**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，对于那些本身就需要考虑大小的比较的意义的特征，就不需要进行独热编码。

另外，不同的机器学习模型对离散型特征的理论上的处理方式也是不同的，有的模型（如线性回归模型）本身无法避免考虑特征的大小比较关系，这时只能将需要不考虑大小关系的特征进行独热编码，对于独热编码后的特征，即使模型也对其考虑大小比较关系，那也没有意义，因为其只有两个取值，本身就没有了大小关系的意义；而有的模型（比如决策树）对于离散型特征的处理本身就可以不考虑大小关系，因为它可以根据离散型特征的每个取值都分裂出一个子节点（形成多叉树）。这时也就没有必要对特征进行独热编码了。但是要格外注意的是，sklearn库中的决策树模型，它无法区分特征是离散型还是连续型，所以相当于对所有的特征都当作连续型特征进行处理，因此对那些不想考虑其大小关系的离散型特征，还是要先进行独热编码。尽管模型对独热编码后的特征也是当作连续型特征来处理，但是（前面已经说过）独热编码后的特征本身已经不具有大小关系的意义了，所以从效果上还是相当于不考虑大小关系的。

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。

**3、特征选择。**上面已经说过，样本的不同特征对样本的输出的影响程度不同。现实中，我们收集的样本的特征可能存在两个问题：一是可能存在对样本输出影响程度非常小的特征（或称为“无关特征”），它们在模型对样本的预测中基本不起作用；二是可能存在具有比较强的相关性的特征，它们对于模型的训练来说是冗余的。

为了解决这两个问题，去除无关特征和冗余特征，我们常常在特征工程部分进行特征选择。

常用的特征选择方法有以下几种：

（1）贪心算法：该方法需要与模型绑定，先从所有特征中选择一个最优特征，使得模型在该特征上的表现最好，将该特征作为初始的特征集。然后迭代，每次向特征集中添加一个特征，使得添加该特征后，模型在该特征集上的表现最好。当然，也可以从所有的特征构成的特征集出发进行，每次剔除一个特征使模型效果变好（或者不会变差）。

（2）基于L1正则：对于某些模型来说，模型的参数就是各特征的系数，它们表征了相应特征对输出结果的重要性。比如线性回归、逻辑回归模型。这时，对模型使用L1正则，会有稀疏化的作用，也就是使比较不重要的特征的系数变成0。这就相当于对样本做了特征选择。

（3）决策树算法：决策树算法其实是一个分类算法，它可以实现样本分类。由于在训练的过程中，决策树每次向下分裂都要选择“最优特征”，使用最优特征进行分割，所以它相当于对特征的重要程度进行了排序，越靠近根节点的节点对应的分裂特征越好，因此它也实现了特征选择的功能。

（4）基于相关性计算：就是计算每个特征跟输出的相关性，相关性越强则特征越好。