机器学习就是用大量的数据训练出来一个模型（也叫学习器），那该怎么训练呢？这就涉及到机器学习的算法。不同的算法得出不同的模型，采用何种模型需要和需求相联系，即你要解决的问题是什么，这个模型能不能很好的解决你的问题（性能度量）。这又涉及到模型的评估与选择。这个模型预测的结果准确率是多少？它能不能适应没有训练过的数据（泛化）？它是不是只针对一些数据有效（过拟合）？我们不能把所有的数据用来训练（训练集），得留一部分数据来进行测试（测试集）。那又该怎么划分训练集和测试集？

模型可以理解成一条规则。如果下雨，地面就是湿，这也可以说是一种模型。

交叉验证法：

把训练集划分成k个集合，每个集合的数据分布类似，用k-1个集合当做训练集，第k个集合当成测试集，此过程重复k次（即第k个集合每次都不一样）。

Python os.listdir(path) :把path路径下的文件或文件夹读取出来，保存并返回该值。 <http://www.runoob.com/python/os-listdir.html>

os.path.isdir(path)

如果path是一个存在的目录，则返回True。否则返回False。

<http://blog.51cto.com/wangwei007/1104940>

DS\_Store (英文全称Desktop Services Store) 是一种由苹果公司的Mac OS X操作系统所创造的隐藏文件，目的在于存贮目录的自定义属性

os.remove() 方法用于删除指定路径的文件。如果指定的路径是一个目录，将抛出OSError。

在Unix, Windows中有效

http://www.runoob.com/python/os-remove.html

.read() 每次读取整个文件，它通常用于将文件内容放到一个字符串变量中。然而 .read() 生成文件内容最直接的字符串表示，但对于连续的面向行的处理，它却是不必要的，并且如果文件大于可用内存，则不可能实现这种处理。

<http://blog.csdn.net/werm520/article/details/6898473>

os.path.exists(path)

如果path存在，返回True；如果path不存在，返回False。

<http://blog.51cto.com/wangwei007/1104940>

分词：

<http://blog.csdn.net/FontThrone/article/details/72782499>

shutil.move( src, dst) 移动文件或重命名

<http://www.cnblogs.com/funsion/p/4017989.html>

怎么知道哪个文本是属于哪一类？

scikit-learn是Python的一个开源机器学习模块，它建立在NumPy，SciPy和matplotlib块之上能够为用户提供各种机器学习算法接口，可以让用户简单、高效地进行数据挖掘和数据分析。

<http://hareric.com/2016/05/22/scikit-learn%E7%9A%84%E5%AE%89%E8%A3%85%E5%92%8C%E5%9F%BA%E6%9C%AC%E4%BD%BF%E7%94%A8%E6%95%99%E7%A8%8B/>

Scikit Learn CountVectorizer 入门实例

<http://blog.csdn.net/guotong1988/article/details/51567562>

fit\_transform()按照一定的算法中得到实际上是一个词向量矩阵

简单的词向量矩阵例子

"dog cat fish","dog cat cat","fish bird", 'bird'

dog dog只出现一次，所以次数为0，dog cat出现两次所以次数为2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Count | dog | cat | fish | bird |
| dog | 0 | 1 |  |  |
| cat |  |  |  |  |
| fish |  |  |  |  |
| bird |  |  |  |  |
|  |  |  |  |  |

存在问题：

随着词料的增多，这个矩阵会越来越大

这个矩阵很可能最后是一个稀疏矩阵，浪费，所以等对矩阵进行降维等处理。

TF-IDF权重策略

如果一个词在一个文件里出现得比较多，而在其他文件出现比较少，那么这个词就适合分类（越重要）

TF:(词频)

一个词语在文件中的出现次数

IDF(逆向文本频率)

一个词重要的度量

IDF = log(总文件数目/保护该词语的文件数目)

TF与IDF的乘积越大，该词就越重要 ，即一个词在一个文件里出现得越多，而在其他文件出现得越少，该词就越重要