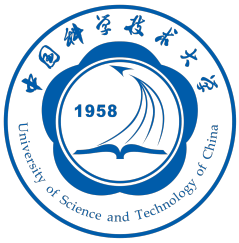


中国科学技术大学
University of Science and Technology of China

2023 年春季学期深度学习导论课程
期末 KAGGLE 竞赛报告

Kaggle 比赛

ICR - Identifying Age-Related Conditions

Ranking: 2446/6430，Top 39%

IKUN 组

组员：**王祥祺** / PB20000152 / †

组员：**徐海阳** / PB20000326 / †

组员：陈心羽 / 非本班人员 / 参与讨论

组员：周俊炜 / 非本班人员 / 参与讨论

†: **主要贡献 – IDEA、代码、讨论、报告**

2023 年 8 月 3 日

目录

1 比赛介绍

1.1 竞赛背景

本次竞赛 ICR - Identifying Age-Related Conditions [1] 由 InVitro Cell Research，LLC（ICR）主办，其专注于再生和预防性个性化医疗。竞赛奖金池为 $60,000。竞赛的主要目标是解决一个关键技术问题：尽管已经使用了诸如 XGBoost 和随机森林等模型来预测医疗状况，但模型的性能仍然不尽人意。在处理涉及生命的紧急问题时，模型需要在不同案例之间稳定而可靠地进行准确预测。

这个问题的挑战在于，由于生物信息学中的数据较少，我们需要从有限的数据中挖掘出有意义的信息，以及在仅有少量训练样本的情况下获得有价值的预测结果。因此，本次竞赛中的任务涉及到少量数据下的特征挖掘，属于类似于 few-shot learning 的问题。

1.2 竞赛目标

本次比赛的主要目标是基于健康特征测量值进行预测，判断一个人是否患有三种特定疾病之一。这是一个二分类问题，即预测结果为某人是否患有三种疾病之一，或者完全没有患上这三种疾病。参赛者的任务是创建一个模型，通过收集与患病条件相关的关键特征，并进行编码，从而在保护患者详细信息的情况下，预测其健康状况。

这样的预测模型可以帮助研究人员发现特定特征的测量值与潜在患者状况之间的关联。

1.3 训练数据与评分标准

本次竞赛提供了包括 train.csv、test.csv、greeks.csv 和 sample_submissions.csv 在内的四个数据集 [2]。数据集包含与三种特定年龄相关的疾病有关的五十多个匿名健康特征。需要注意的是，实际最终排名时，示例测试数据将会被替换为完整的测试集。完整测试集大约包含 400 行数据。

在 train.csv 中，Id 列代表每个病人的标识，而 AB 到 GL 共有 56 个匿名列，代表病人的匿名特征。这些特征中，除了 EJ 列为范畴特征外，其余都是数字特征。最后一列 Class 表示是否患病，即预测目标。此外，train.csv 数据集包含约 335KB 的数据。

此外，还有一个名为 greeks.csv 的数据集，其中的列具有以下含义：Alpha 列表示与年龄相关的疾病类型；而 A、B、D 和 G 列表示与年龄有关的三种疾病，分别对应 Class 1。而 Beta、Gamma 和 Delta 列标识三种实验特性；Epsilon 列标识收集数据的日期。

最终的提交结果将通过平衡的对数损失（balanced logarithmic loss）进行评估。这个评估方法确保每个类别在总体分数中大致具有相同的重要性。

最终的对数损失公式如下，其中，(N_c) 代表类别 c 的数量，(log) 代表自然对数，(y_{ci}) 为 1 表示样本 i 属于类别 c，(p_{ci}) 为样本 i 属于类别 c 的预测概率。

$$Logloss = \frac{-\frac{1}{N_0}\Sigma_{i=1}^{N_0} y_{0i}\log(p_{0i}) - \frac{1}{N_1}\Sigma_{i=1}^{N_1} y_{1i}\log(p_{1i})}{2}$$

同时，提交的每一行的概率并不需要求和为 1，因为在评分之前会重新缩放（每行除以行总和）。为避免对数函数在极端情况下的不稳定性，每个预测概率 p 都会被替换为 $max(min(p, 1-10^{-15}), 10^{-15})$。

train.csv (336.56 kB)

Detail Compact Column 10 of 58 columns

Id	AB	AF	AH	AM	AR
617 unique values	0.08 – 6.16	193 – 28.7k	85.2 – 1.91k	3.18 – 631	8.14
000ff2bfdfe9	0.209377	3109.03329	85.200147	22.394407	8.13868
007255e47698	0.145282	978.76416	85.200147	36.968889	8.13868
013f2bd269f5	0.47003	2635.10654	85.200147	32.360553	8.13868
043ac50845d5	0.252107	3819.65177	120.201618	77.112203	8.13868
044fb8a146ec	0.380297	3733.04844	85.200147	14.103738	8.13868
04517a3c90bd	0.209377	2615.8143	85.200147	8.541526	8.13868
049232ca8356	0.3482495	1733.65412	85.200147	8.377385	15.3124
057287f2da6d	0.269199	966.45483	85.200147	21.174189	8.13868
0594b00fb30a	0.346113	3238.43674	85.200147	28.888816	8.13868
05f2bc0155cd	0.324748	5188.68207	85.200147	12.9686875	8.13868
06055f3f6785	0.504214	6089.31532	85.200147	189.781063	17.0272
06554e7b9979	0.760594	6957.75289	200.089275	23.462872	8.13868
068a4e4bbbab	0.491395	2627.29962	138.377301	48.969764	8.13868
06c0ddf265c5	0.534125	4784.80222	90.704235	8.739734	8.13868
075bd937ab85	0.508487	1632.12907	85.200147	11.864607	17.6172
07760b4cf3f8	0.666588	3657.7816	304.283751	75.418144	8.13868

图 1: train.csv 数据集概览

2 数据处理与模型设计

在本部分，我们将深入探讨数据处理和模型设计的核心原理，以及为什么选择这些方法来应对挑战性的比赛任务。

数据方面，我们首先对数据进行了归一化、KFold 划分等操作，并且针对样本分类不均匀不平衡的现象使用了 ‘balanced_log_loss‘ 和 ‘RandomOverSampler‘ 来使得采样和优化过程更加鲁棒；模型方面，我们使用了集成模型（Ensemble）来进行特征提取和分类，特别地，针对本次比赛的数据集，我们使用了 XGBoost 的决策树算法和针对 Table 类数据的 TablePFN [3]。

2.1 数据处理及采样

数据在机器学习中扮演着至关重要的角色，然而现实中的数据往往不完美。因此，我们首先对数据进行了归一化操作，将不同特征的值缩放到相似的尺度上。这是为了确保某些特征不会因为其数值范围的不同而在模型训练中产生不合理的影响。通过归一化，我们可以使得优化算法更快地收敛，模型的性能也更加稳定。

为了评估模型的性能和泛化能力，我们采用了 KFold 交叉验证方法。这种方法将数据划分为多个训练集和验证集的组合，每个组合都会作为训练和验证的数据集。这样做的好处是，我们可以获得对模型性能的更准确估计，同时避免了过拟合问题。通过交叉验证，我们可以更加自信地评估模型在真实数据上的表现。

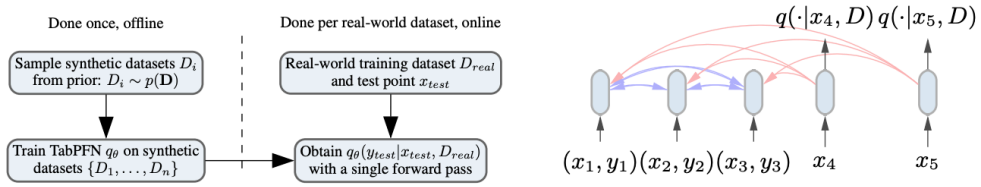
然而，在处理数据时，我们还面临着另一个问题，即样本分类不均匀和不平衡。在这种情况下，常规的模型评估指标可能无法准确地反映模型的性能。因此，我们引入了 ‘balanced_log_loss‘函数来计算损失，该函数考虑了不同类别之间的权重差异，从而更好地衡量模型的质量。此外，我们还使用了 ‘RandomOverSampler‘来进行样本过采样，增加少数类别的样本数量，以平衡类别分布，从而提升模型的分类能力。

2.2 模型设计及思考

在模型设计方面，我们采用了集成学习的方法，这是一种将多个模型组合在一起以提升性能的策略。集成学习的核心思想在于，通过结合多个模型的预测结果，可以获得更准确、稳定的整体预测。这对于解决复杂的分类问题尤为有益。

我们设计了一个名为 ‘Ensemble‘的类来构建集成模型。在这个类中，我们将多个分类器引入模型中，如 XGBoost 和 TablePFN。XGBoost 是一种基于决策树的强大算法，适用于各种类型的数据，具有良好的泛化性能。而 TablePFN 是专门为表格数据设计的神经网络模型，它可以从表格中提取关键特征，为表格数据预测任务提供了一种有效的解决方案。它的具体架构如图2所示。

综合而言，我们的数据处理和模型设计策略是经过深思熟虑的。通过对数据进行归一



(a) Prior-fitting and inference

(b) Architecture and attention mechanism

Figure 1: Left (a): The PFN learns to approximate the PPD of a given prior in the offline stage to yield predictions on a new dataset in a single forward pass in the online stage. Right (b): Training samples $\{(x_1, y_1), \ldots, (x_3, y_3)\}$ are transformed to 3 tokens, which attend to each other; test samples x_4 and x_5 attend only to the training samples. Plots based on Müller et al. (2022).

图 2: TabPFN Architecture

化、交叉验证和采样，我们确保了数据的质量和模型的泛化能力。而集成模型的引入则使得我们可以从多个角度对问题进行建模，提升了模型的整体性能。这些方法的综合应用，使得我们在面对这个具有挑战性的比赛任务时，能够充分发挥机器学习和数据分析的优势，获得更好的结果。

3 代码具体实现

我们参考了比赛过程中 Kaggler 分享的 pipeline，重构了他的代码并且增加了 GPU 训练、推理的支持，速度提升 15x。在这一部分，我们将详细阐述数据处理、模型设计和实际代码实现的步骤。

3.1 数据加载与预处理

我们首先从本地文件加载了训练数据、测试数据、样本提交文件以及一个名为 “greeks” 的数据集。这些数据在后续的模型训练和预测中将会被用到。接下来，我们对数据进行了以下处理：

```
# Transform category label from 'A''B' to 0, 1
first_category = train.EJ.unique()[0]
train.EJ = train.EJ.eq(first_category).astype('int')
test.EJ = test.EJ.eq(first_category).astype('int')

# Prepare K-Fold Cross Validation Data
cv_5 = KF(n_splits=5, shuffle=True, random_state=42)

# Define balanced_log_loss function
def balanced_log_loss(y_true, y_pred):
    # ... (function content) ...
```

在这一步骤中，我们将类别标签由字符’A’ 和’B’ 转换为数值 0 和 1，以便于后续的模型训练和评估。然后，我们使用 KFold 方法划分数据为 5 个训练集和验证集的组合，以进行交叉验证。此外，我们定义了名为 ‘balanced_log_loss‘的函数，用于计算平衡的对数损失，以解决数据不平衡带来的问题。

3.2 集成模型：Ensemble 类

在我们的实现中，集成模型是实现高性能的关键。我们构建了一个名为 ‘Ensemble‘的类，这个类在内部包含了多个分类器，包括了 XGBoost 和 TablePFN 等，通过组合它们的预测结果，达到提升模型性能的目的。

```
class Ensemble():
    def __init__(self, device="cpu"):
        # 初始化函数
        # ... (classifier initialization) ...
```

```
    def fit(self, X, y):
        # 训练函数
        # ... (data preprocessing) ...
        for classifier in self.classifiers:
            if classifier == self.classifiers[2] or
            classifier == self.classifiers[3]:
                classifier.fit(X, y, overwrite_warning=True)
            else:
                classifier.fit(X, y)

    def predict_proba(self, x):
        # 预测函数
        x = self.imputer.transform(x)
        # ... (probability calculation and weighting) ...
```

在 'Ensemble'类的初始化函数中，我们通过设备参数来初始化各个分类器，如 XGBoost 和 TablePFN。这些分类器在集成模型中发挥了关键作用，各自具有不同的特点和优势。

在 'fit'方法中，我们首先对数据进行预处理，填充缺失值，并使用预处理后的数据对每个分类器进行训练。需要注意的是，对于 TablePFN 分类器，我们在特定情况下添加了'overwrite_warning=True'的参数，以避免重写警告。

而在 'predict_proba'方法中，我们对输入的数据进行预测，并计算了每个分类器的概率预测结果。然后，我们将这些概率进行加权处理，得到了最终的集成预测结果。这种加权处理考虑了类别的不平衡性，使得模型对不同类别的样本能够做出更准确的预测。我们通过整合多个分类器的预测结果，充分发挥了集成学习的优势，提升了模型的性能和泛化能力。

3.3 预测与结果

最终，我们使用训练好的集成模型对测试数据进行了预测，得到了概率预测结果。

```
times = greeks.Epsilon.copy()
# ... (data processing) ...

y_pred = m.predict_proba(test_with_time)
probabilities = np.concatenate((
    y_pred[:,:1],
    np.sum(y_pred[:,1:], 1, keepdims=True)
), axis=1)
```

```
p0 = probabilities[:, :1]
```

在这一步骤中，我们首先对时间数据进行了处理，将日期转换为数值表示。然后，我们使用训练好的集成模型 ‘m‘对测试数据进行了概率预测，得到了每个样本属于各个类别的概率。最后，我们将这些概率进行了加权，得到了预测结果。

最终，我们的模型在比赛中的 Private Score 为 0.50025，排名为 2446/6430，位于 top 39%。这表明我们的模型在众多参赛者中取得了不错的成绩，具有一定的竞争力。

4 关于比赛的讨论与思考

4.1 关于比赛结果的讨论

在 Kaggle 社群中，关于 ICR 比赛的讨论持续激烈。由于数据量相对较少，一些参赛者对比赛中的数据问题提出了自己的观点。初始阶段的公共排行榜在训练和评估时出现了明显的过拟合现象，这可能与原始数据的规模有关。此外，训练和测试数据集的规模较小、嘈杂，而且随时间变化，还存在不同的测试分布。因此，一些人认为最终结果可能会是纯粹的随机结果，排名靠前的选手可能是运气成分较大。并且而排名靠前的 winner 用的往往是笨拙且巨量的方法，很难说这些 Top Ranking 的方法最终对于赞助商来说有效的。

此外，公共排行榜和私有排行榜之间存在较大的数据分布变化，导致在公共排行榜上取得较好成绩的模型在私有排行榜上表现平平。尽管如此，也有人认为，实际疾病预测任务中确实存在数据量有限的情况，且病例之间的数据漂移较大，因此预测模式可能会较为困难。这一点也增加了比赛的难度和挑战，也是这场比赛的难点和有趣的点之一，所以不应该过于指责比赛本身。

4.2 关于 SOTA 方法的讨论与思考

引人注目的是，一名胜利者并没有使用任何数据的后处理技巧 [4]。此外，该选手的代码量并不大，但采用了一些令人印象深刻的处理方法，包括：活动正则化、非线性处理方法以及重新加权概率等。该选手的方法核心是基于复杂的深度神经网络，但由于许可要求，可能无法完全公开其代码。

这位选手通过精心设计的网络架构和处理步骤，取得了优异的成绩。他用到的总 code 量并不大，我认为别出心裁的点在于：Activity Regularization，Non Linearty 的处理方法以及 Gating 重新加权概率。

同时，他关注到了两个重要的点，并且做出了相应的应对：

1. 数据分布决定模型显然是过拟合的。因此他在模型中使用了非常高的 Dropout 值：0.75, 0.5 和 0.25。同时，他使用了 10 折交叉验证，每折重复 10-30 次，根据交叉验证的结果为每个折选择 2 个最佳模型。

2. greeks.csv 作用甚微，且 greeks 的测试数据也没有 release 出来。因此他直接没有使用该特征 csv，事实证明对于避免过拟合非常有效。

除此以外，他利用了一种基于注意力的新型架构 [5](Temporal Fusion Transformer)，这将高性能多水平预测与对时间动态的可解释见解相结合，是一种基于 Variable Selection Network 的时序融合 Transformer 架构。

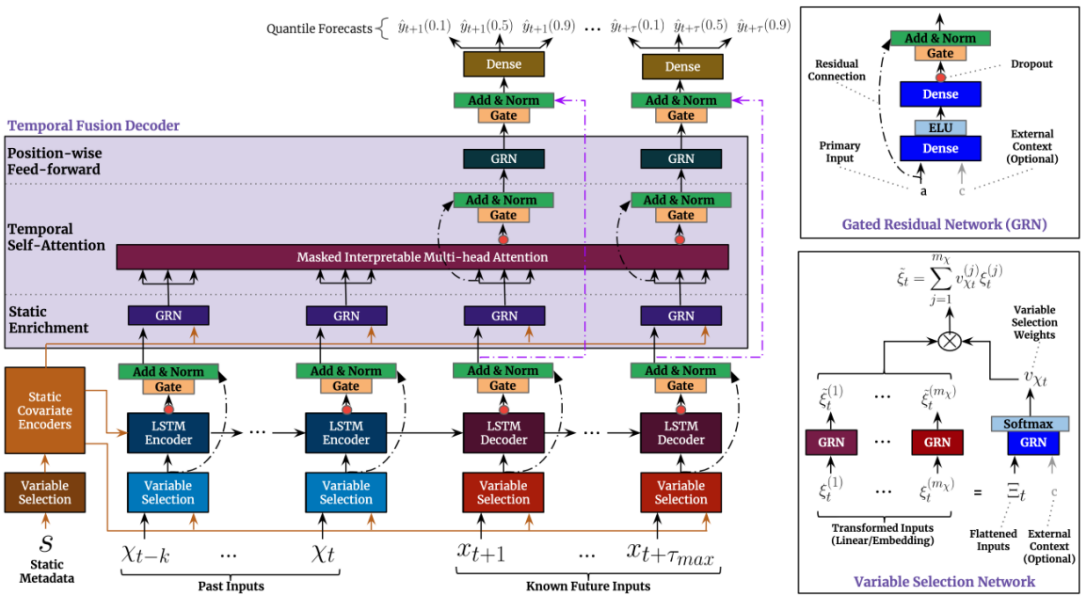


图 3: TFT Architecture

4.3 关于 Few-shot Learning 的讨论与思考

Few-shot learning 是一种令人印象深刻的学习方法，其在数据有限的情况下具有较强的泛化和适应能力。尽管如此，Few-shot learning 仍然面临一些挑战：

1. 样本不平衡：在 Few-shot learning 中，不同类别的样本数量可能存在不平衡问题，这可能导致模型在学习过程中对于少样本类别的学习效果较差，从而影响模型的性能。

2. 迁移学习问题：在 Few-shot learning 中，模型需要将之前学习到的知识迁移到新的类别或任务上。然而，如何进行有效的知识迁移仍然是一个挑战，因为新的类别或任务可能与之前的学习内容存在差异。

3. 鲁棒性问题：尽管 Few-shot learning 具有快速学习和泛化能力，但在真实世界的复杂环境中，模型的鲁棒性仍然是一个关键问题。模型在少量样本上的表现可能会受到干扰、噪声等因素的影响。

总体而言，Few-shot learning 是一种有前途的学习方法，尤其在数据稀缺的情况下具有潜在的应用前景。然而，需要进一步解决样本不平衡、迁移学习和鲁棒性等问题，以实现在真实场景中的稳健

5 总结与致谢

通过参与 InVitro Cell Research 比赛，我们深刻认识到在医学预测领域中数据的有限性和特殊性。数据量的限制以及数据分布的变化增加了模型设计和优化的难度，同时也反映了现实医学问题的挑战性。本次比赛使我们在实际应用中接触到了数据预处理、特征工程和模型设计等环节的实践，拓展了我们的知识领域。

在此，我们要向连德富老师、金开宇助教以及贺颖助教表示诚挚的感谢。感谢连德富老师在课堂上深入浅出地讲解深度学习的核心概念，为我们打开了新的技术视野。感谢金开宇助教和贺颖助教精心的实验设计和耐心指导，让我们在实验中收获了宝贵的经验和技能。

参考文献

[1] C. R. H. Aaron Carman Alexander Heifler Ashley Chow, "Icr - identifying age-related conditions," 2023. [Online]. Available: https://kaggle.com/competitions/icr-identify-age-related-conditions

[2] I. C. Research, "Identifying age-related conditions dataset." [Online]. Available: https://www.kaggle.com/competitions/icr-identify-age-related-conditions/data

[3] N. Hollmann, S. Müller, K. Eggensperger, and F. Hutter, "Tabpfn: A transformer that solves small tabular classification problems in a second," *arXiv preprint arXiv:2207.01848*, 2022.

[4] SAMUEL, "Simple tabpfn approach for score of .15 in 1 min." [Online]. Available: https://www.kaggle.com/code/muelsamu/simple-tabpfn-approach-for-score-of-15-in-1-min/notebook

[5] B. Lim, S. Ö. Arik, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *ArXiv*, vol. abs/1912.09363, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:209414891