电影票房预测系统设计方案

整体方案

系统的构建共分为 4 步：

1. 数据的获取，通过网络爬虫获取票房和电影特征
2. 测试集合和训练集合的划分
3. 特征的处理和选择
4. 学习回归预测模型，主要是模型的选择和超参数的选取
5. 不同模型结果的融合和最终结果的后处理
6. 预测电影票房并构建演示系统

详细设计方案

数据的获取

1. 豆瓣电影

通过爬虫获取静态网页，获取 2011.01.01~2015.11.31 的 437 部电影的票房数据和相应特征，主要的特征有：

- ✧ 电影的名称
- ✧ 电影的宣传海报
- ✧ 电影的简介，故事梗概
- ✧ 电影的类别
- ✧ 电影的演员、导演和编剧

- ✧ 电影的产地
- ✧ 希望观看此电影的人数
- ✧ 电影的片长
- ✧ 电影上映的日期
- ✧ 观众对电影的评价等信息

2. 百度索引指数

百度索引指数没有提供 API，而且百度索引指数使用动态加载技术，并通过 Html5 绘制索引信息，很难通过一般的静态网页爬虫获取。即使获取网页源码，由于使用 Html5 的 Canvas 进行绘制，仍然很难获取对应的数据，为此，我们采用如下方案：

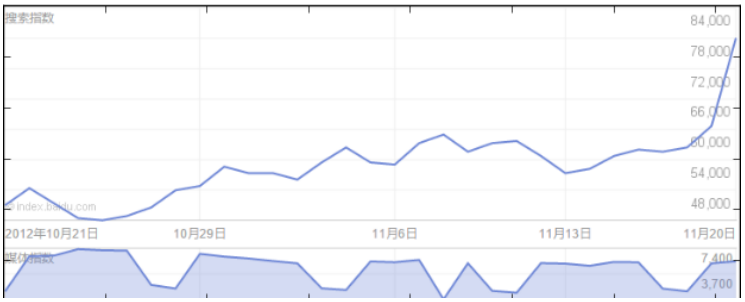
1. 调用浏览器的驱动渲染百度索引指数的页面，然后通过网页截图的方式获得百度索引指数的图片，如下图所示

由于百度有防止 IP 过度访问的限制，我们通过爬虫获取 100 多个 IP 地址，建立代理 IP 池；每隔一定频率更换一次 IP 地址，使用代理 IP 访问百度索引指数网页，突破 IP

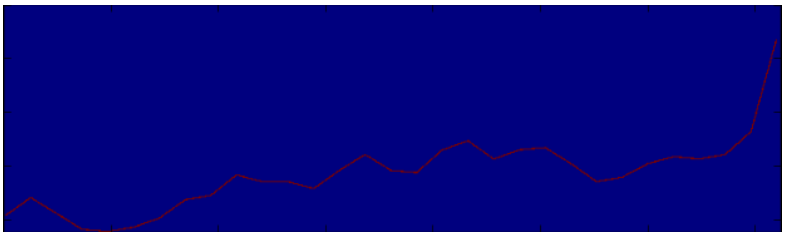
的限制

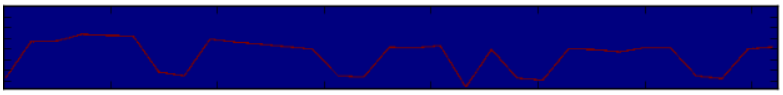
2. 关键区域的分割

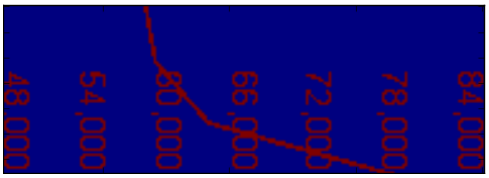
a) 首先获取关键区域，图中包括索引指数和新闻指数



b) 通过二值化，开闭操作，获取曲线以及对应的纵坐标区域，如图





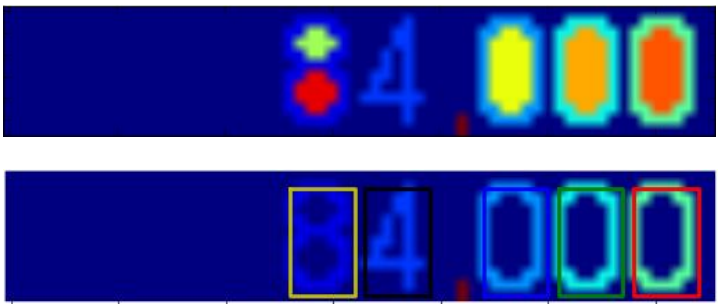


3. 数字的识别

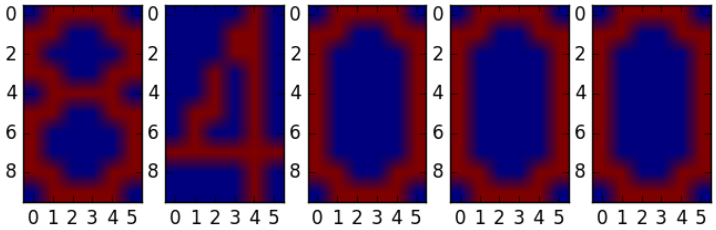
a) 数字的分割

b) 通过连通域算法并求取每个连通域的最小包围盒(去除面积过小的包围盒，并将过宽的包围盒进行分割)将数

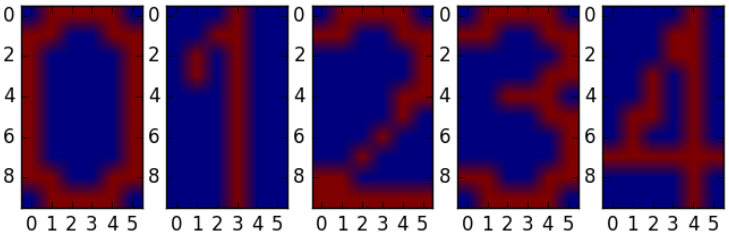
字分开，如图

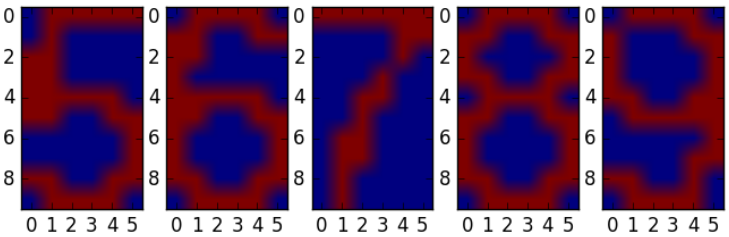


进一步得到

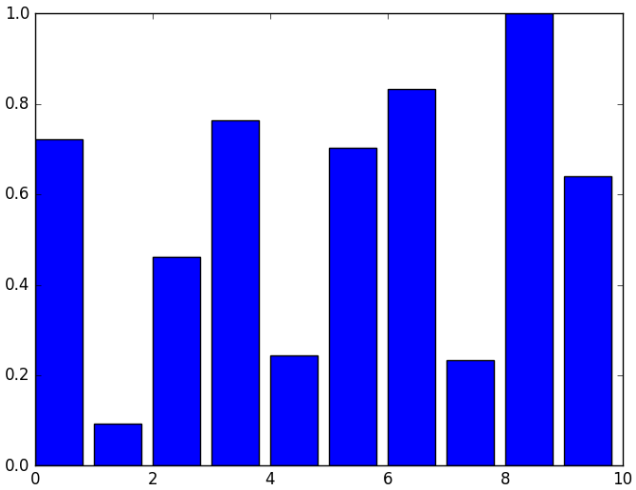


c) 对于单独的数字进行模板匹配得到数字的值，模板如下所示





对 b)中的第一个数字进行识别，得到如下分数，结果为 8



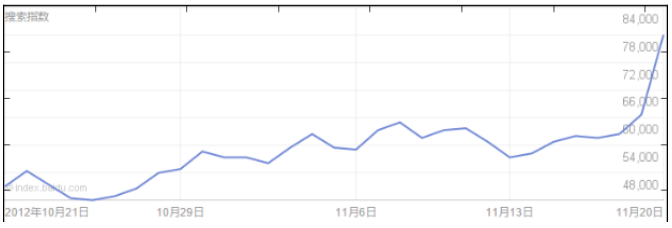
主要的方法是待识别数字与模板进行与操作，求取二者的交集占并集的比例，若分数小于阈值，则识别失败

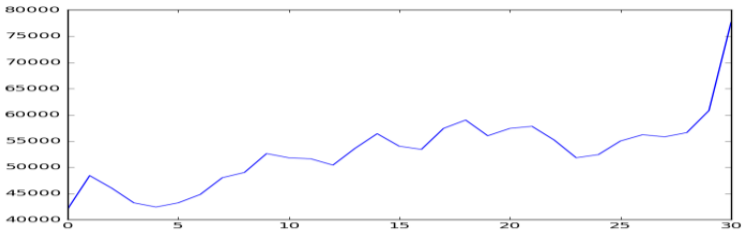
d) 对于 1 中的搜索指数，结果如下：

[84000, 78000, 72000, 66000, 0, 54000, 48000]

e) 对 d)中的结果进行对应调整，求得最稳定的间隔和最小值，得到：【6000, 42000】

4. 每隔一定间隔求取折线的高度，通过间隔和最小值计算出最终的实际值，如下所示





测试集合和训练集合的划分

采用 85%的数据作为训练集合，采用 15%的数据作为测试集合。对于训练集合，采用 10%的数据作为验证数据，90%用于训练数据，即 10-fold cross validation

特征的处理和选择

1. 数值型数据

   如果没有特殊说明，均直接进行归一化，方法如下

$$\bar{x} = \frac{x - min}{max - min}$$

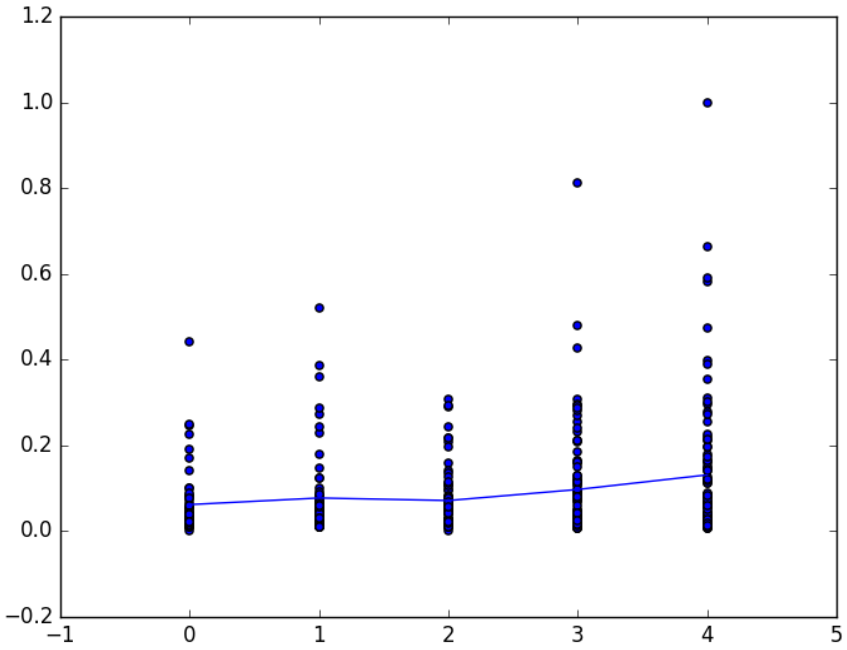
   其中，max、min 为训练集合中 x 的最大值和最小值，x 代表一维特征。对于票房直接除以最大值，而不做其他处理

2. 电影上映的日期

   在与时间相关的预测中，社会的发展、理念的变化、经济的增长、消费水平的提高和消费模式的变化是不得不考虑的因素，尤其是通货膨胀率的影响，可能会使本应该相同的票房产生巨大的差异。所以，在进行所有实验之前，我们首先对票房和时间的关系进行深入的分析，得到如下图所示结果。通过对均值进行线性回归，得到

$$y=0.01573x+0.04596$$

   R^2 score = 0.8307, 其中 x 代表年份减去 2010，y 代表票房。因此，我们将票房数据通过 $y' = (y - 0.04596)/x$ 进行变换。从而去除上述所有因素的影响。也正是因为如此，我们在测试集合和训练集合的划分过程中可以不考虑上映时间这一因素。在有限的未来，经过这样的处理并假设上式适用于之后的电影显然是合理的。



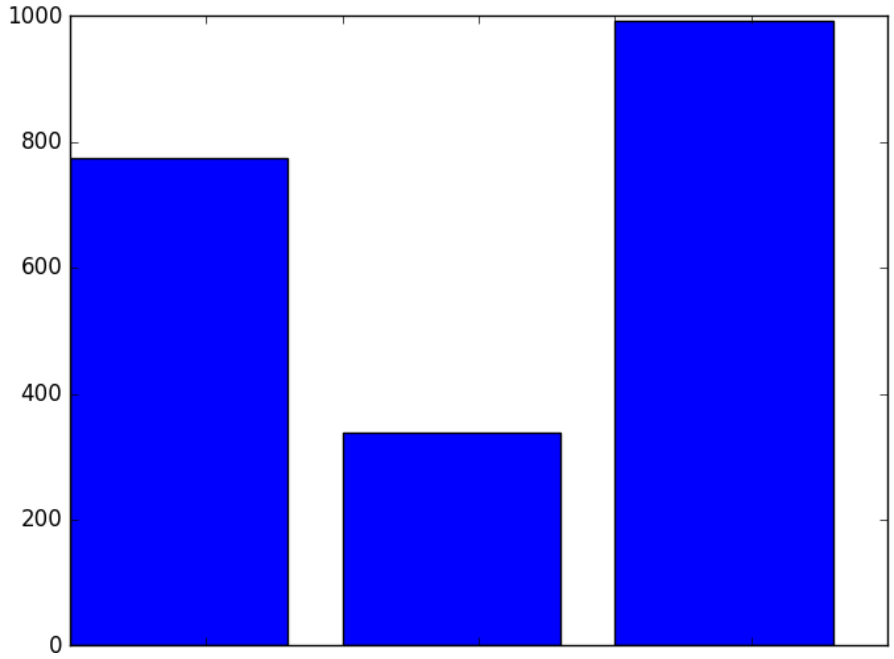
3. 电影的演员、导演和编剧

我们对演员、导演和编剧进行统计，发现在训练集合的所有电影中，演员、导演和编剧的数量与电影数量相近或远大于电影数量。由此可以看出，同一个演员、导演和编剧在不同电影出现的概率接近于 0。当然，这是由于样本数目较小的原因。虽然在本模型中我们没有使用演员、导演和编剧信息。但是，我们提出以下可能的处理方法，步骤如下：

a) 爬取演员、导演和编剧的获奖信息、搜索指数，观众的评价以及对应电影的评分

b) 通过利用上述信息，使用 K-means 或者谱聚类对其进行聚类

c) 使用上述聚类得到的类别作为演员、导演和编剧的特征向量进行后续的学习和处理工作



上图依次为演员、导演、编剧的数目

4. 电影的类别、电影的产地

a) 统计出现过的所有类别，类别数目记为 n

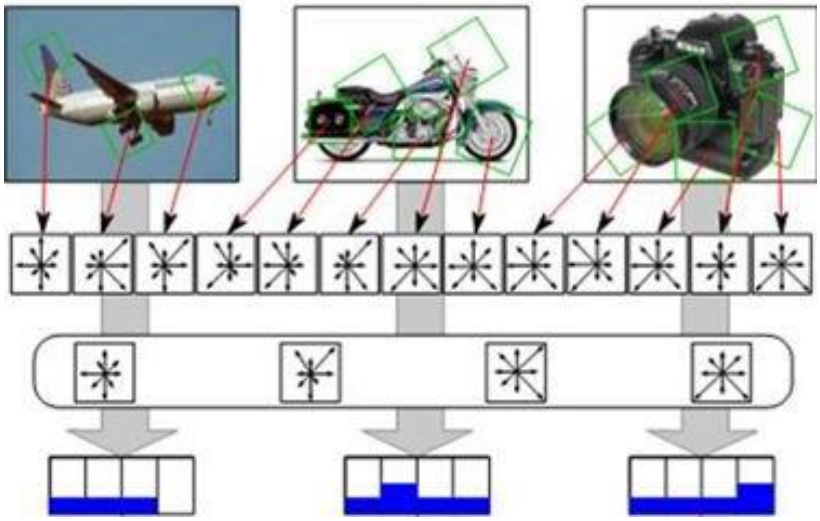
b) 每部电影对应一个 n 维的向量，每一位为 0 或者 1，表示是否属与当前编号对应的类别

5. 电影的宣传海报

我们利用基于内容的图像分析方法来分析海报的内容信息。为了方便地将海报因素与其他影响票房收入的因素结合在一起进行建模，我们利用视觉词袋(Bag-of-visual-words)模型将每一幅图像量化为一个高维度的特征向量。

a) 对于海报图像，提取 SIFT 特征。我们从所有 437 幅海报中提取出总共 326651 个 128 维的特征向量。

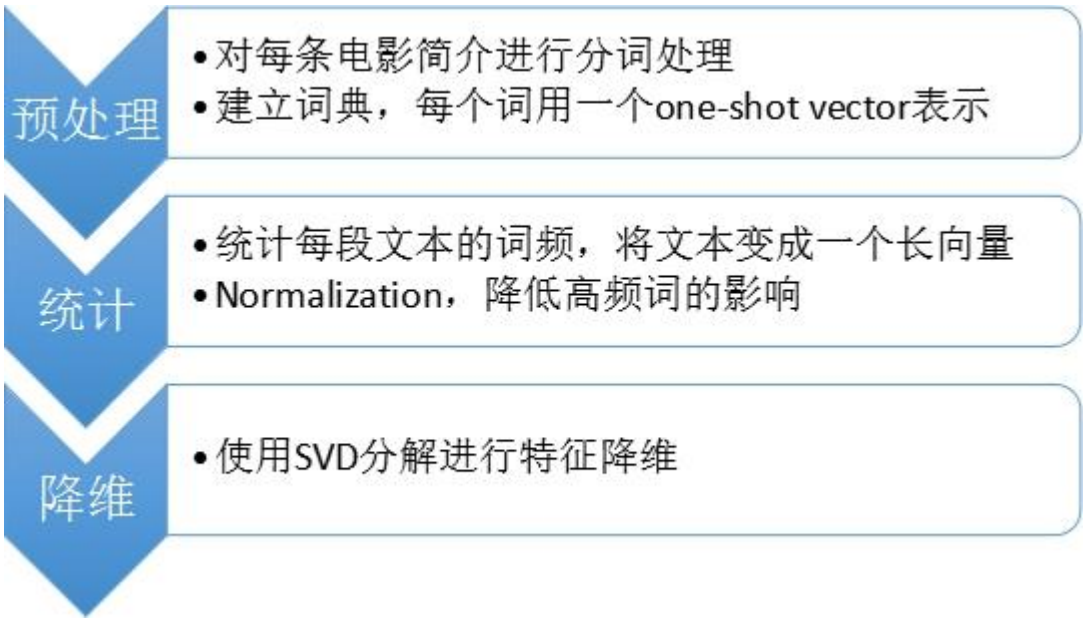
b) 利用视觉词袋模型得到每一幅海报图像特征表达。视觉单词以及视觉词袋模型的示意图如下：



我们利用 K-means 算法将提取的所有 326651 个特征向量聚成 50 个类，构成 50 个单词的视觉词典。每一幅海报图像被表示成视觉词典中单词的统计直方图的形式，也即每一幅图像被表示成一个 50 维的特征向量。再对特征进行归一化处理，使得特征向量中的每个元素位于 0 和 1 之间。

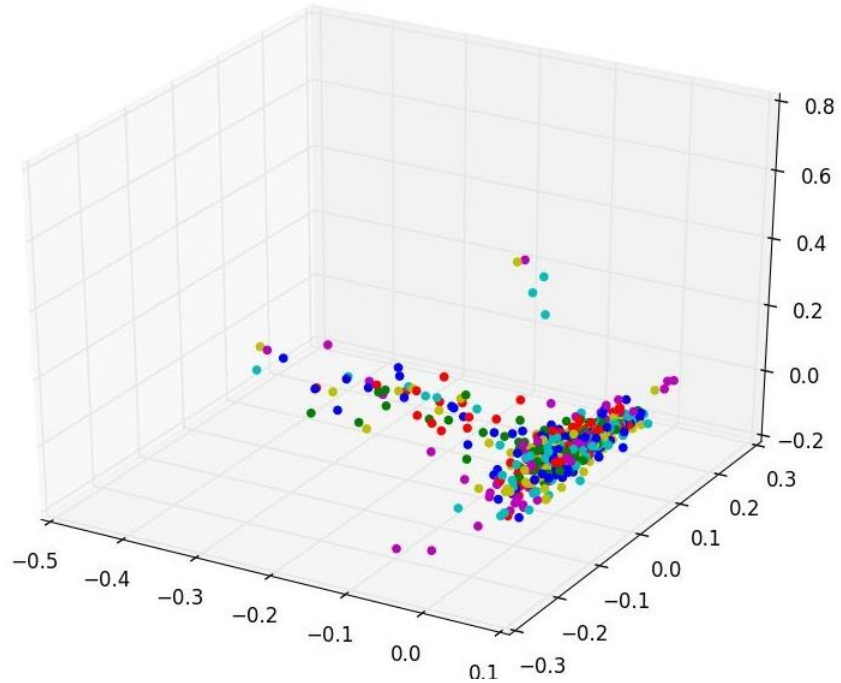
6. 电影的名称，电影的简介，故事梗概

我们使用 one gram 处理上述特征，即把每个文档中的词频直接进行统计，而忽略单词的顺序信息，我们有几百条电影数据，一共包含上万个单词，于是每个文档就成为维度为上万的向量。然后使用特征降维的手段，将上万维空间的长向量映射到几百维的短向量。具体的流程如下：



每个文本都映射到对应的向量之后，就可以作为机器学习算法的输入，进行有监督的回归模型训练。

通过处理，我们发现上述文本信息与电影票房没有很高的关联性，故在后续过程中不再考虑，如图所示



学习回归预测模型

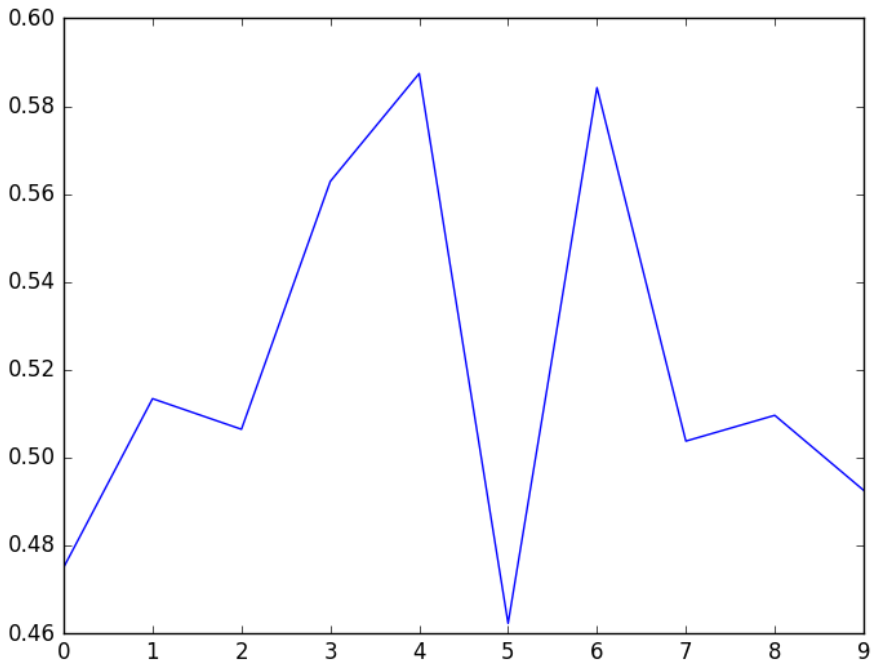
- ✧ 模型的选择和超参数的选取

  使用 K-fold 进行交叉验证，对于每一个模型，获取 K 次训练后验证集合上的误差，选择均值、方差较小的模型或者超参数

- ✧ GradientBoostingRegressor

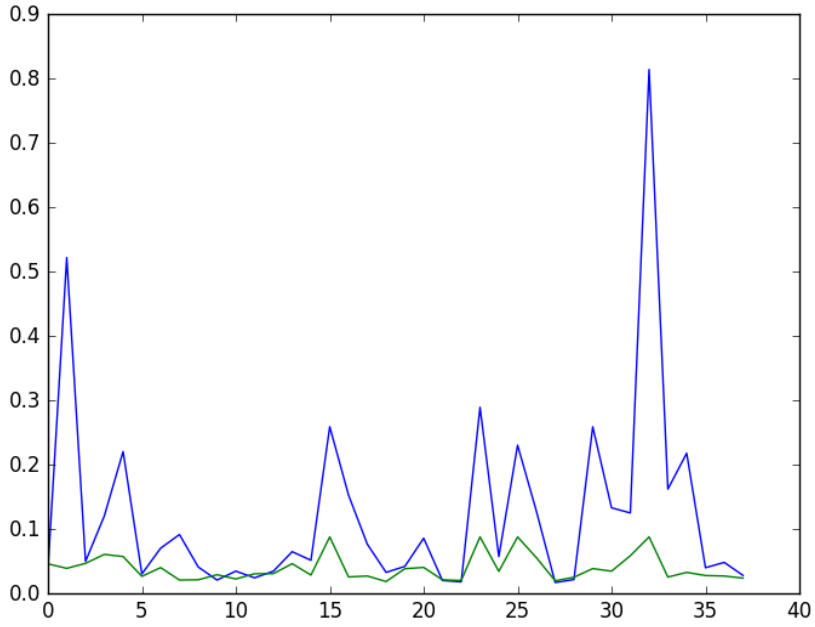
  特征使用百度搜索指数，使用 ensemble 家族中的 GradientBoostingRegressor 做为回归预测模型，参数使用网格搜索在验证集上确定，最终的参数为学习率设为 0.1，学习器个数为 25，目标函数为 $\text{obj} = \frac{abs(y-y\prime)}{y}$ 需要注意的是，为进一步抑制小票房数据的错误率，我们对样本采取不同的权重。具体公式如下：

  $W = \frac{1}{y}$ 上述模型在验证集合上的效果如下所示：



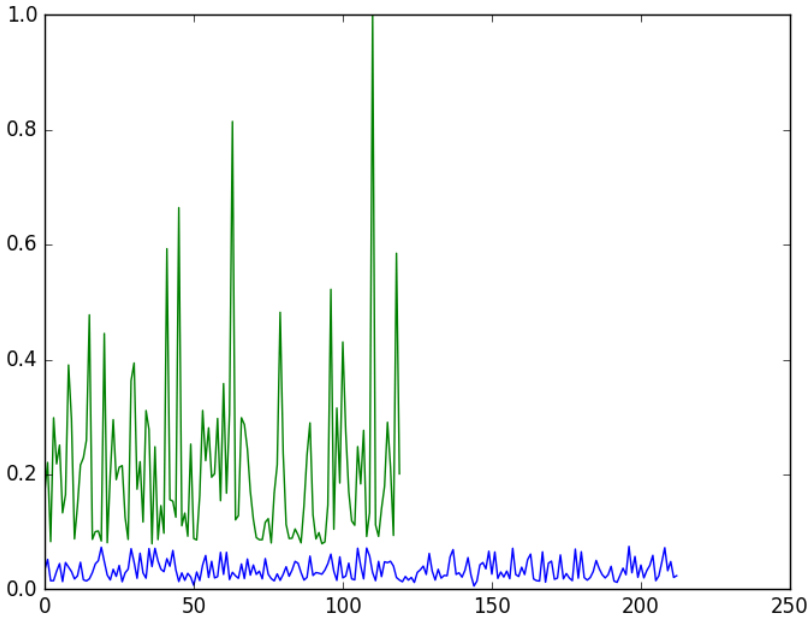
  均值为 0.52，标准差为 0.04

  预测结果如下所示：



✧ K-means 聚类 + 分类别回归

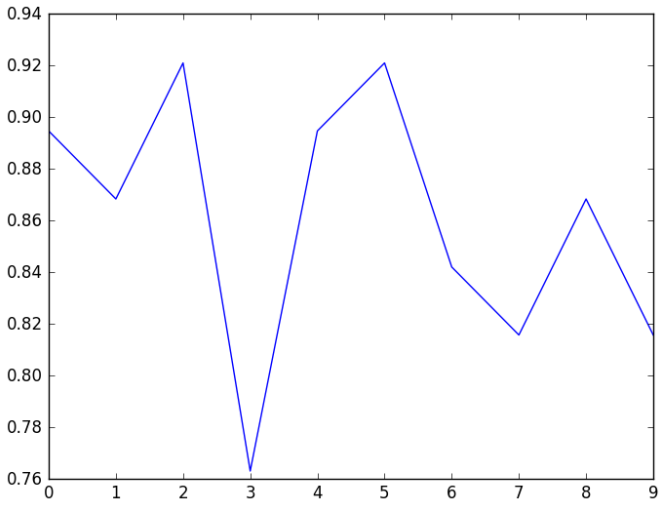
1. 使用 K-means 和默认参数对票房进行聚类，将票房分为高票房电影和低票房电影 2 类，聚类结果如下所示，



2. 从 1 中可以看出，两类数据的数目严重不平衡。这会使分类器倾向于预测较多的一类。未解决上述问题，我们使用 SMOTE 算法对数据量较少的一类数据进行上采样，使得二者数据量相

当

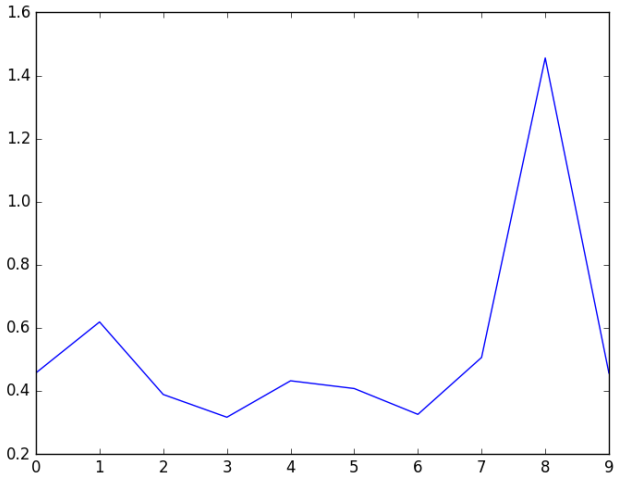
3. 利用上述扩增后的数据集的聚类后的类别作为标签，利用【电影种类，电影产地，电影时长，希望观看此电影的人数】作为特征，使用 AdaBoost 作为分类器进行分类。参数使用参数空间搜索的方法确定，学习器个数为 300。



均值为 0.86，标准差 0.05

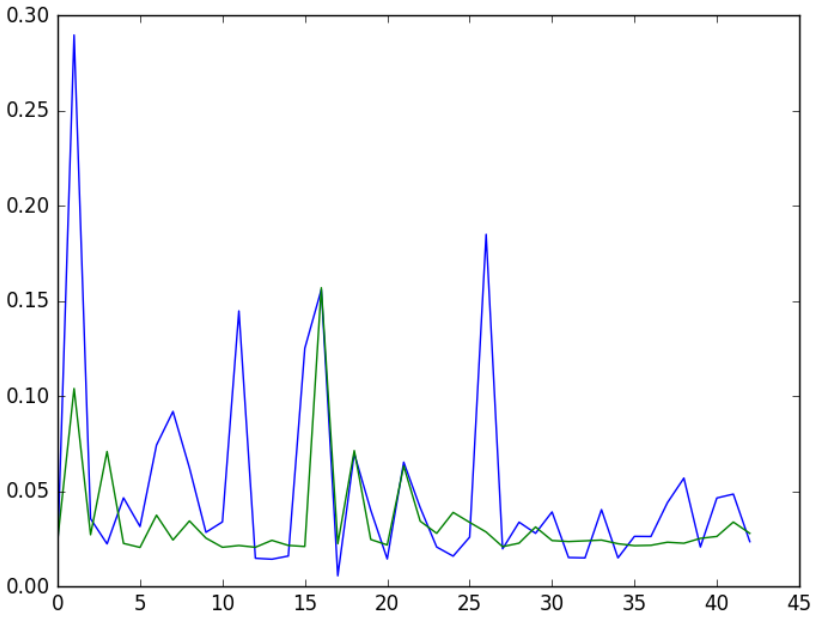
4. 根据聚类的结果分别对高票房和低票房的电影训练一个线性回归模型作为高票房和低票房的回归预测模型，特征为索引指数

回归模型 1：

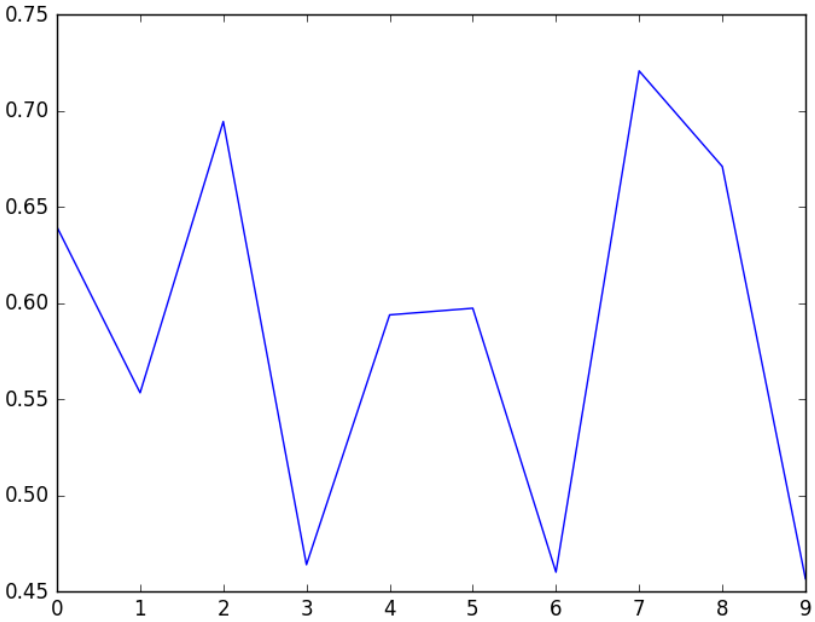


大多数情况下比较理想，均值 0.537，标准差 0.317

预测结果：

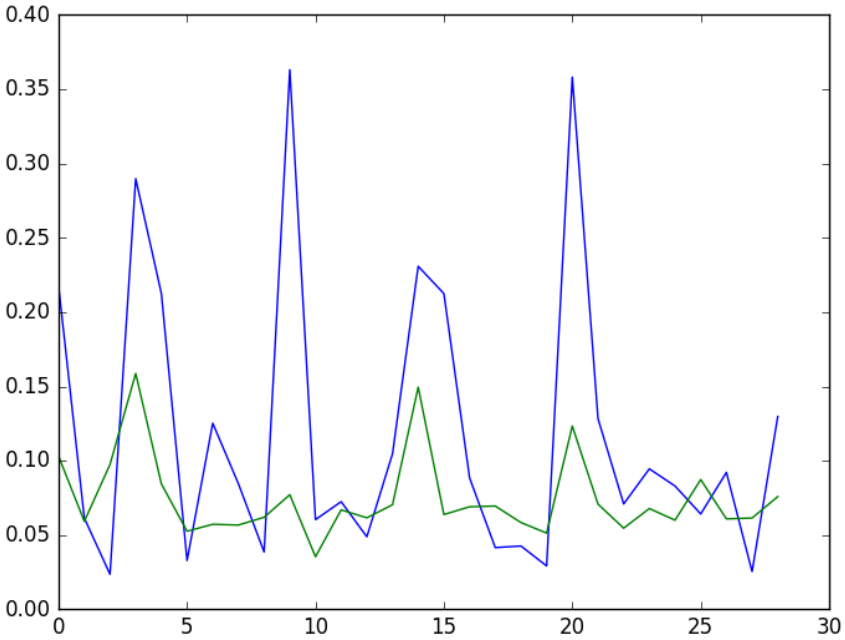


回归模型 2，此处预测结果的调整参数设为 0.5：



均值为 0.585，标准差 0.0942

预测结果：



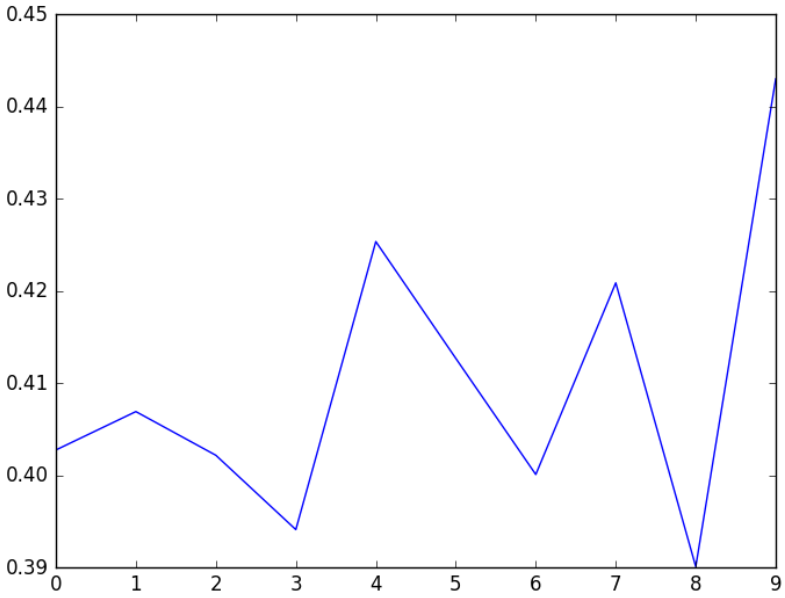
- ✧ 神经网络 MLP

  使用多层感知器对海报信息进行处理。由于训练数据有限，我们使用单隐层的前向反馈神经网络，隐层节点数 32，输出层为回归层

模型融合和后处理

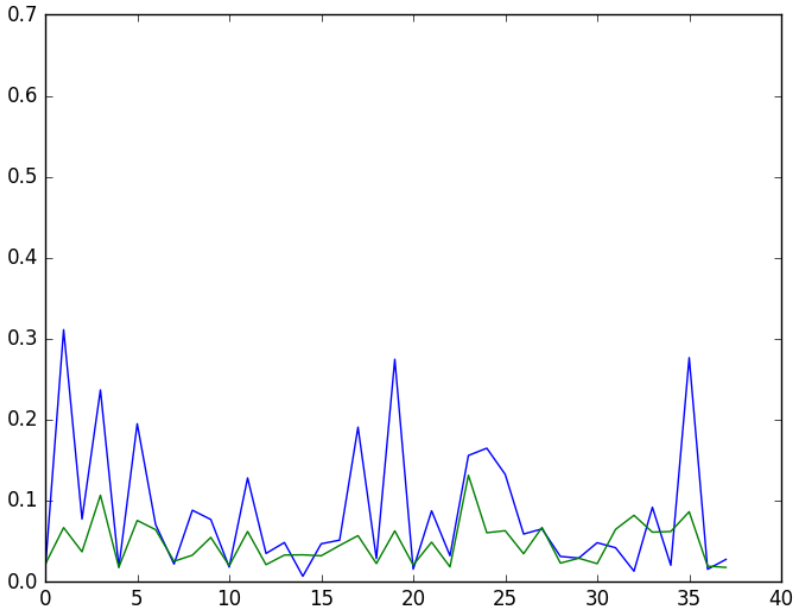
1. 我们从 boosting 算法中获得启示，最后使用加权平均的方式对不同模型的结果进行融合。加权的权重通过使用线性回归模型学习。

   结果如下所示：



均值为 0.41，标准差为 0.015

预测结果：



2. 最后，只需通过之前的公式$y' = (y - 0.04596)/x$ ，进行逆运算，即$y = y'x + 0.04596$，即可得到与时间无关的归一化票房。乘以缩放因子再进行合理的舍入操作，即将末尾的若干非零数字化为零，即可得到最后的预测票房

原型系统的构建

本原型系统提供命令行接口，输入为规定格式的电影信息文件，输出为电影票房预测结果，单位为万元，详细信息见演示系统说明文档