# 基于哔哩哔哩视频网站的热门视频分析

**作者**：廖诗云、黄果、张莉玲、杨春华、程合晴、冯永

**单位**：重庆大学，计算机学院

**案例版权**：该案例归重庆大学计算机学院所有

**涉及的知识点**：数据采集、数据预处理、无监督学习、有监督学习

**案例来源及案例真实性情况**：该案例来源于重庆大学计算机学院专业硕士（电子信息）课程《大数据架构与技术》中的学生精选汇编课程设计。

**摘要** 随着便携式摄影设备、视频剪辑技术的普及，越来越多的“短视频”风靡网络。相对于传统视频类型，“短视频”时间短，制作相对简单，发行方便，极易观看和传播。近年来，在哔哩哔哩视频网站等媒体平台上产生了大量具有社会影响力的短视频作品。哔哩哔哩视频网站每日有数以万计的短视频发布，但成为热门的短视频是为凤毛麟角。本案例对热门视频进行数据分析，总结其中的共通点。本案例核心内容有：（1）从哔哩哔哩视频网站中随机爬取视频的信息；（2）基于爬取到的视频信息，从众多视频中筛选出热门视频；（3）对筛选出的热门视频做分析，探寻其共通点。

**关键词**：数据采集、数据预处理、无监督学习、有监督学习、词频统计、可视化

## 1 引言

该教学案例来源于重庆大学计算机学院专业硕士（电子信息）课程《大数据架构与技术》中的学生精选汇编课程设计。该案例的关键问题为如何衡量一个视频是否热门以及如何分析热门视频。需引导学生进行的主要内容有：（1）从 Bilibili 视频网站中随机爬取视频的信息；（2）基于爬取到的视频信息，从众多视频中筛选出热门视频；（3）对筛选出的热门视频做分析，探寻其共通点。

## 2 背景介绍

随着便携式摄影设备、视频剪辑技术的普及，越来越多的“短视频”风靡网络。相对于传统视频类型，“短视频”时间短，制作相对简单，发行方便，极易观看和传播。近年来，在哔哩哔哩视频网站等媒体平台上产生了大量具有社会影响力的短视频作品。哔哩哔哩视频网站每日有数以万计的短视频发布，但成为热门的短视频是为凤毛麟角，而能够拥有百万粉丝的UP主则更是稀缺。

随着大数据时代的来临，任何事物的发展背后规律都可以通过数据进行深层探索，在本案例中，我们选取哔哩哔哩视频网站的热门视频作为分析对象，以寻找热门视频背后的共通点。

## 3 内容

该案例的主要内容主要分为三个小节，分别为1）从哔哩哔哩视频网站中随机爬取视频的信息并进行数据存储（2）数据预处理；（3）基于爬取到的视频信息，从众多视频中筛选出热门视频；（4）对筛选出的热门视频做分析，探寻其共通点。

3.1 数据采集

（1）数据爬取

Scrapy是一个为了爬取网站数据，提取结构性数据而编写的应用框架。但是

不适用于大规模的分布式应用，因此我们通过改变了Scrapy的队列调度，将起始的网址从start \_urls里分离出来，改为从Redis读取,多个客户端可以同时读取同一个Redis,从而实现了分布式的爬虫，Scrapy内部执行流程如下图所示。

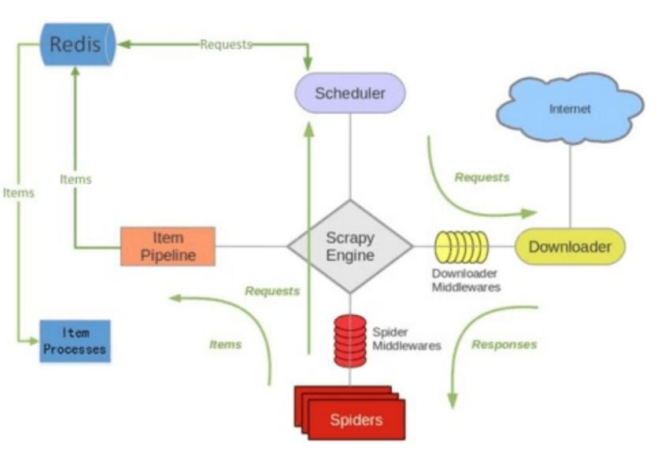


图1 Scrapy执行流程

本案例的爬虫对象为：（1）哔哩哔哩某分区视频的信息，包括视频标题、点赞数、投币数、收藏量、视频时长、观看量等；（2）每日排行榜更新的视频信息，包括标题、点赞数、投币数、收藏量、视频时长。

我们一共爬取了30多万条的视频信息，所有视频信息可在数据集文件下获

取。数据集的格式如下图所示，该图表示了每个视频包含的属性特征。我们主要

使用的信息的有：标题：“title”、时长：“duration”、点赞数：“like”、投

币数：“coin”、收藏数：“favorite”。

1. 数据存储

使用 Mongodb+Redis 对收集到的数据进行存储。

1. 数据使用

处理环境：Spark&Hadoop分布式环境。

分析对象：使用算法从点赞、投币、收藏等信息找出热门视频，再对热门视频的特征进行分析，多角度寻找热门视频的共通点，得出制作热门高质量视频的启发。

3.2 数据预处理

由于获取到的数据共计30万条左右，属性较多，其中有些属性是我们不需要的，所以需要对数据进行清洗和预处理。考虑到主要是数据中特殊字符，以及特定列信息，我们使用正则表达式来完成清洗工作，将数据集保存在 MongoDB 中。

如图2所示，数据预处理这一部分主要是因为爬取下来的数据里面掺杂着很多无用的数据，比如缺失值、特殊符号、英文等等，这些我们无法处理。所以我们使用正则表达式来清理特殊字符，使用 lambda 来去除重复 ID数据为本次数据集预处理信息，具体预处理过程见下。

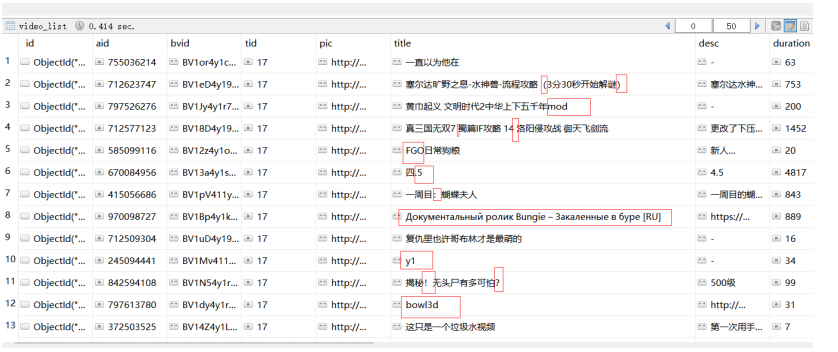


图2 数据集预处理信息

特征删除：考虑到有些特征与本次的研究主题无关（例如视频 id、视频

的具体网址等），所以将其删除。

数据清洗：很多样本的特征存在一些问题，例如很多标题下混入一些无

意义字符及一些特征乱码，不利于分析后面的热门视频，所以通过正则表达

式对存在乱码以及无意义字符进行删除并合并空格。

预处理后的数据即为我们想要的数据，如图3所示。

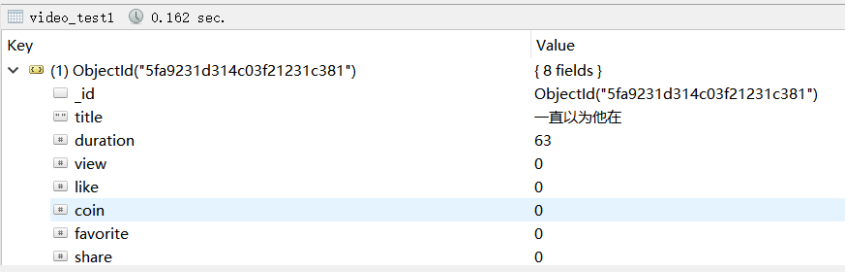


图3 预处理后数据显示

我们选取了其中特定的列，可以看到，title属性中的特殊字符已经被清

理掉了，效果图如下：

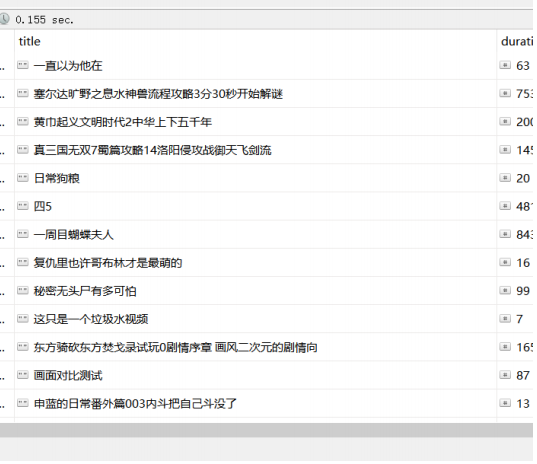


图4 预处理后 title 列信息

3.3 热门视频筛选

如上文所述，我们需要对数据集进行处理，筛选出热门视频，主要利用的是view、like、coin、favorite等属性进行筛选。筛选的方法本节将详细介绍，然后再对得到的热门视频进行分析处理，分析热门视频的相关属性。在此过程中使用的数据集主要有两种形式：一种是直接读取MongoDB到dataframe，还有一种是导出CSV文件格式。

（1）无监督学习

K-Means算法在选取第一个点时为随机选取，之后则是通过计算每个样本与当前已有类聚中心最短距离，这个值越大，表示被选取作为聚类中心的概率较大。最后，用轮盘法选出下一个聚类中心，直到选出k个聚类中心。在实验过程，首先给定了K的一个范围，在此范围内进行迭代，迭代最大值用户自己设置，通过不停的迭代，将训练数据更好的分为K类；计算不同K在迭代过后的损失函数，绘制损失结果图，我们可以发现，在K=3的时候，函数第一次发生变化，也就是说，K最优取3。

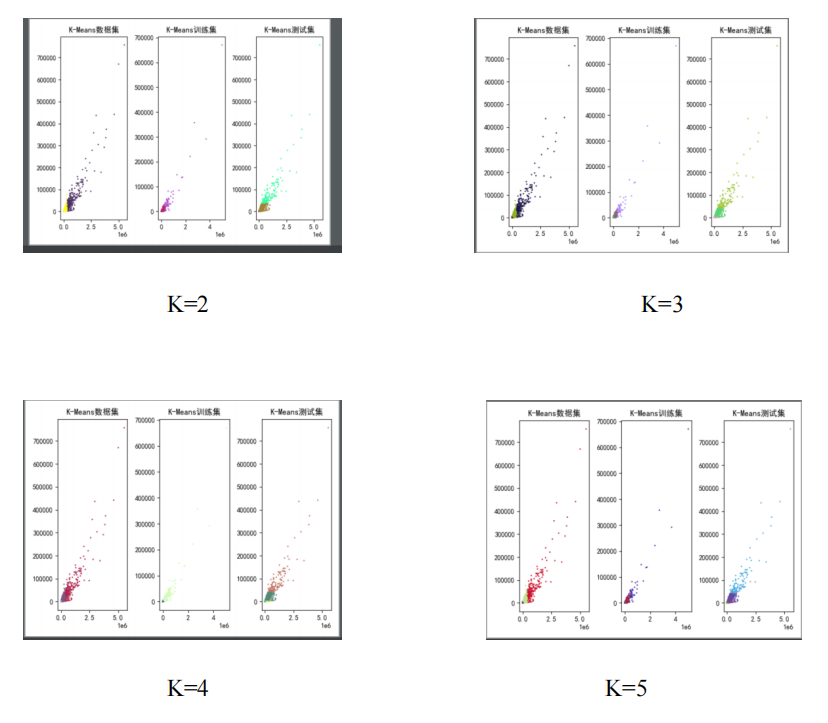
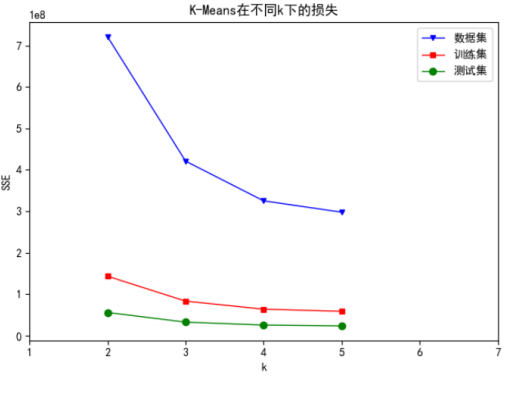


图5 KMeans聚类算法

K 不同时，K-Means 的损失如图所示：

图6 KMeans在不同K值下的损失

使用 K-Means后共筛选出667条热门视频，数据聚类成了三类，分别是热门视频、中性视频、冷门视频。为了更好地与有监督学习进行比较，我们使用CSV保存结果，并输出 type 属性，其中0 代表冷门，1 代表中性，2 代表热门。

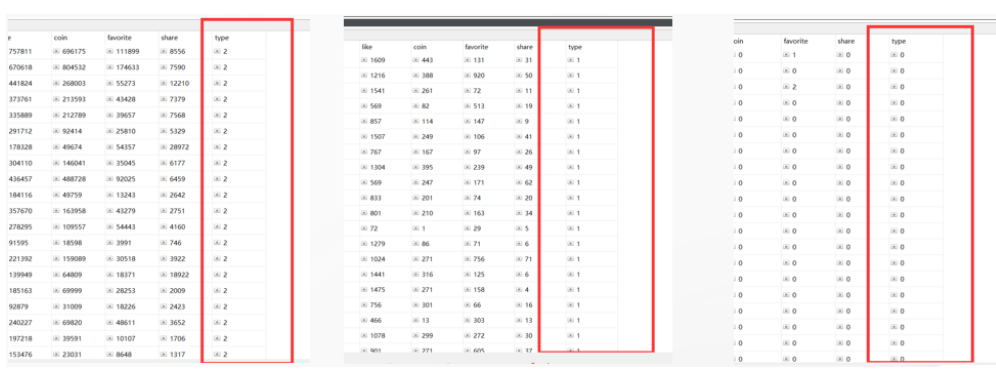
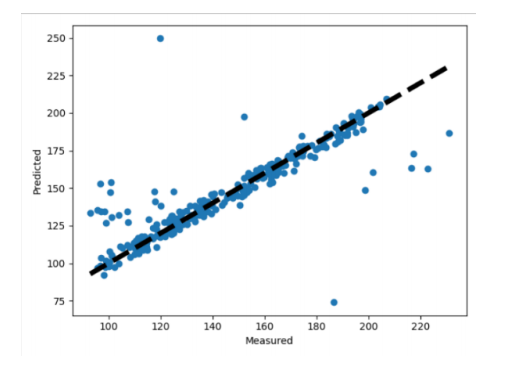


图7 视频数据结果

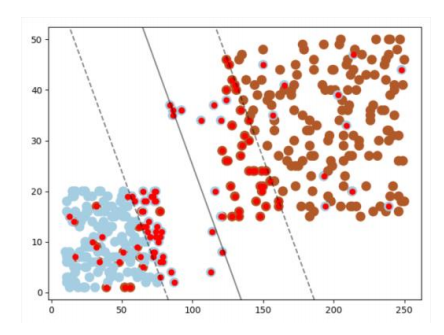
由于数据在处理时是线性可分的，故在无监督学习过程我们采用了KMeans 算法，实验证明，这个算法也是比较适用这个情况，问题在于K-Means在迭代聚类时，花费的时间比较大，所以下面我们进行有监督学习实验，来进一步处理，训练数据来源于B站的每日更新热门榜单。

（2）有监督学习

使用训练好的线性分类器对爬取的数据进行筛选，按照哔哩哔哩官方榜单的标准，选取作为筛选阈值，共筛选出880条热门视频用训练好的SVM分类器对爬取的视频信息进行筛选，共筛选出548条热门视频训练集训练好的线性分类器。

图8

训练集训练好的SVM 分类器

图9

3.3热门视频分析

通过前期的筛选，我们已得到所有的热门视频信息。为了寻求这些视频之间的共通点，下面，我们选取了标题以及播放时长进行分析，以此来判断热门视频在标题和播放视频的有何相似，为今后想成为 up 主拍视频提供一些方向建议。

（1）热门视频标题词频统计

通过对热门视频标题词频统计，其K-Means的统计结果如下图所示，其中，热门视频标题含“好吃”、“美食”、“简单”的次数达到最高，分别在50 次、

30 次和 28 次之间。

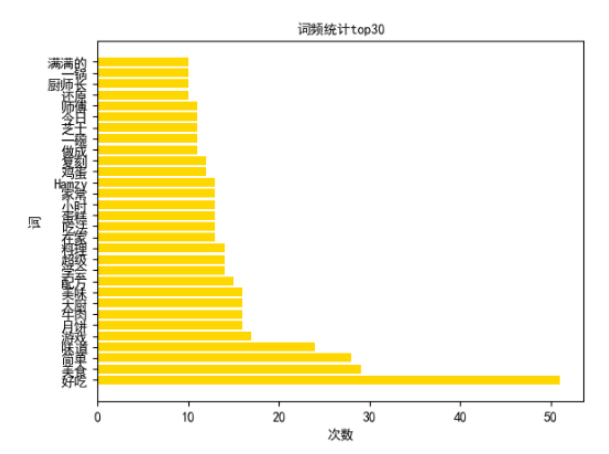


图10 K-Means筛选后词频统计结果

而对SVM方法筛选的热门视频标题词频统计，统计结果如下图11所示，仍是“好吃”、“简单”、“美食”占据词频前三，但是较之KMeans词频统计结果， 前三词频的位序发生变化，并且“好吃”该词频数更高，为86次。

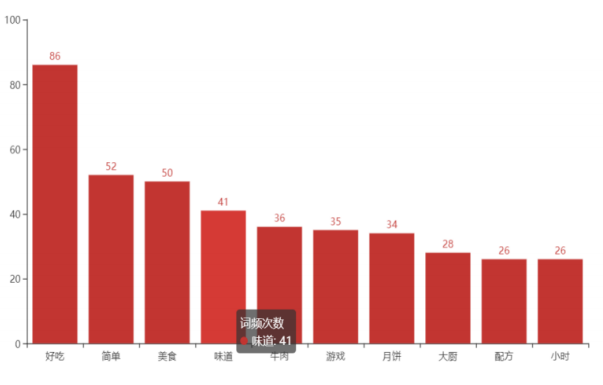


图11 SVM筛选后词频统计结果

（2）热门视频播放时长分析

图 12为通过KMeans分类出的视频时长统计。由图可得，冷门视频小于2分钟的数量远远大于热门及一般视频，而多于11分钟的视频热门视频也没有冷门视频及一般视频多，由此可得，2~11 分钟为制作视频的最佳时长。

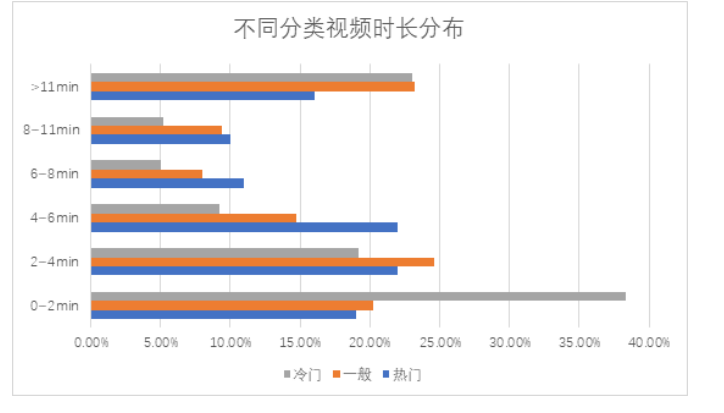


图12

图 13为通过SVM与线性回归统计出热门视频的时长对比，由图可得，两种算法统计的时长大致相同，总体上可得 SVM 统计出热门视频的时长稍微比线性分类时长大。

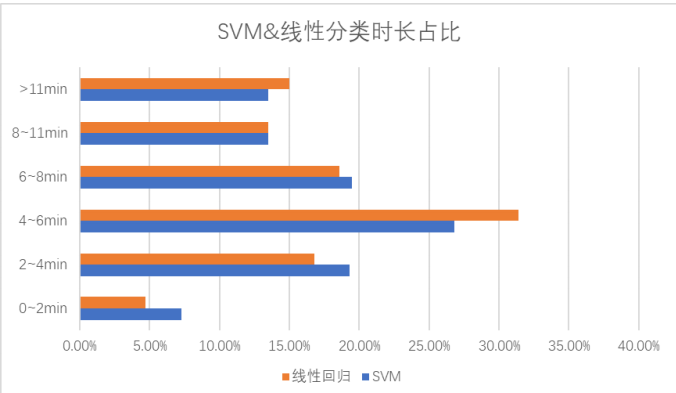


图13

## 4 小结

本案例通过筛选热门视频，进行热门视频标题词频统计以及播放时长分析，探讨了短视频时代下热门视频的共通点。本案例主要内容包括从哔哩哔哩视频网站中随机爬取视频的信息、筛选出热门视频以及热门视频分析。该案例主题新颖，结合了当下火爆的短视频背景，并使用了数据挖掘方法，可以充分增强学生的实践能力与理论基础。另外，本案例的内容仅为指导性的过程，在实际教学中，可保持基本研究内容不变，鼓励学生引入其它的数据预处理、数据挖掘、机器学习方法完成任务。

## 附录

1. 本案例提供配套的PPT、视频、数据集与代码等，发布于Github，链接为：https://github.com/Wanghui-Huang/CQU\_bigdata。

2. 本案例涉及到数据预处理以及多种机器学习算法，建议使用python语言进行编写，推荐的工具包有pandas（数据读取与预处理库），scikit-learn（机器学习算法库），matplotlib（可视化绘图库）。

3. 本案例参考文献如下：

[1] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016.

参考内容：数据预处理理论，KMeans，SVM

1. 吕云翔,张扬编.Python网络爬虫与数据采集[M].人民邮电出版社,2021.

参考内容：数据采集理论