种情况也会出现在其他上下文功能的Source\_screen\_name和Source\_type。因为有如此大的差异，理论上没有允许在测试集上应用在训练集上训练的模型. 这就是为什么参赛者一起对训练和测试集进行了特征工程。

训练集中有35996首不同的歌曲，测试集有224753首和59873首不同的歌曲只存在于测试集中。这意味着测试集有26.64%的新歌，通常称为冷启动。这些歌曲在12.52%的测试中出现。例子。这是一项艰巨的任务。冷启动用户也存在问题。中有30755个不同的用户

训练集和25131个不同的用户在测试集中3648只存在于测试集中（14.52%的冷启动用户，出现在7.20%的测试示例中）。冷启动这项任务中的问题非常严重，我们需要

更注意获得一个健壮的模型，以避免过度拟合。

# 特征工程（F表构建特征函数）

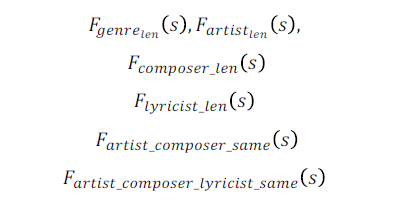
## 歌属性：

有两个与歌曲的原始特性相关的csv文件（song.csv和song\_extra\_info.csv）。我们提取了几个特征来自Song\_Extra\_info.csv。

从ISRC提取歌曲的年份和国家：来自维基百科，知道国际标准的前两个字符记录代码（ISRC）是两个字符的国家代码，并且ISRC的后两个字符表示参考年。我们提取歌曲的年份和国家代码来自ISRC的歌曲。



统计如下特征，类别、歌唱家、作曲家、作词家；因为有些歌曲可以包含多个类别、歌唱家、作曲家、作词家；我们计算每首歌中类别、艺术家、作曲家、作词家的数量。在一些歌曲中，艺术家、作曲家和作词的人可能是一样的。我们生成两个功能，指示如果一首歌中的艺术家和作曲家是同一个人，一首歌中的作曲家和作词是同一个人。

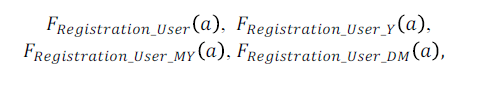


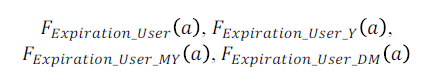
## 用户属性：

用户年龄：在数据探索中，我们可以看到年龄中的异常值属性，包括小年龄（年龄=0）、负年龄（年龄<0）和年龄大（年龄>130）。在这种方法中，我们将年龄小于0岁的设置为0，大于75的设置为76.



用户注册和到期日期：从描述中，记录历史从2015-01-01到2017-03-31。我们因此，计算到期剩下有多少天的特征，注册日期至2017-03-31，有效期至日期和2017-03-31。此外，我们生成注册日期和到期日期的年和月；





构建用户和歌曲之间的年龄差距特征：此特征表示差距在用户的年龄和歌曲的年份之间。年轻人更喜欢听新歌。



统计每个用户听过的歌曲数目：



## 特征表示：

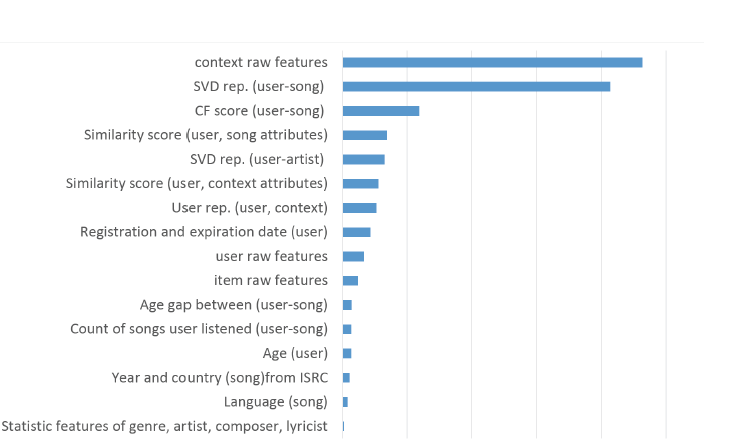
由于用户和歌曲之间的强烈互动，我们当然应该派生基于交互的特性。然而，由于用户和歌曲的维度很大，我们无法对用户或歌曲通过一个热编码的行为。这需要我们为用户、歌曲和用户歌曲对找到适当的表示。在推荐区，基于邻里的Top-N协同过滤和基于模型的矩阵分解（mf）是两个强大的家族，我们可用于生成功能。基于项目的CF将计算每个项目的前n个最相似项基于相似性度量（例如余弦相似性、Jaccard相似性等）作为那个项目的邻居。然后它将计算每个基于用户购买的内容和项目的邻居。同样，基于用户的CF将构造每个用户的前n个最相似的用户作为其邻居用户。另一方面，矩阵因式分解方法同时考虑了用户和物品进入一个低维度的潜在空间。在那个空间里，用户将接近他购买的物品。最多的有效的MF方法是奇异值分解（SVD）

SVD潜空间的维数k是一个超参数。我们通过交叉验证进行评估：在最终结果中，选择k= 30。利用方程计算的隐向量表示我们生成一个30维用户表示和一个30维歌曲表现。我们也计算点这两个表示的乘积表示用户歌曲对。

将用户表示的各个维度和歌曲维度相乘表示，以生成另一个30维用户歌曲成对表示。这些特征可以表明有多相似在每一个潜在空间维度的用户和歌曲。

同时，类似上面所提到的用户与歌曲生成隐向量方法，解决方案中还有生成用户与歌手隐向量；方法类似，不再说明。从artist属性中，通过分隔符（, | \\ & \ / + ; ，feat）分割歌手。一首歌可以有多个歌手。

## 重要特征：



（通过运行lightgbm程序，统计得出，不加仔细分析）

# 总结：

特征工程是一个非常重要和不可替代的预测分析项目的组成部分。尤其是当数据集的大小不是很大，特征工程非常在大多数情况下有助于提高性能。一些功能从特征工程没有帮助，甚至没有伤害模型的性能。显然，什么时候数据集很大，时间成本可能成为一个问题，因为

方法，我们生成特性，然后评估并可能保留它们。成本与绩效增长的平衡在这种情况下要仔细评估。

由于训练和测试集之间的分布不同，我们使用训练集和测试集上计算的特征，因此

实际上，将未来的信息泄漏到我们的模型中。这明显可能会影响测试的性能。例如，如果我们生成统计每首歌的用户数的功能，性能将增加验证集，但减少测试集合。原因是这个特性在验证和测试集明显不同。然而，我们必须减轻不同分配的影响处理特性工程过程时泄漏风险。到限制这种风险，我们需要格外小心以避免过度处理。