四．实验方法：

本部分，我们介绍进行实验的一些方法与策略。我们将本文提出的单层的自动选择模型，与2017年kraska在《The Case for Learned Index Structures》中提出的rmi递归索引模型。实验中比较的重要指标包括：索引学习准确率，内存占用，查询时间。其中：

准确率的度量采取打分的模式进行衡量。上述提到的两种学习索引方法，都是基于各种回归模型以及一些样条插值函数进行实现的。为了衡量模型效果，我们设计了打分函数，针对于一些用于分析回归模型的经典评估参数，例如中值，方差等进行了设计，其中包括：均方误差，绝对中值误差等。对于每一参数Si，我们设计打分函数score(i)进行分数评估。在最后计算每个模型Mi整体评分时，在每个参数Si已有得分scorei(Si)的基础上，赋予每个参数权重Wi，调整各项，使得当求和后总体得分score越小（接近于0的正数）时，模型效果越好。我们将求得的和作为损失的度量，准确率则是1-score。这一指标是一切指标的前提保障。

内存占用部分，我们对于查询过程进行了内存占用的监视计算。内存占用主要分为两大类：1. 数据读取；2. 利用训练好的模型中已有参数，输出查询结果。其中kraska模型较为特殊，它支持人为选定各层模型，或者由自身选择最优搭配。因此在这两种模式下，内存占用不同。实际上，采取最优搭配的方法在提升了自身准确率的前提下，同时提高了自身的内存占用以及需要额外的查询时间。

查询时间，记录下从数据读取完毕，进入模型进行学习索引过程，到最终得出被查询位置的最小相关字段（学习到的索引）所用的时间。在保障了查询准确率的基础上，必须使得查询时间最小化。需要注意的是，某些优化方法在保证准确率的前提下，牺牲了额外的查询时间。这一操作牺牲部分了正常正确查询的时间，当需要进行纠错，例如回表等对于误差进行处理时，能优化纠错时间。但是在接下来的实验中可以看到，对于大容量的工作负载，绝大多数查询仍是正确的，现有的方法需要在模型性能，查询准确率，纠错方法的选择，以及查询时间四者之间制定合理的权衡策略。

实验环境准备：本对比试验在cpu为i7-9050处理器，GTX1650 GPU，windows10环境下进行实验测试。其中对于kraska的rmi模型的测试，需要配置rust语言环境，在命令行模式下利用cargo命令进行测试并进行实验相关优化参数的设置。

实验数据的处理方法：我们采用了某网站上2018-2019年各种类书籍销售数据，包括书籍名字，书籍单价，书籍所属种类，书籍出售数目，总体销售金额等。数据大小约为300M，涵盖10000000条以上数据，并且处理为整形数据集和浮点型数据集，涵盖了整形以及浮点型的查询需求。我们还提取了wiki百科中2008-2012年间用户查询的记录，数据类型为字符串类型查询，大小约为1.4G。我们对于不同大小的工作负载，针对不同的数据分布生成了不同的数据集，保证涵盖高斯分布，指数分布，均匀分布，随机分布等基本分布类型。至此，我们用于训练输入的数据就从数据类型，数据分布，数据量上，尽可能测试现实世界中的数据集的各种情形。其中，为了增加数据真实性，我们在自主生成数据的过程中也加入了噪声。

不同模型对于数据的处理方式：本文中提出的单层模型学习索引，首先读取上文中提到的全部数据用于训练。我们将输入的数据分成多组，分别进行训练。对于每组读取进来的数据，分别训练我们提供的所有模型。这些模型包括：线性回归，逻辑回归，多项式回归，弹性网络回归，决策树回归，极限树回归，梯度提升树回归，还有一些其他的bagging族的一些回归模型。数据进入模型进行训练，得出每个模型的一系列评估参数。利用这些评估参数，进行打分的计算。对于一轮输入的数据，会得到该轮的所有模型的打分列表。进行排序之后，选取小于人为设定阈值的，几个排序后表现优秀的模型，对于这组数据打上取的模型标号的标签。对于下一轮数据，重复上述过程，直到标注完所有数据。修改后的数据被保留，当查询到来时，直接根据数据的标签，进入对应模型序号的，训练好的模型。利用已有的模型参数，迅速计算出最优模型的输出。

对于kraska的rmi模型，读入数据类型为二进制数据。数据内部有特殊符号作为key（X）和相关字段postion（y）的分隔符。其中X与y是用于拟合模型，进行fit（X，y）操作的两个输入。这种二进制文件读取速度较快，文件处理过程占用内存较小，但是修改，纠错代价消耗更高，需要额外的数据还原过程，回到例如csv等文件形式，进行数据的标注，修改，或者查询后的回表核对，纠错等操作。

对于多层回归的层次结构模型，没有对于数据的二次加工过程，直接读取全部数据进行模型训练学习，模型顶层是两层的nn，由16\*16的全连接层组成，下面是由三阶样条函数和 通过层次的递归结构，利用各层之间的输出，逐层缩小相关字段的范围，并最终直接输出最小的相关字段。这一方法由于需要直接读取全部数据，占用内存较大，处理，训练过程较慢。并且由于层次结构的存在，存储的参数也相对较多。由于不存在数据二次处理步骤，训练后的数据不用保存，查询到来后，直接通过保存的参数就可以得到学习索引之后的相关字段。

查询过程对比实验：

1. 内存占用对比试验：
2. 准确率对比试验：
3. 查询时间对比试验：