**一、概述：**

利用机器学习解决实际问题时，一般的流程是：

收集数据 –> 数据预处理 –> 特征工程 –> 建模和训练数据 –> 预测。

其中特征工程是非常重要的一个步骤。它是基于你对于数据和模型的理解，对数据的原始特征进行处理，以及通过原始特征的组合或映射来构造新的特征，使得新的特征更能适应模型对输入的要求，从而使模型达到更好的性能。

实际上，特征工程可以看成是模型的改造。比如，在线性模型中将样本的特征用代替，则线性模型就变成了二次模型。

**二、特征工程中常见操作：**

**1、特征编码。**模型需要输入的样本的各维特征都是数值，因此在建模之前，必须对特征进行编码，也就是将字符串类型的特征转换为数值，一般我们只需将字符串对应成离散数值0,1,2,3,…即可。在此之后，可能需要对离散型特征进行“去大小化”或对连续型特征“离散化”。

“去大小化”是什么意思呢？当该离散型特征的取值只具有不同类别的意义，而没有比较大小的意义时，比如“科目”特征，有数学、语文、英语三个取值，我们可以简单地将其数值化，将三个科目分别映射为1,2,3。但这样做的问题就是使这三个只具有类别意义的取值变得具有大小的意义，而模型可能会学到比较大小的信息，这些信息就是虚假的信息。

解决这个问题的通常做法是“独热编码（one-hot-encoding）”，它的意思是将一个特征拆成多个新特征，每个新特征对应于原特征的一个取值，然后样本在每个新特征上的取值为1（若样本在原特征上取值为该新特征）或0（若样本在原特征上取值不为该新特征）。

注意，并不是所有的离散型特征都需要进行独热编码，对于那些本身就具有大小的比较的意义的特征，就不需要进行独热编码。

特征编码的另外一种操作是连续型特征的离散化。就是将连续型特征的取值按区间映射成离散的取值，比如某特征取值范围为(18,39)，可以将该特征进行如下离散化：(18,24] ->1,(24,30]->2,(30,39]->3。

连续性特征的离散化操作可以增加模型的非线性型，同时也可以有效地处理数据分布的不均匀的特点。

**2、特征缩放。**样本的不同特征的实际意义是影响预测结果的不同因素。不同特征对预测结果的影响程度不同。特征对预测结果的影响是怎么体现的呢？对于连续型的特征，就是该特征产生一定的变化，会使预测结果发生多大的变化。显然，如果特征发生很大的变化，但预测结果基本没有改变，就说明该特征对于预测结果的影响很小，那么该特征对于该预测问题也就没有太大用了。按照我们正常的理解，在衡量特征对预测结果的影响程度时，应该使用“相对变化”而不是“绝对变化”，比如特征1取值范围在1e3量级，特征2取值范围在1e1量级，如果特征1变化100跟特征2变化10能够使预测结果产生相同大小的变化，那么我们能认为特征2产生的变化较小所以特征2对预测结果的影响程度更大吗？不能，因为特征1产生的相对变化较小，所以特征1对预测结果的影响程度更大。

在实际中，我们的模型通常不会计算特征的相对变化，而是计算绝对变化。比如很多模型是基于距离的，直接计算距离的话，若两个特征的值域范围相差较大，那么距离的大小基本上由值域较大的特征决定，值域较小的特征基本上起不到什么作用。因此我们需要在特征工程阶段，对特征进行缩放，也就是标准化。标准化的方法主要有两种： “最大最小标准化”和“z-score标准化”。

（1）最大最小标准化：



其中min(x)和max(x)分别是x的最小值和最大值。

（2）z-score标准化：



其中mean(x)和std(x)分别是x的均值和标准差。