# 实验1 音乐流行度预测

2024年3月28日

截止日期：4月10日24:00

## 任务描述

构建线性回归模型，对音乐流行度进行预测。

## 数据集简介

人类与歌曲和音乐有着紧密的联系。音乐可以改善情绪，减轻疼痛和焦虑，为情感表达提供机会。研究表明，音乐可以在许多方面有益于我们的身体和心理健康。最近，多项研究已经开展，以了解歌曲和其受欢迎程度之间的关系。本数据集中，歌曲用多种指标度量对进行描述并记录为表格，任务是预测歌曲流行度。数据集包含18835个样本，每个样本包含13个属性与1个实值标记，属性包括连续数据属性和离散属性，属性信息见下表。要注意数据之间存在多重共线性，这增加了任务的挑战难度。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 变量名 | 属性/标记 | 连续/离散 | 类型 | 上下界 | 离散值个数 |
| Song\_name |  |  | string |  |  |
| Song\_popularity | Label | Continuous | int | (0,100) |  |
| Song\_duration\_ms | Attribute | Continuous | int | (12k,1.8m) |  |
| acousticness | Attribute | Continuous | double | (0,1) |  |
| daceability | Attribute | Continuous | double | (0,0.99) |  |
| energy | Attribute | Continuous | double | (0,1) |  |
| instrumentalness | Attribute | Continuous | double | (0,1) |  |
| key | Attribute | Discrete | int |  | 12 |
| liveness | Attribute | Continuous | double | (0.01,0.99) |  |
| loudness | Attribute | Continuous | double | (-38.8,1.58) |  |
| audiomode | Attribute | Discrete | int |  | 2 |
| speechiness | Attribute | Continuous | double | (0,0.94) |  |
| tempo | Attribute | Continuous | double | (0,242) |  |
| timesignature | Attribute | Discrete | int |  | 6 |
| audiovalence | Attribute | Continuous | double | (0,0.98) |  |

**目标**：根据歌曲指标预测歌曲流行度。

## 教学要求

1. 掌握线性回归模型的原理与构建
2. 掌握数据集预处理的常用方法，包括属性转换、归一化、共线性（multicollinearity）的检测与处理等。其中关于共线性要求自学，可参考 [1] [2]。
3. 掌握常用评估方法（本实验中为最基本的单次留出法）
4. 掌握实验结果的展示方式

## 实验要求

1. 使用Python编程构建线性回归算法，预测音乐流行度。
2. 数据集预处理：
   1. 离散属性连续化
   2. 连续属性归一化normalization
   3. 共线性的检测与处理
3. 使用80%的数据集进行训练，20%的数据集进行测试。需要报告训练误差和测试误差。请注意，需要保证模型能够在测试集上进行准确的预测。
4. 使用可视化技术来展示你的模型和数据集。可以使用Python的Matplotlib库来完成该任务。（可选）

* 对数据集的展示，比如，属性值的分布情况、属性与标记之间的相关性等。
* 对模型的展示，比如在二维空间中展示模型对应的线

1. 报告模型的性能和准确率。可以使用Python的Scikit-Learn库中的mean\_squared\_error函数来计算均方误差（MSE）。
2. 提交代码与报告，其中报告内容包括以下几个部分：
   1. 预处理
   2. 实验设置，包括实验评估方法与性能度量
   3. 实验结果，包括模型性能，可视化结果展示
   4. 结果分析
   5. 代码附录

## 提交要求

1. 提交的文件为压缩文件，其中包含：.py代码文件或者.ipynb（不接受其它后缀的代码文件）、实验报告（doc或.docx），可能包含：模型文件（训练代码和预测代码分开的话需要）、运行脚本（视自己情况而定）
2. 提交的报告为doc或docx，代码需要复制后放在附录中。
3. 提交的代码需要一键运行，或者命令行（脚本）直接运行。（脚本运行需要注意路径）
4. 模型使用pickle库保存。用法参照：

<https://blog.csdn.net/vincent_duan/article/details/121276754>

1. 不要包含太多库导致批改设备运行时报错（没用到的库尽量不要导入）。

参考文献

|  |  |
| --- | --- |
| [1] | 流病与统计, “一看就懂的多重共线性,” [联机]. Available: https://zhuanlan.zhihu.com/p/355241680 |
| [2] | 喀喀湖, “线性回归的决定系数,” [联机]. Available: https://blog.csdn.net/weixin\_44005185/article/details/121273629 |

## 附录 关于共线性问题的相关说明

共线性检测是回归问题中一个重要的问题，可以使用多种方法进行检测，以下是一个使用Python语言实现的基于方差膨胀因子（VIF）的共线性检测代码：

import pandas as pd

import numpy as np

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

# 假设数据存在一个名为"df"的DataFrame对象，其中包含四个属性x1、x2、x3和x4。

# X为输入属性，y为输出属性

X = df[['x1', 'x2', 'x3', 'x4']].values

y = df['y'].values

# 计算VIF

vif = pd.DataFrame()

vif["variables"] = ['x1', 'x2', 'x3', 'x4']

vif["VIF"] = [variance\_inflation\_factor(X, i) for i in range(X.shape[1])]

# 输出结果

print(vif)

在该代码中，使用了statsmodels库中的variance\_inflation\_factor函数来计算VIF。如果任何一个属性的VIF值大于5，就可以认为存在共线性问题。

共线性可以影响模型的稳定性和可靠性，因此需要对其进行处理。以下是一个基于主成分分析（PCA）的共线性处理代码，用于去除共线性：

import pandas as pd

import numpy as np

from sklearn.decomposition import PCA

# 假设数据存在一个名为"df"的DataFrame对象，其中包含四个属性x1、x2、x3和x4。

# X为输入属性，y为输出属性

X = df[['x1', 'x2', 'x3', 'x4']].values

y = df['y'].values

# 使用PCA进行特征降维

pca = PCA(n\_components=X.shape[1])

X\_pca = pca.fit\_transform(X)

# 输出每个主成分的贡献率

print(pca.explained\_variance\_ratio\_)

# 输出每个主成分的权重

print(pca.components\_)

# 输出降维后的属性矩阵

print(X\_pca)

在该代码中，使用了sklearn库中的PCA类进行特征降维处理。通过调整n\_components参数，可以选择需要保留的主成分数。一般来说，可以保留贡献率大于某个阈值的主成分，例如0.9或0.95。通过输出每个主成分的贡献率和权重，可以对降维后的属性矩阵进行解释。

PCA（主成分分析）的主要目标是找到可以最大程度解释数据方差的线性组合。在PCA中，保留主成分的原则通常有两种方式：

1. 保留方差解释率（explained variance ratio）大于某个阈值的主成分。方差解释率是每个主成分解释的数据方差占总方差的比例。因此，保留方差解释率大于某个阈值的主成分可以尽可能多地保留数据的信息，同时减少特征的数量。
2. 保留主成分的累计方差解释率（cumulative explained variance ratio）大于某个阈值。累计方差解释率是前k个主成分的方差解释率之和。这种方法可以在保留足够多的数据信息的同时，减少主成分的数量。

在选择保留主成分的方法时，需要根据具体问题和数据集的特点来决定。一般来说，可以通过绘制方差解释率或累计方差解释率的图像来帮助确定保留的主成分数量。

方差解释率是主成分分析（PCA）中用来衡量每个主成分对总方差的贡献程度的一个指标。具体地，方差解释率是每个主成分解释的数据方差占总方差的比例。

在PCA中，主成分按照解释方差的大小进行排序。第一个主成分解释的方差最大，第二个主成分解释的方差次之，以此类推。方差解释率表示每个主成分解释的方差占总方差的比例，通常使用百分比表示。

方差解释率越高，说明相应的主成分解释了更多的数据方差，也就是说，这个主成分包含了更多的信息。因此，我们可以通过方差解释率来选择保留哪些主成分，以达到降维的目的，并尽量保留数据的信息。

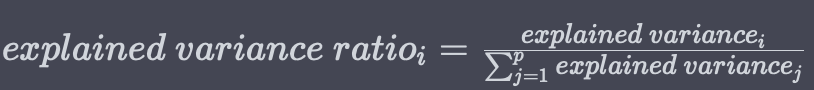
可以使用Python中的PCA类计算每个主成分的方差解释率。例如，在scikit-learn库中，可以使用pca.explained\_variance\_ratio\_属性来获取每个主成分的方差解释率。

主成分解释的方差是指每个主成分能够解释的数据方差。在主成分分析（PCA）中，每个主成分是原始属性的线性组合，可以通过计算它们的方差来衡量它们对数据的解释能力。

设第i个主成分的方差为$Var(z\_i)$，则第i个主成分解释的方差为：



其中，p是原始属性的数量，$\sum\_{j=1}^p Var(z\_j)$表示所有主成分的方差之和。方差解释率（explained variance ratio）是第i个主成分解释的方差占所有主成分解释方差之和的比例，即：



方差解释率衡量了每个主成分对总方差的贡献程度，是主成分的重要性指标。在选择保留哪些主成分时，可以根据它们的方差解释率进行决策。