# 行政案件数据挖掘大作业

## 任务介绍

本项目爬取了中国裁判文书网与openLaw网站的裁判文书数据，进行数据清洗和处理，使用TF-IDF算法提取案件关键词，再结合数据统计分析得到的文书属性值，综合利用集合算法训练模型，分析出哪些因素对政府在行政复议中败诉的影响较大。同时，利用从案件中法院意见提取到的关键词，基于word2vec，建立了一个案件判决结果分类模型。

## 1.1 行政复议介绍

行政复议是指公民、法人或者其他组织认为行政主体的具体行政行为侵犯了其合法权益，依法向行政复议机关提出复查该具体行政行为的申请，行政复议机关依照法定程序对申请的具体申请行为进行合法审查、适当性审查，并作出行政复议决定的一种法律制度。

如果有以下情形的，可以申请行政复议：

１、行政复议审查的具体行政行为对公民、法人和其他组织合法权益造成的侵害的具体行政行为，都可以在行政复议中受到审查。

２、行政复议审查的部分抽象行政行为为行政复议的申请人，在提起对具体行政行为进行复议的同时，可以对该具体行政行为所依据的行政规定，提出进行审查的请求。

## 数据来源

## 2.1 数据网站

我们项目的数据源主要来自中国裁判文书网和openLaw网站。

裁判文书在互联网公开是2014年司法公开改革的重要风向标。2014年1月1日，《最高人民法院关于人民法院在互联网公布裁判文书的规定》（下称《规定》）的实施，标志着由最高院在全国的范围内统一组织裁判文书网上公开工作。值得注意的是，该《规定》第四条明确了除“涉及国家秘密、个人隐私”、“涉及未成年人违法犯罪”、“调解结案”、“其他不宜互联网公布”的五种裁判文书外，人民法院的生效裁判文书应当在互联网公布。

作为裁判文书网的补充，我们还使用了OpenLaw作为我们的另一个数据源，OpenLaw是一个面向律师、法官、检察官、法学教师、学者、学生以及从事法律相关的工作人员的 NGO 开放型组织，用户可以共同分享法律专业知识以及智慧和经验成果。这个网站最大的特点是添加了评价和批注功能，增加了司法审判的大数据研究报告，这些都对我们的数据分析提供了帮助。

## 2.2 数据维度

我们对收集到的数据进行整理，剔除无效数据后，整合得到了如下的几个维度，现在依次介绍他们的含义：

* 时间：按年划分，表示该文书的年份。
* 文书类型：表示该文书的类型，裁定书、判决书、通知书以及其他
* 省份：表示该文书所属省份，按省份简称划分。
* 判决结果：表示判决结果，分为胜诉或败诉。
* 庭审程序：表示该文书的庭审程序，分为一审、二审、再审等。
* 标题：字面意思，表示该文书的标题。
* 原告：表示该文书的原告，内容为姓名。
* 被告：表示该文书的被告，内容为姓名。
* 庭审过程：表示该审判的庭审过程。
* 法院意见：表示该审判的法院意见。
* 判决结果：表示该审判的详细判决结果，具有更详细的说明。

## 2.3 统计分析

我们抓取了2016年、2017年、2018年和2019年每年各1000个案件，总共4000个案件，案件中涵盖了各种类型，包括判决、裁定书，包括各省份的文书等等。剔除无效数据后，最终有效案件数为3937件。

首先我们确定了一部分能够直接通过统计分析得到的案件属性数据。由于使用的参数较多，在此，我们仅列举案件的部分特征。

### 2.3.1 文书类型特征

通过对各案件数据进行统计分析，可以发现不同的文书类型对应的胜诉、败诉结果是不同的，其中裁定书的败诉比例最高，约为33%，判决书的败诉比例相对较低，约为11%。在四年的过程中，裁定书败诉比例呈现出逐年递减的趋势。

由此可知，文书类型与判决结果之间存在着一定的相关性。

### 2.3.2 省份特征

通过对案件所属省份进行分类，我们统计了各省的败诉比例，绘制了如下的各省败诉比例统计图。从该图中可以看出，不同省份的败诉比例差距还是比较大的。但这并不是说败诉比例最高的陕西、宁夏与广西政府更容易败诉。观察数据，我们发现这些省份的案件信息明显少于其他省份，即统计样本数少，并不能完全客观地反映现实的判决情况。对于案件数目较多的北京市与广东省来说，两个地区的败诉率接近，都略低于20%。从以上分析，我们认为省份信息也可能是影响判决结果的重要因素。

### 2.3.3 法院分类特征

在数据中，存在着各个不同等级的法院，如下所示：

北京市高级人民法院

北京市第一中级人民法院

北京市高级人民法院

北京市房山区人民法院

山东省淄博市中级人民法院

北京市第三中级人民法院

北京市高级人民法院

北京市高级人民法院

北京市高级人民法院

北京市高级人民法院

北京市高级人民法院

北京市第二中级人民法院

江苏省南通市中级人民法院

上海市高级人民法院

为了方便后面的计算，我们将各法院按照层次划分为5个等级，作为法院分类特征：

（1） 中级人民法院；

（2） 高级人民法院；

（3） 最高人民法院

（4） 地区性人民法院

（5） 其他法院

### 2.3.4 原告分类特征

数据中，存在着如下明显不同的原告类别：

信阳市南湾湖风景区南湾街道办事处郑家冲九组

孙友

陈建元、周和余、余国雄、余秀宝

广东省韶关市人民政府

无锡市辉宏企业管理咨询有限公司

根据原告的信息，我们可以将其划分为个体、法人、集体三类不同的原告类型，作为特征。

### 2.3.5 被告分类特征

数据中，部分被告信息展示如下：

北京市怀柔区人民政府

北京市公安局

北京市通信管理局、中华人民共和国工业和信息化部

北京市公安局东城分局、北京市公安局

如皋市综合行政执法局、南通市城市管理综合行政执法局、如皋市民生住房开发有限公司

上海市徐汇区住房保障和房屋管理局、上海市住房和城乡建设管理委员会

上海市监狱管理局、上海市人民政府

信阳市人民政府

无锡市司法局、江苏省司法厅

中华人民共和国公安部

湘潭市公安局岳塘分局、湘潭市人民政府

通过分析，我们可以按照如下分类提取被告分类特征：

（1） 人民政府

（2） 公安局

（3） 税务局

（4） 其他局类部门（例如环保局，质监局）

（5） 街道办事处

（6） 各级委员会

（7） 人民政府与公安局

（8） 人民政府与其他

（9） 其余未分类

### 2.4 文本关键词提取

我们通过文本分类及分析，统计了所有行政复议政府**胜诉**的案例。为了更好地利用案件信息，我们利用TF-IDF算法，从文书内容中提取出：**征收、房屋、公积金补偿、行政诉讼、住房、法定、缴存**等关键词，各关键词在所有文书中的权重值如下：

征收 0.10249672233712526

房屋 0.06957056061323234

公积金 0.06720771036491174

补偿 0.059355922205223385

住房 0.0466519481406368

缴存 0.04272978754307336

对行政复议政府诉讼的文书进行关键词提取，**败诉**的关键词有：**征收、房屋、补偿、土地**等相关词汇。其中，各收集到的关键词在所有文书中的权重值如下：

征收 0.10947084143682247

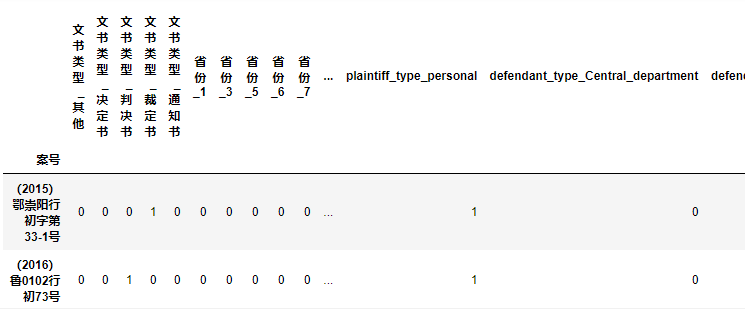
房屋 0.09000447721048628

补偿 0.07685444098719868

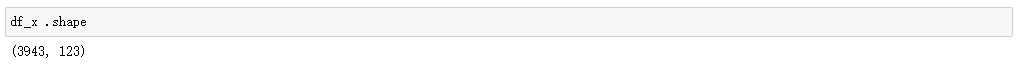
土地 0.03763413620783219

### 2.5 特征构成

将统计与关键词提取中总结的所有分类项均转化为如下的0-1变量



总计有123个特征，均为0-1变量



训练集为败诉胜诉的0-1变量



## 3 模型训练与分析

我们利用5种分类算法进行集成，利用KFold来避免标签泄露。

## 3.1 部分算法简介

### 3.1.1 xgboost算法

xgboost是一种集成学习算法，它是一个加法模型，基模型一般选择树模型，但也可以选择其它类型的模型如逻辑回归等。

xgboost属于梯度提升树(GBDT)模型这个范畴，GBDT的基本想法是让新的基模型（GBDT以CART分类回归树为基模型）去拟合前面模型的偏差，从而不断将加法模型的偏差降低。相比于经典的GBDT，xgboost做了一些改进，从而在效果和性能上有明显的提升。第一，GBDT将目标函数泰勒展开到一阶，而xgboost将目标函数泰勒展开到了二阶。保留了更多有关目标函数的信息，对提升效果有帮助。第二，GBDT是给新的基模型寻找新的拟合标签（前面加法模型的负梯度），而xgboost是给新的基模型寻找新的目标函数（目标函数关于新的基模型的二阶泰勒展开）。第三，xgboost加入了和叶子权重的L2正则化项，因而有利于模型获得更低的方差。第四，xgboost增加了自动处理缺失值特征的策略。通过把带缺失值样本分别划分到左子树或者右子树，比较两种方案下目标函数的优劣，从而自动对有缺失值的样本进行划分，无需对缺失特征进行填充预处理。

### 3.1.2 lightgbm算法

LightGBM也是GBDT算法的一个改进版本。LightGBM使用的是histogram算法，占用的内存更低，数据分隔的复杂度更低。其思想是将连续的浮点特征离散成k个离散值，并构造宽度为k的Histogram。然后遍历训练数据，统计每个离散值在直方图中的累计统计量。在进行特征选择时，只需要根据直方图的离散值，遍历寻找最优的分割点。

### 3.1.3 随机森林

随机森林是一个用随机方式建立的，包含多个决策树的集成分类器。其输出的类别由各个树投票而定（如果是回归树则取平均）。随机森林的生成过程如下：

1. 从原始样本中采取有放回抽样的方法选取n个样本；

2. 对n个样本选取a个特征中的随机k个，用建立决策树的方法获得最佳分割点；

3. 重复m次，获得m个决策树；

4. 对输入样例进行预测时，每个子树都产生一个结果，采用多数投票机制输出。

## 3.2 训练模型

通过交叉训练，观察各个分类算法对于败诉胜诉分类的损失函数的多寡，结果如下：

|  |  |
| --- | --- |
| xgboost | 0.3534 |
| lightgbm | 0.3564 |
| svr | 0.4577 |
| 岭回归 | 0.3672 |
| 随机森林 | 0.3541 |

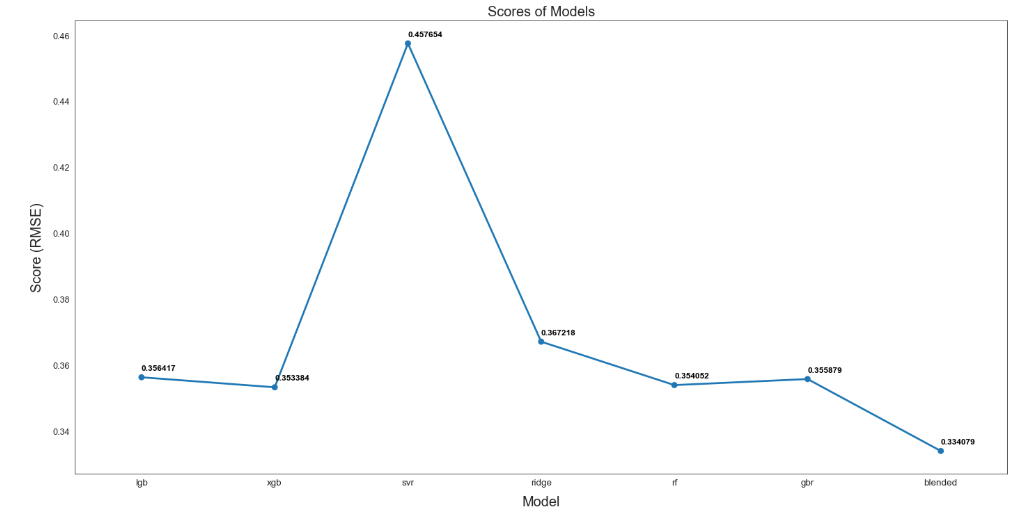
根据其损失函数对综合学习器进行加权，得到一个强学习器，此强学习器对于各个算法进行stacking集成后线性回归。

最后我们得到6个分类器，5个基础的分类器和一个集成的强分类器，再利用损失函数加权形成最终的第七个分类器，这是为了使得学习器更加稳健，不至于过拟合。

模型权重如下：

|  |  |
| --- | --- |
| Xgboost | 0.1 |
| Lightgbm | 0.1 |
| Svr | 0.2 |
| 岭回归 | 0.1 |
| 随机森林 | 0.05 |
| 强学习器 | 0.35 |

最终得到的综合学习器，损失函数值为0.33407，比任何一个弱学习器都要小。

**3.3 预测和拟合**

### 3.3.1 预测

因为使用的是线性回归模型，在测试AUC和ACC时将预测值-0.5后，大于0得标记为1，小于0的标记为0。

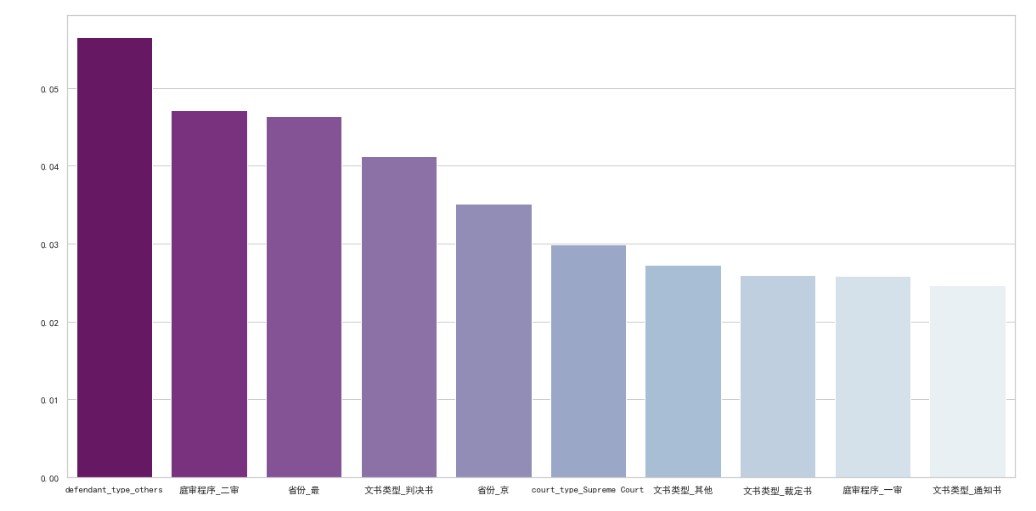
预测结果：

**准确率**: 84.32%

**AUC值**: 66.03%

### 3.3.2 评估各个变量对于模型的影响

此处以xgboost为例，观察模型中各个特征值的重要性



由图可见，模型中最为重要的参数是

（1）被告是否为“其他类型”

（2）庭审程序是否二审

（3）省份是否分类为“最高人民法院”

（4）文书类型是否判决书

（5）省份是否分类为北京

（6）法院类型是否为“最高人民法院”

（7）文书类型是否为“其他”

（8）文书类型是否为“裁定书”

（9）庭审程序是否为一审

（10文书类型是否为“通知书”

Top20的因素及其重要性大小为

|  |  |  |
| --- | --- | --- |
| 庭审程序 | 二审 | 0.047097 |
| 省份 | 最 | 0.046405 |
| 文书类型 | 判决书 | 0.041229 |
| 省份 | 京 | 0.035096 |
| court\_type | Supreme Court | 0.029846 |
| 文书类型 | 其他 | 0.027301 |
| 文书类型 | 裁定书 | 0.025965 |
| 庭审程序 | 一审 | 0.025830 |
| 文书类型 | 通知书 | 0.024640 |
| court\_type | high | 0.022351 |
| defendant\_type | other\_bru | 0.017598 |
| 庭审程序 | 其他 | 0.014235 |
| defendant\_type | Central\_department | 0.013651 |
| plaintiff\_type | legal entity | 0.013541 |
| 庭审程序 | 再审 | 0.013217 |
| 省份 | 粤 | 0.012884 |
| 省份 | 赣 | 0.012867 |
| 省份 | 琼 | 0.012767 |
| 省份 | 陕 | 0.012700 |

### 3.3.3数据量对于模型评估的影响

此处仍然以xgboost为例，探讨横截面特征数和样本总体数据量对于归类算法准确率与AUC的影响

1. 横截面特征数  
   在横截面特征上，我们分别剔除“TF-IDF特征”，“省份特征”，“法院特征”，“原告被告特征”来观察缺失了指定特征后，模型的预测效果变化。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 缺失TF-IDF | 缺失省份特征 | 法院特征 | 原告被告特征 |
| Acc | 0.8483 | 0.8316 | 0.8427 | 0.8346 |
| Auc | 0.6851  . | 0.6373 | 0.6718 | 0.6751 |

可以发现，“省份特征”， “原告被告特征”对于准确率和auc的影响较大，而单纯的“TF-IDF特征”，“法院特征”产生的影响可以忽略不计。

1. 样本比例

将样本划分为训练集和测试集，根据划分比例的不同，准确率和auc都会产生变化

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| Acc | 0.817722 | 0.816223 | 0.845309 | 0.833333 | 0.833164 | 0.830516 | 0.817457 | 0.820602 | 0.798253 |
| Auc | 0.6389730. | 0.642228 | 0.684038 | 0.658294 | 0.659978 | 0.659046 | 0.625662 | 0.628386 | 0.612770 |

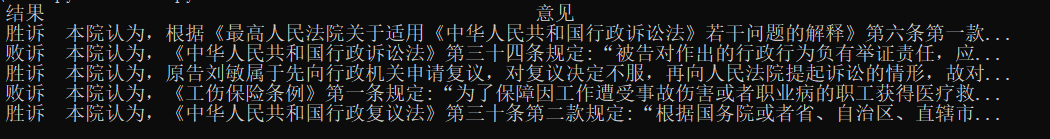
可以发现ACC和AUC值在样本集划分在0.5-0.6左右时最高，由于行政复议法院公文的同质性，过低的样本比例会产生过拟合问题，而过高的样本比例会导致训练集的数据不足，同样会降低预测的品质

## 案件分类模型

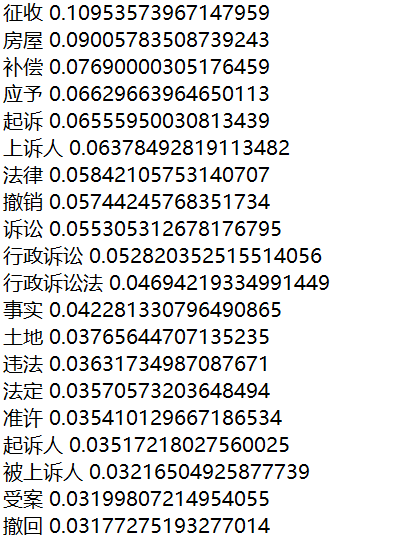
在以上实验之外，我们另外搭建了由jieba、word2vec、LogisticRegression构成的案件分类模型。该模型通过建立停顿词表，对每个案件的法院意见进行分词，整理得到包含所有案件法院意见分词的分词表。根据该表，可将法院意见映射为100维的矩阵，即法院意见向量化，进而利用该矩阵进行逻辑回归，建立案件结果分类模型。

## 4.1 jieba分词

Python的jieba库是一款中文分词工具。在经过数据预处理后，我们将所有的行政案件分为胜诉、败诉两大类，每一类信息中的“法院意见”作为文本内容。利用jieba库对每一条法院意见进行分词处理。



为了优化分词结果，我们需要创建stopwords停顿词文本，过滤无效的字、词。在本分类任务中，与“行政”类似的一些中性法律词汇，对分类结果的影响不大，却大量出现于法院意见之中，所以需要将这些词汇添加到停顿词中进行过滤。直接对败诉案件法院意见进行关键词提取，得到的结果如下：

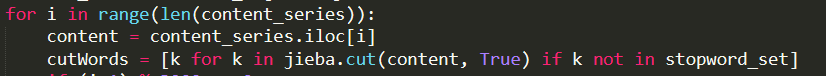


从关键词中可知，部分词汇是中性词汇，对判决结果无影响，为此我们需要更新停顿词，过滤无效词汇。部分停顿词列举如下：

**人民法院、第一款、认定、被上述人、涉案、证据、并无不当、上述人**

## 4.2 使用jieba库创建cutwords\_list

通过读取训练集中3947条法院意见，我们利用jieba库的cut方法获得所有法院意见的分词列表，将所有非停顿词的分词添加至cutwords\_list文本文件中。

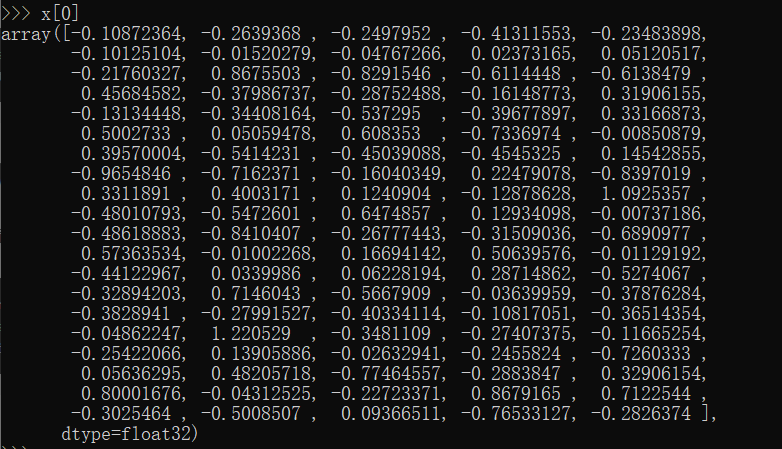


## 4.3 Word2vec模型

Word2vec模型通过建立词汇、向量间的映射，可以将文本库中所有的词向量化，从而定量地度量不同词汇间的的关系，挖掘词之间的关系。

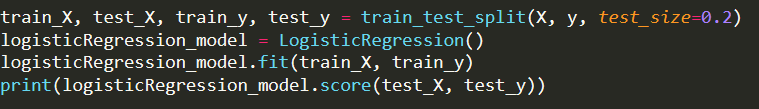
对于每一条法院意见，我们通过word2vec模型，获取意见的每一个分词在word2vec模型的相关性向量。然后把一篇文章的所有分词在word2vec模型中的相关性向量求和取平均数，即此篇文章在word2vec模型中的相关性向量。在实例化word2vec对象时，将关键词参数设置为100，即相关性矩阵大小为100维。

法院意见相关性矩阵举例如下：



## 4.4 模型训练

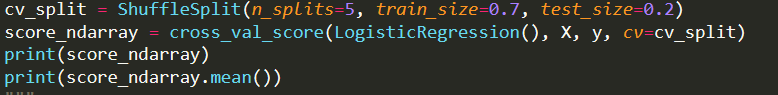
将3937条法院意见按照7：3的比例划分为训练集与测试集，调用sklearn.preprocessing库的LabelEncoder类的fit\_transform方法对新闻分类做***标签编码***。



回归结果值如下所示：



为了更好地验证模型的分类效果，进一步使用交叉验证的方法，进行验证。  
调用sklearn.model\_selection库的ShuffleSplit方法实例化交叉验证对象。  
调用sklearn.model\_selection库的cross\_val\_score方法获得交叉验证每一次的得分。



交叉验证回归结果如下：



从交叉验证结果看，训练后的逻辑回归模型进行案件分类的成功率可达91%。

## 分析总结

本项目对爬取的案件数据进行预处理，通过统计分析的方法确定了一部分案件特征，再利用TF-IDF算法进行关键词提取，获得了胜诉、败诉案件中的常见关键词，综合利用这两大类特征作为集合算法模型的特征输入，最终确定了与政府败诉相关的特征：庭审程序是否二审，法院是否为最高人民法院，省份是否为北京等等。明确了这些特征与政府败诉存在着一定的相关性。

最后，我们使用了word2vec模型实现了文本内容向量化，搭建了一个案件分类模型用于判断政府是否败诉。从分类结果看，该模型能够很好地分辨出政府是否败诉。但使用该模型亦存在着弊端，即我们无法从向量化的数据中明确到底是哪些词汇影响了分类结果。

通过这次大作业，我们系统地经历了一次数据挖掘的流程，深刻体会到数据预处理对于数据挖掘的重要意义，体会到数据源的重要性。合理有效地组织数据是数据挖掘成功的关键。