神经网络何时能胜过提升树表格数据？

邓肯·麦克尔弗雷什*[1, 2]，苏杰·坎达加勒[3]，乔纳森·瓦尔韦德[4]，维沙克·普拉萨德[5]，本杰明·费尔[6]，钦梅赫格德[6]，加内什·罗摩克里希南[5]，迈卡·戈德布鲁姆[6]，科林·怀特[1, 7]

[1]Abacus.AI，[2]斯坦福，[3]Pinterest，[4]马里兰大学
[5]印度孟买理工学院，[6]纽约大学[7]加州理工学院

抽象的

表格数据是机器学习中最常用的数据类型之一。尽管最近表格数据的神经网络 (NN) 取得了进展，但关于 NN 在表格数据上的表现是否通常优于梯度提升决策树 (GBDT) 的讨论仍在继续，最近的几篇论文认为 GBDT 在表格数据上的表现始终优于 NN，反之亦然。在这项工作中，我们退一步思考，质疑这场争论的重要性。为此，我们进行了迄今为止最大规模的表格数据分析，比较了 176 个数据集中的 19 种算法，我们发现“NN 与 GBDT”之争被过分强调了：对于数量惊人的数据集，要么 GBDT 和 NN 之间的性能差异可以忽略不计，要么对 GBDT 进行轻度超参数调整比在 NN 和 GBDT 之间进行选择更重要。一个显著的例外是最近提出的先验数据拟合网络 TabPFN：尽管它实际上仅限于大小为 3000 的训练集，但我们发现它的平均表现优于所有其他算法，即使在随机抽样 3000 个训练数据点时也是如此。接下来，我们分析了数十个元特征，以确定什么*特性*数据集的复杂度使得 NN 或 GBDT 更适合表现良好。例如，我们发现 GBDT 在处理倾斜或重尾特征分布和其他形式的数据集不规则性方面比 NN 好得多。我们的见解可以作为从业者的指南，以确定哪些技术可能最适合他们的数据集。最后，为了加速表格数据研究，我们发布了 TabZilla 基准套件：我们研究的 36 个“最难”数据集的集合。我们的基准套件、代码库和所有原始结果可在 https://github.com/naszilla/tabzilla。

1 简介

表格数据集是按行和列组织的数据，由通常为连续、分类或序数的不同特征组成。它们是实践中机器学习中最古老、最普遍的数据集类型之一。[8，61]，由于其在医学领域的广泛应用[三十八，64]， 金融 [6，14]， 在线广告 [二十九，四十八，56]，以及许多其他领域[10，11，65]。

尽管神经网络（NN）架构在表格数据方面取得了进展[5，53]，关于 NN 在表格数据上的表现是否普遍优于梯度提升决策树 (GBDT) 仍然存在激烈的争论，有多项研究认为 [5，三十九，43，53，57] 或反对 [8，二十七，二十八，61] 神经网络。这与计算机视觉和自然语言理解等其他领域形成了鲜明对比，在这些领域，神经网络的发展速度远远超过了竞争方法。[9，19，20，四十二，67]。

*电子邮件：duncan@abacus.ai ，crwhite@caltech.edu 。SK 和 JV 在 Abacus.AI 时完成的工作。

第 37 届神经信息处理系统会议（NeurIPS 2023）数据集和基准跟踪。

几乎所有先前的表格数据研究都使用了少于 50 个数据集，或者没有正确调整基线[四十七，63]，这些发现的普遍性受到质疑。此外，许多先前研究的底线是回答“就数据集的平均排名而言，NN 和 GBDT 哪个表现更好”的问题，而不是寻找更细致的见解。

在本文中，我们采取了一种完全不同的方法，重点关注以下几点。首先，我们通过研究算法选择的重要性，质疑“NN 与 GBDT”之争的重要性。其次，我们分析了*特性*数据集使得 NN 或 GBDT 更适合表现良好。我们采用数据驱动的方法来回答这些问题，通过比较进行迄今为止最大的表格数据分析19 种算法，每种算法最多有 30 个超参数设置，涵盖 176 个数据集，包括来自 OpenML-CC18 套件的数据集[7] 和 OpenML 基准测试套件 [二十六]。为了评估不同数据集之间的性能差异，我们考虑了数十个元特征。我们对每个数据集使用 10 倍，以进一步降低结果的不确定性。

我们发现，对于相当一部分数据集，非常简单的基线算法的表现与顶级算法相当；此外，对于大约三分之一的数据集，在 CatBoost 或 ResNet 上进行轻度超参数调整比在 GBDT 和 NN 之间进行选择更能提高性能。这些结果表明，对于许多表格数据集，没有必要尝试许多不同的 NN 和 GBDT：在许多情况下，强大的基线或经过良好调整的 GBDT 就足够了。

我们确实发现，在所有 176 个数据集中，表现最好的两种算法是 CatBoost [54] 和 TabPFN [33]。后者的性能尤其有趣，因为 TabPFN 是一个最近提出的先验数据拟合网络，可以在不到一秒的时间内对小数据集进行训练和推理，但其运行时间和内存使用量与训练样本的数量成二次方关系。令人惊讶的是，我们表明，即使对于大型数据集，随机选择 3000 个训练数据点也足以让 TabPFN 实现非常强大的性能。

接下来，我们进行分析，以发现*特性*数据集解释了哪些方法或方法系列成功或失败。我们计算了各种元特征与算法性能的相关性，并证明了这些相关性具有预测性。我们的主要发现如下（另见图 5）：数据集*规律性*预测 NN 的表现优于 GBDT（例如，特征分布的倾斜度和重尾性较小）。此外，GBDT 在较大的数据集上往往表现更好。

最后，为了加速表格数据研究，我们发布了TabZilla 基准测试套件： 我们研究的 176 个数据集中“最难”的数据集的集合。我们选择简单基线无法取胜的数据集，以及大多数算法无法达到最佳性能的数据集。

我们的工作为处理表格数据的研究人员和从业者提供了大量工具。我们在一个界面中提供了最大的算法和数据集开源代码库，以及一组“最难”的数据集和原始结果（超过 500K 个经过训练的模型），https://github.com/naszilla/tabzilla让研究人员和从业者能够更轻松地比较方法。最后，研究人员可以利用我们的元特征洞察来发现表格算法的失败模式，而从业者则可以利用我们的元特征洞察来帮助确定哪些算法在新的数据集上表现良好。

我们的贡献。我们的主要贡献总结如下：

- 我们对表格数据进行了迄今为止最大规模的分析，在 176 个数据集上比较了 19 种方法，训练了超过 50 万个模型。我们发现，对于相当一部分数据集，简单的基线表现最佳，或者对 GBDT 进行轻度超参数调整比在 NN 和 GBDT 之间进行选择更重要，这表明“NN 与 GBDT”之争被过分强调了。
- 在分析了数十个元特征之后，我们对使数据集更适合 GBDT 或 NN 的属性提出了一些见解。
- 我们发布了 TabZilla Suite：一套 36 个“硬”数据集，旨在加速表格数据研究。我们开源了基准套件、代码库和原始结果。

相关工作。表格数据集是机器学习实践中最古老、最常见的数据集类型之一。8，61]，由于其应用范围广泛[6，11，14，三十八，56GBDT [22] 迭代构建决策树集合，每棵新树拟合先前树的损失残差，使用梯度下降来最小化损失。XGBoost [十三], LightGBM [40]和Catboost [54] 是三种广泛使用的高性能变体。Borisov 等

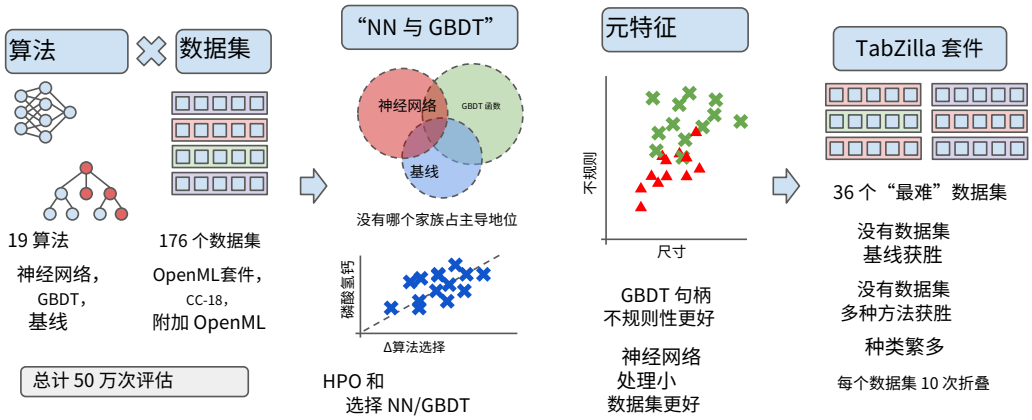

图 1：我们工作的概述。我们首先对表格数据进行迄今为止最大规模的研究（左）；我们分析了算法选择（“NN 与 GBDT”）以及元特征的重要性（中间）；并根据我们的研究，发布了 TabZilla，这是最难的表格数据集的集合。

等人描述了神经网络的三种表格数据方法[8]: 数据转换方法[三十，70]，基于架构的方法[5，12，二十九，53]（包括变压器[二十七，三十六，62]）以及基于正则化的方法[三十七，三十九，60]。最近的几篇论文将 GBDT 与表格数据的 NN 进行了比较，结果发现*任何一个*NNs [5，二十七，三十九，53]*或者*GBDT [8，二十七，二十八，61] 表现最佳。

也许与我们最相关的工作是由 Grinsztajn 等人完成的。[二十八]，他们研究了为什么基于树的方法在表格数据上的表现优于神经网络。他们的工作与我们的工作有一些不同。首先，他们考虑了 7 种算法和 45 个数据集，而我们考虑了 19 种算法和 176 个数据集。其次，他们的数据集大小范围从 3 000 到 10 000，或者 7 个正好是 50 000，而我们的数据集大小范围从 32 到 1 025 009（见表6）。此外，他们还进一步控制了他们的研究，例如通过限制大小与特征的比率，删除高基数分类特征，以及删除低基数数值特征。虽然这有利于成为一项更可控的研究，但他们的分析遗漏了我们的一些观察结果，例如 GBDT 在“不规则”数据集上的表现优于 NN。最后，虽然 Grinsztajn 等人深入研究了一些元特征，例如数据集平滑度和无信息特征数量，但我们的工作考虑了更多数量级的元特征。同样，虽然每种方法都有自己的优势，但我们的工作能够为从业者发现更多潜在的见解、相关性和收获。据我们所知，唯一考虑了超过 50 个数据集的相关工作是 TabPFN [33]，其中考虑了 179 个大小为 2 000 或更小的数据集。参见 附录 C进行更长时间的讨论

相关工作。

二　阿里go 的姐妹 律动对于表格 D 艾塔

在本节中　n，埃贝西亚t 一个大 规模研究科技　泥曲是　r 涵盖多种对象的表格数据集五里吨数据集。　r　n 升s我试图回答以下两个问题　问题达

1. gor tm 怎么做我l　s （和算法　家庭）c　比较　阿克塞种各样的数据集？
2. 白色 在广告属性　ataset 与表现优于其他算法（和家族）相关吗？

阿尔戈里米数据和数据集　实施的。我们展示了 19 种算法的结果，包括流行的 elines。这些方法最近的特征技艺精湛　包括三种 GBDT：CatBoost [54],LightGBM 11神经网络 [40]，A nd XGBoost[十三]；　l 网络：DA　网 [12]，FT-Transformer [二十七]，两个MLP [二十七]，节点　[53]，ResNet[二十七]，　圣人 [62]、STG [69], TabNet [5], 标签PFN [33]和 VIME [70]；以及五基线：决策 Tr　ee[55]，KNN[17]，逻辑回归[18]，随机森林[四十五] 和 SVM 算法，因为 [16]我们选择这些　它们的流行性、多样性和强大的性能。对于 TabPFN，运行时和内存使用量与 输入（即训练样本）。为了在大小大于 3000 的数据集上运行，我们只需从完整训练数据集中随机抽取大小为 3000 的样本。我们将此变体表示为 TabPFN *。

表 1：98 个数据集上的算法性能（见表 11以获得扩展结果）。列显示算法系列（GBDT、NN、PFN 或基线）、所有数据集的排名、平均归一化准确度（平均准确度）、各折叠归一化准确度的标准差（标准准确度）以及每 1000 个实例的训练时间（秒）。这些数量的最小值/最大值/平均值/中位数取自所有数据集。

算法	班级	秩			平均加速度。			标准配件		次/1000实例	
		最小值	最大值	平均值	平均值	平均值	平均值			意思是	医学。
CatBoost	国别分类法	1	19	6.12	5	0.87	0.93	0.30	0.22	21.70	2.08
表PFN*	PFN	1	19	6.43	5	0.84	0.92	0.26	0.19	0.25	0.01
XGBoost	国别分类法	1	19	7.85	6.5	0.81	0.89	0.33	0.22	0.81	0.37
残差网络	神经网络	1	21	8.73	8	0.75	0.83	0.30	0.21	16.01	9.34
节点	神经网络	1	21	9.08	9	0.74	0.81	0.26	0.20	138.36	117.04
圣	神经网络	1	21	9.09	8	0.73	0.86	0.31	0.24	169.54	146.16
FTTransformer	神经网络	1	20	9.29	9	0.76	0.80	0.31	0.21	27.67	18.40
随机森林	根据	1	21	9.5	9	0.76	0.83	0.32	0.22	0.35	0.24
轻量级GBM	国别分类法	1	21	9.61	9	0.76	0.84	0.36	0.21	0.87	0.34
支持向量机	根据	1	20	10.34	11.5	0.69	0.76	0.26	0.19	30.40	1.67
数据网络	神经网络	1	20	11.07	11	0.73	0.79	0.32	0.23	68.82	60.15
MLP-RTDL	神经网络	1	21	11.10	12	0.65	0.72	0.28	0.16	14.27	7.30
星火	神经网络	1	21	13.19	14	0.56	0.63	0.29	0.17	18.44	15.79
决策树	根据	1	21	13.32	15	0.59	0.68	0.35	0.25	0.03	0.01
多层感知处理器	神经网络	1	21	13.49	15	0.57	0.57	0.29	0.18	18.39	11.20
线性模型	根据	1	21	13.76	16	0.51	0.53	0.31	0.24	0.04	0.03
塔格网	神经网络	1	21	14.24	16	0.54	0.60	0.39	0.25	34.95	29.90
KNN	根据	1	21	15.32	17	0.45	0.51	0.29	0.21	0.01	0.00
维梅	神经网络	3	21	16.73	19	0.37	0.32	0.27	0.18	16.81	14.86

我们在 OpenML 的 176 个分类数据集上运行了该算法。66我们的目标是纳入最近研究表格数据的热门论文中的大多数分类数据集 [8，二十七，三十九，61]，包括来自 OpenML-CC18 套件的数据集[7]，OpenML 基准测试套件 [二十六] 和其他 OpenML 数据集。由于我们的实验规模（总共训练了 538 650 个模型），我们限制了每个实验的运行时间（如下所述），这排除了使用大于 1.1M 的数据集。表 6显示所有数据集的汇总统计信息。CC-18 和 OpenML Benchmarking Suite 均被视为对算法进行公平、多样化评估的首选标准，因为它们具有严格的选择标准和广泛的数据集多样性。[7，二十六据我们所知，我们的 19 个算法和 176 个数据集是*任何一个*算法*或者*数据集（TabPFN 除外 [33]) 是最近的表格数据集文献中考虑的最大数量，也是单个开源存储库中可用的最大数量。

元特征。我们使用 Python 库 PyMFE 提取元特征 [4]，其中包含 965 个元特征。元特征的类别包括：“一般”（例如数据点、类别或数字/分类特征的数量）、“统计”（例如所有特征分布的最小、平均或最大偏度或峰度）、“信息论”（例如目标的香农熵）、“标志”（基线（例如 1-最近邻）在数据集子样本上的表现）和“基于模型”（某些模型对数据的拟合的汇总统计数据，例如决策树模型中的叶节点数量）。由于其中一些特征具有长尾分布，我们还在分析中包括每个严格正元特征的对数。

实验设计。对于每个数据集，我们使用 OpenML 提供的十个训练/测试折叠，这样我们就可以将测试折叠的结果与使用相同 OpenML 数据集的其他工作进行比较。由于我们还需要验证拆分才能运行超参数调整，因此我们将每个训练折叠分为训练和验证集。对于每个算法和每个数据集拆分，我们运行算法长达 10 小时。在此期间，我们使用最多 30 个超参数集（一个默认集和 29 个随机集，使用 Optuna [3]）。每个参数化算法在 32GiB V100 上最多需要两个小时才能完成一次训练/评估周期。与之前的工作一致，我们感兴趣的主要指标是*准确性*，我们报告在验证集上具有最大性能的超参数设置的测试性能。我们还考虑了对数损失，它与准确率高度相关，但包含的关系明显较少。我们还包括 F1 分数和 ROC AUC 的结果附录 D.与之前的工作类似[21，68]，无论何时

表 2：算法在 57 个大小小于或等于 1250 的数据集上的性能。列显示所有数据集的排名、平均归一化准确度 (Mean Acc.)、各折叠归一化准确度的标准差 (Std. Acc) 以及每 1000 个实例的训练时间。最小值/最大值/平均值/中位数取自所有数据集。

算法	秩 分钟	最大限度	意思是	医学	平均加速度。 意思是	医学。	标准配件 意思是	医学	次/1000实例 意思是	医学
表PFN*	1	18	4.88	3	0.84	0.93	0.35	0.26	0.00	0.00
CatBoost	1	18	5.37	4	0.85	0.91	0.39	0.30	26.22	2.75
残差网络	1	19	6.75	6	0.77	0.79	0.42	0.30	23.67	13.87
随机森林	1	18	7.65	7	0.76	0.82	0.40	0.29	0.47	0.32
圣	1	19	7.67	6	0.74	0.87	0.42	0.31	197.41	181.62
FTTransformer	1	18	7.93	7	0.75	0.78	0.42	0.32	32.93	26.39
XGBoost	1	17	8.30	8	0.74	0.80	0.42	0.30	0.95	0.61
节点	1	19	8.35	8	0.73	0.75	0.36	0.28	173.55	144.45
支持向量机	1	18	9.54	11	0.68	0.72	0.35	0.28	23.90	0.42
MLP-RTDL	1	19	9.77	10	0.64	0.69	0.39	0.31	21.48	12.21
轻量级GBM	1	19	10.00	10	0.68	0.71	0.45	0.38	0.64	0.23
线性模型	1	19	10.21	11	0.61	0.71	0.38	0.29	0.06	0.05
数据网络	1	18	10.74	10	0.68	0.69	0.41	0.34	83.57	71.19
决策树	1	19	11.44	十三	0.60	0.67	0.45	0.32	0.02	0.01
多层感知处理器	1	19	11.49	十三	0.57	0.54	0.38	0.30	27.88	16.81
星火	1	19	11.49	12	0.57	0.64	0.40	0.34	21.22	18.24
KNN	1	19	13.12	15	0.46	0.51	0.38	0.32	0.00	0.00
塔格网	3	19	14.54	16	0.42	0.40	0.52	0.49	41.83	34.35
维梅	2	19	14.88	17	0.33	0.27	0.36	0.29	18.95	16.43

我们对数据集进行平均，使用到最小值 (ADTM) 指标的平均距离，该指标由 0-1 缩放组成（在选择最佳超参数之后，这有助于防止出现异常值 [二十八]）。最后，为了查看每种方法在同一数据集的不同折叠上的方差，我们报告了每种方法在所有 10 个折叠上的平均（缩放）标准差。

2.1 相对算法性能

在本节中，我们将回答以下问题：“单个算法和算法系列在各种数据集上的表现如何？”我们特别考虑 GBDT 和 NN 之间的差异是否显著。

没有哪种单独的算法能够占据主导地位。我们首先比较所有数据集上所有算法的平均排名，同时排除使用上述实验设置时遇到大量算法内存或超时问题的数据集。因此，我们考虑一组 98 个数据集（我们将在下一节和下一节中包括所有 176 个数据集的结果）。附录 D.2）。如上一节所述，对于每个算法和数据集拆分，我们报告在验证集上进行调整后的测试集性能；参见表格1。

出奇，*几乎每个算法在至少一个数据集上排名第一，在至少一个其他数据集上排名最后*。正如预期的那样，基线方法往往表现不佳，而神经网络和 GBDT 的平均表现则更好。事实上，所有算法中最好的 CatBoost 仅获得了 5.06 的平均排名，这表明没有一种方法可以在大多数数据集中占据主导地位。

TabPFN 的表现非常出色。在表格1，我们发现 TabPFN 的性能几乎与 CatBoost 相同。这尤其令人惊讶，原因有二。首先，TabPFN 每 1000 个实例的平均训练时间为 0.25 秒：是所有非基线中最快的，比 CatBoost 快两个数量级。其次，回想一下，为了在大型数据集上运行 TabPFN，我们只需使用来自完整训练数据集的大小为 3000 的随机样本运行 TabPFN。尽管对于许多数据集，TabPFN 仅看到训练数据集的一部分，但它仍实现了近乎顶级的性能。

接下来，在表 2，我们针对 57 个最小数据集计算同一张表：大小最多为 1250 的数据集。现在，我们发现 TabPFN 在所有算法中取得了最好的平均性能，同时训练时间也最快。然而，需要注意的是*推理*TabPFN 的时间是

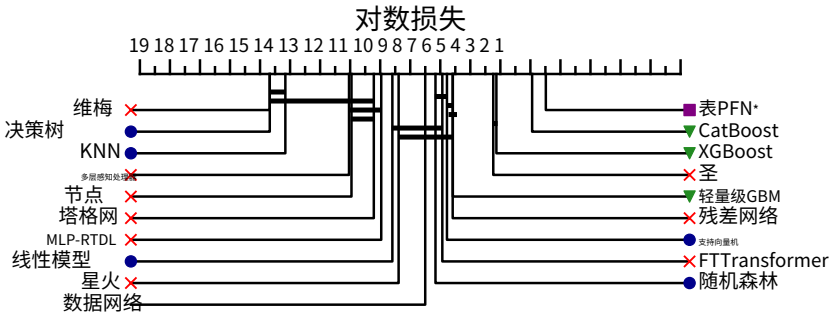

图 3：根据 98 个数据集上的平均对数损失等级比较所有算法的临界差异图。每个算法的平均等级在轴上显示为一条水平线。*没有显著差异*由水平黑条连接。算法系列由算法名称旁边的标记表示：红色“X”表示神经网络，蓝色圆圈表示基线算法，绿色三角形表示 GBDT，紫色方块表示 PFN。

每 1000 个实例平均耗时 2.36 秒，高于其他算法。我们在 TabPFN 上给出了进一步的讨论和结果附录 D.3，包括针对 TabPFN 和 CatBoost 随机抽取 1k 或 3k 训练点的消融研究。

性能与运行时间。在图 2，我们绘制了所有算法的准确率与运行时间的关系图，取所有数据集的平均值。总体而言，神经网络需要最长的运行时间，并且通常优于基线方法。另一方面，GBDT 同时需要很少的运行时间，同时还能实现强大的性能：它们始终优于基线方法，

并且始终需要更少的运行时间比 NN 更好。我们再一次看到了 TabPFN 的出色表现，它比任何 GBDT 或 NN 都能以更短的训练时间实现最高的准确率。

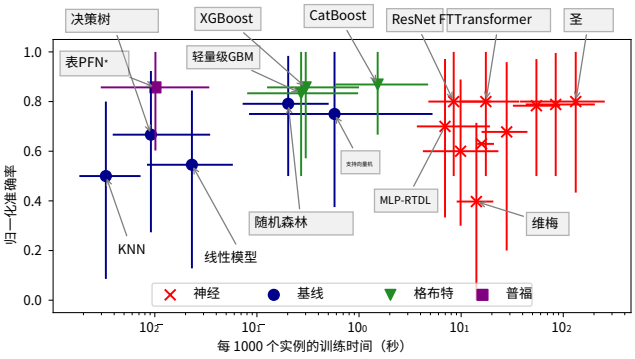

图 2：98 个数据集中每种算法的运行时间中位数与归一化准确率中位数。条形图涵盖所有数据集的第 20 到第 80 个百分位数。

统计上显著的表现
曼斯差异。表格1节目
很多 算法有 类似 表现 .否接下来，我们确定 埃粒米内*统计上 点火 义重大*（页 <0。05）
在上述 98 个数据集上，算法之间的性能差异。首先，我们使用 Friedman 检验来确定每个算EA
法之间的性能差异是否显著。23]；我们可以拒绝原假设（p<0。05）对于此测试；p 值小于 蚂蚁
10-20我们使用 Wilcoxon 符号秩检验来确定哪些算法对具有显著的性能差异（p<0。05）[15母鸡
]，我们使用 Wilcoxon 检验和 Holm-Bonferroni 校正来计算 nce
乌特
进行多重比较 [三十五]。由于测试准确率指标中存在许多平局，因此使用测试对数损失指 我们
标。请参阅图 3（并参见图 9Fo r 是 F1 分数）。在这些图中， 这
每种算法的平均排名 hm 显示为o在房子里 水平轴；如果算法之间存在差异 是
不重要（页 ⩾0。05）然后算法通过水平线连接起来。我们发现 TabPFN 的平均表现优于所有 唱子
其他算法罗斯 98 天 数据集，这个结果是统计的 莉
显著。请注意图 3和表格1那是前者 塞斯
准确度，而后者使用对数损失。

GBDT 与 NN。虽然表 11告诉我们哪个 个人方法平均表现最佳 格，
现在我们来思考一个老问题，“对于表格数据，GBDT 是否比 NN 更好？”我们 普利特
将 19 种算法分为三类*家庭*：GBDT（CatBoost、XGBoost、LightGBM）、NN（DANet、 FT-
Transformer、两个 MLP、NODE、ResNet、SAINT、STG、TabNet 和 VIME）和基线（Decis 离子

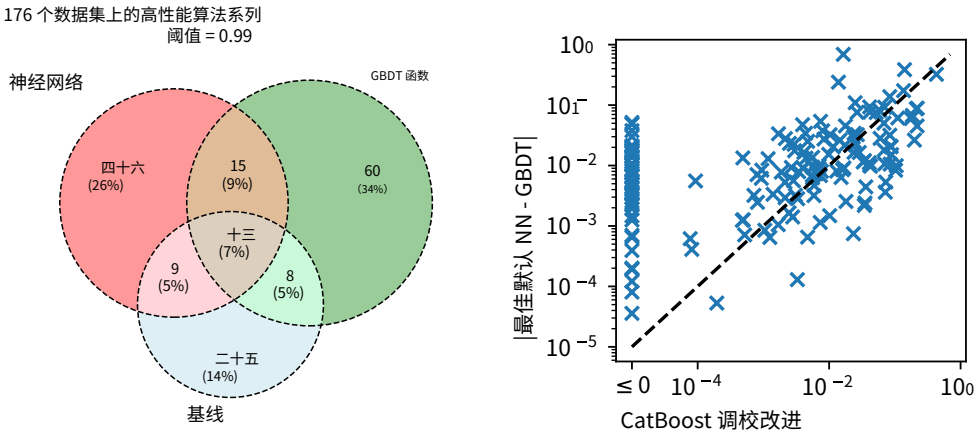

图 4：左图：所有 176 个数据集中每个算法在每个算法类别中表现“高性能”的数字数据集维恩图。如果算法在 0-1 缩放后的测试准确率至少为 0.99（我们在附录 D.2）。右图：超参数调整对 CatBoost 的性能改进，与使用默认超参数的最佳神经网络和最佳 GBDT 之间的绝对性能差异相比。每个点表示一个数据集的归一化对数损失，虚线上或虚线下的点表示由于调整而带来的性能改进与 NN-GBDT 算法选择之间的差异一样高。

树、KNN、线性模型、随机森林、SVM）。†如果一个算法的 0-1 比例测试准确率至少达到，我们就说它是“高性能”的0。99，然后我们确定哪些算法系列（GBDT、NN、基线）具有高性能算法；参见图 4令人惊讶的是，三向维恩图在 GBDT、NN 和基线之间相对平衡，尽管 GBDT 总体上占有优势。附录 D.2，我们进行相同的分析，使用阈值0。9999。在这种情况下，GBDT 是大多数数据集中唯一表现优异的算法系列。由于这些优势不到 0.01%，因此对从业者来说可能并不重要。

算法选择与调整。接下来，我们确定是选择最佳算法系列更重要，还是简单地对总体表现良好的算法（例如 CatBoost 或 ResNet）进行轻度超参数调整更重要。我们考虑一种场景，从业者可以决定 *(A)* 使用默认超参数测试几种算法，或者 *(b)*优化单个模型的超参数，例如 CatBoost 或 ResNet。我们计算 *(A)* 或者 *(b)* 导致更好的性能。具体来说，我们使用默认超参数测量表现最好的 GBDT 和 NN 之间的性能差异，以及性能差异

使用默认超参数的 CatBoost 与通过 30 次随机迭代调整的 CatBoost 之间的比较
在验证集上搜索；参见图 4（右），然后看看附录 D.2对于 ResNet 的相同分析。令人惊讶的是，对于大约三分之一的数据集，轻度超参数调整比 GBDT-vs-NN 选择产生了更大的性能提升。再一次，这表明对于很大一部分数据集，没有必要确定 GBDT 或 NN 哪个更好：对 CatBoost 或 ResNet 等算法进行轻度调整可以带来同样多的性能提升。在下一节中，我们将探讨*为什么*数据集可能更适合神经网络或 GBDT。

2.2 元特征分析　　姐姐

在本节中，我们讨论技术r th问题是，“一款产品的哪些特性优于集合与某些算法的性能差异rs或算法家族与个人性能之技术其他产品？”我们用三个不同的量来相关，通过计算d来回答这个问间的相关性。　　图描述不同产品之间的性能差异。　　题，以及相对
英石
阿尔法图。

†由于 TabPFN 的工作 你贝斯特何神经网络都截然不同（或　　任何其他算法），我们将其排除原理与本节的分析一致　　ion 和下一节讨论‘GBDT 与 NN’时。

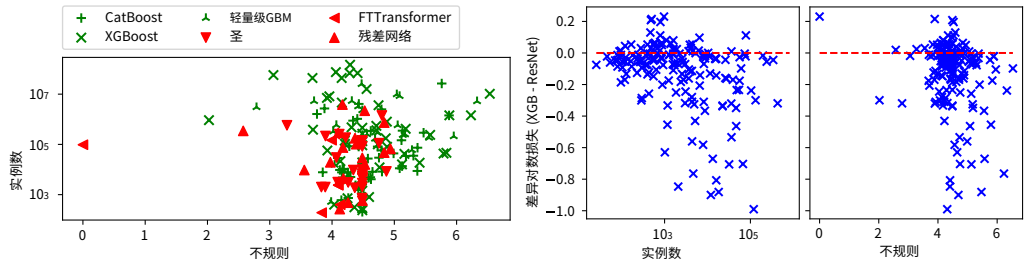

图 5：左图：所有 176 个数据集上最佳算法在元特征上的散点图。纵轴表示数据集大小，横轴结合了与不规则性相关的五个数据集元特征。右图：XGBoost 和 ResNet 之间归一化对数损失差异的散点图，按数据集大小（中间子图）和不规则性（右子图）划分。不规则性特征是五个标准化数据集属性的线性组合：特征协方差矩阵的最小特征值（-0.33）、所有特征标准差的偏度（0.23）、所有特征范围的偏度（0.22）、所有特征调和平均值的四分位距（0.21）以及所有特征峰度的标准差（0.21）。

为了评估 NN 和 GBDT 之间的性能差异，我们计算了最佳 NN 和最佳 GBDT 之间的归一化对数损失的差异，我们将其称为 $\Delta_{\ell\ell}$. 我们计算相关性 $\Delta_{\ell\ell}$所有数据集中的各种元特征；参见图 14和表 17在 附录 D.4接下来，为了确定每个算法各自的优势和劣势，我们计算各种元特征与单个算法相对于所有其他算法的性能的相关性；参见表3和图 5最后，我们计算元特征与表现最佳算法对之间的性能差异的相关性， 第 2.1 节：CatBoost、XGBoost、SAINT 和 ResNet。参见图 5和 [0377]表 22。

为了证明这些元特征*预测*，我们使用留一法训练和评估元学习模型：保留一个数据集进行测试，其余 175 个数据集用于训练，并取所有 176 个可能的测试集的平均值；参见附录 D.4. 在本节的其余部分，我们陈述并讨论元特征分析的主要发现。

神经网络在较大的数据集上的表现相对较差。在我们的元特征分析过程中，我们发现 GBDT 的表现比具有更大数据集的 NN 和基线更好。图 5结果表明，在 7 个最大的数据集上，XGBoost 与所有 19 种算法相比均取得了最佳性能，并且 GBDT 总体表现良好。在表3有点令人惊讶的是，数据集大小是与 LightGBM 和 XGBoost 的相对性能最负相关的元特征。最后，表 17表明 GBDT 家族的表现也与*比率*大小与特征数量的关系。值得注意的是，所有这些分析

与所有算法的性能有关，其中包括新发布的 TabPFN [33]， A
由于其精心设计的先验，GBDT 在小规模数据集上表现优异。另一方面，当数据集大小与特征数量之比较高时，GBDT 表现优异，因为决策树中的所有分割都是使用更多数据点计算的。我们上述的一些发现得到了先前研究的支持，例如 Grinsztajn 等人。[二十八] 表明，增加（无信息）特征与数据集大小的比率会损害 ResNet 的性能，我们的结果表明

同样的趋势（表 23)。另一方面，NN 整体上呈现出相反的趋势（表 17),
至少在附录 D.3.4. 注意数 F电变压器在 Grinsztajn 等 等[二十八]。 我们提供额外的分析
据集，这并不意味着 H在，而总体趋势表明所有 GBDTGBDT 在更大的数据集上表现优于
是唯都优于 所有 NN o 神经网络。例如，在较小的数
TabPFN 和 TabNet 是 bo 唯神经网络，但 TabPFN 执行部分 据集上表现更好， 在选择特
和 TabNet 执行部分 一种 乌拉 更大的数据集上表现良好。重要的 定于算法的分析（例如 GBDT
新的算法表格1，表 2， 电 是商务 业者应该关注算法 附录 D.3.4） 与 NN 的趋势）时需要注意的
一个 dA 而不是一般的“ 是。

GBDT 有利于不规则数据 塔套。另一个趋势是“不规则”数据集。所有三个元特征分析都是顶
GBDT 和 NN 一致，我们发 f 当 Comp CatBoost 和 XGBoost 优于 级 GBDT 的对
现特征分布 H Re 时，重尾、倾斜或具有高变异性 sNet 和 SAINT 在数据集上的
应收账 表现（见[0377]表 22完整

表 3：与每个表现最佳算法的性能最相关的元特征，计算为 176 个数据集中元特征与归一化对数损失之间的皮尔逊相关性。显示了与算法性能具有最大绝对皮尔逊相关性的元特征，相关性以 95% 置信区间表示，使用 Fisher 变换计算。

阿尔及利亚	元特征描述	纠正。
CatBoost	特征的噪声程度：（ $\sum_{\text{我年代我}-}\sum_{\text{我}}MI$（我，））/ $\sum_{\text{我}}MI$（我，），在哪里$年代_{我}$是熵特征$我$，和MI（$我$，）是特征之间的相互信息$我$和目标$是$。	[0.25, 0.34]
XGBoost	对数实例数。	[-0.27，-0.18]
轻量级GBM	对数实例数。	[-0.36，-0.28]
残差网络	任何数字特征和目标之间的平均典型相关性。	[-0.28，-0.19]
FTTransformer	目标类别的数量。	[0.23, 0.32]
圣	特征的噪声程度。	[0.19, 0.29]

细节）。也就是说，一些数据集的特征分布都具有相似的偏度，而其他数据集的特征分布则更不规则，偏度范围较大。在后一种数据集上，GBDT 的表现优于 NN。我们还发现，当数据集的类别不平衡程度更高时（尤其是在 SAINT 上），GBDT 的表现会更好。在图 5，GBDT 在最不规则的数据集上表现最佳，通过五个元特征的线性组合计算，每个元特征测量特征分布的偏度或峰度。

底线。总的来说，我们回答了本文的标题问题：GBDT 在更“不规则”的数据集、大型数据集以及大小与特征数量比率较高的数据集上的表现优于 NN。当从业者面对新的数据集时，基于所有分析 第 2 部分，我们给出以下建议：首先尝试简单的基线，然后对 CatBoost 进行轻度超参数调整。令人惊讶的是，这通常已经会带来强劲的表现。下一步，从业者可以尝试基于数据集的元特征与强劲表现最相关的 NN 和其他 GBDT，使用以下分析表格1和附录 D.3.4。

3 TabZilla 基准测试套件

为了加速表格数据研究，我们发布了 TabZilla 基准套件：收集了我们在 2016 年研究的 176 个数据集中“最难”的 36 个数据集。第 2 部分我们使用以下三个标准。

对于基线算法来说很难。正如所讨论的第 2.1 节在我们的实验中，简单的基线在很大一部分数据集上表现非常好。因此，为了选择我们的硬数据集套件，我们删除任何数据集，使得基线（如在第 2 部分) 的归一化对数损失与表现最佳的算法相差 20% 以内。此标准并不完美；例如，如果一个数据集太难以至于所有 19 种算法都无法达到非平凡性能，则它不满足该标准。但是，考虑到可用信息，该标准是衡量数据集难度的良好指标。

对于大多数算法来说都很难。此标准旨在包含大多数算法无法达到最佳性能的数据集。具体而言，如果 19 种算法中第四好的对数损失比最高对数损失至少差 7%，则该数据集满足此标准。换句话说，此标准将包含一种、两种或三种算法在性能方面能够脱颖而出的数据集。例如，如果十种算法的性能都能够达到与表现最佳的算法相差 7% 以内，我们可以合理地假设该数据集可能不“难”。有趣的是，此标准涵盖了上一个标准中的大多数数据集。

对于 GBDT 来说很难。前两个标准导致数据集中 GBDT 主要是表现最佳的方法。考虑到我们在第 2 部分。然而，对于表格数据领域的进步，如果只关注 GBDT 已经表现良好的数据集，就会对 GBDT 表现不佳的数据集视而不见。因此，我们添加了所有 GBDT 表现比表现最佳的算法差 10% 的数据集，以实现更大的数据集多样性。

表 4：TabZilla 基准测试套件。列显示用作选择标准的硬度指标、数据集属性和表现最佳的算法。“标准峰度”表示所有特征峰度的标准差。符合我们选择标准的硬度指标以粗体显示。

	硬度指标			数据集属性				前 3 名算法。	
数据集	根据	第四佳	国别分类法	*否*	# 壮举。	标准峰度	第一	第二	第三
信用-g	0.26	0.13	0.12	1 000	21	1.92	ResNet	FTTransformer	CatBoost
丛林象棋	0.30	0.18	0.17	44,819	7	0.08	圣	塔格网	轻量级GBM
迷你布诺	0.20	0.09	0.00	130 064	51	12162.65	轻量级GBM	XGBoost	CatBoost
艾伯特	0.42	0.28	0.00	425 240	79	1686.90	CatBoost	XGBoost	残差网络
电	0.46	0.38	0.00	45312	9	2693.51	轻量级GBM	XGBoost	FTTransformer
电梯	0.36	0.08	0.05	16,599	19	2986.50	塔格网	XGBoost	CatBoost
吉列尔莫	0.35	0.60	0.00	20 000	4 297	钠	XGBoost	随机森林	塔格网
希格斯	0.41	0.10	0.07	98 050	二十九	15.53	ResNet	XGBoost	轻量级GBM
野毛	0.22	0.18	0.00	34 465	119	1100.34	LightGBM	XGBoost	CatBoost
100 种植物纹理	0.20	0.11	0.00	1,599	65	17.66	CatBoost	XGBoost	残差网络
扑克牌	0.58	0.98	0.00	1 025 009	11	0.08	XGBoost	CatBoost	KNN
概率	0.39	0.38	0.00	672	10	0.95	CatBoost	深度调频	MLP-RTDL
索科莫布	0.24	0.10	0.00	1 156	6	钠	XGBoost	CatBoost	残差网络
听力学	0.43	0.03	0.00	226	70	钠	星火	XGBoost	残差网络
拼接	0.30	0.03	0.00	3 190	61	钠	轻量级GBM	XGBoost	CatBoost
车辆	0.05	0.10	0.10	846	19	15.16	表格PFN	支持向量机	数据网络
澳大利亚	0.15	0.08	0.00	690	15	2.00	CatBoost	XGBoost	表PFN
生物反应	0.07	0.07	0.00	3 751	1,777	328.77	轻量级GBM	XGBoost	CatBoost
手势阶段	0.08	0.08	0.00	9,872	33	52.18	LightGBM	XGBoost	CatBoost
快速约会	0.18	0.14	0.00	8,378	121	36.43	XGBoost	CatBoost	轻量级GBM
不可知论者	0.12	0.11	0.00	4,562	49	钠	XGBoost	CatBoost	轻量级GBM
航空公司	0.20	0.18	0.00	539 382	8	2.01	轻量级GBM	XGBoost	CatBoost
人造角色	0.13	0.11	0.00	10218	8	0.63	XGBoost	轻量级GBM	CatBoost
绞痛	0.13	0.11	0.00	368	二十七	4.00	CatBoost	XGBoost	FTTransformer
信用审批	0.12	0.08	0.00	690	16	74.77	CatBoost	表PFN	XGBoost
炉	0.10	0.07	0.08	294	14	NaN	DeepFM	标签转换器	不结盟运动
茉莉花	0.13	0.13	0.00	2 984	145	47.60	CatBoost	XGBoost	轻量级GBM
KC1	0.14	0.07	0.00	2 109	22	28.34	CatBoost	XGBoost	FTTransformer
淋巴	0.14	0.08	0.00	148	19	17.04	XGBoost	数据网络	圣
傅立叶函数	0.00	0.07	0.07	2 000	77	0.64	支持向量机	圣	星火
音素	0.10	0.15	0.00	5 404	6	1.23	XGBoost	轻量级GBM	随机森林
qsar-biodeg	0.08	0.08	0.05	1 055	四十二	93.24	选项卡PFN	CatBoost	圣
天平	0.07	0.05	0.16	625	5	0.02	表PFN	圣	多层感知处理器
cnae-9	0.11	0.04	0.10	1 080	857	钠	标签转换器	星火	MLP-RTDL
泽尔尼克	0.00	0.04	0.10	2 000	四十八	1.42	支持向量机	数据网络	残差网络
僧侣问题-2	0.04	0.00	0.17	601	7	南圣		残差网络	MLP-RTDL

TabZilla 的特点。表4显示所有数据集、其统计数据以及排名前三的算法。仅根据标准，数据集特征就多种多样，大小从 148 到 100 多万不等，特征峰度方差（我们的不规则性度量之一）的范围也很大。在表 5，我们在基准测试套件上比较了所有算法的性能。排名前五的算法分别是 XGBoost、CatBoost、LightGBM、ResNet 和 SAINT，平均排名分别为 3.27、3.86、6.06、6.14 和 6.37。为了加速表格数据的研究，我们在 OpenML 中发布了 TabZilla 作为集合，并开源了我们计算的所有元特征和结果。在附录 B，我们提供了完整的数据集文档，包括数据表[24]。

4 结论和未来工作

在这项工作中，我们进行了迄今为止最大规模的表格数据分析，比较了 176 个数据集中的 19 种方法。我们发现“NN 与 GBDT”之争被过分强调了：对于数量惊人的数据集，简单的基线方法与所有其他方法的表现相当，或者对 GBDT 进行轻度超参数调整比选择最佳算法更能提高性能。另一方面，平均而言，GBDT 的表现确实优于 NN。我们还分析了数据集的哪些属性使 NN 或 GBDT 更适合表现良好。例如，GBDT 在处理各种类型的数据不规则性方面比 NN 更好。最后，根据我们的分析，我们发布了 TabZilla，这是我们研究的 176 个数据集中 36 个“最难”的数据集的集合：对于基线、大多数算法和 GBDT 来说都很难。发布 TabZilla 的目标是通过专注于改进文献中当前的盲点来加速表格数据研究。

我们的工作提供了大量工具来加速表格数据的研究。例如，开发用于表格数据的新神经网络的研究人员可以使用我们的开源存储库立即将他们的方法与 176 个数据集中的 19 种算法进行比较。研究人员还可以使用我们的元特征分析来改进当前或未来算法的弱点；例如，使神经网络对数据不规则性更具鲁棒性是自然而然的下一步。此外，研究集成的研究人员

表 5：36 个数据集的 Tabular Benchmark Suite 中的算法性能。列显示所有数据集的排名、平均归一化对数损失 (Mean LL)、不同折叠的归一化对数损失的标准差 (Std. LL) 以及每 1000 个实例的训练时间。这些数量的最小值/最大值/平均值/中位数取自所有数据集。

	秩				平均 LL		标准 LL		次/1000实例	
算法	最小值	最大值	平均值	平均值	平均值	平均值			意思是	医学
XGBoost	1	15	3.69	2	0.07	0.04	0.11	0.06	2.02	0.28
CatBoost	1	十三	4.14	3	0.09	0.06	0.11	0.07	26.46	1.15
表PFN*	1	十三	5.29	4	0.19	0.13	0.11	0.07	0.48	0.01
轻量级GBM	1	15	6.62	6.5	0.14	0.09	0.20	0.09	1.23	0.36
残差网络	1	14	6.77	6	0.20	0.17	0.13	0.08	8.27	5.22
圣	1	18	7.00	6	0.20	0.15	0.12	0.08	130.18	92.42
数据网络	2	16	8.04	8	0.19	0.16	0.15	0.11	58.70	52.74
FTTransformer	2	15	8.24	8	0.25	0.20	0.13	0.12	17.41	12.64
MLP-RTDL	2	19	9.42	8	0.37	0.25	0.16	0.13	6.33	4.21
随机森林	2	17	9.43	9	0.29	0.28	0.23	0.09	0.35	0.24
支持向量机	1	18	10.00	10	0.30	0.22	0.14	0.08	19.73	2.81
星火	1	18	10.29	11	0.33	0.26	0.11	0.06	15.99	15.29
塔格网	1	19	11.03	10	0.39	0.29	0.28	0.11	27.02	27.10
节点	6	17	11.40	11	0.38	0.35	0.08	0.07	153.72	124.27
多层感知处理器	3	18	11.42	11.5	0.45	0.39	0.15	0.14	8.86	4.36
线性模型	4	19	11.68	11.5	0.48	0.39	0.12	0.07	0.04	0.02
维梅	5	18	13.91	15	0.51	0.48	0.09	0.08	20.79	15.18
决策树	6	19	14.26	16	0.63	0.62	0.39	0.22	0.11	0.01
KNN	4	19	15.42	17	0.71	0.74	0.36	0.21	0.03	0.00

方法可以根据每个数据集的元特征对模型进行不同的加权。最后，我们开源的大量数据集和元特征集合可以让研究人员更轻松地设计新的元学习[33] 或针对表格数据的预训练模型 [三十二，四十四，71]，还有一些有趣的想法可以扩展，例如回归数据集、时间序列预测数据集，研究不确定性量化，研究分类特征百分比对NN的影响，以及研究包括正则化方法在内的更全面的超参数优化。

作者贡献

DM、SK、JV、VPC、GR、MG 和 CW 为原始论文做出了贡献，该论文被 NeurIPS 数据集和基准测试 Track 2023 接受。随后，BF 和 CH 加入进来，显著扩展并改进了第 2 节和附录 D 中的结果。

参考

[1] Steven Adriaensen、Herilalaina Rakotoarison、Samuel Müller 和 Frank Hutter。使用先验数据拟合网络进行高效贝叶斯学习曲线外推。*神经信息处理系统年度会议（NeurIPS）论文集*，2023年。

[2] Rishabh Agarwal、Levi Melnick、Nicholas Frosst、Xuezhou Zhang、Ben Lengerich、Rich Caruana 和 Geoffrey E Hinton。神经加性模型：利用神经网络进行可解释的机器学习。*神经信息处理系统年度会议（NeurIPS）论文集*，2021年。

[3] 秋叶卓也、佐野章太郎、柳濑俊彦、太田健、小山正则。 Optuna：下一代超参数优化框架。在*第 25 届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集*，第 2623–2631 页，2019 年。

[4] 埃德西奥·阿尔科巴萨、费利佩·西凯拉、阿德里亚诺·里沃利、路易斯·PF·加西亚、杰斐逊·T·奥利瓦、安德烈·CPLF·德卡瓦略。 Mfe：走向可重复的元特征提取。*机器学习研究杂志*, 21(111):1–5, 2020。

[5] Sercan Ö Arik 和 Tomas Pfister。Tabnet：专注的可解释表格学习。在*AAAI 人工智能会议 (AAAI) 论文集*，2021年。

[6] Kumar Arun、Garg Ishan 和 Kaur Sanmeet。基于机器学习方法的贷款批准预测。*IOSR 计算机工程学杂志*《细胞与分子生物学杂志》, 18(3): 18–21, 2016。

[7] Bernd Bischl、Giuseppe Casalicchio、Matthias Feurer、Frank Hutter、Michel Lang、Rafael G Mantovani、Jan N van Rijn 和 Joaquin Vanschoren。 Openml 基准测试套件。*arXiv 预印本 arXiv:1708.03731*，2017年。

[8] Vadim Borisov、Tobias Leemann、Kathrin Seßler、Johannes Haug、Martin Pawelczyk 和 Gjergji Kasneci。深度神经网络和表格数据：一项调查。*arXiv 预印本 arXiv:2110.01889*，2021年。

[9] Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell 等。语言模型是小样本学习器。*神经信息处理系统的进展*, 33: 1877–1901，2020年。

[10] Anna L Buczak 和 Erhan Guven.网络安全入侵检测的数据挖掘和机器学习方法调查。*IEEE 通信调查和教程*《细胞与分子生物学杂志》, 18(2): 1153–1176, 2015.

[11] Varun Chandola、Arindam Banerjee 和 Vipin Kumar.异常检测：一项调查。*ACM 计算调查 (CSUR)*, 41(3):1–58, 2009。

[12] Jintai Chen、Kuanlun Liao、Yao Wan、Danny Z Chen 和 Jian Wu。Danets：用于表格数据分类和回归的深度抽象网络。*AAAI 人工智能会议 (AAAI) 论文集*，2022年。

[13] Tianqi Chen 和 Carlos Guestrin。Xgboost：可扩展的树提升系统。*第 22 届 acm sigkdd 知识发现和数据挖掘国际会议论文集*，第 785–794 页，2016 年。

[14] Jillian M Clements、Di Xu、Nooshin Yousefi 和 Dmitry Efimov。使用表格财务数据进行信用风险监控的顺序深度学习。*arXiv 预印本 arXiv:2012.15330*，2020年。

[15] 威廉・杰伊・科诺弗。*实用非参数统计*，第 350 卷。约翰威利父子公司，1999 年。

[16] Corinna Cortes 和 Vladimir Vapnik。支持向量网络。*机器学习*，1995 年。

[17] Thomas Cover 和 Peter Hart.最近邻模式分类。*IEEE 信息理论汇刊*，1967 年。

[18] David R Cox.二元序列的回归分析。*皇家统计学会杂志：B 系列（方法论）*，1958 年。

[19] Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。Bert：用于语言理解的深度双向转换器的预训练。*arXiv 预印本 arXiv:1810.04805*，2018 年。

[20] Alexey Dosovitskiy、Lucas Beyer、Alexander Kolesnikov、Dirk Weissenborn、翟晓华、Thomas Unterthiner、Mostafa Dehghani、Matthias Minderer、Georg Heigold、Sylvain Gelly 等。一张图片胜过 16x16 个单词：用于大规模图像识别的 Transformers。*国际学习表征会议（ICLR）论文集*，2021年。

[21] Matthias Feurer、Katharina Eggensperger、Stefan Falkner、Marius Lindauer 和 Frank Hutter。 Auto-sklearn 2.0：通过元学习实现解放双手的自动化机器学习。*机器学习研究杂志*，2022年。

[22] Jerome H Friedman. 贪婪函数逼近：梯度提升机。*统计年鉴*，第 1189–1232 页，2001 年。

[23] 米尔顿・弗里德曼。使用秩来避免方差分析中隐含的正态性假设。*美国统计协会杂志*，1937 年。

[24] Timnit Gebru、Jamie Morgenstern、Briana Vecchione、Jennifer Wortman Vaughan、Hanna Wallach、Hal Daumé Iii 和 Kate Crawford。数据集的数据表。*ACM 通讯*, 64(12):86–92, 2021年。

[25] ES Gelsema 和 LN Kanal。*模式识别实践 IV：多范式、比较研究和混合系统*. ISSN。Elsevier Science，2014 年。

[26] Pieter Gijsbers、Erin LeDell、Janek Thomas、Sébastien Poirier、Bernd Bischl 和 Joaquin Vanschoren。一个开源的 automl 基准。*arXiv 预印本 arXiv:1907.00909*，2019年。

[27] Yury Gorishniy、Ivan Rubachev、Valentin Khrulkov 和 Artem Babenko。重新审视表格数据的深度学习模型。*神经信息处理系统年度会议（NeurIPS）论文集*，2021年。

[28] Leo Grinsztajn、Edouard Oyallon 和 Gael Varoquaux。为什么基于树的模型在典型的表格数据上仍然优于深度学习？*第三十六届神经信息处理系统数据集和基准会议*，2022年。

[29] 郭惠峰、唐瑞明、叶云明、李振国、何秀强。 Deepfm：一种基于分解机的用于 ctr 预测的神经网络。在*国际计算机辅助教学*，2017年。

[30] John T Hancock 和 Taghi M Khoshgoftaar。神经网络分类数据调查。 *大数据杂志*, 7(1):1–41, 2020。

[31] 何开明、张翔宇、任少卿、孙健。深度残差学习在图像识别中的应用。*IEEE 计算机视觉和模式识别会议论文集*，第 770–778 页，2016 年。

[32] Stefan Hegselmann、Alejandro Buendia、Hunter Lang、Monica Agrawal、Xiaoyi Jiang 和 David Sontag。Tabllm：使用大型语言模型对表格数据进行小样本分类。 *国际人工智能与统计会议论文集（AISTATS）*，2023年。

[33] Noah Hollmann、Samuel Müller、Katharina Eggensperger 和 Frank Hutter。Tabpfn：一款可在一秒钟内解决小型表格分类问题的转换器。*国际学习表征会议（ICLR）论文集*，2023年。

[34] Noah Hollmann、Samuel Müller 和 Frank Hutter。半自动化数据科学 GPT：引入 caafe 进行情境感知自动化特征工程。*国际机器学习会议 (ICML) 论文集*，2023年。

[35] Sture Holm.一种简单的顺序拒绝多重检验程序。*斯堪的纳维亚统计学杂志*，第 65-70 页，1979 年。

[36] Xin Huang、Ashish Khetan、Milan Cvitkovic 和 Zohar Karnin。Tabtransformer：使用上下文嵌入的表格数据建模。*arXiv 预印本 arXiv:2012.06678*，2020年。

[37] Alan Jeffares、Tennison Liu、Jonathan Crabbé、Fergus Imrie 和 Mihaela van der Schaar。Tangos：通过梯度正交化和特殊化对表格神经网络进行正则化。 *国际学习表征会议（ICLR）论文集*，2023年。

[38] Alistair EW Johnson、Tom J Pollard、Lu Shen、Li-wei H Lehman、Mengling Feng、Mohammad Ghassemi、Benjamin Moody、Peter Szolovits、Leo Anthony Celi 和 Roger G Mark。Mimic-iii，一个可以免费访问的重症监护数据库。*科学数据*, 3(1):1–9, 2016。

[39] Arlind Kadra、Marius Lindauer、Frank Hutter 和 Josif Grabocka。经过良好调整的简单网络在表格数据集上表现出色。*神经信息处理系统年度会议（NeurIPS）论文集*，34，2021年。

[40] 柯国林、孟奇、Thomas Finley、王泰峰、陈炜、马伟东、叶启伟、刘铁燕。 Lightgbm：一种高效的梯度提升决策树。在*神经信息处理系统年度会议（NeurIPS）论文集*，2017年。

[41] Gurnoor Singh Khurana、Samuel Dooley、Siddartha Venkat Naidu 和 Colin White。Forecastpfn：医疗保健的通用预测。*ICLR 2023 健康领域时间序列表征学习研讨会*，2023 年。

[42] Alex Krizhevsky、Ilya Sutskever 和 Geoffrey E Hinton。利用深度卷积神经网络进行 Imagenet 分类。*神经信息处理系统年度会议（NeurIPS）论文集*，2012 年。

[43] Roman Levin、Valeriia Cherepanova、Avi Schwarzschild、Arpit Bansal、C Bayan Bruss、Tom Goldstein、Andrew Gordon Wilson 和 Micah Goldblum。使用深度表格模型进行迁移学习。*国际心肺复苏术*，2023年。

[44] Roman Levin、Valeriia Cherepanova、Avi Schwarzschild、Arpit Bansal、C Bayan Bruss、Tom Goldstein、Andrew Gordon Wilson 和 Micah Goldblum。使用深度表格模型进行迁移学习。*国际学习表征会议（ICLR）论文集*，2023年。

[45] Andy Liaw、Matthew Wiener 等人。通过随机森林进行分类和回归。*R 新闻*, 2(3): 18–22, 2002。

[46] Guido Lindner 和 Rudi Studer。Ast：使用 cbr 方法支持算法选择。在 *第三届欧洲数据挖掘与知识发现原理会议论文集*, PKDD '99, 第 418-423 页，柏林，海德堡，1999 年。Springer-Verlag。

[47] Zachary C Lipton 和 Jacob Steinhardt。机器学习学术界令人不安的趋势：一些机器学习论文存在缺陷，可能会误导公众并阻碍未来的研究。*队列*，2019年。

[48] H Brendan McMahan、Gary Holt、David Sculley、Michael Young、Dietmar Ebner、Julian Grady、Lan Nie、Todd Phillips、Eugene Davydov、Daniel Golovin 等。广告点击预测：来自战壕的视角。在*知识发现和数据挖掘（KDD）年度会议论文集*，第 1222–1230 页，2013 年。

[49] Samuel Müller、Matthias Feurer、Noah Hollmann 和 Frank Hutter。Pfns4bo：贝叶斯优化的上下文学习。*国际机器学习会议 (ICML) 论文集*，2023年。

[50] Samuel Müller、Noah Hollmann、Sebastian Pineda Arango、Josif Grabocka 和 Frank Hutter。Transformer 可以进行贝叶斯推理。在*国际学习表征会议（ICLR）论文集*，2022 年。

[51] Thomas Nagler。先验数据拟合网络的统计基础。*国际机器学习会议 (ICML) 论文集*，2023 年。

[52] Fabian Pedregosa、Gaël Varoquaux、Alexandre Gramfort、Vincent Michel、Bertrand Thirion、Olivier Grisel、Mathieu Blondel、Peter Prettenhofer、Ron Weiss、Vincent Dubourg 等人。 Scikitlearn：用 Python 进行机器学习。*机器学习研究杂志*《国际神经病学杂志》12：2825–2830，2011。

[53] Sergei Popov、Stanislav Morozov 和 Artem Babenko。用于表格数据深度学习的神经无意识决策集成。*国际学习表征会议（ICLR）论文集*，2020年。

[54] 柳德米拉·普罗霍连科娃、格列布·古谢夫、亚历山大·沃罗别夫、安娜·维罗妮卡·多罗古什、安德烈·古林。 Catboost：具有分类特征的无偏提升。*神经信息处理系统年度会议（NeurIPS）论文集*，2018 年。

[55] J. Ross Quinlan. 决策树的归纳。*机器学习*，1986 年。

[56] Matthew Richardson、Ewa Dominowska 和 Robert Ragno。预测点击次数：估算新广告的点击率。*第十六届万维网国际会议论文集*，第 521–530 页，2007 年。

[57] Ivan Rubachev、Artem Alekberov、Yury Gorishniy 和 Artem Babenko。重新审视表格深度学习的预训练目标。*arXiv 预印本 arXiv:2207.03208*，2022年。

[58] Mostafa A Salama、Aboul Ella Hassanien 和 Kenneth Revett。神经网络和粗糙集在元学习中的应用。*模因计算*, 5: 165–177, 2013。

[59] Bernhard Schäfl、Lukas Gruber、Angela Bitto-Nemling 和 Sepp Hochreiter。 Hopular：用于表格数据的现代霍普菲尔德网络。*arXiv 预印本 arXiv:2206.00664*，2022年。

[60] Ira Shavitt 和 Eran Segal。正则化学习网络：表格数据集的深度学习。 *神经信息处理系统的进展*，2018 年 31 日。

[61] Ravid Shwartz-Ziv 和 Amitai Armon。表格数据：深度学习并不是你所需要的全部。 *信息融合*，81：84–90，2022年。

[62] Gowthami Somepalli、Micah Goldblum、Avi Schwarzschild、C Bayan Bruss 和 Tom Goldstein。Saint：通过行注意和对比预训练改进表格数据的神经网络。 *arXiv 预印本 arXiv:2106.01342*，2021年。

[63] Bojan Tunguz。霍普拉尔的麻烦，2022 年。

[64] Dennis Ulmer、Lotta Meijerink 和 Giovanni Cinà。信任问题：不确定性估计无法对医疗表格数据进行可靠的洪水检测。在*机器学习在健康领域的应用*，第 341-354 页。PMLR，2020 年。

[65] Christopher J Urban 和 Kathleen M Gates.深度学习：心理学家的入门书。 *心理方法*，2021年。

[66] Joaquin Vanschoren、Jan N Van Rijn、Bernd Bischl 和 Luis Torgo。 Openml：机器学习中的网络科学。*ACM SIGKDD 探索简讯*，2014 年。

[67] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。你只需要关注。*神经信息处理系统的进展*，第 5998–6008 页，2017 年。

[68] Martin Wistuba、Nicolas Schilling 和 Lars Schmidt-Thieme。学习超参数优化初始化。*在2015 IEEE 数据科学与高级分析国际会议 (DSAA)*，2015 年。

[69] Yutaro Yamada、Ofir Lindenbaum、Sahand Negahban 和 Yuval Kluger。使用随机门进行特征选择。*在国际机器学习会议 (ICML) 论文集*，2020年。

[70] Jinsung Yoon、Yao Zhang、James Jordon 和 Mihaela van der Schaar。Vime：将自我和半监督学习的成功扩展到表格领域。*神经信息处理系统的进展*, 33: 11033–11043，2020年。

[71] Bingzhao Zhu、Xingjian Shi、Nick Erickson、Mu Li、George Karypis 和 Mahsa Shoaran。Xtab：表格转换器的交叉表预训练。*arXiv 预印本 arXiv:2305.06090*，2023年。

更广泛的社会影响声明

我们的工作目标是对表格数据进行分析，包括“GBDT 与 NN”的重要性，以及为处理表格数据的研究人员和从业者提供大量工具。我们没有看到我们的工作对社会产生任何负面影响。事实上，我们的工作表明，在令人惊讶的大部分数据集上，不需要训练资源密集型神经网络：简单的基线或调整 CatBoost 就足以达到最佳性能。我们甚至预测哪些数据集更适合 GBDT：更大的数据集、具有高大小与特征数量比的数据集和“不规则”数据集。我们希望我们的工作将对从业者和研究人员产生积极影响：通过提供迄今为止最大的分析和开源代码库以及“硬”数据集的基准套件，我们的工作既可以加速未来的研究，又可以使未来工作的比较更加严格和全面。

B 数据集文档

在本节中，我们概述了数据集的文档。有关完整详细信息（包括用法和教程），请参阅https://github.com/naszilla/tabzilla。

B.1 作者责任和许可

如果发生侵权，我们作为作者将承担所有责任。我们的存储库的许可证是Apache 许可证 2.0。有关详细信息，请参阅For more information, seehttps://github.com/naszilla/tabzilla/blob/main/LICENSE。

B.2 维护计划

数据可在 OpenML 上获取https://www.openml.org/search?type=study&study_type=task&id=379&sort=tasks_included。

我们计划积极维护基准套件，并欢迎社区的贡献。

B.3 行为准则

我们的行为准则来自贡献者契约 2.0 版。请参阅 https://www.contributor-covenant.org/version/2/0/code_of_conduct.html。

B.4 数据表

我们附上了一份数据表 [24] 适用于 TabZilla，可在此处以及https://github.com/naszilla/tabzilla。

数据表创建的动机

* 为什么要创建数据表？（例如，是否有特定的任务需要完成？是否存在需要填补的特定空白？）发布 TabZilla 基准套件的目的是通过引入一组“硬”数据集来加速表格数据的研究。具体来说，简单的基线无法达到最佳性能，大多数算法（我们尝试的 19 种算法中）都无法达到最佳性能。我们发现，如今表格研究中使用的数据集中，有相当高比例的数据集使得简单的基线可以达到与领先方法一样高的准确率。

* 该数据集是否已被使用？如果是，结果在哪里，以便其他人进行比较（例如，已发表论文的链接）？所有单独的数据集都已在 OpenML 中发布，其中许多数据集已在之前的表格数据工作中使用。然而，我们的工作将这些数据集集中到一个“硬”套件中。

* 数据集可用于哪些（其他）任务？所有这些数据集都是表格分类数据集，因此据我们所知，它们不能用于表格分类以外的任何用途。

* 谁资助了创作数据集？该基准测试套件由 Abacus.AI、斯坦福大学、Pinterest、马里兰大学、孟买印度理工学院、纽约大学和加州理工学院的研究人员创建。数据集计算本身的资金来自 Abacus.AI。

* 还有其他评论吗？没有任何。

数据表组成

* 实例是什么？（即示例；例如文档、图像、人物、国家）是否有多 种类型的实例？（例如电影、用户、评分；人物、他们之间的互动；节点、边）每个实例都是一个表格数据点。每个点的组成取决于其数据集。例如，三个数据集由扑克牌、用电量和植物纹理组成。

* 总共有多少个实例（如果适用，每种类型有多少个实例）？参见表格4详细了解每个数据集的实例数量。

* 每个实例由哪些数据组成？“原始”数据（例如未处理的文本或图像）？特征/属性？是否有与实例相关的标签/目标？如果实例与人相关，是否可以识别亚群（例如按年龄、性别等）以及他们的分布如何？原始数据托管在 OpenML 上。在我们的存储库中，我们还包含在训练表格数据模型之前运行的标准预处理脚本。这些数据与人无关。

* 个别实例中是否缺少任何信息？如果是，请提供描述，解释此信息缺失的原因（例如，因为不可用）。这不包括故意删除的信息，但可能包括，例如，删节的文本。没有缺失的信息。

* 各个实例之间的关系是否明确（例如，用户的电影评分、社交网络链接）？如果是，请描述如何明确这些关系。各个实例之间没有关系。

* 数据集是否包含所有可能的实例，还是它是来自更大集合的实例样本（不一定是随机的）？如果数据集是一个样本，那么更大的集合是什么？样本是否代表更大的集合（例如，地理覆盖范围）？如果是，请描述如何验证/核实这种代表性。如果它不代表更大的集合，请描述原因（例如，为了覆盖更多样化的实例，因为实例被保留或不可用）。

我们为基准套件选择了以下数据集。我们从 176 个数据集开始，选择这些数据集的目的是为了涵盖最近流行的研究表格数据的论文中的大多数分类数据集 [8，二十七，三十九，61]，包括来自 OpenML-CC18 套件的数据集[7]，OpenML 基准测试套件 [二十六]，以及其他 OpenML 数据集 [66]。由于我们的实验规模（总共训练了 538 650 个模型），我们将数据集限制在 1.1M 以下。CC-18 和 OpenML Benchmarking Suite 都被视为进行公平、多样化评估的首选标准

由于其严格的选择标准和数据集的多样性，跨算法[7，二十六] 在这 176 个数据集中，我们选择了 36 个数据集作为我们的套件，如下所述第 3 部分。

* 是否有推荐的数据拆分方式（例如，训练、开发/验证、测试）？如果有，请提供这些拆分方式的描述，并解释其背后的原理。

我们使用 OpenML 中的 10 倍样本，建议报告这 10 倍样本的平均性能，我们和 OpenML 都是这么做的。如果需要验证集，我们建议另外使用我们使用的验证拆分，如中所述第 2 部分。

* 数据集中是否存在任何错误、噪声源或冗余？如果有，请提供描述。没有已知错误、噪声源或冗余。

* 数据集是否自成体系，或者是否链接到或以其他方式依赖外部资源（例如网站、推文、其他数据集）？如果链接到或依赖外部资源，a) 是否保证这些资源会长期存在并保持不变；b) 是否有完整数据集的官方存档版本（即包括数据集创建时存在的外部资源）；c) 是否有任何与外部资源相关的限制（例如许可证、费用）可能适用于未来的用户？请提供所有外部资源的描述及其相关限制，以及链接或其他访问点（视情况而定）。数据集是独立的。

还有其他评论吗？没有任何。

收集流程

* 使用了哪些机制或程序来收集数据（例如，硬件设备或传感器、人工管理、软件程序、软件 API）？这些机制或程序是如何验证的？我们没有创建单独的数据集。但是，我们为基准套件选择了以下数据集。我们从 176 个数据集开始，选择这些数据集的目的是包括最近流行的研究表格数据的论文中的大多数分类数据集 [8，二十七，三十九，61]，包括来自 OpenML-CC18 套件的数据集[7]，OpenML 基准测试套件 [二十六]，以及其他 OpenML 数据集 [66]。由于我们的实验规模（总共训练了 538 650 个模型），我们将数据集限制在 1.1M 以下。CC-18 和 OpenML Benchmarking Suite 均被视为对算法进行公平、多样化评估的首选标准，因为它们具有严格的选择标准和广泛的数据集多样性[7，二十六]。

在这 176 个数据集中，我们选择了 36 个数据集作为我们的套件，如下所述第 3 部分。

* 与每个实例相关的数据是如何获取的？数据是直接可观察的（例如，原始文本、电影评分）、由受试者报告的（例如，调查回复），还是间接从其他数据推断/得出的（例如，词性标签、基于模型的年龄或语言猜测）？如果数据是由受试者报告的或间接从其他数据推断/得出的，那么数据是否经过验证/核实？如果是，请描述如何验证/核实。数据集是根据以下三个标准选择的：第 3 部分。

* 如果数据集是来自更大集合的样本，那么采样策略是什么（例如，确定性、具有特定采样概率的概率性）？如前所述，数据集是根据以下三个标准选择的：第 3 部分。

* 谁参与了数据收集过程（例如，学生、众包工作者、承包商）以及他们如何获得报酬（例如，众包工作者获得多少报酬）？ TabZilla Benchmark Suite 的创建是由本文的作者完成的。

* 数据是在什么时间段收集的？这个时间段是否与实例相关数据的创建时间段相匹配（例如，最近抓取的旧新闻文章）？如果不是，请描述与实例相关的数据的创建时间段。TabZilla Benchmark Suite 的构建时间范围是 2023 年 4 月 15 日至 2023 年 6 月 1 日。

数据预处理

* 是否对数据进行了任何预处理/清理/标记（例如，离散化或分桶、标记化、词性标记、SIFT 特征提取、实例删除、缺失值处理）？如果是，请提供描述。如果没有，您可以跳过本部分的其余问题。

我们同时包括原始数据和预处理数据。我们通过将每个 NaN 归结为相应特征的平均值来预处理数据。我们将所有其他预处理（例如缩放）留给算法本身。

* 除了预处理/清理/标记的数据外，是否还保存了“原始”数据（例如，以支持未预料到的未来用途）？如果是，请提供“原始”数据的链接或其他访问点。

原始数据可从以下网址获取：https://www.openml.org/search?type=study&study_type=task&id=379&sort=tasks_included。

* 是否有用于预处理/清理/标记实例的软件？如果有，请提供链接或其他访问点。

我们的自述文件包含有关数据预处理的大量部分，如下：https://github.com/naszilla/tabzilla#openml-datasets。

* 此数据集收集/处理程序是否实现了本数据表第一部分所述创建数据集的动机？如果没有，其局限性是什么？我们希望此基准测试套件的发布能够实现加速表格数据研究的目标，并使研究人员和从业人员更容易设计和比较算法。时间将告诉我们我们的套件是否会被社区采用。

* 任何其他评论没有任何。

数据集分布

* 数据集将如何分发？（例如，网站、API、GitHub 上的 tarball；数据是否有 DOI 并且是否有冗余存档？）OpenML 上的基准测试套件位于https://www.openml. org/search?type=study&study_type=task&id=379&sort=tasks_included。

* 数据集何时发布/首次分发？它以什么许可证（如果有）分发？该基准套件于 2023 年 6 月 1 日公开，并根据 Apache License 2.0 分发。

* 数据有版权吗？数据没有版权。

* 是否有任何费用或访问/出口限制？没有任何费用或限制。

* 还有其他评论吗？没有任何。

数据集维护

* 谁在支持/托管/维护数据集？此项工作的作者正在支持/托管/维护该数据集。

* 数据集会更新吗？如果会，更新频率是多少？更新者是谁？我们欢迎表格数据社区的更新。如果创建了新算法，作者可以打开拉取请求以包含他们的方法。

* 如何传达更新？（例如邮件列表、GitHub）更新将在 GitHub README 上发布https://github.com/naszilla/tabzilla。

* 如果数据集过时了，如何传达？如果数据集过时了，将会在 GitHub README 上告知https://github.com/naszilla/tabzilla。

* 如果其他人想要扩展/增强/构建此数据集，是否有机制供他们这样做？如果有，是否有流程来跟踪/评估这些贡献的质量。将这些贡献传达/分发给用户的流程是什么？其他人可以在 GitHub 上创建拉取请求，其中包含我们基准套件的可能扩展，这些请求将逐案批准。例如，新硬表格数据集的作者可以使用新数据集在我们的代码库中创建 PR。这些更新将再次在 GitHub README 上传达。

法律和道德考虑

* 是否进行了任何伦理审查流程（例如，由机构审查委员会进行）？如果是，请提供这些审查流程的描述，包括结果，以及任何支持文档的链接或其他访问点。没有伦理审查过程。我们注意到我们的基准测试套件由 OpenML 上已公开的现有数据集组成。

* 数据集是否包含可能被视为机密的数据（例如，受法律特权或医患保密保护的数据，包括个人非公开通信内容的数据）？如果是，请提供描述。数据集不包含任何机密数据。

* 数据集中是否包含直接查看时可能具有冒犯性、侮辱性、威胁性或可能引起焦虑的数据？如果是，请说明原因这些数据均不得具有冒犯性、侮辱性、威胁性，或以其他方式引起焦虑。

* 数据集与人有关吗？如果没有，你可以跳过本部分的其余问题。数据集与人无关。

* 还有其他评论吗？没有任何。

C 其他相关工作

梯度增强决策树。GBDT 迭代地构建决策树集合，每棵新树都拟合先前树的损失残差，并使用梯度下降来最小化损失。自 2001 年创建以来，GBDT 一直是对表格数据进行建模的强大工具 [22]，并且许多作品都提出了高性能的 GBDT 变体。XGBoost (eXtreme Gradient Boosting) [十三] 使用加权分位数草图和稀疏性感知，使其能够扩展到大型数据集。LightGBM（轻梯度提升机）[40] 使用基于梯度的单边采样和独有的特征捆绑来创建更快、更轻量的 GBDT 实现。CatBoost（分类增强）[54] 引入了有序提升，一种处理分类特征的新方法，以及处理缺失值和异常值的更好方法。

用于表格数据的神经网络。Borisov 等人在针对表格数据的深度学习的调查中，描述了三种用于神经网络的表格数据方法[8] 数据转换方法[三十，70] 试图将数据编码为更适合神经网络的格式。基于架构的方法为表格数据设计专门的神经架构 [12，二十九，53]，其中很大一部分是基于 Transformer 的架构[5，二十七，三十六，62基于正则化的方法专门定制正则化方法来提高给定架构的性能[三十七，三十九，60值得注意的是，最近的一项研究设计了一个基于潜在单元归因的表格设置中的正则化新框架[三十七]，而另一项近期的研究表明，寻找应用于简单神经网络的 13 种正则化技术的最佳组合/混合可实现出色的性能。虽然正则化不是我们当前工作的重点，但包括正则化方法在内的后续工作将是一个令人兴奋的方向。

最近一个值得注意的表格数据神经网络是 TabPFN [33，三十四]，一个先验数据拟合的网络[50，51] 用于表格数据。TabPFN 是一个*元学习*可证明近似贝叶斯推理的算法，可以在不到一秒的时间内对新数据集进行预测。PFN 最近被应用于学习曲线外推[1]，预测[41]和贝叶斯优化[49]。

GBDT 与 NN。最近的几篇研究成果将 GBDT 与表格数据的 NN 进行了比较，结果发现*任何一个*神经网络[5，二十七，三十九，53] 或 GBDT [8，二十七，二十八，61] 表现最佳。Shwartz-Ziv 和 Armon 在 30 个数据集上比较了 GBDT 和 NN，发现 GBDT 的平均表现更佳，而将两者结合起来可获得更好的性能 [61Kadra 等人在 40 个数据集上比较了 GBDT 和 NN，发现经过适当调整的神经网络平均表现最佳 [三十九]。

Gorishniy 等人[二十七] 引入类似 ResNet 的 [31] 架构，以及基于 Transformer 的 FT-Transformer [67] 架构。通过对 11 个数据集的实验，他们得出结论，GBDT 和 NN 之间仍然没有普遍的赢家。Borisov 等人。[8] 在五个表格数据集上将经典机器学习方法与 11 种深度学习方法进行了比较，得出的结论是 GBDT 仍然具有优势。在迁移学习设置中，Levin 等人。[43] 发现，当有预训练数据时，神经网络比 GBDT 具有决定性的优势。

也许与我们最相关的工作是由 Grinsztajn 等人完成的。[二十八]，他们研究了为什么基于树的方法在表格数据上的表现优于神经网络。他们的工作与我们的工作有一些不同。首先，他们只考虑了 7 种算法和 45 个数据集，而我们考虑了 19 种算法和 176 个数据集。其次，他们的数据集大小范围从 3 000 到 10 000，或者 7 个正好是 50 000，而我们的数据集大小范围从 32 到 1 025 009（见表6）此外，他们还进一步控制他们的研究，例如通过限制尺寸与特征的比率，通过删除

表 6：我们实验中使用的所有 176 个数据集的汇总统计数据。左列显示实例数、特征数、目标类数以及任何类的最小频率与任何类的最大频率之比（最小-最大类频率）。右列显示特征类型的数量。除类频率比之外的所有统计数据均四舍五入为最接近的整数。

	# 安装。	# 壮举。	# 类	最小-最大类频率。	# 特征类型		
					编号	垃圾桶。	猫。
意思是	30567	223	6	0.48	206	二十五	17
标准	106943	786	12	0.35	781	144	119
分钟	三十二	2	2	2e-05	0	0	0
25%	596	9	2	0.14	4	0	0
50%	2218	21	2	0.46	10	0	0
75%	11008	61	6	0.82	50	2	8
最大限度	1025009	7200	100	1.00	7200	1555	1555

高基数分类特征，并删除低基数数值特征。虽然这有利于成为一项更受控制的研究，但他们的分析忽略了我们的一些观察结果，例如 GBDT 在“不规则”数据集上的表现优于 NN。最后，虽然 Grinsztajn 等人深入研究了一些元特征，例如数据集平滑度和无信息特征数量，但我们的工作考虑了更多数量级的元特征。同样，虽然每种方法都有自己的优势，但我们的工作能够为从业者发现更多潜在的见解、相关性和收获。

D 附加实验

我们进行了额外的实验，包括数据集统计（附录 D.1), 其他结果来自 第 2.1 节（附录 D.2）以及来自第 2.2 节（附录 D.4）。

D.1 数据集统计

表 6显示了我们在实验中使用的所有 176 个数据集的汇总统计信息。我们的数据集中大约有一半具有二元分类目标（以及多达 100 个目标类别），大约有一半的训练集的实例少于 2300 个 - 尽管许多数据集有数万个实例。

D.2 来自的附加实验第 2.1 节

在本节中，我们给出了额外的实验第 2.1 节，包括相对绩效表、训练时间分析和关键差异图。

D.2.1 相对绩效表

首先，我们根据每个性能指标显示所有调整后的算法的排名，并按数据集取平均值：对数损失（表 7)、F1 得分（表 8) 和 ROC-AUC (表 9）这些类似于 表格1，但指标不同。排名的计算方法是，首先对每个数据集的所有 10 个分割的调整后性能进行平均，然后根据其平均性能对每个数据集的每个算法进行排名。对于这些排名，两个算法之间的平局由*最低*（最佳）排名。因此，如果多个算法对数据集的平均准确率最高，则它们都会获得排名 1。

有些算法甚至表现良好*没有*超参数调整。接下来，我们使用与表格1和前面的表格，但其默认超参数集显示为单独的算法。我们展示测试准确率的排名结果。参见 表 10。许多性能最佳的算法，包括 CatBoost、XGBoost、LightGBM 和 ResNet，无论是否进行超参数调整，都表现得相当好。

表 7：根据 98 个数据集的对数损失对算法的性能。列显示所有数据集的排名、平均归一化对数损失 (平均 LL)、各折叠的归一化 LL 标准差 (标准 LL) 以及每 1000 个实例的训练时间。最小值/最大值/平均值/中位数取自所有数据集。

算法	班级	秩			平均 LL			标准 LL		次/1000实例	
		最小	最大	平均值	平均	平均	平均	平均		意思是	医学
表PFN*	PFN	1	16	5.02	4	0.06	0.01	0.07	0.04	0.25	0.01
CatBoost	国别分类法	1	16	5.73	5	0.05	0.02	0.08	0.06	13.89	1.66
XGBoost	国别分类法	1	17	6.36	6	0.05	0.03	0.08	0.06	0.73	0.37
圣	神经网络	1	21	8.49	8	0.12	0.06	0.09	0.08	202.59	173.23
残差网络	神经网络	1	18	9.67	9	0.12	0.08	0.10	0.08	16.12	8.97
支持向量机	根据	1	20	9.70	10	0.15	0.06	0.08	0.05	49.83	1.20
轻量级GBM	国别分类法	1	21	9.76	9	0.13	0.07	0.21	0.09	0.83	0.27
数据网络	神经网络	1	20	10.08	10	0.12	0.09	0.13	0.08	71.58	61.35
FTTransformer	神经网络	1	19	10.85	11.50	0.14	0.09	0.11	0.09	29.58	18.48
星火	神经网络	1	20	11.17	11	0.18	0.07	0.07	0.05	18.82	15.85
随机森林	根据	1	21	12.26	十三	0.19	0.13	0.22	0.07	0.29	0.22
线性模型	根据	1	21	12.28	十三	0.24	0.10	0.10	0.06	0.04	0.03
MLP-RTDL	神经网络	1	21	13.31	14	0.28	0.18	0.17	0.12	13.75	7.96
节点	神经网络	1	20	13.71	14	0.23	0.19	0.04	0.03	196.82	176.16
多层感知处理器	神经网络	1	21	14.27	15	0.29	0.21	0.15	0.11	18.29	10.95
塔格网	神经网络	1	21	14.55	16.50	0.40	0.25	0.38	0.19	34.62	29.69
维梅	神经网络	3	21	16.27	18	0.40	0.37	0.09	0.07	16.92	14.64
KNN	根据	1	21	17.17	18	0.51	0.42	0.37	0.20	0.01	0.00
决策树	根据	1	21	17.46	20	0.56	0.58	0.54	0.41	0.02	0.01

表 8：根据 98 个数据集的 F1 得分，算法的性能。列显示所有数据集的排名、平均归一化 F1 得分（平均 F1）、归一化 F1 得分的标准差（标准 F1）以及每 1000 个实例的训练时间。最小值/最大值/平均值/中位数取自所有数据集。

算法	班级	秩			平均 F1			标准 F1		次/1000实例	
		最小值	最大值	平均值	平均值	平均值	平均值			意思是	医学
CatBoost	国别分类法	1	19	6.43	5	0.87	0.93	0.29	0.22	21.00	2.08
表PFN*	PFN	1	20	6.46	5.50	0.83		0.93	0.25	0.19	0.25	0.01
XGBoost	国别分类法	1	19	7.78	6	0.82	0.89	0.32	0.22	0.83	0.37
残差网络	神经网络	1	21	8.74	8.50	0.76		0.82	0.29	0.19	16.04	9.34
节点	神经网络	1	21	9.23	9	0.75	0.81	0.25	0.19	140.71	117.04
圣	神经网络	1	21	9.26	9	0.73	0.85	0.30	0.23	171.14	144.37
FTTransformer	神经网络	1	19	9.32	9	0.76	0.82	0.30	0.19	27.94	18.40
随机森林	根据	1	21	9.46	9	0.77	0.83	0.31	0.21	0.36	0.25
轻量级GBM	国别分类法	1	21	9.62	9	0.76	0.83	0.35	0.21	0.86	0.31
支持向量机	根据	1	20	10.30	11	0.69	0.77	0.25	0.17	29.99	1.73
MLP-RTDL	神经网络	1	21	10.96	12	0.66	0.73	0.27	0.16	14.29	7.30
数据网络	神经网络	1	20	11.01	11	0.74	0.81	0.31	0.22	69.54	60.20
决策树	根据	1	21	13.19	15	0.61	0.71	0.34	0.24	0.03	0.01
星火	神经网络	1	21	13.24	14	0.57	0.65	0.28	0.18	18.43	15.76
多层感知处理器	神经网络	1	21	13.69	15	0.57	0.58	0.29	0.18	18.42	11.20
线性模型	根据	1	21	13.83	16	0.52	0.53	0.30	0.24	0.04	0.03
塔格网	神经网络	1	21	14.22	16	0.56	0.60	0.38	0.26	34.82	29.16
KNN	根据	1	21	15.21	17	0.46	0.55	0.28	0.21	0.01	0.00
维梅	神经网络	3	21	16.87	19	0.36	0.34	0.27	0.18	17.02	14.96

表 9：根据 ROC-AUC 计算的算法性能，超过 98 个数据集。列显示所有数据集的排名、平均归一化 ROC-AUC 分数（平均 AUC）、归一化 ROC-AUC 分数的标准差（标准 AUC）以及每 1000 个实例的训练时间。最小值/最大值/平均值/中位数取自所有数据集。

算法	班级	秩			平均AUC			标准AUC		次/1000实例	
		最小	最大	平均值	平均	平均	平均	平均		意思是	医学
表PFN*	PFN	1	17	5.30	3	0.91	0.97	0.17	0.07	0.25	0.01
CatBoost	国别分类法	1	20	6.43	5	0.91	0.96	0.18	0.08	20.51	1.94
XGBoost	国别分类法	1	19	7.17	6	0.90	0.96	0.19	0.09	0.84	0.38
残差网络	神经网络	1	21	8.23	7	0.84	0.93	0.19	0.12	15.83	8.78
圣	神经网络	1	21	8.35	7.50	0.84	0.93	0.19	0.12	170.31	145.99
随机森林	根据	1	20	8.64	8	0.87	0.94	0.19	0.10	0.41	0.28
数据网络	神经网络	1	19	9.36	9	0.83	0.91	0.19	0.08	64.15	57.12
轻量级GBM	国别分类法	1	21	10	9	0.83	0.91	0.22	0.10	0.89	0.29
节点	神经网络	1	21	10.55	11	0.81	0.90	0.20	0.14	160.58	131.56
FTTransformer	神经网络	1	21	10.74	11.50	0.79	0.88	0.20	0.13	27.73	18.00
支持向量机	根据	1	21	11.26	12.50	0.75	0.87	0.22	0.10	61.16	2.01
MLP-RTDL	神经网络	1	21	11.31	11	0.73	0.83	0.21	0.12	15.05	7.01
星火	神经网络	1	21	12.85	14	0.66	0.79	0.24	0.15	18.58	15.98
线性模型	根据	1	21	13.62	15	0.62	0.75	0.23	0.17	0.04	0.03
多层感知处理器	神经网络	1	21	14.13	15	0.65	0.71	0.23	0.14	18.17	11.16
塔格网	神经网络	1	21	14.46	16	0.63	0.75	0.32	0.19	35.06	29.32
决策树	根据	1	21	15.66	17	0.53	0.59	0.30	0.23	0.02	0.01
KNN	根据	1	21	16.40	18	0.52	0.56	0.25	0.19	0.01	0.00
维梅	神经网络	1	21	16.74	18	0.49	0.50	0.30	0.20	17.84	15.55

现在我们展示一个类似表格1，但包含 176 个数据集。请参阅表 11。请注意，有些算法并未在所有 176 个数据集上完成，但算法的顺序与表格1. 我们还包括 NAM 的部分结果 [2], DeepFM [二十九] 和 TabTransformer [三十六]。

D.2.2 训练时间分析

在本节中，我们将分析每种算法所需的相对训练时间。这里我们只考虑具有以下特征的算法：*默认*超参数，因此不需要进行调整。表 12根据每 1,000 个训练样本的平均训练时间显示所有算法的排名；与以前一样，我们分别展示了小于或等于 1,250 的数据集的结果表 13。这些排名是通过首先取所有 176 个数据集的所有 10 倍中每 1 000 个样本的平均训练时间，然后根据这个平均训练时间对每个数据集的每个算法进行排名来计算的。

D.3 其他 TabPFN 结果

在本节中，我们展示了额外的结果，其中包括两个修改版的 TabPFN，一个使用 3000 个训练点的随机子集（称为 TabPFN-3k 或 TabPFN*在主论文中），以及使用 1000 个随机采样训练点的版本（TabPFN-1k）。作为消融，我们还测试了使用 1000 个训练点子集的 CatBoost 版本（CatBoost-1k）。

该分析包括 98 个数据集和 21 个算法（包括 TabPFN-3k、TabPFN-1k 和 CatBoost-1k）的子集，其中所有算法都会为所有数据集产生结果。对于类似于 表格1，但使用 TabPFN 和 CatBoost 消融，请参见表 14（其中性能指标是准确度）。另请参阅对数损失的相同表格（表 15) 和 F1 得分 (表 16）接下来，我们展示类似于图 3，但性能指标是准确度：图 6。最后，我们展示了准确率与时间的关系图，类似于图 2，但其中包括 TabPFN 和 CatBoost 消融：图 7。

D.3.1 HPO 图

在图 4中，我们绘制了 CatBoost 上超参数调整的性能改进，并与使用默认超参数的最佳神经网络和最佳 GBDT 之间的绝对性能差异进行了比较。现在，我们也为 ResNet 绘制了相同的图。参见图 8。

表 10：根据准确率，包括使用默认超参数参数化的算法在内的算法在 104 个数据集上的性能。列显示所有数据集的排名、平均归一化准确率 (M ean Acc.)、各倍归一化准确率的标准差 (Std. Acc.) 以及每 1000 个实例的训练时间。最小值/最大值/平均值/中位数取自所有数据集。

算法	秩 分钟	最大限度	意思是	医学	平均加速度。 意思是	医学。	标准配件 意思是	医学	次/1000实例 意思是	医学
CatBoost	1	三十	8.05	6.0	0.91	0.95	0.20	0.12	30.27	2.22
CatBoost（默认）	1	31	10.12	7.0	0.87	0.93	0.20	0.12	21.95	1.53
XGBoost	1	三十	10.77	8.5	0.87	0.93	0.22	0.13	0.94	0.42
残差网络	1	三十四	11.51	11.5	0.84	0.90	0.20	0.11	16.07	10.04
XGBoost（默认）	1	三十四	12.02	10.5	0.85	0.91	0.22	0.13	1.20	0.61
节点	1	33	12.17	12.0	0.82	0.88	0.18	0.12	146.89	118.94
圣	1	三十四	12.27	10.0	0.81	0.91	0.20	0.15	168.14	144.84
FTTransformer	1	31	12.78	13.0	0.83	0.87	0.21	0.14	29.47	18.63
随机森林	1	33	13.00	12.0	0.83	0.88	0.21	0.15	0.36	0.25
轻量级GBM	1	三十四	13.19	12.0	0.85	0.90	0.23	0.14	1.06	0.37
ResNet（默认）	1	三十五	13.94	14.0	0.79	0.87	0.22	0.14	15.26	8.67
支持向量机	1	三十二	14.24	16.0	0.78	0.88	0.18	0.13	29.22	1.37
LightGBM（默认）	1	三十四	14.38	12.5	0.80	0.88	0.23	0.16	1.30	0.50
SAINT （默认）	1	三十五	14.48	12.0	0.76	0.88	0.20	0.15	135.22	107.82
NODE（默认）	1	三十四	14.61	15.0	0.77	0.86	0.18	0.12	67.54	50.85
数据网络	1	33	15.08	15.0	0.83	0.87	0.21	0.15	69.29	60.15
随机森林（默认）	1	三十五	16.22	15.0	0.76	0.82	0.20	0.14	0.50	0.39
MLP-RTDL	1	三十四	16.28	18.0	0.74	0.83	0.18	0.11	14.33	7.62
FTTransformer（默认）	1	三十五	18.01	20.0	0.71	0.80	0.21	0.14	25.92	15.35
STG	1	三十四	18.73	19.0	0.70	0.81	0.20	0.11	18.47	16.00
决策树	1	三十五	18.93	20.0	0.73	0.79	0.23	0.18	0.03	0.01
SVM（默认）	1	三十五	19.63	24.0	0.64	0.74	0.19	0.12	1.09	0.38
多层感知处理器	1	三十四	19.66	21.5	0.69	0.73	0.20	0.11	18.61	11.92
线性模型	1	三十五	19.95	22.0	0.64	0.71	0.22	0.15	0.04	0.03
MLP-rtdl（默认）	1	三十五	20.11	21.0	0.63	0.78	0.20	0.13	13.07	6.33
DANet（默认）	1	33	20.19	22.0	0.71	0.74	0.21	0.15	45.04	38.53
塔格网	1	三十五	20.56	22.5	0.68	0.78	0.25	0.18	35.09	30.83
决策树（默认）	1	三十五	21.88	23.0	0.63	0.70	0.24	0.16	0.02	0.01
KNN	1	三十五	22.88	25.0	0.59	0.62	0.20	0.15	0.01	0.00
TabNet（默认）	1	三十五	23.52	25.0	0.60	0.70	0.29	0.22	28.39	25.93
MLP（默认）	1	三十五	23.78	27.0	0.56	0.59	0.24	0.18	17.37	11.30
KNN（默认）	1	三十五	24.61	26.5	0.54	0.57	0.21	0.16	0.01	0.00
维梅	3	三十四	25.16	27.0	0.53	0.59	0.17	0.13	17.10	14.97
STG（默认）	1	三十五	26.17	30.0	0.44	0.38	0.18	0.11	16.39	13.74
VIME（默认）	6	三十五	30.78	33.0	0.23	0.05	0.23	0.14	15.75	14.10

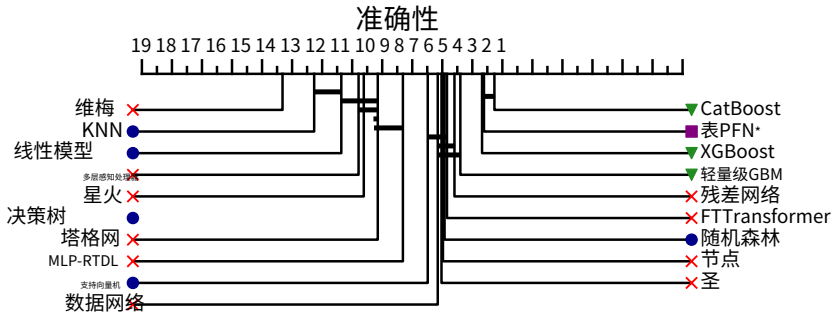

图 6：根据平均准确率等级比较所有算法的临界差异图
超过 98 个数据集。每个算法的平均排名在轴上显示为一条水平线。 *算法没有显著差异*A 由一条水平黑条连接。 r

表 11：21 种算法在所有 176 个数据集上的表现（按准确率计算）。列显示所有数据集的排名、平均归一化准确率（平均准确率）和归一化准确率的标准差（标准准确率）。最小值/最大值/平均值/中位数取自所有数据集。最右边的列显示每个算法在 176 个数据集中成功运行的数据集数量。

算法	秩 分钟	最大限度	意思是	医学	平均加速度。 意思是	医学。	标准配件 意思是	医学	
CatBoost	1	18	5.21	4.0	0.88	0.94	0.22	0.11	165
XGBoost	1	19	5.60	4.0	0.88	0.96	0.24	0.12	174
残差网络	1	20	6.80	7.0	0.79	0.88	0.22	0.10	174
轻量级GBM	1	21	6.98	6.0	0.83	0.92	0.27	0.12	165
圣	1	20	7.55	7.0	0.76	0.88	0.25	0.13	138
节点	1	20	7.57	7.0	0.76	0.84	0.23	0.14	141
随机森林	1	20	8.10	8.0	0.77	0.84	0.22	0.10	173
FTTransformer	1	17	8.15	8.0	0.76	0.81	0.25	0.14	148
支持向量机	1	19	8.41	8.0	0.74	0.85	0.21	0.14	143
数据网络	1	20	8.73	8.0	0.77	0.83	0.26	0.15	147
MLP-RTDL	1	19	9.57	10.0	0.67	0.75	0.20	0.09	176
深度调频	1	21	10.89	11.5	0.64	0.74	0.26	0.19	90
塔格网	1	21	11.05	10.0	0.64	0.75	0.29	0.13	168
多层感知处理器	1	21	11.36	12.0	0.62	0.64	0.21	0.12	175
决策树	1	21	11.40	12.0	0.60	0.65	0.25	0.13	175
标签转换器	1	21	11.52	12.0	0.58	0.66	0.16	0.10	124
星火	1	21	11.55	12.0	0.60	0.69	0.22	0.11	164
线性模型	1	21	12.30	14.0	0.51	0.57	0.23	0.13	168
KNN	1	21	12.83	14.0	0.53	0.58	0.22	0.13	167
维梅	1	21	14.58	16.0	0.41	0.40	0.21	0.12	163
不结盟运动	1	21	15.94	17.0	0.35	0.26	0.25	0.17	80

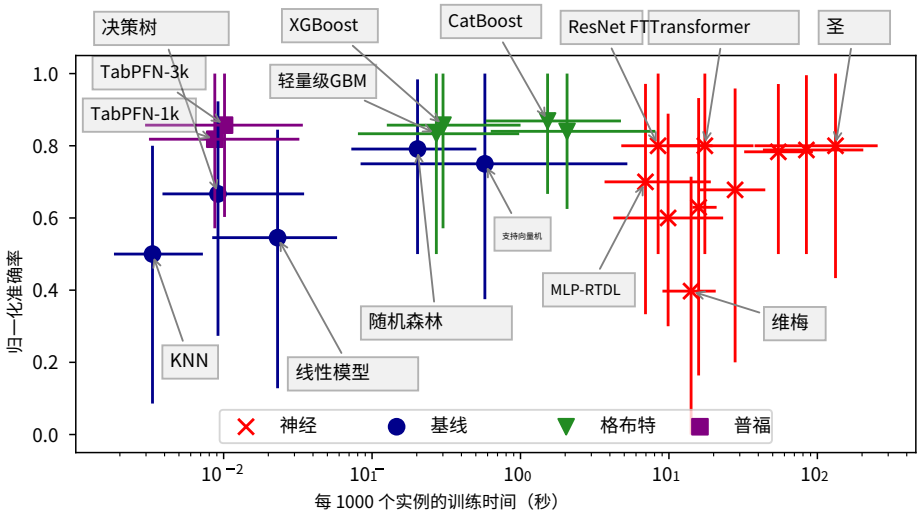

图 7：每个算法的中位运行时间与中位归一化准确率，包括两个变量 国家标准 TabPFN，超过 98 个数据集。条形图涵盖所有数据集的第 20 到第 80 个百分位数。

表 12：根据每 1,000 个样本的平均训练时间，对所有 176 个数据集的所有算法进行排名。排名越低表示训练时间越短。排名列显示所有数据集的最小、最大和平均排名。右列显示所有 10 个训练组合中每 1,000 个样本的平均训练时间，以及每个算法考虑的数据集数量。

算法	秩 *最小最大平均值*	平均列车时间 （s/1000 个样本）	# 数据集
KNN	1 5 1.66	0.04	167
决策树	1 8	2.33 0.18	175
线性模型	1 4	2.45 0.05	168
随机森林	1 8	5.05 0.47	173
XGBoost	1 16	5.70 1.68	174
轻量级GBM	4 14	6.67 2.64	165
支持向量机	3 20	6.74 3.76	141
CatBoost	1 19	7.32 15.36	165
MLP-RTDL	2 17	9.23 9.21	176
深度调频	7 16	9.64 5.52	90
多层感知处理器	3 17	10.43 12.19	175
残差网络	4 16	10.46 11.58	174
标签转换器	5 19	12.79 17.28	122
维梅	8 19	13.12 18.25	156
星火	8 20	13.18 15.26	164
FTTransformer	7 18	13.47 21.97	148
塔格网	9 20	15.37 26.41	160
数据网络	11 21	17.20 42.10	146
节点	7 21	17.44 60.76	141
不结盟运动	12 21	18.34 129.18	79
圣	10 21	18.38 119.49	124

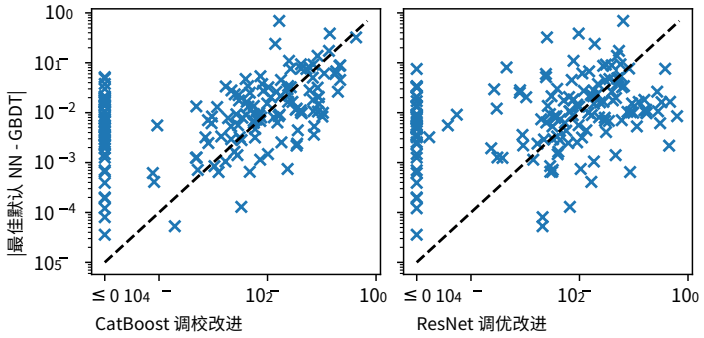

图 8：CatBoost（左）和 ResNet（右）的超参数调整（横轴）的性能改进与使用默认超参数（纵轴）的最佳神经网络和最佳 GBDT 之间的绝对性能差异相比。每个点表示不同的数据集，所有值均为归一化对数损失。虚线上的点表示，由于超参数调整而导致的性能改进与由于调整而导致的差异相同

NN-vs-GBDT 算法选择。

D.3.2 关键差异 ce d图表

图 9出现四次关键 d 且排伊夫根据 F1 分数绘制的曲线图。注意相 我们重复弗里德曼检验，我们名为 o 的情况会发生变 第f同的数据集和算法。然而，这些测试 的发现不太可能非常小（p<10 化，因为 聚醋酸乙烯酯没有校正 −20）。

我们比较了 nce每种算法*家庭*（GBDT，N Ns 和基线）。这里我们使用 所有 176 个数据集。我们使 他是这里的方法与之前的方法相同 节。请参阅图 10（对数损失）用 t 和图 11（F1 分数）。

表 13：根据每 1,000 个样本的平均训练时间，对 57 个大小小于或等于 1,250 的数据集的所有 22 种算法进行排名。排名越低表示训练时间越短。排名列显示所有数据集的最小、最大和平均排名。右列显示所有 10 个训练折中每 1,000 个样本的平均训练时间，以及每个算法考虑的数据集数量。

算法	秩 *最小*	*最大*	平均列车时间 *平均值*（s/1000 个样本）	# 数据集
表PFN	1	3	1.16 0.00	57
KNN	1	3	2.00 0.00	57
决策树	1	4	2.88 0.02	57
线性模型	3	5	4.25 0.06	57
支持向量机	4	8	5.23 0.22	57
轻量级GBM	5	12	6.47 0.85	57
随机森林	5	9	6.84 0.70	57
XGBoost	6	9	7.35 1.25	57
CatBoost	8	19	9.67 17.94	57
MLP-RTDL	9	17	11.51 20.26	57
残差网络	10	16	12.14 22.70	57
维梅	9	16	12.49 18.03	57
星火	9	15	12.56 18.99	57
多层感知处理器	9	17	13.26 26.30	57
FTTransformer	11	18	13.86 29.09	57
塔格网	10	18	15.04 32.14	57
数据网络	15	19	17.39 53.91	57
节点	十三	19	17.40 61.94	57
圣	16	19	18.51 155.07	57

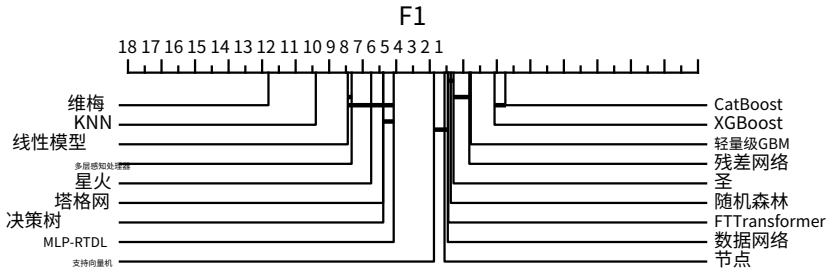

图 9：根据 F1 分数比较所有算法的关键差异图。每个算法的平均排名在轴上显示为一条水平线。*没有显著差异*由一条水平的黑条连接。

D.3.3 维恩图

回想一下第 2 部分对于我们的 Venn diag GB收道 洛TS瓦斯普亮19 艾尔里特卡毫秒家庭: elines 决（CatBoost、XGBoost、LightGBM）、NN（li KNN斯特德第 2 节), 一d巴保树，（LinearModel、RandomForest、SVM）。要使用最小平缓艾尔交 射频漫融 操作系统，我们缩放比例，如中所述第 2 部分。 在图古尔 埃4，我们ID 算法是“高如果它达到了至少0。99，然后我们确定哪个 n
算法系列（GBDT、NN、基线）具有高性能算法。现在我们计算相同的维恩图，将高性能的定义收紧为0。9999缩放准确度。
看图 12。在这种情况下，GBDT 是 42% 数据集的唯一高性能算法， F
而 NN 是 30% 数据集中唯一高性能算法系列。然而，由于这些差异小于 0.1%，因此对从业者来说可能并不重要。

表 14：算法在 98 个数据集上的表现，其中算法包括两个修改版的 TabPFN。列显示算法系列（GBDT、NN、baseline 或 PFN）、所有数据集的排名、平均归一化准确度（Mean Acc.）、各折叠归一化准确度的标准差（Std. Acc.）以及每 1000 个实例的训练时间（以秒为单位）。这些数量的最小值/最大值/平均值/中位数取自所有数据集。

算法	班级	秩			平均加速度。			标准配件		次/1000实例	
		最小值	最大值	平均值	平均值	平均值	平均值			意思是	医学。
CatBoost	国别分类法	1	19	6.12	5	0.87	0.93	0.3	0.22	21.7	2.08
TabPFN-3k	PFN	1	19	6.43	5	0.84	0.92	0.26	0.19	0.25	0.01
CatBoost-1k	国别分类法	1	21	7.02	5.5	0.85	0.9	0.3	0.23	6.76	2.4
XGBoost	国别分类法	1	19	7.85	6.5	0.81	0.89	0.33	0.22	0.81	0.37
TabPFN-1k	PFN	1	21	7.94	7	0.8	0.91	0.27	0.19	0.25	0.01
残差网络	神经网络	1	21	8.73	8	0.75	0.83	0.3	0.21	16.01	9.34
节点	神经网络	1	21	9.08	9	0.74	0.81	0.26	0.2	138.36	117.04
圣	神经网络	1	21	9.09	8	0.73	0.86	0.31	0.24	169.54	146.16
FTTransformer	神经网络	1	20	9.29	9	0.76	0.8	0.31	0.21	27.67	18.4
随机森林	根据	1	21	9.5	9	0.76	0.83	0.32	0.22	0.35	0.24
轻量级GBM	国别分类法	1	21	9.61	9	0.76	0.84	0.36	0.21	0.87	0.34
支持向量机	根据	1	20	10.34	11.5	0.69	0.76	0.26	0.19	30.4	1.67
数据网络	神经网络	1	20	11.07	11	0.73	0.79	0.32	0.23	68.82	60.15
MLP-RTDL	神经网络	1	21	11.1	12	0.65	0.72	0.28	0.16	14.27	7.3
星火	神经网络	1	21	13.19	14	0.56	0.63	0.29	0.17	18.44	15.79
决策树	根据	1	21	13.32	15	0.59	0.68	0.35	0.25	0.03	0.01
多层感知处理器	神经网络	1	21	13.49	15	0.57	0.57	0.29	0.18	18.39	11.2
线性模型	根据	1	21	13.76	16	0.51	0.53	0.31	0.24	0.04	0.03
塔格网	神经网络	1	21	14.24	16	0.54	0.6	0.39	0.25	34.95	29.9
KNN	根据	1	21	15.32	17	0.45	0.51	0.29	0.21	0.01	0.0
维梅	神经网络	3	21	16.73	19	0.37	0.32	0.27	0.18	16.81	14.86

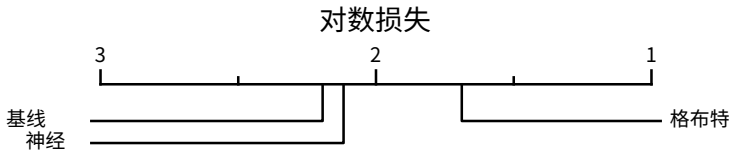

图 10：根据对数损失比较三种算法类型的临界差异图。每种算法的平均排名在轴上显示为一条水平线。*没有显著差异*由一条水平的黑条连接。

D.3.4 数据集规模分析

在本节中，我们研究数据集大小与性能之间的关联。第 2.2 节我们表明，与 NN 和基线相比，GBDT 在较大的数据集上表现相对较好；这是基于归一化对数损失和数据集大小之间的负相关性。然而，比较单个算法的性能比比较算法系列更有参考价值。例如，图 13比较了三种算法的排名：CatBoost、SAINT 和 TabNet，以及数据集大小。在左侧面板（CatBoost 减去 TabNet）中，CatBoost 在所有数据集上的表现都优于 TabNet，最大数据集大小约为 1500。对于较大的数据集，CatBoost 和

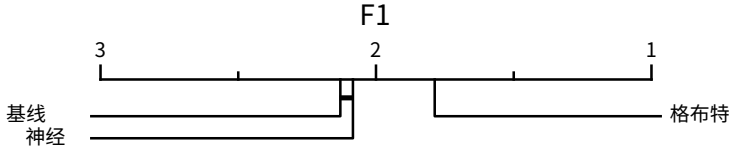

图 11：根据 F1 得分比较三种算法类型的临界差异图。每种算法的平均排名在轴上显示为一条水平线。*没有显著差异*由一条水平的黑条连接。

表 15：算法在 98 个数据集上的表现，其中算法包括两个修改版的 TabPFN。列显示算法系列（GBDT、NN、基线或 PFN）、所有数据集的排名、平均归一化对数损失（平均 LL）、各折叠归一化 LL 的标准差（标准 LL）以及每 1000 个实例的训练时间（秒）。这些数量的最小值/最大值/平均值/中位数取自所有数据集。

算法	班级	秩			平均 LL			标准 LL		次/1000实例	
		最小值	最大值	平均值	平均值	平均值	平均值			意思是	医学
TabPFN-3k	PFN	1	16	5.02	4	0.06	0.01	0.07	0.04	0.25	0.01
CatBoost	国别分类法	1	16	5.73	5	0.05	0.02	0.08	0.06	13.89	1.66
CatBoost-1k	国别分类法	1	18	5.93	5	0.05	0.02	0.08	0.05	5.59	1.84
TabPFN-1k	PFN	1	17	6.21	5	0.07	0.02	0.08	0.04	0.25	0.01
XGBoost	国别分类法	1	17	6.36	6	0.05	0.03	0.08	0.06	0.73	0.37
圣	神经网络	1	21	8.49	8	0.12	0.06	0.09	0.08	202.59	173.23
残差网络	神经网络	1	18	9.67	9	0.12	0.08	0.1	0.08	16.12	8.97
支持向量机	根据	1	20	9.7	10	0.15	0.06	0.08	0.05	49.83	1.2
轻量级GBM	国别分类法	1	21	9.76	9	0.13	0.07	0.21	0.09	0.83	0.27
数据网络	神经网络	1	20	10.08	10	0.12	0.09	0.13	0.08	71.58	61.35
FTTransformer	神经网络	1	19	10.85	11.5	0.14	0.09	0.11	0.09	29.58	18.48
星火	神经网络	1	20	11.17	11	0.18	0.07	0.07	0.05	18.82	15.85
随机森林	根据	1	21	12.26	十三	0.19	0.13	0.22	0.07	0.29	0.22
线性模型	根据	1	21	12.28	十三	0.24	0.1	0.1	0.06	0.04	0.03
MLP-RTDL	神经网络	1	21	13.31	14	0.28	0.18	0.17	0.12	13.75	7.96
节点	神经网络	1	20	13.71	14	0.23	0.19	0.04	0.03	196.82	176.16
多层感知处理器	神经网络	1	21	14.27	15	0.29	0.21	0.15	0.11	18.29	10.95
塔格网	神经网络	1	21	14.55	16.5	0.4	0.25	0.38	0.19	34.62	29.69
维梅	神经网络	3	21	16.27	18	0.4	0.37	0.09	0.07	16.92	14.64
KNN	根据	1	21	17.17	18	0.51	0.42	0.37	0.2	0.01	0.0
决策树	根据	1	21	17.46	20	0.56	0.58	0.54	0.41	0.02	0.01

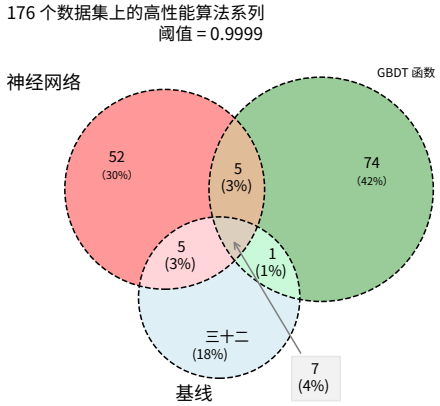

图 12：在所有 176 个数据集中，算法系列“高性能”的数据集数量的维恩图。如果算法在 0-1 缩放后的测试准确率高于某个阈值，则该算法是高性能的。而图 4使用了阈值0.99，这个图使用了阈值0.9999。

TabNet；这表明对于较小的数据集，应该选择 CatBoost 而不是 TabNet，而对于较大的数据集，这两种算法是相当的。另一方面，中心面板（CatBoost 减去 SAINT）表明 CatBoost 和 SAINT 在所有数据集上都具有相当的性能 *取决于*对于大小为 1 500 的数据集；对于较大的数据集，CatBoost 的表现优于 SAINT。这表明，对于非常大的数据集，应该选择 CatBoost 而不是 SAINT，但对于小数据集，两种算法是相当的。

主要结论这些发现的结论是，从业者不应该专注于选择要关注的算法系列，例如 NN 或 GBDT。例如，TabPFN 和 TabNet 都是神经网络，但 TabPFN 在较小的数据集上表现相对较好，而 TabNet 在较大的数据集上表现相对较好。相反，他们可以参考我们的结果元数据集来决定

表 16：算法在 98 个数据集上的表现，其中算法包括两个修改版的 TabPFN。列显示算法系列（GBDT、NN、基线或 PFN）、所有数据集的排名、平均归一化 F1 分数损失（平均 F1）、归一化 F1 的标准差（标准 F1）以及每 1000 个实例的训练时间（秒）。这些数量的最小值/最大值/平均值/中位数取自所有数据集。

算法	班级	秩				平均 F1		标准 F1		次/1000实例	
		最小值	最大值	平均值	平均值	平均值	平均值			意思是	医学
CatBoost	国别分类法	1	19	6.43	5	0.87	0.93	0.29	0.22	21.0	2.08
TabPFN-3k	PFN	1	20	6.46	5.5	0.83	0.93	0.25	0.19	0.25	0.01
CatBoost-1k	国别分类法	1	21	7.09	6	0.85	0.91	0.29	0.22	6.94	2.43
XGBoost	国别分类法	1	19	7.78	6	0.82	0.89	0.32	0.22	0.83	0.37
TabPFN-1k	PFN	1	21	8.01	7	0.8	0.91	0.26	0.19	0.25	0.01
残差网络	神经网络	1	21	8.74	8.5	0.76	0.82	0.29	0.19	16.04	9.34
节点	神经网络	1	21	9.23	9	0.75	0.81	0.25	0.19	140.71	117.04
圣	神经网络	1	21	9.26	9	0.73	0.85	0.3	0.23	171.14	144.37
FTTransformer	神经网络	1	19	9.32	9	0.76	0.82	0.3	0.19	27.94	18.4
随机森林	根据	1	21	9.46	9	0.77	0.83	0.31	0.21	0.36	0.25
轻量级GBM	国别分类法	1	21	9.62	9	0.76	0.83	0.35	0.21	0.86	0.31
支持向量机	根据	1	20	10.3	11	0.69	0.77	0.25	0.17	29.99	1.73
MLP-RTDL	神经网络	1	21	10.96	12	0.66	0.73	0.27	0.16	14.29	7.3
数据网络	神经网络	1	20	11.01	11	0.74	0.81	0.31	0.22	69.54	60.2
决策树	根据	1	21	13.19	15	0.61	0.71	0.34	0.24	0.03	0.01
星火	神经网络	1	21	13.24	14	0.57	0.65	0.28	0.18	18.43	15.76
多层感知处理器	神经网络	1	21	13.69	15	0.57	0.58	0.29	0.18	18.42	11.2
线性模型	根据	1	21	13.83	16	0.52	0.53	0.3	0.24	0.04	0.03
塔格网	神经网络	1	21	14.22	16	0.56	0.6	0.38	0.26	34.82	29.16
KNN	根据	1	21	15.21	17	0.46	0.55	0.28	0.21	0.01	0.0
维梅	神经网络	3	21	16.87	19	0.36	0.34	0.27	0.18	17.02	14.96

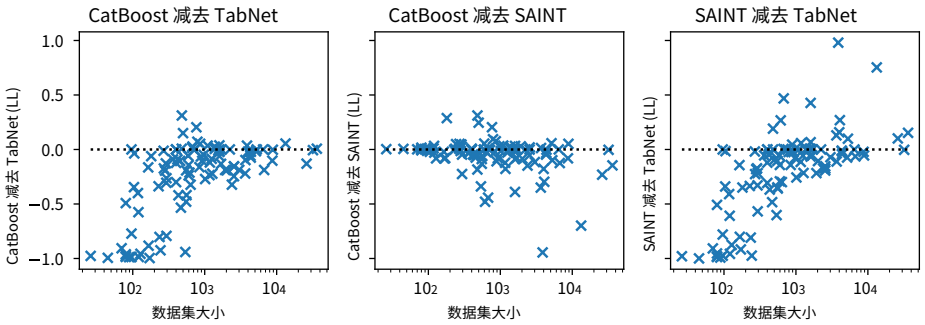

图 13：三种算法（CatBoost、SAINT 和 TabNet）之间的归一化对数损失差异（越低越好），以数据集大小绘制。虚线上方的点表示第二种算法的对数损失较低，这意味着性能优于第一种算法。

哪种算法适合他们的具体用例。总体趋势有助于选择有效的算法，但还不够。

D.4 附加实验　　条款　从第 2.2 节

回想一下第 2.2 节 神经　那Δ　ℓℓ表示规范方法的差异。表 17和图 zed 日志 最佳设置属性与网络和最佳G最大绝对　苯二氮14节目　　数据相关　　与　　Δℓℓ。

为了评估预测训练/测试收力　数据集属性，我们训练几个二进制 决定　使用 ural net 的树程序（GBDT）和其他任和　结果：1 如果Δℓℓ>0 (这 18显示了性最好的模型击败了在此比较务，不同深度包括视觉 。桌子 能准确性 o 我们还包括一个　决策　中训练的树的最佳模描述　水平；XGBoost 模型，mple 深度 3 决策　对于 型。最后，我们n 的 si 树，数字　15。这 决策树分类

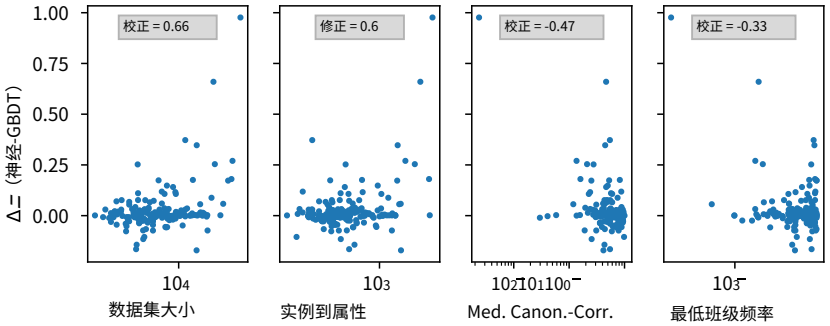

图 14：最佳 NN 与 GBDT 之间的归一化对数损失差异（Δ *ℓℓ*）的四个数据集属性，针对所有 176 个数据集的所有 10 个分割。相关性越大意味着属性值越大，与最佳 GBDT 相比，最佳神经网络的对数损失越大（性能越差）。所有数据集属性均以对数刻度绘制。

表 17：与最佳 NN 和 GBDT 之间的归一化对数损失差异具有最大绝对相关性的选定元特征（Δ *ℓℓ*），涵盖了所有 176 个数据集。较高的相关值表明*更大*元特征的值与*更差*NN 性能和*更强* GBDT 性能。

描述	更正Δ *ℓℓ*
记录实例数。	0.63
数据集大小与特征数量的比例。	0.55
每个特征和目标之间的中值典型相关性的对数。	-0.41
最小目标类别频率的对数。	-0.35

前五种算法中哪一种表现最好。请注意，决策划分完全基于
最大化树中该点的信息增益。

最后，我们展示与算法对之间的对数损失差异最相关的元特征。我们考虑了每个系列中表现最好的两种算法：CatBoost、XGBoost、ResNet 和 SAINT。参见[0377]表 22和表 23分别用于统计和一般元特征。

D.5 回归数据集上的实验

虽然我们的实验侧重于分类数据集，但 TabZilla 代码库还能够处理回归数据集，这些数据集具有连续目标变量，而不是分类或二进制变量。我们使用 12 种算法对 17 个表格回归数据集进行实验，使用的实验设计和参数与第 2 部分。通过最大化 R 平方 (R2) 指标，每个算法都针对每个数据集进行调整。这些实验中使用的回归数据集已在最近使用表格数据的机器学习研究中使用 [二十七，三十六，53，59]；每个数据集

对应于 OpenML 任务，并且可以像分类数据集一样进行预处理
用过 任何其使数据库认证（Ope 这些实验中使用的 ets 实体是“银行票据
“UCI” nML 任务 361002） “例如anSkulls”（5040 ），in W （190420），“W 伊斯康辛州-乳房-癌症 - 细胞gy-features”（3610 03）“体脂”(5514)、“ca 利福nia”（361089），“chsc ase-foot”（5012），“克利夫特”（2285），“学院”（3599 42）、“cpu-small”（4 883），“数据集-销售”(190 418)，“kin8nm”(2280)，“莉芙er 障碍”（52948 ），eta“米(4729)、“mv”（4 774）、“pbc”（4850）和“ 老兵”(4828)。

桌子 19什在测试集上的最新 十二选1个一般结论的7 注册回归数据集，acc 根据 R2 第 i 个度量 计算排名。ost算法的表g 算法是 电子模拟我们的研究结果分类数据集秒：球 我们 ll 和糟糕的*阿特勒 负责恩*数据集；然而， GBDT 性能帕特里维利恩，尤其是 Cat 嘘声吨。

表 18：树模型在表格数据集上预测最佳神经网络是否会胜过最佳 GBDT 模型的测试准确率。结果汇总了 176 个训练/测试分割，每个分割中都有一个数据集系列用于测试。每种模型类型的任何模型使用的数据集属性数量列在最左侧一列。顶行包括决策树 (DT $\leqslant n$) 最大深度 n；下面一行是用于比较的 XGBoost 模型。

模型	测试准确度 （意思是 ±标准差	编号 元特征
DT=1	0.54 ±0.28	3
DT ⩽3	0.60 ±0.29	21
DT ⩽5	0.59 ±0.28	183
DT ⩽7	0.64 ±0.29	382
DT ⩽9	0.66 ±0.27	423
DT ⩽11	0.66 ±0.30	529
DT ⩽十三	0.65 ±0.29	561
DT ⩽15	0.68 ±0.30	594
DT ⩽ ∞	0.67 ±0.30	685
XGBoost	0.74 ±0.33	675

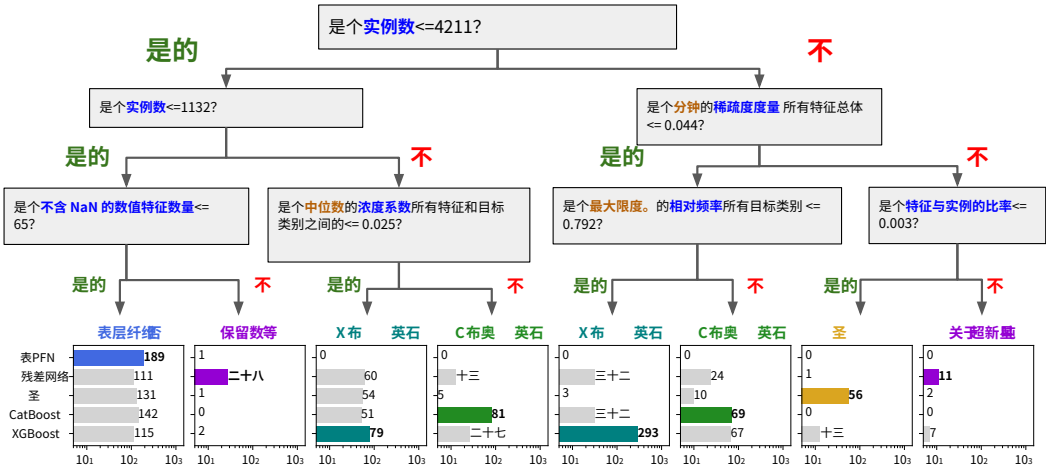

图 15：基于我们在 176 个数据集上的实验，选择最佳算法的决策树。决策树从我们的实验中选择了五种表现最佳的算法：ResNet、SAINT、TabPFN、CatBoost 和 XGBoost。请注意，决策分割完全基于在 176 个数据集中最大化树中该点的信息增益。IQR 表示四分位距，CanCor [四十六]表示典型相关性。

D.6 分位数缩放的附加结果

在本节中，我们讨论数据集预处理。不同的论文使用各种不同的预处理方法，算法本身内部的“内置”预处理技术数量也各不相同。因此，我们的主要结果通过一致的轻量级预处理（输入 NaN 值）来最大限度地减少混杂因素。在本节中，我们将在 tabzilla 基准测试套件上比较 13 种算法，这些算法在所有连续特征上都使用了和不使用分位数缩放（最流行的技术之一）。我们使用 scikit-learn 中的 QuantileTransformer [52]。我们使用与主要实验相同的计算设置和实验设计。参见[0377]表 20。我们发现分位数缩放可以改进简单算法：决策树、MLP、随机森林和 SVM，但对高性能算法影响不大。

D.7 附加超参数优化实验

在我们的马 在为了炒作而进行的实验 机器人参数优化 （Hb），我们进行了 30 次迭代 on随机
搜索 艾拉算法我ms。在这个系列中 电脑 我 我们影响 oF广告意大利 HPO 为 f 或猪算法 頁面:

表 19：12 种算法在 17 个表格回归数据集上的表现。列显示所有数据集的排名、平均归一化 R2（平均 R2）以及各折叠归一化 R2 的标准差（标准 R2）。这些数量的最小值/最大值/平均值/中位数取自所有数据集。

算法	秩				平均 R2		标准 R2	
	最小值	最大值	平均值	平均值	平均值	平均值		
CatBoost	1.0	7.0	3.0	3.0	0.96	0.98	0.11	0.02
轻量级GBM	1.0	11.0	4.35	3.0	0.93	0.97	0.15	0.03
随机森林	2.0	10.0	4.94	4.0	0.89	0.94	0.17	0.03
XGBoost	1.0	10.0	5.29	6.0	0.89	0.99	0.18	0.03
星火	1.0	11.0	6.24	7.0	0.82	0.89	0.10	0.02
线性模型	1.0	12.0	6.71	7.0	0.75	0.89	0.21	0.04
多层感知处理器	1.0	12.0	6.82	7.0	0.78	0.95	0.23	0.04
节点	1.0	12.0	7.56	8.5	0.47	0.5	0.16	0.16
塔格网	3.0	12.0	7.65	7.0	0.68	0.9	0.48	0.04
决策树	2.0	12.0	7.94	8.0	0.7	0.84	0.30	0.07
KNN	2.0	12.0	8.06	8.0	0.64	0.88	0.11	0.03
维梅	3.0	12.0	9.18	10.0	0.62	0.77	0.15	0.07

XGBoost、CatBoost、LightGBM 和 RandomForest。由于需要大量计算资源，我们没有对任何神经网络方法进行额外的 HPO 实验。

对于每个算法，我们使用默认的 Optuna 运行 100 次 HPO 迭代 [3] 算法（树结构 Parzen 估计器），优化对数损失。我们在 TabZilla 基准套件中的所有 36 个数据集上运行这些 HPO 实验。所有超参数范围都可以在我们的存储库中的文件夹中查看https://github.com/naszilla/tabzilla/tree/main/TabZilla/models。

[0377]表 21显示了这些 HPO 实验（算法后缀“(HPO)”）的性能，与默认超参数（后缀“(default)”）的性能相比，以及经过 30 次随机超参数搜索迭代后的性能（如我们的主要结果所示）（无后缀）。正如预期的那样，额外的超参数调整提高了 XGBoost、CatBoost、LightGBM 和 RandomForest 的性能。

D.8 用于识别重要数据集属性的前向特征选择

在本节中，我们提出了一种不同的方法来确定哪些数据集属性与算法之间的性能差异有关。这里我们使用贪婪的前向特征选择[二十五] 来识别重要的数据集属性。在这些实验中，我们研究了使用元特征预测 CatBoost 和 ResNet（两种非常有效的 GBDT 和 NN 算法）之间归一化对数损失差异的问题。

从高层次来看，贪婪前向特征选择会按顺序选择元特征，从而提高元模型的性能。为了评估性能，我们使用留一数据集交叉验证：每个数据集为整个元数据集贡献 10 倍，因此每倍包含 10 个用于验证的实例和所有剩余用于训练的实例。

在这些实验中，我们首先使用 XGB 回归器拟合整个元数据集来选择 200 个特征以供考虑。然后我们使用贪婪前向选择，在 Python 包 mlxtend 中实现（https://rasbt.github.io/mlxtend），使用 XGB 回归器作为元模型。前五个选定的特征如下（按顺序）：

1. 根据 Shapiro-Wilk 检验，特征数量呈正态分布。
2. 所有特征的最小值的中值。
3. 所有特征的稀疏度的中值。[58]
4. 所有特征平均值的四分位距。
5. 所有特征的调和平均值。

表 20：13 种算法在硬数据集基准测试套件中所有 36 个数据集上的性能，包括对每个连续特征应用分位数缩放和不应用分位数缩放的情况。带有后缀“-QSCALE”的算法使用分位数缩放，而不带此后缀的算法使用原始连续特征。算法根据归一化对数损失进行排名，列显示排名、归一化对数损失和训练时间，类似于表格1。

算法	秩			平均 LL		平均值		标准 LL		次/1000实例	
	最小值	最大值	平均值	平均值	平均值					意思是	医学
XGBoost	1	22	4.78	3	0.07	0.04	0.10	0.06	2.02	0.28
量子计算	1	24	4.83	3	0.07	0.04	0.10	0.06	1.02	0.26
CatBoost	1	19	5.60	4	0.09	0.06	0.11	0.06	26.46	1.15
CatBoost-QSCALE	1	二十七	6.57	4	0.12	0.08	0.10	0.06	2.40	1.19
轻量级GBM	1	24	9.53	8	0.13	0.09	0.19	0.08	1.23	0.36
残差网络	1	23	9.89	9	0.19	0.17	0.12	0.08	8.27	5.22
ResNet-QSCALE	1	二十五	10.43	9	0.22	0.17	0.13	0.09	8.63	5.74
圣	1	二十八	10.52	8	0.20	0.14	0.12	0.08	130.18	92.42
数据网络	2	23	11.63	12	0.18	0.15	0.14	0.11	58.70	52.74
圣-QSCALE	1	24	11.74	10	0.23	0.15	0.09	0.05	90.53	42.05
FTTransformer	4	24	12.48	11	0.25	0.20	0.13	0.11	17.41	12.64
随机森林-QSCALE	3	二十六	12.77	11	0.27	0.23	0.16	0.08	0.20	0.15
DANet-QSCALE	2	二十七	13.33	12	0.20	0.17	0.18	0.14	59.44	57.77
FTTransformer-QSCALE	5	24	13.81	12	0.27	0.22	0.14	0.09	19.00	15.93
随机森林	4	二十七	14.46	14	0.28	0.27	0.22	0.09	0.35	0.24
MLP-RTDL	2	三十	14.56	12	0.36	0.25	0.16	0.13	6.33	4.21
支持向量机	1	二十八	15.03	14	0.29	0.21	0.14	0.08	19.73	2.81
SVM-QSCALE	6	22	15.25	17	0.21	0.17	0.13	0.12	8.85	1.91
星火	1	二十八	15.61	17	0.32	0.23	0.10	0.06	15.99	15.29
MLP-RTDL-QSCALE	1	二十九	15.92	14.5	0.39	0.31	0.18	0.14	7.35	4.85
塔格网	1	三十	16.88	16	0.37	0.27	0.26	0.12	27.02	27.10
节点	9	二十七	17.50	17	0.37	0.34	0.08	0.07	153.72	124.27
节点—QSCALE	6	二十八	17.76	18	0.36	0.33	0.08	0.07	171.73	147.36
多层感知处理器	3	二十九	17.81	18.5	0.43	0.40	0.15	0.14	8.86	4.36
线性模型	4	二十九	17.97	18.5	0.47	0.37	0.11	0.07	0.04	0.02
线性模型	5	三十	18.06	17	0.47	0.38	0.12	0.08	0.01	0.01
维梅	6	二十八	21.44	24	0.49	0.48	0.09	0.08	20.79	15.18
决策树—QSCALE	8	三十	21.54	23	0.60	0.59	0.36	0.20	0.05	0.01
决策树	9	31	21.86	23	0.62	0.62	0.38	0.20	0.11	0.01
KNN	6	31	23.61	二十五	0.68	0.70	0.35	0.21	0.03	0.00
KNN-QSCALE	5	三十	23.94	25.5	0.70	0.75	0.34	0.23	0.01	0.00

表 21：所有算法在硬数据集基准测试套件中所有 36 个数据集上的性能，包括 HPO 100 次迭代后的性能（后缀为“(HPO)”的算法）和默认超参数（后缀为“(default)”的算法）。没有任何后缀的算法名称表示 30 次随机超参数搜索迭代后的性能，就像我们的主要结果一样。算法根据归一化对数损失进行排名，列显示排名、归一化对数损失和训练时间，类似于表格1。

	秩			平均 LL			标准 LL		次/1000实例	
算法	最小值	最大值	平均值	平均值	平均值	平均值			意思是	医学
XGBoost（HPO）	1	三十	6.28	4	0.03	0.02	0.04	0.03	6.36	1.54
XGBoost	1	二十八	6.83	5	0.03	0.02	0.04	0.03	2.02	0.28
CatBoost（HPO）	1	二十八	6.88	6	0.05	0.02	0.04	0.03	16.75	1.82
CatBoost	1	20	7.23	6	0.05	0.02	0.04	0.02	26.46	1.15
LightGBM（HPO）	1	22	7.38	4.5	0.04	0.02	0.04	0.03	0.64	0.20
XGBoost（默认）	1	二十七	9.06	8	0.07	0.03	0.03	0.03	1.76	0.41
CatBoost（默认）	1	33	11.17	10	0.10	0.04	0.03	0.02	29.53	0.97
轻量级GBM	1	三十	11.94	11	0.06	0.03	0.07	0.04	1.23	0.36
残差网络	1	二十九	12.20	12	0.10	0.04	0.04	0.04	8.27	5.22
圣	1	三十四	13.33	11	0.08	0.05	0.04	0.03	130.18	92.42
LightGBM（默认）	2	三十四	13.81	12	0.07	0.05	0.05	0.04	1.46	0.61
数据网络	2	31	14.81	14	0.06	0.05	0.05	0.04	58.70	52.74
FTTransformer	2	二十九	15.69	15	0.10	0.06	0.04	0.03	17.41	12.64
SAINT （默认）	2	三十七	16.18	十三	0.08	0.05	0.04	0.04	111.07	83.68
随机森林（HPO）	4	三十五	16.29	十三	0.14	0.08	0.08	0.03	0.33	0.20
ResNet（默认）	2	三十五	17.00	16	0.13	0.06	0.06	0.04	7.28	4.72
MLP-RTDL	2	三十七	17.92	16	0.18	0.09	0.06	0.04	6.33	4.21
随机森林	5	33	18.03	16	0.15	0.10	0.08	0.03	0.35	0.24
星火	1	三十五	18.48	20	0.14	0.06	0.04	0.03	15.99	15.29
支持向量机	1	三十四	18.62	18	0.11	0.07	0.04	0.03	19.73	2.81
塔格网	1	三十二	20.30	18	0.14	0.12	0.09	0.06	27.02	27.10
随机森林（默认）	4	三十四	20.57	21	0.23	0.10	0.03	0.02	0.32	0.27
多层感知处理器	3	三十六	21.06	21.5	0.21	0.12	0.05	0.04	8.86	4.36
FTTransformer（默认）节点	5	三十六	21.17	23	0.19	0.07	0.05	0.04	15.71	11.42
MLP-rtdl（默认）	11	三十二	21.27	20	0.19	0.12	0.03	0.02	153.72	124.27
DANet（默认）	1	三十九	21.69	21.5	0.29	0.13	0.11	0.06	5.82	3.89
线性模型	6	三十七	21.74	23	0.12	0.09	0.08	0.05	40.59	38.95
NODE（默认）	8	三十七	22.09	22.5	0.26	0.13	0.04	0.03	0.04	0.02
SVM（默认）	11	三十四	22.70	23	0.21	0.14	0.03	0.02	52.26	42.19
TabNet（默认）	2	三十五	23.57	二十六	0.15	0.08	0.04	0.02	4.19	0.80
MLP（默认）	2	三十九	24.94	二十七	0.22	0.14	0.11	0.07	24.04	23.40
维梅	4	三十四	25.28	二十六	0.34	0.17	0.10	0.07	8.13	4.44
STG（默认）	10	三十八	25.91	二十九	0.23	0.18	0.04	0.03	20.79	15.18
决策树	14	三十七	26.81	二十七	0.25	0.14	0.03	0.02	13.72	13.20
KNN	6	三十七	27.29	二十八	0.29	0.21	0.13	0.08	0.11	0.01
决策树（默认）	15	三十九	29.27	31	0.28	0.26	0.12	0.09	0.03	0.00
VIME（默认）	18	三十九	31.91	三十三	0.60	0.67	0.30	0.15	0.12	0.02
KNN（默认）	6	三十九	32.00	33	0.38	0.31	0.09	0.02	20.10	12.79
			33.48	三十六	0.59	0.55	0.22	0.17	0.03	0.00

表 22：与算法对之间的归一化对数损失差异（算法 1 的损失减去算法 2 的损失）最相关的数据集元特征。对 CatBoost、XGBoost、ResNet 和 SAINT 成功运行的所有 133 个数据集的所有 10 个分割进行相关性计算。列出了每对算法具有最大绝对相关性的 10 个数据集属性。属性名称与 PyMFE 使用的命名约定相对应。

算法 1	算法 2	纠正。	属性名称
CatBoost	残差网络	- 0.25	所有特征的最大偏度。
CatBoost	残差网络	- 0.24	所有特征的偏度范围。
CatBoost	残差网络	- 0.23	所有特征峰度标准差的对数。
CatBoost	残差网络	- 0.23	所有特征偏度的标准差的对数。
CatBoost	残差网络	0.22	所有特征对之间协方差绝对值的中值的对数。
CatBoost	残差网络	0.21	所有特征的标准差中值的对数。所有特征的方差中值的对
CatBoost	残差网络	0.21	数。
CatBoost	残差网络	0.20	所有特征的最大值的中值的对数。
CatBoost	残差网络	0.20	经过 10 倍 CV 训练的朴素贝叶斯分类器的最佳性能。
CatBoost	圣	0.26	使用信息量最少的特征对单节点决策树进行 10 倍 CV 拟合，在所有 10 倍中表现最佳。
CatBoost	圣	0.25	使用最少信息特征的单节点决策树拟合的 10 倍 CV 的平均性能。
CatBoost	圣	0.24	对于使用最少信息量特征的单节点决策树拟合的 10 倍 CV，10 倍以上的中位性能。
CatBoost	圣	- 0.24	单节点决策树 10 倍 CV 的最差性能的对数 使用最具信息量的特征进行拟合。
CatBoost	圣	- 0.23	使用单节点决策树拟合的性能峰度的对数 最具信息量的特征，超过 10 倍 CV。目标的
CatBoost	圣	0.23	香农熵的对数。
CatBoost	圣	-	精英最近邻在 10 倍 CV 中的最佳性能的对数为 0.23。
XGBoost	残差网络	- 0.29	所有特征的最大偏度。
XGBoost	残差网络	- 0.28	所有特征的偏度范围。
XGBoost	残差网络	-	所有特征峰度标准差的对数为 0.28。
XGBoost	残差网络	- 0.27	所有特征偏度的标准差的对数
XGBoost	残差网络	0.25	所有特征对之间的绝对协方差中值的对数。
XGBoost	残差网络	0.24	所有特征的中位标准差的对数。所有特征的中位方
XGBoost	残差网络	0.23	差的对数。
XGBoost	残差网络	0.23	所有特征的中值最大值的对数。
XGBoost	残差网络	0.22	朴素贝叶斯分类器在 10 倍 CV 中的最佳性能。
XGBoost	残差网络	0.22	朴素贝叶斯分类器在 10 倍 CV 中的最佳性能的对数。
XGBoost	圣	- 0.23	特征的噪声程度：（$\sum_{我}\sum_{年代_{我}} - {}_{我}MI(我,)) / \sum_{我} MI(我,)$，在哪里$年代_{我}$ 是特征的熵$我$， 和$MI(我,)$是特征之间的相互信息$我$和目标$是$。
XGBoost	圣	- 0.22	所有特征峰度标准差的对数。
XGBoost	圣	- 0.20	所有特征偏度的标准差的对数。
XGBoost	圣	- 0.20	使用以下方法对单节点决策树进行 10 倍 CV 拟合，性能最佳 最具信息量的功能。
XGBoost	圣	- 0.20	所有特征的最大偏度。0.19 目标的香农熵
XGBoost	圣		的对数。
XGBoost	圣	- 0.19	所有对之间绝对相关性的标准差的对数 特征。
XGBoost	圣	- 0.18	所有特征的偏度范围。0.18
XGBoost	圣		对随机属性进行训练的决策树的性能范围的对数，超过 10 倍 CV。

表 23：与算法对之间的归一化对数损失差异（算法 1 的损失减去算法 2 的损失）最相关的数据集元特征。对 CatBoost、XGBoost、ResNet 和 SAINT 成功运行的所有 133 个数据集的所有 10 个分割进行相关性计算。列出了每对算法具有最大绝对相关性的 10 个数据集属性。属性名称与 PyMFE 使用的命名约定相对应。

算法 1	算法 2	纠正。 属性名称
CatBoost	残差网络	- 数据集大小的 0.14 对数。
CatBoost	残差网络	0.14 特征数量与数据集大小之比的对数。
CatBoost	残差网络	- 0.14 数据集大小与特征数量之比的对数。
CatBoost	残差网络	0.06 数值特征的数量。 特征的数
CatBoost	残差网络	0.06 量。
CatBoost	残差网络	- 0.06 每个目标类别的相对频率范围。
CatBoost	残差网络	- 任何目标类别的最大频率的 0.06 对数。
CatBoost	残差网络	- 0.05 所有目标类别的相对频率的四分位距。
CatBoost	残差网络	- 0.04 任何目标类别的最大相对频率。
CatBoost	圣	- 0.19 所有目标类别的相对频率的标准差。 0.18 所有目标类别的相对
CatBoost	圣	频率的峰度。
CatBoost	圣	- 0.18 所有目标类别的最大相对频率。
CatBoost	圣	- 0.18 所有目标类别的平均相对频率。
CatBoost	圣	- 0.17 所有目标类别的中位相对频率的对数。
CatBoost	圣	0.16 目标类别的数量。
CatBoost	圣	0.15 目标类别相对频率的偏度。
XGBoost	残差网络	- 数据集大小的 0.20 对数。
XGBoost	残差网络	- 数据集大小与特征数量之比的对数为 0.20。
XGBoost	残差网络	0.20 特征数量与数据集大小之比的对数。所有目标类别的最小
XGBoost	残差网络	0.11 相对频率的对数。数值特征的数量。
XGBoost	残差网络	0.08
XGBoost	残差网络	0.08 所有目标类别的中位相对频率。 所有目标类别
XGBoost	残差网络	0.07 的最小相对频率。
XGBoost	圣	- 0.17 所有目标类别的相对频率的标准差。
XGBoost	圣	- 数据集大小的 0.16 对数。
XGBoost	圣	- 0.16 所有目标类别的相对频率的四分位距。0.15 数据集大小与特征
XGBoost	圣	数量比率的对数。
XGBoost	圣	- 特征数量与数据集大小之比的对数为 0.15。
XGBoost	圣	- 0.15 所有目标类别的最大相对频率。
XGBoost	圣	- 0.13 所有目标类别的相对频率范围。
XGBoost	圣	- 0.13 所有目标类别的相对频率的平均值。
XGBoost	圣	- 0.12 所有目标类别的相对频率的中位数。