机器学习一般步骤：（以kaggle的titanic为例）

Step1：拿到数据考虑数据的正负样本是否均衡，如果不均衡，通过上抽样或下抽样或者调整样本的权重，来获得均衡的训练样本。

Step2：从整体上把握原始数据中各特征的特点和与目标值的关系，初步判断该特征是否重要，可以通过数据可视化的形式形象表现出来

Step3：缺失值处理：

1. 如果数据集很多，但有很少的缺失值，可以删掉带缺失值的行
2. 如果该属性相对学习来说不是很重要，可以对缺失值赋均值或者众数。比如在哪儿上船Embarked这一属性（共有三个上船地点），缺失俩值，可以用众数赋值
3. 对于标称属性，可以赋一个代表缺失的值，比如‘U0’。因为缺失本身也可能代表着一些隐含信息。比如船舱号Cabin这一属性，缺失可能代表并没有船舱
4. 使用回归 随机森林等模型来预测缺失属性的值。因为年龄在该数据集里是一个相当重要的特征，所以保证一定的缺失值填充准确率是非常重要的，对结果也会产生较大影响。一般情况下，会使用数据完整的条目作为模型的训练集，以此来预测缺失值。对于当前的这个数据，可以使用随机森林来预测也可以使用线性回归预测。我使用的是随机森林预测模型，选取当期数据集中的数值属性作为特征（因为sklearn的模型只能处理数值属性阿，当然后面会把其他属性都处理成数值属性，然后就可以更新这块预测年龄的代码啦）。

Step4：数据变换：

scikit learn中的模型都只能处理数值型的数据，需要将这些原始的字符串类型的数据转为数值型数据。所有数据通常可以分成两种类型：定量与定性。定量的属性（数值属性）通常蕴涵着可排序性，比如在泰坦尼克号数据集中，年龄就是一个定量属性。定性属性（标称 序数 二元属性）的值是一些符号或事务的名称，每个值代表某种类别编码或状态，不是可测量量，是不具有排序意义的，比如Embarked(登船地点)。

一 定性属性的数据变换

对于字符串型的定性属性转换，如果单纯的用数字来代替的化，比如对于Embarked的三个值S Q C分别用1 2 3来代替，模型会把它当成是有顺序的数值属性，对于一些根据距离来确定分类的算法来说，就不能准确运行啦。那么应该怎么将定性属性转为数字呢？

（1）dummy varibles(不知道中文应该说成啥。。虚设属性？)

什么是dummy呢，举个栗子，Emarked属性的取值有三个S Q C，分别代表三个上船地点。dummy这个属性呢，就是向数据集里再加入三个属性暂且命名为Embarked\_S Embarkde\_Q 和Embarked\_C，如果一个人是在S地点上船的，那么这三个属性的值就是（1，0，0），在Q点上船的就是（0，1，0），每个属性都是二元属性，1代表是，0代表否。所以dummy适用于值范围相对较少的属性。

这样就会3个dummy属性加到数据集里啦，用df.info()看一下：

（2）factorizing(因子分解？)

用dummy可以处理像Embarked这样的值域范围较小的标称属性。对于Cabin（船舱号，A43 B55这种）这种标称属性，用dummy就不好处理了。pandas提供了一个factorize()函数，用以将标称属性的字符串值映射为一个数字，相同的字符串映射为同一个数字。不同于dummy，这种映射最后只生成一个属性。对于Cabin属性，我们可以将其分成两部分，字符串+数字，新建两个属性。对于字符串（A-E & U），可以用factorize()将其处理成数字。

二 定量属性的数据变换

1）数据规范化

数据规范化通过将数据压缩到一个范围内（通常是0-1或者-1-1）赋予所有属性相等的权重。对于涉及神经网络的分类算法或者基于距离度量的分类和聚类，规范化特别有用。规范化方法有多种，如rescaling logarithmic normalize等，但是有些时候并不需要规范化，比如算法使用相似度函数而不是距离函数的时候，比如随机森林，它从不比较一个特征与另一个特征，因此也不许要规范化，

StandardScaler将数值压缩到[-1,1]区间，计算公式为(2x - max(x) - min(x)) / (max(x) - min(x)).

（2）Binning

就像直方图的bin将数据划分成几块一样，我们也可以将数值属性划分成几个bin，这是一种连续数据离散化的处理方式。我们使用pandas.qcut()函数来离散化连续数据，它使用分位数对数据进行划分，可以得到大小基本相等的bin。因为是bin,所以属性都是一个个区间，代表这个数据属于哪个区间。对于这样的数据，我们需要factorize下，转为数值型数据。

Step5：深挖数据，提取特征的派生属性

经常采用的一种方式是先自动产生大量的候选特征，然后根据它们与分类类别的信息增益等方法来选取最好的特征。但是，运行包含大量特征的学习器来寻找有用的特征组合太耗时，也容易导致过拟合。还是需要人为的介入特征的选择中。

什么是派生属性呢？派生属性就是从原始数据中得到的一些属性，比如上一节从Age属性经过Factorize得到的Age\_bin属性就是一个派生属性，当然这种派生只是非常简单的派生。为什么要对这些属性做各种各样的统计和处理呢，这其实是特征工程的一部分，先构建足够多可能会对结果有意义的属性，然后再从这些候选集中选择我们想要的特征。特征工程非常繁琐，但是对数据挖掘非常重要，一般来说，做一个数据挖掘项目，百分之八十的努力要用在特征工程上。除了基本的转换和interaction属性，我们也要创造性的从原始属性中发现新属性。比如电话号码，可以从中提取出来国家和区域特征。

一 研究业务逻辑提取特征

对于一些字符串类型的属性，比如Name我们可以从中提取出来一些可以揭示其社会地位的称号，再者对于Ticket属性也需要处理，数字部分也有一定的规律：以1开头的一般是一等舱2开头的是二等舱3开头的是三等舱，4-9开头的大都是三等舱。以上这些数据告诉我们处理Ticket是有意义的，能够发现其内部蕴涵的信息。

二 简单组合属性提取特征

一些属性可以从它本身的数据里提取一些信息，有些属性则需要和其他属性组合来产生信息。比如对淘宝上的一个商品来说，购买数/点击率可以反应商品的转化率，也是商品的一个非常重要的特征。

对于Titanic来说，我们用Age\*Pclass组合产生一个属性，虽然没有一个名词来解释它，但是从结果数据上来看，我们增大了年纪大的人的权重也提高了高等舱的权重，从最后幸存的结果上看，这个组合还是有意义的。除了这两个属性之外，我们还可以对其他数值属性进行数学运算，以得到更大的候选特征集。

这个过程自动产生大量的特征，这里用了9个特征产生了176个特征，可能这些特征有些过于多了，但是它只是一个候选集，我们可以通过一些处理筛选掉一些特征。当然有些模型也很适合大量特征的训练集，比如随机森林（有论文验证，随机森林是分类算法中表现最好的模型）。

产生的这些特征可能高度相关于原始特征，线性模型处理这类特征时会产生multicollinearity问题，可以用计算这些特征的皮尔逊相关系数，筛选相关性特征。如果用随机森林模型训练的话，可以不需要这个步骤。

三 用PCA进行维归约

通过上面三个部分的处理，我们得到了具有大量特征的维度很高的数据集，特征较多不能直接用来作为模型输入，一是因为这些特征间具有多重共线性，可能 会导致空间的不稳定；二是因为高维空间本身具有稀疏性，一维正态分布有68%的值落于正负标准差之间，而在十维空间上只有0.02%；三是由于过多的属性 会使挖掘需要很长时间。对于一些模型来说，比如使用L1（Lasso），当有大量属性时效果很好，因为它可以有效忽略掉噪声变量。而一些模型则容易过拟 合。

数据归约技术可以用来得到数据集的规约表示，它小得多，但仍接近于保持原始数据的完整性。也就是说，在归约后的数据集上进行数据挖掘将更加有效，仍然产生几乎相同的数据分析结果。

PCA（主成份分析）是一种维归约的方法，它搜索k个最能代表数据的n维正交向量，将原始数据投影到一个小的多的空间上，导致维归约。PCA通过创建一个替换的较小的变量集组合属性的基本要素

实验发现，PCA对于线性模型（不使用Lasso）非常有效，但是对于随机森林模型没有提高。

Step6：进行算法的学习训练：

经过之前的特征工程，现在已经有237个特征，数目过多的特征会使模型过拟合，幸运的是，随机森林在训练之后可以产生一个各个特征重要性的数据集，我们可以利用这个数据集，确定一个阈值，选出来对模型训练帮助最大的一些属性。

可以看到Tiltle\_Mr Title\_id Gender这三个属性相当重要。而与Title有关的属性都是我们对姓名进行分析得到的，可见一些字符串属性中可能会藏有非常重要的信息，在特种工程中要非常重视而不是将其抛弃。因为我们的原始属性非常少，所以产生的重要属性大都是原始属性的数学组合，派生变量可能并不是必需的，这主要和模型有关，但大多数时候派生变量是没有什么坏处的。对于随机森林这种训练数据想对容易的模型来说，可能一些原始的属性直接用来进行训练也会产生很好的效果，但是作为一道学习题，当然是什么处理办法都要尝试一遍，积累经验啦。

当然在得到重要的特征后，我们就要把不重要的特征去掉了，以提高模型的训练速度（阈值可调的小一点，以保留更多的特征）

Step7；参数选择

上面部分都是用的随机森林的默认参数，但是模型的参数是可调的，我们要调整参数以获得更好的训练。scikit learn 提供了两种参数优化的方法，也是其他工具通用的方法，一是GridSearch，另一个是RandomizedSearch。在这两种情况下，都可以指定每个参数的取值范围，创建一个字典。将参数字典提供给search方法，它就会执行模型所指定的值的组合。对于GridSearch，它测试参数每一个可能的组合。 RandomizedSearch允许指定有多少不同的组合要测试，然后随机选择组合。如果正在使用模型关键参数很多，RandomizedSearch很有用可以帮助节省时间。

Step8：模型的评价：

主要是偏差方差折衷与测试误差和训练误差的关系。我们应该调整模型来达到测试误差的最小值。sklearn.learning\_curve模块可以完成这个功能

<http://www.cnblogs.com/north-north/tag/kaggle/>

计算AUC

|  |
| --- |
| def\_auc(y\_true,y\_score): |
|  | """Alternative implementation to check for correctness of |
|  | `roc\_auc\_score`.""" |
|  | pos\_label = np.unique(y\_true)[1] |
|  |  |
|  | # Count the number of times positive samples are correctly ranked above |
|  | # negative samples. |
|  | pos = y\_score[y\_true == pos\_label] |
|  | neg = y\_score[y\_true != pos\_label] |
|  | diff\_matrix = pos.reshape(1, -1) - neg.reshape(-1, 1) |
|  | n\_correct = np.sum(diff\_matrix > 0) |
|  |  |
|  | return n\_correct / float(len(pos) \* len(neg) |