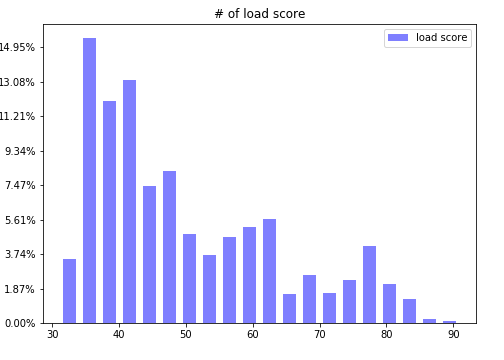
**第一部分：**

在不同数据库中，特征和指标的分布（53522）：

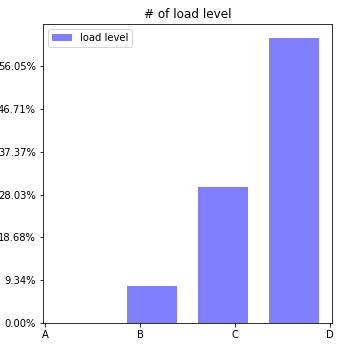
**① 频度分布**

所有库的数据分布统计：

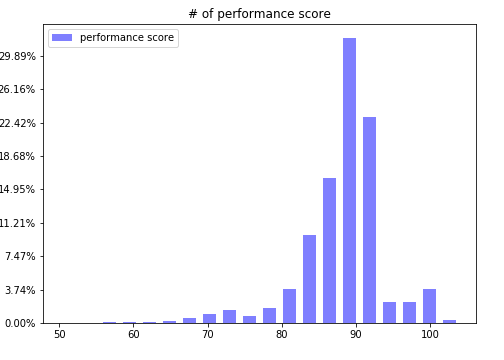
Load-Score:



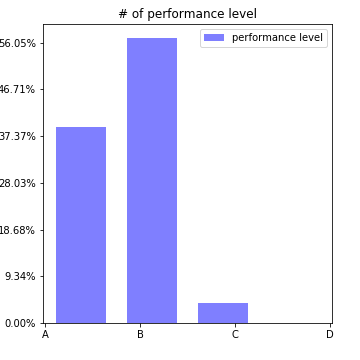
Load-Level:



Perf-Score:



Perf-Level:

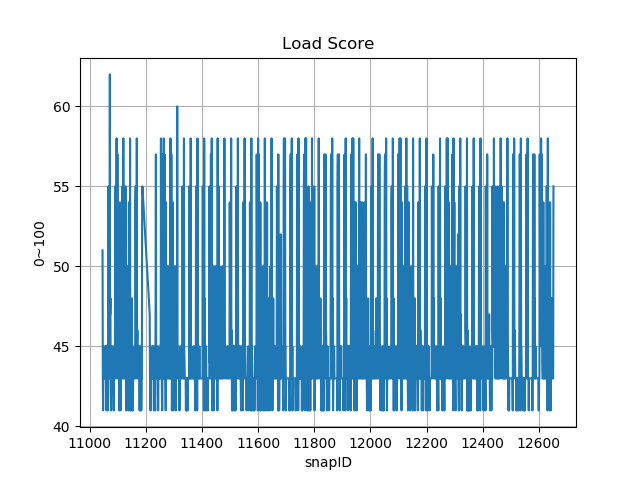


**② 时序分布**

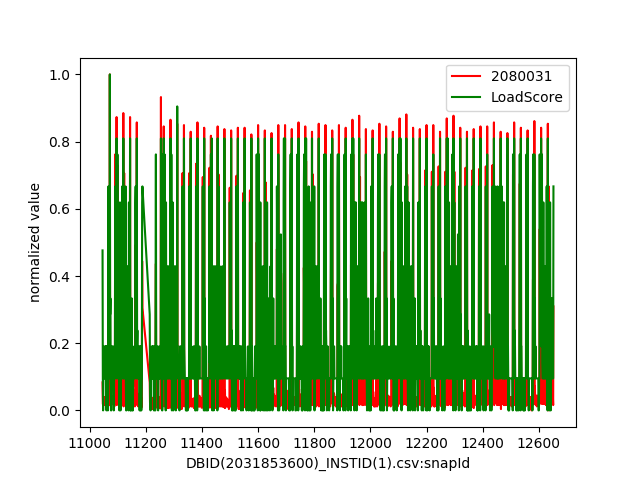
多数数据库数据文件，从08-07的0点开始记录，每一小时采集一次，10-09的23点结束，每个数据文件一般有1510条数据（中间08-09日没有记录）。不同数据库的数据文件的分布不尽相同，这里举例给出具体的给定的DBID数据文件。

Load-Score:

DBID = 2031853600



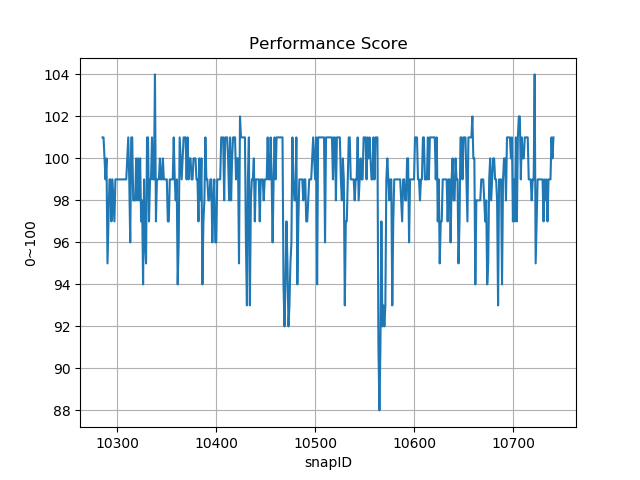
特征编号“2080020”到“2080034”等15个特征与Load-Score相关。最大信息系数方法选择"2080031"(每秒SQL执行次数)作为与Load-Score相关的最重要特征。



绿色曲线为打分的Load-Score，红色曲线为选择的最重要特征时序分布。从曲线走势看，两者具有很高的正相关性，另外绿色曲线在局部有对红色曲线做平滑处理的效果。

Perf-Score:

DBID = 1227435885 (1)



特征编号“2080040”到“2080065”等26个特征与Perf-Score相关。最大信息系数方法选择" 2080049"( CPU占DB时间比例（%）)作为与Perf-Score相关的最重要的特征。

绿色曲线为打分的Perf-Score，红色曲线为选择的最重要特征时序分布。从曲线走势看，两者具有很高的正相关性，另外绿色曲线在局部有对红色曲线做平滑处理的效果。

当然不同数据库数据文件的特征和指标的分布不尽相同，特征选择的结果也不同，我们选取了50个数据库数据文件的各自的Load-Score-Top10特征和Perf-Score-Top10特征，并统计了各个特征被选取的总次数，依此选取了总体的两种Score下的Top10特征。详细的结果参考文件：MIC\_result.csv。

Load-Score-Top10:



Perf-Score-Top10:



两个表格的第一行为特征选择Top10结果，第二行为每个特征对应在50个数据库数据文件中被选取的总次数。

特征选择主要的作用还是能从大维度的数据中选择出对任务有不可忽略的影响的特征；所以下一步的工作在原始数据库数据文件上做特征选择。另外考虑数据的时序性，可以采用基于时序的特征选择的方法。

**第二部分：**

非时序的回归与分类

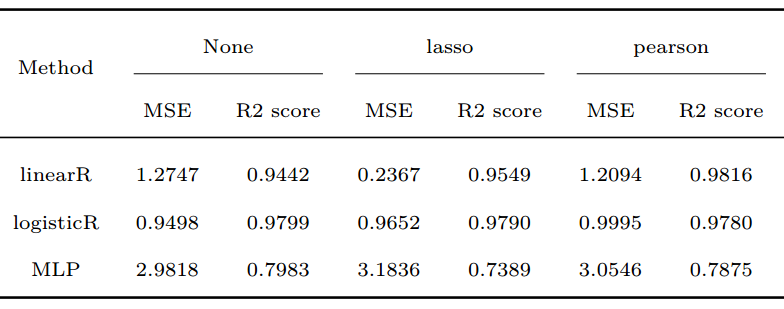
由于依据经验模型挑选的41个特征都是很重要的，这里做特征选择主要是：如第一部分所述对重要的特征进行排序；另一方面比较用少量的特征做回归和分类对性能的影响。由于这里的特征选择结果不再具有稀疏性，我们选择前90%的特征，具体的讲：Load用13个特征；Performance用23个特征。

**①** 对Score数据进行回归，回归的特征选择Lasso，Pearson

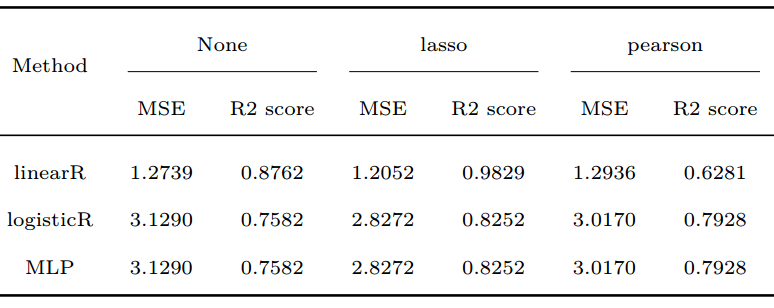
数据归一化下：MAE平均绝对误差，R2-score它提供了一个衡量未来样本很可能被模型预测的度量值。最佳的可能的比分是 1.0，可以任意的差（负数）。

回归采用linear-R（线性回归）、用sig-mod函数构造的logistic-R（逻辑回归）、MLP（多层感知机-神经网络）

Load-Score:



Perf-Score:



\*（高维数据的表示）

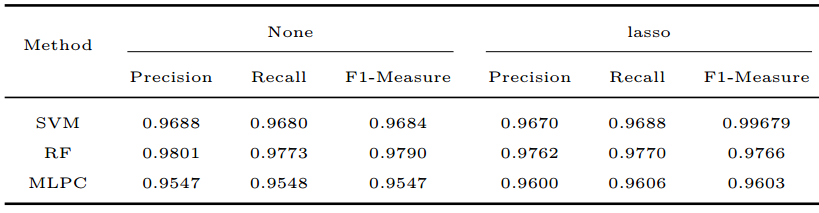
**②** 对Level数据进行分类，分类的特征选择采用relief-F方法，用Lasso和Pearson时，将多标签[‘A’, ‘B’, ‘C’, ‘C’]转换为[0, 1, 2, 3]。

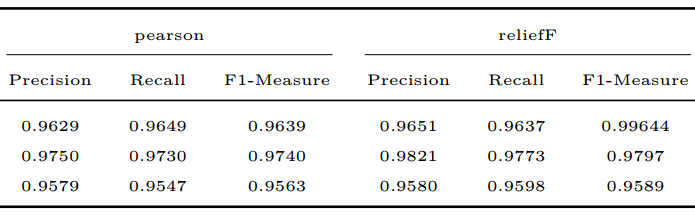
分类标签A~D: Precision准确率，Recall召回率、F1-Measure可以解释为加权平均准确率和召回率, 最佳值为1，最差值为0。

F1 = 2 \* (precision \* recall) / (precision + recall)

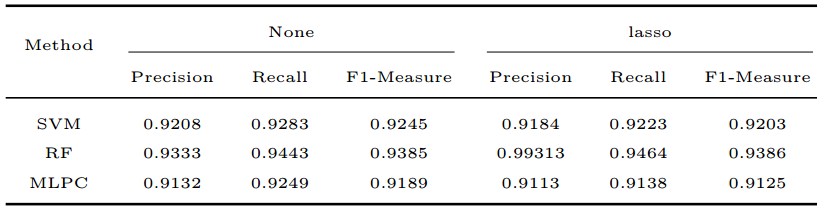
多标签分类采用：SVM（支持向量机）、RF（随机森林）、MLPC（基于多层感知机的分类器）

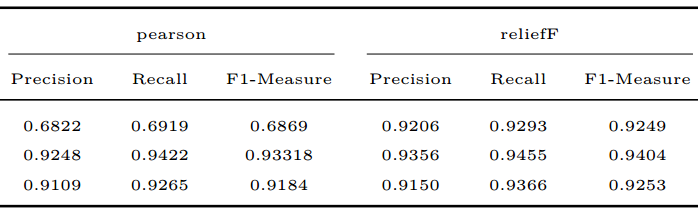
Load-Level:





Perf-Level:





**第三部分：**

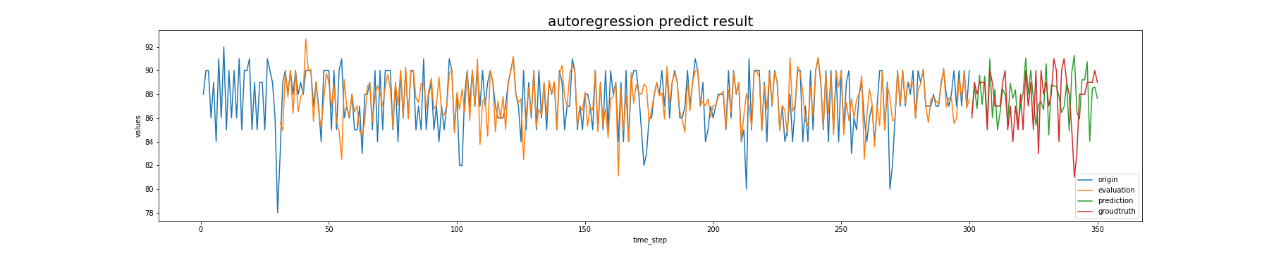
基于时序的回归与分类预测。

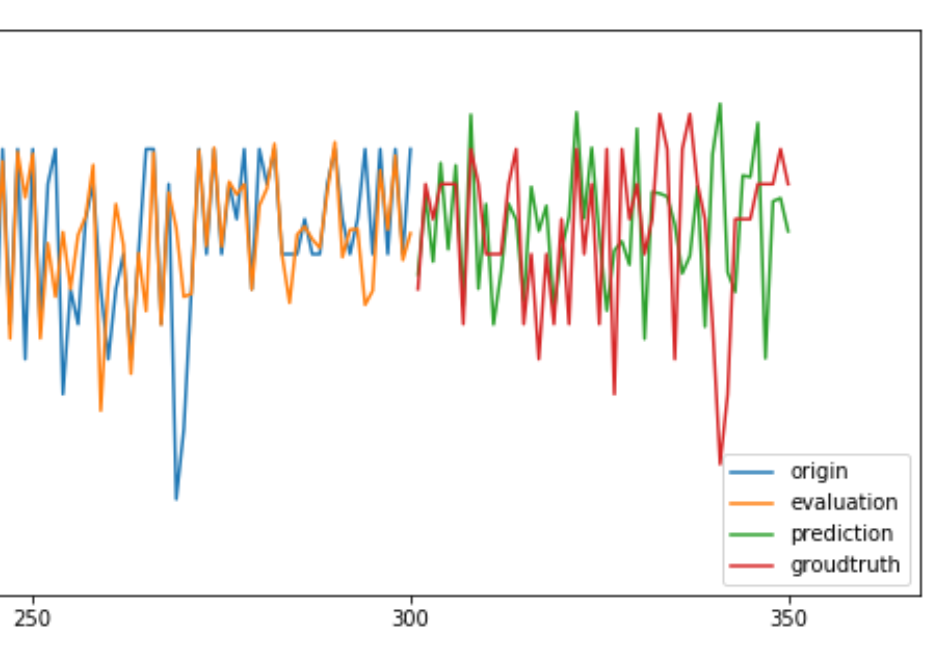
数据库数据文件中样本是时序采样的，具有时序性，做时序模型也便于后续的异常预测的需求。这里采用两种时间序列预测模型：Auto-Regression（自动回归模型）、Long-Short-Term-Memory（长短记忆网络）。

测试数据库数据文件：DBID = 172908691 (1) ；输入全部（41）维度的数据，对Perf-Score进行预测。

**① Auto-Regression**

在时间序列上，先后输入前300个样本（12.5天），则可以对后续的50个样本点的Perf-Score进行预测（2天），结果如下：





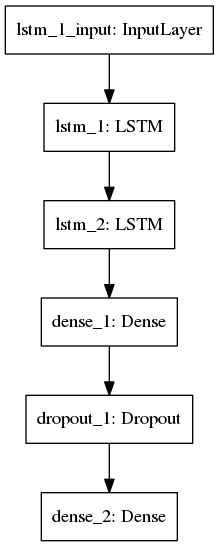
时间点300之前：蓝色曲线为原始Perf-Score数据，橙色曲线为AR模型拟合值；时间点在301~350之间：红色曲线为真实值，即原始Perf-Score数据，绿色曲线为AR模型预测值。

对同一个数据库文件，进行切分，然后做5组上述基于AR模型的时间序列预测。测试的结果如下：

AR模型的预测误差均值为2.037（2.0372412109375002），预测的R2-Score=-0.3982（-0.39822152014623935）。从图中也可以看出预测的误差很大，所以可以认为这样的方法下，长时间的预测是无效的。因此，我们考虑向后预测一个小时的情形：

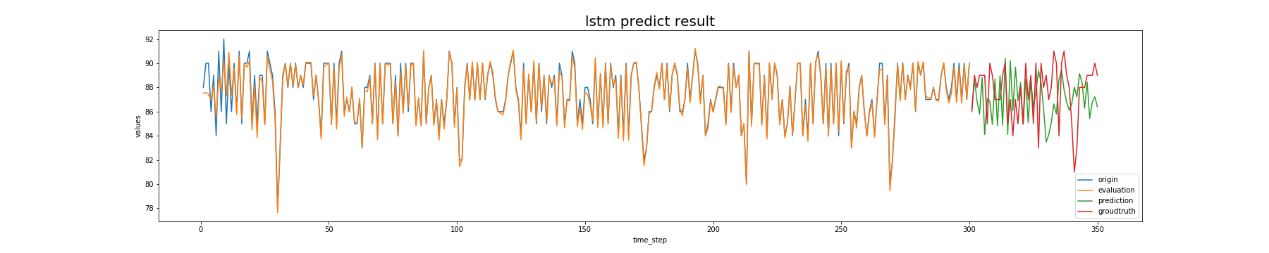
在时间序列上，先后输入前50个样本（2天），则可以对后续的1个样本点的Perf-Score进行预测（1小时），结果如下：

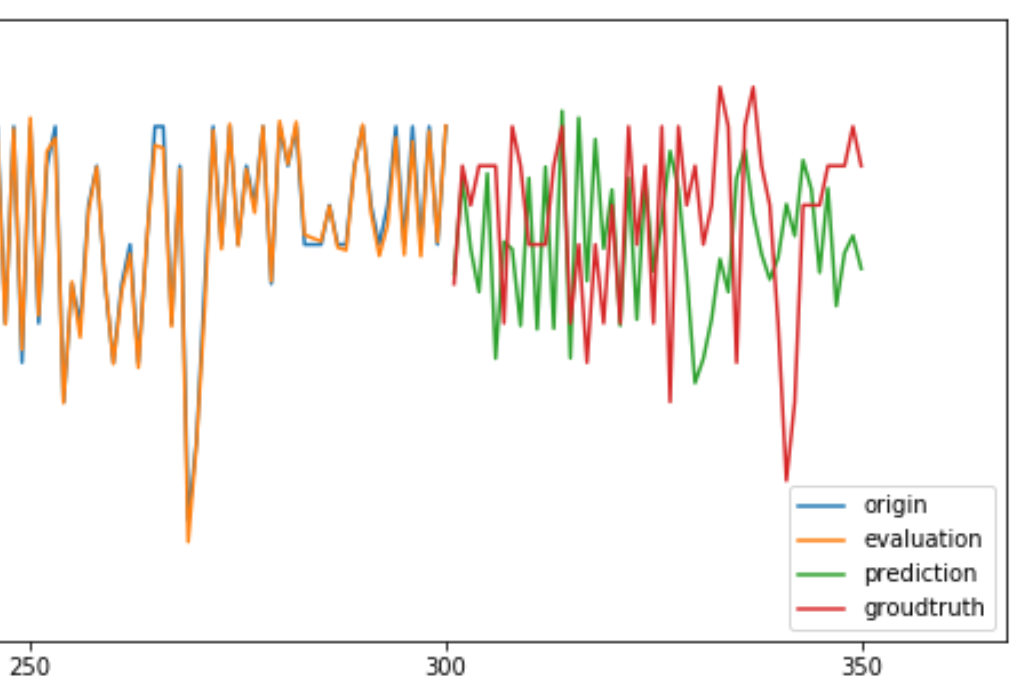
**② LSTM**



长短记忆网络采用右图的整体结构：输入为全部41维度的样本点，然后是两层的LSTM网络层，

在时间序列上，先后输入前300个样本（12.5天），则可以对后续的50个样本点的Perf-Score进行预测（2天），结果如下：





时间点300之前：蓝色曲线为原始Perf-Score数据，橙色曲线为AR模型拟合值；时间点在301~350之间：红色曲线为真实值，即原始Perf-Score数据，绿色曲线为AR模型预测值。

同样的，对同一个数据库文件，进行切分，然后做5组上述基于LSTM模型的时间序列预测。测试的结果如下：

LSTM模型的预测误差均值为1.713（1.7127156066894531）； 预测的R2-Score为0.01288（0.012876979251902898）。从图中也可以看出预测的误差很大，所以可以认为这样的方法下，长时间的预测是无效的。因此，我们考虑向后预测一个小时的情形：

在时间序列上，先后输入前50个样本（2天），则可以对后续的1个样本点的Perf-Score进行预测（1小时），结果如下：

依据各阈值对预测值进行归类，分类正确率为91.8%（0.9178082191780822）。

**第四部分：**

从原始数据库数据文件中选出不同的或更多的特征，再次评估上述回归和分类的效果。