本项目用的为决策树模型

决策树模型分为分类树和回归树，两者都是if-then的树形原理，最大的区别为，对于分类树，最后一层叶子结点才是分类标签，其他时候的结点都不是，而回归树不同，回归树的每一个结点都是待回归属性。

决策树模型的逻辑是，从根节点出发，对实例的每一个特征进行判断，根据判断结果，将实例分配到其子节点中，此时，每个节点又对应着该特征的一个取值，如此递归的对实例进行判断和分配，直至将实例分配到叶子结点中，其基本楼诚遵循简单且直观的“分而治之”的策略。

本质而言，决策树模型是一个 if - then 的规则集合，根节点到叶子结点的每一条路径构建成了一条规则。这个规则有一个重要的性质——互斥且完备：

* 完备性：每个实例**都**有一条规则路径所覆盖
* 互斥性：每个实例**只**有一条规则路径所覆盖

构建决策树规则路径的依据是条件概率，这一条件概率分布定义在特征空间的一个划分上，决策树的路径就是一个划分单元，决策树分类时将该节点的实例强行分配到条件概率大的一个类别中。

决策树学习的本质是从训练数据集中归纳出一组分类规则，这种决策规则有可能一个也没有，也可能有很多个，这时候需要选择一个与数据集矛盾较小的决策树规则，同时又需要很好的泛化效果。

决策树使用损失函数来表示这一目标，这个损失函数通常是正则化的极大似然函数。决策树的学习策略就是以损失函数为目标函数的最小化。由于从很多决策树中选择决策树是一个NP难问题（多项式复杂程度的非确定性问题，无法按部就班的直接求解，通常使用验证解的方式近似求解），所以决策树学习算法通常是递归的选择特征对数据集进行分割，以达到最好的分类结果。

学习算法原理：构造根节点，将所有训练数据集都放到根节点，选择一个最优特征，将训练数据集分割成子集，使得训练集在当前按条件下有最好的分类。如果这些子集已经可能很好的分类，那么构建叶节点，如果还不能很好的分类，继续对其分割，构造相应的结点，如此递归进行，直至所有训练数据集被基本正确分类，或者没有合适的特征为之。

数据类型为需要决策树的数据类型