特征工程步骤

# 数据探索分析（Exploratory Data Analysis）

数据的探索分析EDA，是对数据进行初步的统计分析，统计数据的分布、异常、相互关系，目标是让我们了解这些数据能告诉我们什么。可以用来指导我们进行模型的选择，比如说帮助我们初步的决定哪些特征需要被使用，哪些特征需要被剔除。

## 1、初步分析阶段

1. 准备好样本、特征、label
2. 查看样本集中样本总个数、特征总个数
3. 查看正负样本个数，如果正负样本相差不多，则属于样本均衡的建模问题，如果其中一类明显高于另外一类，则属于样本不均衡的建模问题
4. 先整体查看一下
   1. 每个特征的缺失值数量以及缺失值所占比例
   2. 每个特征的类型：离散型、连续型，按照不同的类型进行分组，分别进行处理
5. 接下来，按照上面的分组，进行单个特征评估

## 2、单个特征评估

* *分析过程按照以下顺序依次进行，没有出现的情况就跳过*
* *分析方法是推荐使用，如有其他方式达到相同效果也可行*
* *分析工具自由选择*

### 2.1 连续型特征

1. 查看变量的缺失值（missing value，空值）个数以及所占比例

* 剔除掉缺失率>80%并且未来依然会高缺失的变量
* 如果“缺失”包含了业务含义，保留该变量，并且直接将缺失值填充为默认值。
* 如果“缺失”没有业务含义
  1. 连续型变量：偏正态分布，用均值填充，可以保持数据的均值；偏长尾分布，使用中值填充，避免受异常值的影响。编程上可以用sklearn.preprocessing中的Imputer实现。
  2. 若缺失量比较大，不适合使用平均值填补。一般通过使用其它变量预测或者直接将缺失值设置为默认值的方法填补；

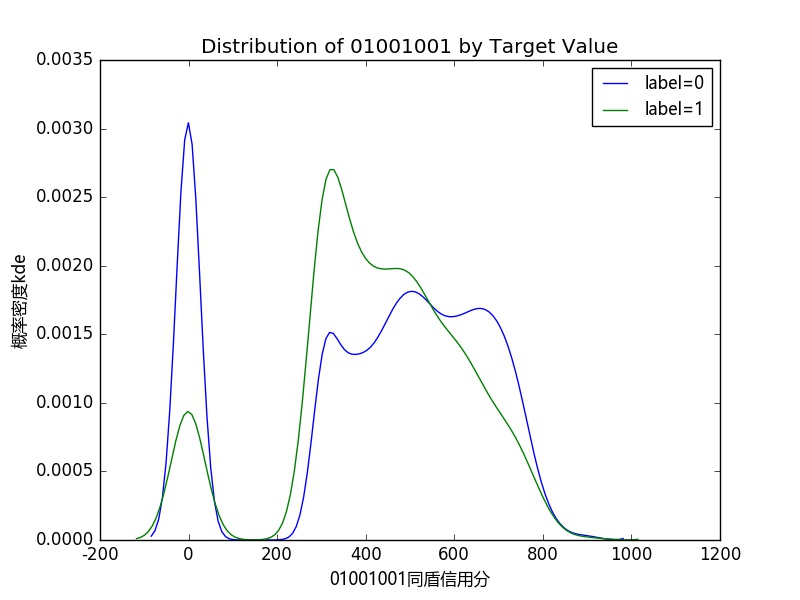
1. 画概率密度图，查看好坏样本在该特征值上的分布

根据变量的值，统计label=0好样本（=1坏样本）的比例，画概率密度图：

* X轴：特征值
* Y轴：具有当前特征值的label=1（或=0）样本占总label=1（或=0）样本的比例
* 编程可以用seaborn. distplot(kde=False, hist=True, norm\_hist=Truee)实现。

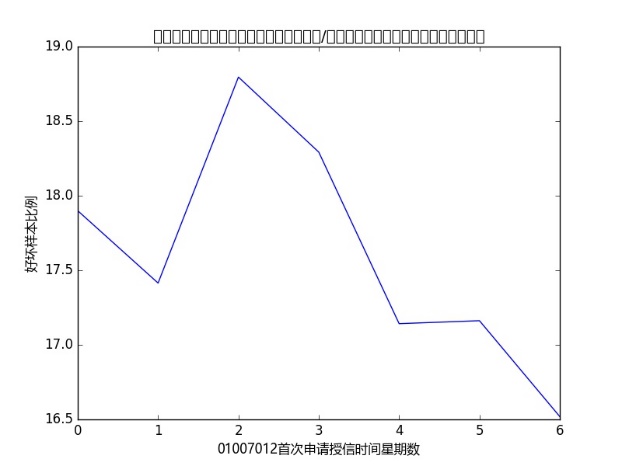
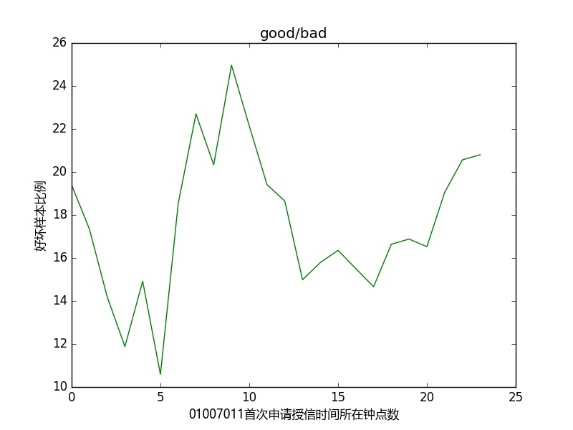
可以看出

* 异常值/极端值
* 查看好坏样本在该特征值上的分布：好（坏）样本集中在哪个范围内？



1. 画“好坏样本比”曲线：计算出每一个特征值上的好坏样本比。可以看出该特征对好坏样本的区分度。

* X轴：该特征的每一个特征值
* Y轴：好坏样本比：好样本个数/坏样本个数
* 从下左图，比较纵坐标的值，可以看出该特征对好坏样本还是有一定区分度的
* 从下右图，比较纵坐标的值，可以看出该特征对好坏样本基本上没有区分度



1. 极端值分析

* 在上一步画图的时候，就可以看出来有哪些极端值
* 除非是计算或者人为错误，极端值需要从**业务层面**进行分析
  1. 分别计算“拥有极端值的那些样本中，坏样本的占比R1” 和“拥有非极端值的那些样本中，坏样本的占比R2”
  2. 如果R1明显大于R2，则说明极端值有业务意义，建议保留，特别是从反欺诈的角度来分析一下
* 如果是计算或者人为错误，极端值的影响可以通过分箱被中和，对模型训练没有影响；也可以将极端值设置成缺失值/空值，然后用处理缺失值的方法进行处理

1. 计算变量的方差

* 如果方差接近于0，也就是说样本在这个特征上基本上没有差异，这个特征对于样本的区分并没有什么用，剔除；

1. 查看变量与label之间的相关性

皮尔逊系数/斯皮尔曼系数：衡量2个变量之间的线性相关性。

* .00-.19 “very weak”
* .20-.39 “weak”
* .40-.59 “moderate”
* .60-.79 “strong”
* .80-1.0 “very strong”

结合特征的概率密度图来决定，该特征对于样本的区分有没有作用。

1. 查看变量之间的相关性

去除冗余变量，并结合业务含义取舍

* 皮尔逊系数/斯皮尔曼系数：衡量2个变量之间的线性相关性。如果>0.8，说明2个变量有明显线性关系，只保留一个，保留与label的皮尔逊系数较大的那个变量或者保留lightgbm AUC最大的那个；
* PCA分析：将相关变量分为组，从分好的组里再选择权重系数较高的变量留下；

1. 计算变量IV值（也可以考虑Gini系数），通过IV值高低将变量进行排序

* <0.02 unpredictive
* 0.02 – 0.1 weak (maybe acceptable)
* – 0.3 medium
* >0.3 strong

IV值低的变量根据业务分析决定是否剔除，一般<0.02的变量可剔除；

1. 计算AUC

* 计算该变量经过lightgbm之后的AUC，AUC>0.53可以考虑留下

1. 其他考虑因素

* KS
* ROC
* 稳定性：PSI

5、6、7、8、9要跟业务结合在一起，综合考虑，衡量一个变量采用与否。

### 离散型特征

1、查看变量的缺失值（missing value，空值）个数以及所占比例

* 剔除掉缺失率>80%并且未来依然会高缺失的变量
* 如果“缺失”包含了业务含义，保留该变量，并且直接将缺失值填充为默认值。
* 如果“缺失”没有业务含义
  1. 离散型变量，用众数填充；
  2. 若缺失量比较大，不适合使用中位数或者平均值填补。一般通过使用其它变量预测或者直接将缺失值设置为默认值的方法填补；

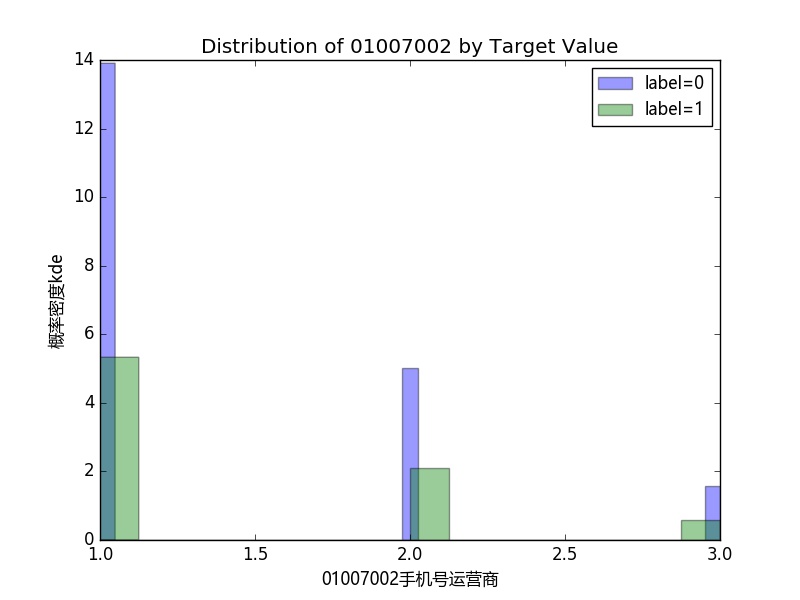
1. 画概率密度图，查看好坏样本在该特征值上的分布

根据变量的值，统计label=0好样本（=1坏样本）的比例，画概率密度图：

* X轴：特征值
* Y轴：具有当前特征值的label=1（或=0）样本占总label=1（或=0）样本的比例
* 编程可以用seaborn. distplot(kde=False, hist=True, norm\_hist=Truee)实现。

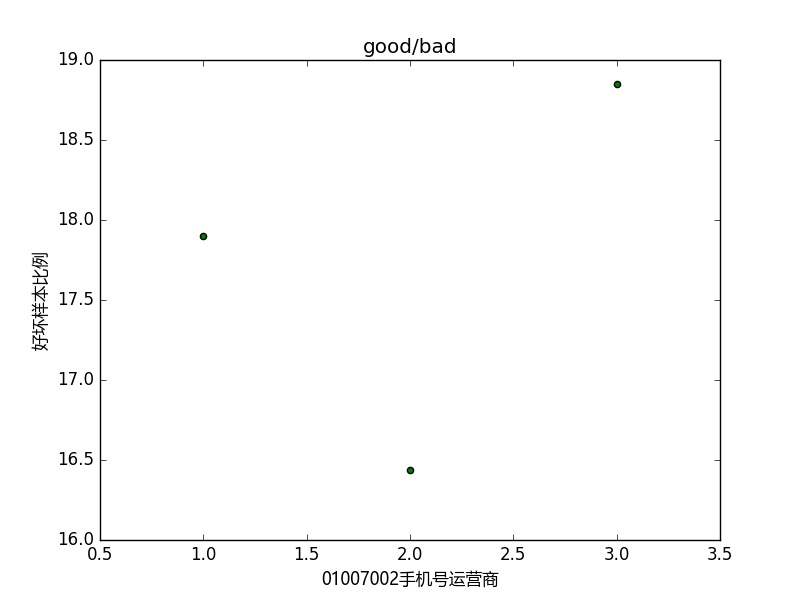
可以看出

* 该离散特征unique value的个数以及值
* 异常值/极端值
* 查看好坏样本在该特征值上的分布：好（坏）样本集中在哪个范围内？



1. 画“好坏样本比”散点图：计算出每一个特征值上的好坏样本比。可以看出该特征对好坏样本的区分度

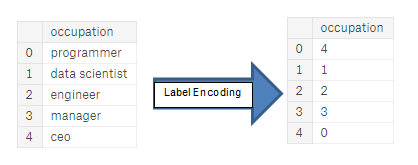
* X轴：该特征的每一个特征值
* Y轴：好坏样本比：好样本个数/坏样本个数
* 从下图，比较纵坐标的值，可以看出该特征对好坏样本基本上没有区分度



1. 对离散值进行编码

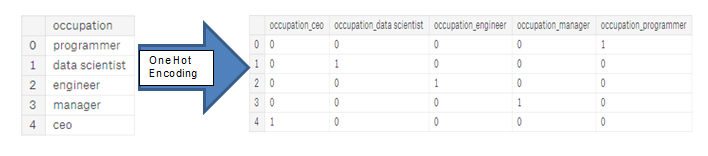
由于离散值有数值型、字符型等，其中字符型没有办法直接入模计算，所以要对其进行编码，全部变成数值型, 将定性数据编码为定量数据。

* Label encoding 将变量中的每个unique value指定为一个整数，该方法不会创建新的列。编程上可以用sklearn中的 LabelEncoder实现。



注意：编码给每一个unique value的整数，我们并不在意它的具体数值，设置成任意值都可以，只要不重复。但是这里需要注意一个问题，如果一个离散特征的unique value个数>2，该编码方法会出现问题：上面的例子，在用树模型的时候，树模型会将（0，1，2，3，4）当成有数值意义的整数来看待，可能会根据（是否>3）的原则，将（0,1,2）分成一组，（3,4）分成一组，也就是（ceo,data\_scientist,engineer）是一组，（manager，programmer）是一组。但是实际上，我的编码（0，1，2，3，4）之间是没有数值上的大小关系的，所以这样分组就有问题。所以对于unique value个数>2的离散特征，One-hot encoding的编码形式是比较好的选择。

* One-hot encoding 将变量中的每个unique value变成一列，该方法生成了新的列。编程上可以用pd.get\_dummies(df)或者sklearn.preprocessing中的OneHotEncoder实现。

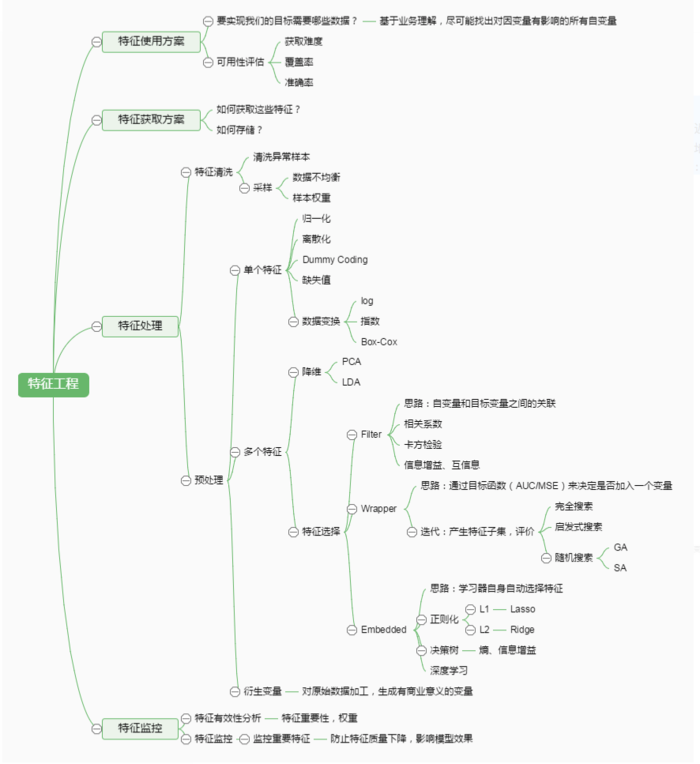


在这里，建议当离散变量的unique value=2时，用label encoding编码方法；当unique value>2时，用one-hotencoding编码方法

* 要注意到的是，之所以需要对离散特征进行处理，是针对模型来说的，模型学习过程中需要对特征的序进行比较时，那么就需要对离散特征进行上述处理。

1. 其他类似连续特征处理方式

# 特征工程



记住一点，与模型构建和超参数调优相比，特性工程具有更大的投资回报！！！

前面做完特征的初步分析之后，接下来是重头戏：特征工程，一般会做两件事情：从现有的特征中构造出新的特征、从现有的特征中筛选合适的特征。下面主要介绍2种特征构造方法：Polynomial features，Domain knowledge features

## 多项式特征（Polynomial features）

我们在进行分类和回归预测的时候，有时需要根据已有的特征构建更多的特征，然后对特征再进行特征选择，通过增加一些输入数据的非线性特征来增加模型的复杂度，这种做法通常是有效的。一个简单通用的办法是使用**多项式特征**，可以获得特征的更高维度和互相间关系的项。

多项式特征，是一个简单的特征构造方法，该方法是根据已有的特征产生新的特征，可以通过sklearn.preprocessing 中的PolynomialFeatures 实现。这些由多个单独变量组合而成的特征称为交互项。换句话说，虽然两个变量本身对目标的影响不大，但将它们组合成一个单独的交互变量可能会显示出与目标的强关系。

举一个例子：

* 原特征是(X\_1, X\_2)，计算原特征与label的pearson线性相关系数；
* 用PolynomialFeatures(degree=2)构造多项式特征，构造出来的新特征就是(1, X\_1, X\_2, X\_1^2, X\_1X\_2, X\_2^2)；
* 再计算这些新特征与label的pearson线性相关系数，查看是否有提高；
* 如果有，可以考虑将这些新生成的多项式特征合并到原特征里；
* 为了查看这些多项式特征有无效果，需要进行对比实验
  1. 用原特征训练lgb模型，记录AUC
  2. 加入多项式特征之后，再训练lgb，记录AUC
  3. 查看后者AUC是否较前者AUC有提高，如果有，说明多项式特征起了作用。