在根目录下，BTREE存储B树的具体实现，draw存储绘图用到的代码及其数据，实验从内存占用**memory.py**，查询时间**time.py**，准确率**currency.py**三个层面进行。其中包括了以神经网络为顶层的系统结构和btree以及kraska的对比实验。Kraska，multi文件夹中是以多项式为顶层结构的对比试验。

RMI文件夹中是我们复现的递归学习索引模型

new\_index\_learning目录下为系统架构代码。其中experiment文件夹下为实验绘图代码，存储的训练好的多种模型，供自动选择调度。new\_index\_learning目录中是模型构建文件，包括了决策树回归decision\_tree.py及其类文件**decision\_tree\_class.py**，弹性回归类**elastic\_class.py**，极限树回归类**extraforest\_class.py**，梯度提升树回归类**GradientBoostingRegressor\_class.py**，**knn**类，线性回归类**linear\_class.py**，多项式回归类**muti\_class.py**，随机森林回归类**randomforest\_regression.py**，调用svr内核的方法类**rf\_class**以及**rbf\_class**。**Score\_config.py**存储了打分函数中的一些超参，以及回归模型常见的几种评估指标。每个类提供了方法便于进行模型测试，保存并加载模型，记录时间，返回预测结果等功能。在**getscore.py**中，针对csv文件，进行数据预处理，多个回归模型训练，最终利用打分函数返回每一轮中选出的最优模型名字以及对应评价指标取值。这里对数据进行了预处理，增强了模型的弹性，可以针对数据集的规模灵活变化。并且，为了每次提供更多种类模型，将模型筛选规则定为可调的阈值。**Rundata.py**文件功能相似，负责读取规格化，处理规整后的数据集，支持了一次显示多种推荐的模型。**Produce\_data.py**为模型调试生成测试数据集。Trian文件是训练类，负责调度模型，读取用户设置的超参，进行模型训练并保存到**trainedModel.py**中，并提供方法可以返回输入文件所对应的标签列表。Test文件中调用了train函数，利用返回的标签列表和源数据，训练随机森林分类器，进行批量查询。查询过程中利用标签以及保存的模型进行查询，并提供了io占用检测，时间监测，以及纠错功能。训练时根据多次训练经验，启发式的确定错误波动范围很小，采用了在错误位置领域内遍历的方法进行纠错。

db目录下有四个文件，db.py为个人封装的sql操作，可以自行修改，main.py是连接调用的一个简单示例，参见.md文档，createbin.py（数据二进制存储），createdata.py（数据正常格式存储）文件均为构建训练数据集的函数，只是当时为了便于测试所写，对于RMI最顶层使用三次样条插值或是多项式拟合，仅需修正NN为相应模型即可