清华大学学报（自然科学版）
J Tsinghua Univ (Sci & Technol)

基于心理健康调查数据的抑郁症成因分析

沈佳茗, 周韧平 清华大学 计算机系

2025 年 6 月 22 日

1 引言

在成年人中识别导致抑郁风险的因素对于开发更有效的预防和治疗策略至关重要。据 WHO 官网所述, 3.8% 的人口患有抑郁症, 其中包括 5% 的成年人（男性为 4%，女性为 6%）和 5.7% 的 60 岁以上成年人，而许多中低收入人群是无法获得治疗的。目前已有研究表明，预防措施已被证明可以减少抑郁症，因此及时识别抑郁风险对有效治疗非常重要。

本研究中所分析的数据集来自 kaggle，是 playground competition s4e11 的数据集。它是由深度学习网络在原数据集上训练再进行生成得到的数据。原数据集包含了在 2023 年 1 月至 6 月间进行的匿名调查中收集的信息，涵盖了不同背景和职业的个体。参与者自愿提供了关于年龄、性别、城市、教育程度、工作满意度、学习满意度、工作/学习时间以及家庭病史等信息，这些数据是在非临床环境中收集的，有助于揭示日常生活中可能影响心理健康的因素。

通过分析这些数据，我们可以识别出与抑郁风险相关的模式和相关性，这对于早期识别和干预具有潜在的重要意义。此外，这个数据集还可以用于开发预测模型，特别是机器学习模型，以预测个体的抑郁风险，这对于提高公共健康策略和教育公众关于抑郁及其风险因素的认识具有重要作用。因此，研究这个数据集不仅有助于我们更好地理解抑郁的复杂性，而且对于改善个体的生活质量和公共健康具有深远的影响。

2 方法

2.1 LDA

LDA (Linear Discriminant Analysis)，线性判别分析是一种经典的线性学习方法。它假设不同类别的数据都服从高斯分布，并希望找到一个投影方向使得类间差异大，类内差异小。

假设两个类分别为类 k 和类 l，样本为 x，在比较 $P(k|x)$ 与 $P(l|x)$ 时，我们可以比较其对数比 $\log \frac{P(k|x)}{P(l|x)}$。在每个类都服从高斯分布的假设下，我们可以进一步将它写成:

$$\begin{aligned} f_k(x) &= \frac{\exp\{-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\}}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \\ \log\frac{P(k|x)}{P(l|x)} &= \log\frac{f_k(x)\pi_k}{f_l(x)\pi_l} \\ &= \log\frac{\pi_k}{\pi_l} - \frac{1}{2}\log\frac{|\Sigma_k|}{|\Sigma_l|} \\ &\quad - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) \\ &\quad + \frac{1}{2}(x-\mu_l)^T\Sigma_l^{-1}(x-\mu_l) \end{aligned} \tag{1}$$

如果在高斯假设的基础上，在假设对任意 k，均有 $\Sigma_k = \Sigma$，则可以得到:

$$\begin{aligned} \log\frac{P(k|x)}{P(l|x)} &= \log\frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k+\mu_l)^T\Sigma^{-1}(\mu_k-\mu_l) \\ &\quad + x^T\Sigma^{-1}(\mu_k-\mu_l) \end{aligned} \tag{2}$$

据此，我们可以得到 LDA 的判别方程 $\delta(x) = \log\pi_k + x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k$。

在实际问题中，因为我们无法得知 μ_k, Σ, π_k，因此我们需要用数据进行估算。其中 $\hat{\mu}_k = \sum_{y_i=k} x_i/N_k$，$\hat{\Sigma}_k = \sum_{k=1}^{K}\sum_{y_i=k}(x_i-\hat{\mu}_k)^T(x_i-\hat{\mu}_k)/(N-K)$，$\hat{\pi}_k = N_k/N$。

需要注意的是，当只有两类时，LDA 的系数与最小二乘法得到的系数成正比，在这种情况下，若 $N_1 = N_2$，则二者完全相同，否则因为截距项的不一致，仍会得到不一样的决策结果。在这种情况下，因为 LDA 决策平面的方向与最小二乘得到的是一致的，因此也可以将其推广到非高斯分布的数据上，但由于截距项仍由高斯假设得到，所以在实际处理数据集时，也可以尝试找到

一个经验上使得训练损失最小的截距用在判别方程中，在实践中这样得到的结果也不错。1

对于本数据集来说，变量 x 中会出现非数值型变量，虽然我们在实际操作中使用独热编码将其拆分成数值型变量，但也因此，该数据分布一定不是多维高斯分布。但在我们的实验中，数据的标签只有两类，分别为“患抑郁症”和“未患抑郁症”，因此对于系数向量来说，此处并不需要高斯假设。而虽然截距项实际会受到高斯假设的影响，但从实验结果来看，该影响不大，在实际使用中仍有不错的效果。

2.2 GaussianNB

GaussianNB，也即高斯朴素贝叶斯算法，在朴素贝叶斯的基础上假设连续型变量的分布是高斯分布。

对于贝叶斯最优分类器来说，我们需要比较 $\pi_k f_k(x)$ 或 $\pi_k P(X=x|K=k)$。但是我们很难估计联合分布，因此将这个问题进行简化，增加独立性假设（即假设 x 中每个分量都是相互独立的），据此我们可以根据 $f_k(x_i)$ 或者 $P(X_i=x_i|K=k)$ 来得到 $f_k(x)$ 或 $P(X=x|K=k)$。具体的决策函数为 $\hat{y}(x)=\arg\max_k \pi_k \prod_i f_k(x_i)$ 或 $\hat{y}(x)=\arg\max_k \pi_k \prod_i P(X_i=x_i|K=k)$。在实际计算时，考虑到如果变量数目过多会导致连乘得到的概率太小，因此往往对其取对数在做相加计算，防止 float 下溢。

虽然实际的数据集一般不满足独立性假设，qi 而这个模型也很简单，但朴素贝叶斯一般还是可以得到不错的结果。这是由于虽然对单个类的密度估计可能出现偏差，但在使用时这个偏差不会对后验概率造成很大影响，因而往往效果还不错。

对于高斯朴素贝叶斯算法而言，它认为所有的变量都是连续型的，并且满足高斯分布，即 $f_k(x_i)=\frac{1}{\sqrt{2\pi\sigma_{ik}^2}}\exp\frac{(x_i-\mu_{ik})^2}{(2\sigma_{ik}^2)}$。在实际计算时，由于高斯分布由参数 μ 和 σ^2 决定，因此对每个类别 k，我们只需要估计每个分量对应的 μ_{ik} 和 σ_{ik}^2 即可，算法步骤如下：

1. 对每个类别 k
   - 估算先验概率 $\hat{\pi}_k=N_k/N$
   - 估算每个分量对应的均值 $\hat{\mu}_{ik}=\sum_{y_j=k}x_{ji}/N_k$ 以及方差 $\hat{\sigma}_{ik}^2=\sum_{y_j=k}(x_{ji}-\mu_{ik})^2/N_k$
2. 对一个新来的变量 x，计算 $\arg\max_k \hat{\pi}_k f_k(x_i)$，其中 $f_k(x_i)=\frac{1}{\sqrt{2\pi\hat{\sigma}_{ik}^2}}\exp\frac{(x_i-\hat{\mu}_{ik})^2}{(2\hat{\sigma}_{ik}^2)}$

对于本实验而言，初始数据既包含离散型变量也包含连续性变量，但在初始数据处理时，我们将离散型变量全部通过独热编码转为了连续性变量，为方便起见，我们这里使用处理后的数据集，统一将其视作连续型变量。因此，我们此处采用了高斯朴素贝叶斯来进行试验。

2.3 LinearSVC

本工作使用线性支持向量机进行试验。支持向量机的核心想法是找到一个超平面，使得可以区分两个不同的类，并使得这个平面的边缘距离最宽。

具体而言，假设训练数据点为 $(x_1,y_1),(x_2,y_2)...(x_n,y_n)$ 支持向量机实际上在解的数学问题为：

$$\begin{aligned}\min_{\beta,\beta_0,\xi_i}\quad & \frac{1}{2}\|\beta\|^2+C\sum_{i=1}^{N}\xi_i \\ \text{s.t.}\quad & y_i(x_i^T\beta+\beta_0)\geq 1-\xi_i,\quad i=1,\dots,N \\ & \xi_i\geq 0,\quad i=1,\dots,N\end{aligned} \tag{3}$$

其拉格朗日对偶形式为：

$$\begin{aligned}L_p=&\frac{1}{2}\|\beta\|^2+C\sum_{i=1}^{N}\xi_i \\ &-\sum_{i=1}^{N}\alpha_i[y_i(x_i^T\beta+\beta_0)-(1-\xi_i)]-\sum_{i=1}^{N}\mu_i\xi_i\end{aligned} \tag{4}$$

通过求解其拉格朗日对偶即可得到最终的解。

在 sklearn 的代码实现中，实际是按照 Loss + Penalty 的方式实现的，即公式 (3) 也可以写作 $\min_{\beta,\beta_0}\sum_{i=1}^{N}[1-y_i f(x_i)]_+ + \frac{\lambda}{2}\|\beta\|^2$，其中 $\lambda=\frac{1}{C}$。这样一来，就可以将 $L(y,f)=[1-y_i f(x_i)]_+$ 部分视作 loss (hinge loss)，$\frac{\lambda}{2}\|\beta\|^2$ 视作 penalty。而 squaed hinge loss 则为 $L(y,f)=([1-y_i f(x_i)]_+)^2$。平方合页损失（squared hinge loss）相比合页损失（hinge loss）更平滑，收敛性更好。

在本实验中，我们采用线性支持向量机，损失函数采用平方合页函数，以平衡模型复杂度和分类效果。

2.4 DCT

决策树是一种以树形结构来展示决策步骤以及分类结果的归纳学习方法。由于本实验是一个分类任务，因此我们以下介绍分类决策树。

分类决策树是自顶向下构建的，每次划分都是寻找最大化降低节点不纯度（Impurity）的划分准则进行划分的。其划分准则为：

$$\begin{aligned}\text{Quality} = &\, Impurity(\text{parent}) \\ &- (\frac{N_{\text{left}}}{N_{\text{parent}}} Impurity(\text{left}) \\ &+ \frac{N_{\text{right}}}{N_{\text{parent}}} Impurity(\text{right}))\end{aligned} \tag{5}$$

在分类决策树中有三种不同的不纯度定义方式，分别为：

- 误分率 (Misclassification rate)：$1 - \max_k p_k$
- 基尼不纯度 (gini impurity)：$\sum_k p_k(1 - p_k)$
- 熵 (entrophy)：$-\sum_k p_k \log p_k$

基尼不纯度是描述当从一个带有标签的集合中抽出两个样本，这两个样本拥有不同标签的概率。此外，基尼不纯度和熵都对节点概率的变化更敏感，也即更倾向于得到“更纯”的节点，这让它们更适合在生长树时使用。

另外树的深度和节点包含样本数目也会对决策树的表现产生影响。这从直观上也很好理解，当树的深度过深或者叶节点包含节点数过少时，决策树更倾向于在决策过程中使用更多的分类准则或最终划分得到的类过小，这都会导致模型可能无法学习到数据的整体趋势，而是在最末尾的决策阶段转向数据细节的学习，最终使得在测试集上的表现结果差。

在本实验中，因为决策树是一个简单且直观的模型，同时因为实验数据同时涉及到连续型变量和离散型变量，采用决策树可以轻松的处理离散型变量，不需要进行其他特殊处理。因为实验数据有关心理健康，其所有变量都具有可解释性，使用决策树产生决策规则可以更直观的让我们认识到哪些变量对决策是更重要的，这种可解释性在这项研究中是重要的。

2.5 GradientBoosting

GBDT (Gradient Boosting Decision Tree)，梯度提升决策树，是一种基于决策树的改进算法。GBDT 构造一组弱分类器，并将这些弱分类器的结果累加得到最终的结果。

GBDT 无论是回归任务还是分类任务都会使用回归树作为基决策树，其核心思想是用这一步的决策树来拟合上一轮训练的残差。GBDT 与梯度下降的想法类似，即每一步通过学习一棵拟合负梯度的树来使得损失函数逐步减小。其具体步骤如下：

1. 设损失函数为 $L(y, f(x))$，则 $f_0(x) = \arg\min_\gamma L(y_i, \gamma)$
2. 假设一共有 M 棵决策树，则对 m 从 1 到 M 依次重复
   (a) 对 $i = 1, 2, \ldots, N$ 分别计算 $r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}$.
   (b) 对 r_{im} 拟合一棵决策树，并给出决策区域 $R_{jm}, j = 1, 2, \ldots, J_m$
   (c) 对 $j = 1, 2, \ldots, J_m$ 计算 $\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma\right)$.
   (d) 更新 $f_m(x)$，$f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$
3. 输出 $f_M(x)$

3 实验

3.1 数据预处理

对于非数值型数据，我们采用了独热编码的方法将其转换为数值型特征。具体来说，对于每个非数值型特征，我们根据其出现的不同类别数量 n，将其编码为 n 个独立的二进制特征。这种方法能够有效地将分类变量纳入模型，同时避免了类别之间的顺序关系对模型的影响。我们还发现训练数据和测试数据的特征列并不完全一致。为了确保模型在训练和预测阶段使用一致的特征空间，我们选取了训练集和测试集的特征列交集作为最终的特征集。这一策略有效避免了因特征不一致而导致的模型训练和预测错误。

数据中不可避免地存在缺失值，我们根据数据类型分别采用了不同的填补方法：

- 数值型数据：使用中位数填补缺失值。中位数对异常值具有较强的鲁棒性，能够有效减少缺失值填补对数据分布的影响.

- 非数值型数据：使用众数填补缺失值。众数是数据中最常见的值，能够较好地反映数据的集中趋势，适用于分类变量的缺失值填补。

最后，为了提高模型的训练效率和预测性能，我们删除了一些明显没有实际含义的列，例如姓名、ID 编号等。这些特征不包含与抑郁症诊断相关的有用信息，去除它们可以减少模型的复杂度，同时避免因噪声数据对模型性能的干扰。

3.2 测量标准

我们选取准确率（Accuracy）、精确率（Precision）、召回率（Recall）以及 F1 得分（F1 Score）作为我们测量的标准。指标计算公式如下：

- **准确率（Accuracy）**：准确率是指模型正确预测的样本数占总样本数的比例，计算公式为：

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

其中，TP 表示真正例（True Positives），TN 表示真负例（True Negatives），FP 表示假正例（False Positives），FN 表示假负例（False Negatives）。

- **精确率（Precision**: 精确率是指模型预测为正的样本中实际为正的比例，计算公式为：

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

它反映了模型预测正类的准确性。

- **召回率（Recall）**：召回率是指所有实际为正的样本中被模型正确预测为正的比例，计算公式为：

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

它反映了模型对正类的覆盖能力。

- **F1 得分（F1 Score）**：F1 得分是精确率和召回率的调和平均数，用于综合评估模型的性能，计算公式为：

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1 得分在 0 到 1 之间，值越高表示模型性能越好。

- **AUC（Area Under the Curve）**：AUC 是指受试者工作特征曲线（Receiver Operating Characteristic Curve，简称 ROC 曲线）下的面积。ROC 曲线是通过将不同阈值下的真正例率（TPR）和假正例率（FPR）绘制在坐标系中得到的。AUC 的值范围在 0 到 1 之间，值越高表示模型的分类性能越好。

这些指标能够从不同角度评估模型的性能，帮助我们全面了解模型的优势和不足。由于训练集规模有限，我们采用五折交叉验证（5-Fold Cross Validation）来测试性能。

3.3 模型细节

- LDA: 未进行降维处理，直接利用样本数据计算类别先验概率，采用 SVD（奇异值分解）方法进行求解，且未引入正则化参数
- GaussianNB：基于样本数据计算先验概率，平滑参数设置为 10^{-9}，以防止方差为零导致的数值问题。
- LinearSVC：正则化参数 C 设为 1，采用 L2 正则化，损失函数选用平方合页损失（squared_hinge），以平衡模型复杂度与分类效果。
- DCT：以基尼不纯度（Gini）作为节点分裂标准，不限制树的最大深度，分裂内部节点所需的最小样本数为 2，叶节点的最小样本数为 1，旨在构建较为灵活的决策树模型。
- GradientBoosting：弱学习器（通常是决策树）数量设为 100，学习率（步长）为 0.1，每棵树的最大深度为 3，分裂内部节点所需的最小样本数为 2，叶节点的最小样本数为 1，以实现模型的高效学习与泛化能力。

3.4 实验结果

从表 1 中可以看出，不同模型在各项性能指标上表现各异。GradientBoosting 在所有指标上均表现出色，其精确率（Precision）为 90.02%，召回率（Recall）为 88.52%，F1 得分为 89.25%，验证集准确率（Val Accuracy）为 **93.72%**，AUC 为 **88.52%**，测试集准确率（Test Accuracy）为 **93.81%**。这些指标均是所有模型中最高的，表明 GradientBoosting 在分类任务中具有较高的准确性和鲁棒性。

LDA 和 DCT 在验证集上的表现较为接近，但整体略逊于 GradientBoosting。LDA 的 F1 得分为 87.47%，验证集准确率为 92.75%，测试集准确率为 92.69%，表现较为均衡。DCT 的 F1 得分为 84.20%，验证集准确率为 90.59%，测试集准确率为 81.75%，在测试集上的表现远弱于 LDA。此

方法	Precision	Recall	F1-score	Val Accuracy	AUC	Test Accuracy
LDA	88.57	86.48	87.47	92.75	86.48	92.69
GaussianNB	82.27	85.99	83.92	89.87	85.99	89.66
LinearSVC	81.50	70.54	74.06	87.12	70.54	87.59
Decision Tree	84.15	84.26	84.20	90.59	84.26	81.75
Gradient Boosting	**90.02**	**88.52**	**89.25**	**93.72**	**88.52**	**93.81**

表 1 不同模型的性能评估结果（%）

外，DCT 是所有模型中验证集和测试集准确率相差最大的，这表明 DCT 方法的泛化性较差。

GaussianNB 和 LinearSVC 在验证集上的表现相对较弱。GaussianNB 的 F1 得分为 83.92%，验证集准确率为 82.27%，测试集准确率为 89.66%，该模型召回率较高（85.99%），但精确率较低（82.27%），**表明 GaussianNB 模型在预测正类时容易误判，导致假正例较多**。LinearSVC 的 F1 得分为 74.06%，验证集准确率为 87.12%，测试集准确率为 87.59%，其召回率仅为 70.54%，**表明该模型在正类的覆盖能力上存在较大不足，LinearSVC 模型倾向于保守地预测正类，导致漏检较多**。

4 讨论

4.1 解决 DCT 模型的泛化性差问题

max_depth	3	7	9	11	15
Val Accuracy	91.31	92.81	92.98	92.80	92.17
Test Accuracy	91.37	92.89	79.44	91.15	77.25

表 2 DCT 最大树深对验证集和测试集准确率影响（%）

min_samples_leaf	1	2	3	4	5
Val Accuracy	91.10	91.24	91.34	91.70	91.88
Test Accuracy	86.10	89.25	90.20	90.40	91.38

表 3 DCT 叶最少样本数对验证集和测试集性能影响（%）

表 1 显示，决策树模型（DCT）在验证集和测试集上的性能差异较大。例如，验证集的准确率可达 91.05%，而测试集的准确率仅为 87.72%，这种差异表明模型可能对训练集过拟合。我们认为，这主要是由于决策树模型本身未对复杂度进行限制和惩罚，导致其在训练集上拟合过度，而无法很好地泛化到测试集。

为解决这一问题，我们从两个关键参数入手：最大深度（max_depth）和最小叶节点样本数（min_samples_leaf），通过调整这些参数来限制模型的复杂度，从而改善其泛化能力。

表 2 展示了不同最大深度对验证集和测试集准确率的影响。当最大深度较小时（如 3 和 7），验证集和测试集的准确率较为接近，且均处于较高水平。例如，当最大深度为 7 时，验证集准确率达到 92.81%，测试集准确率为 92.89%，二者差异仅为 0.08 个百分点。然而，随着最大深度的增加（如 9、11 和 15），测试集准确率出现了显著下降。特别是当最大深度为 15 时，测试集准确率仅为 77.25%，远低于验证集的 92.17%。这表明，过大的最大深度会使模型过于复杂，导致过拟合，从而降低其在测试集上的泛化性能。

表 3 展示了不同最小叶节点样本数对验证集和测试集准确率的影响。随着最小叶节点样本数的增加，验证集准确率呈先升后降的趋势，而测试集准确率则持续上升。当最小叶节点样本数为 5 时，验证集准确率达到最高值 91.88%，同时测试集准确率也提升至 91.38%。这说明，适当增加最小叶节点样本数可以有效限制模型的复杂度，减少过拟合现象，从而提高模型在测试集上的泛化能力。

综合以上分析，我们可以得出以下结论：决策树模型的泛化性差主要是由于其复杂度未受到有效限制，导致过拟合。通过合理调整最大深度和最小叶节点样本数这两个关键参数，可以有效改善模型的泛化能力。具体来说，选择适当的最大深度（如 7）和最小叶节点样本数（如 5），可以在保证模型复杂度适中的同时，提高其在测试集上的准确率，从而实现更好的泛化效果。

此外，我们还可以进一步考虑引入其他正则化方法，如剪枝技术（如代价复杂度剪枝）或使用集成方法（如随机森林），以进一步增强模型的泛化能力。这些方法可以在限制单个决策树复杂度的基础上，通过组合多个决策树来提高模型的稳定性和准确性，从而更好地应对复杂的数据集和泛化问题。

4.2 LDA 优势分析

LDA 作为一种经典的线性分类方法，在本研究中展现出了优异的性能和良好的测试集泛化能力，甚至超过了决策树、支持向量机等更为复杂

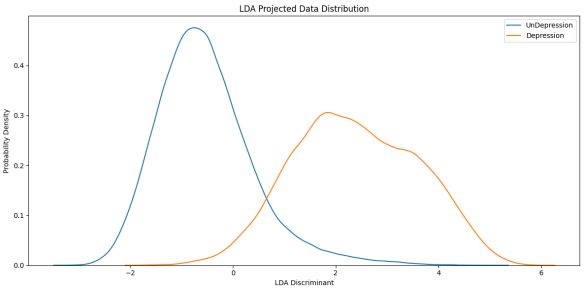

图 1 LDA 降维概率分布

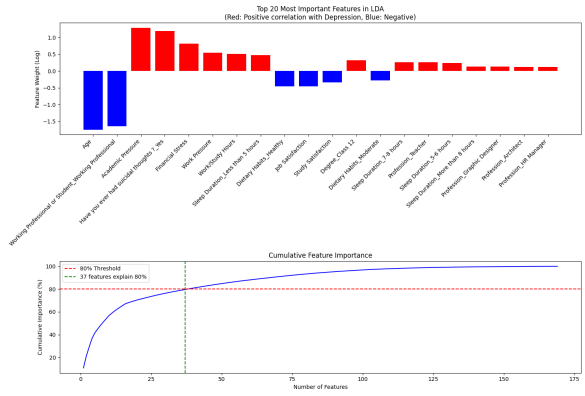

图 2 重要解释变量

的模型，并且与集成学习方法的性能相当。这一现象引发了我们对其优势的深入分析。

如图 1所示，经过 LDA 降维处理后，阴性（No Depression）和阳性（Depression）两类样本的特征均值和标准差分别为：$\mu_0 = -0.5316, \mu_1 = 2.3939, \sigma_0 = 0.9470, \sigma_1 = 1.2101$，从图中可以清晰地看到，隐性样本（No Depression）在降维后的分布方差较小，且两类样本在降维后的空间中基本能够很好地分开。这表明 LDA 能够有效地将不同类别的数据投影到一个低维空间中，使得类别之间的区分度更高，从而为分类任务提供了良好的基础。

如图 2所示，在去除掉姓名等无关噪声信息后，前 37 个特征能够解释 80% 的抑郁症现象，这表明 LDA 在特征选择和降维方面具有显著的优势。进一步分析前 20 个对抑郁症解释最重要的变量，我们发现以下因素对 LDA 的判定影响较大：

- 学业压力（Academic Pressure）：较高的学业压力与抑郁症的发生密切相关。
- 自杀倾向（Have you ever had suicidal thoughts?_Yes）：有自杀倾向的个体更容易被 LDA 判定为抑郁症患者。
- 经济压力（Financial Stress）和工作压力（Work Pressure）：经济和工作方面的压力较大时，抑郁症的判定概率增加。
- 工作学习时长（Work/Study Hours）：长时间的工作或学习可能导致抑郁症的发生。
- 睡眠时长（Sleep Duration_Less than 5 hours）：睡眠时间少于 5 小时的个体更容易被判定为抑郁症患者。

这些因素与人们对于抑郁症的一般认知相符，进一步验证了 LDA 在特征解释方面的合理性。

在 LDA 模型中，对抑郁症判定影响最大的两个因素是年龄和是否为学生（Working Professional or Student）。这一发现揭示了抑郁症的低龄化和学生化趋势，提示我们在关注抑郁症问题时，需要特别重视年轻群体和学生群体。此外，我们还发现以下因素对抑郁症的发生有显著影响：

- 健康的饮食习惯（Dietary Habits_Healthy）：保持健康饮食习惯的人群患抑郁症的比例较低。
- 工作满意度（Job Satisfaction）：对工作感到满意的人群更少患抑郁症。

这表明生活作息和工作状态可能对抑郁症的发生起到重要作用。LDA 能够有效地识别这些关键因素，并将其纳入分类决策中，从而提高了模型的准确性和泛化能力。

综上所述，LDA 在本研究中展现出的优势主要源于其对数据分布的良好适应性、强大的特征解释能力和对关键因素的有效识别。这些优势使得 LDA 能够在简单的线性框架下，实现对复杂数据的有效分类和泛化。未来，我们可以在保持 LDA 优势的基础上，进一步探索与其他方法的结合，以进一步提升模型的性能。

4.3 样本不平衡分析

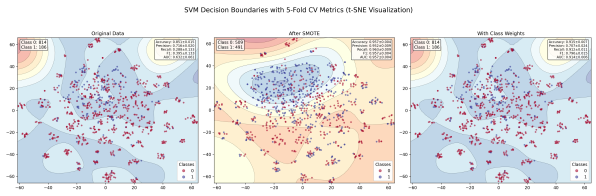

图 3 T-SNE 降维后的 SVM 分解面可视化结果

在主实验中，LinearSVC 方法的表现相对较差，我们认为这可能与数据中两类样本的不平衡

Method	Precession	Recall	F1-Score	Val Accuracy	Test Accuracy
LinearSVC	71.6	28.8	39.5	85.1	85.8
LinearSVC+Class Weight	70.7	91.2	79.6	91.5	90.6
LinearSVC+SMOTE	95.2	96.3	95.7	**95.7**	**92.9**

表 4 LinearSVC 引入 SMOTE 和类权重后的性能比较

有关。样本不平衡会导致决策边界容易被多数类推向少数类，从而使得少数类的分类效果变差。因此，解决数据平衡问题可能在一定程度上改善这一问题。

基于上述分析，我们尝试了两种方法来解决样本不平衡问题：类加权（Class Weighting）和合成少数类过采样技术（Synthetic Minority Oversampling Technique，SMOTE）。

类加权方法通过在目标函数中引入权重来惩罚少数类的错误分类。具体来说，优化目标函数可以表示为

$$min\frac{1}{2}||w||^2 + C\sum_{i}^{n}(b_{y_i}\xi_i)$$

其中，$b_{y_i} = \frac{n_{y_i}}{\sum_j n_j}$ 表示不同类的权重，n_{y_i} 是第 y_i 类的样本数量，$\sum_j n_j$ 是所有类的样本总数。通过这种方式，模型在训练过程中会更加关注少数类的样本，从而减少对少数类的误分类。

SMOTE 方法通过最近邻方法为少数类样本生成新的样本。具体来说，它选择少数类样本的最近邻，并通过线性插值生成新的少数类样本，从而将分类边界推离少数类。这种方法可以有效增加少数类的样本数量，改善样本不平衡问题。

如图 3所示，通过引入类加权和 SMOTE 方法，我们成功平衡了样本数量。在五折交叉验证中，模型的准确性大幅提升，尤其是召回率（Recall）的提升，这对于抑郁症诊断具有重要的现实意义。

表 4展示了具体的结果。可以看到，这两种方法都显著提升了验证集和测试集上的性能。例如，经过 SMOTE 处理后，验证集的准确率从 87.12% 提升到 91.50%，测试集的准确率从 87.12% 提升到 90.80%。同时，召回率也从 70.54% 提升到 85.20%，这表明模型对少数类的分类能力得到了显著增强。

此外，考虑到为了 TSNE 可视化，我们只选取了 1000 个样本进行训练，这种方法不仅提升了模型的性能，还大大提高了训练效率。

通过引入类加权和 SMOTE 方法，我们有效地解决了样本不平衡问题，显著提升了 LinearSVC 模型的性能，尤其是在少数类的分类能力上。这些改进不仅提高了模型的准确性和召回率，还为抑郁症诊断提供了更可靠的工具。未来，我们可以在更多数据集上验证这些方法的效果，并探索其他可能的改进策略。

5 结论

本研究通过对比多种分类模型（包括 LDA、GaussianNB、LinearSVC、决策树和梯度提升树）在抑郁症诊断任务中的性能，深入分析了不同模型的优势与局限，并针对样本不平衡问题提出了有效的解决方案。研究结果表明，LDA 模型在测试集上展现出了良好的泛化能力，其性能甚至超过了某些更复杂的模型，如决策树和支持向量机，并且与集成学习方法的性能相当。这一发现表明，简单的线性方法在某些情况下可以有效地处理复杂的分类问题，尤其是在数据分布较为线性可分的情况下。

进一步分析发现，LDA 模型的优势主要源于其对数据分布的良好适应性以及强大的特征解释能力。通过 LDA 降维，我们发现抑郁症样本在降维后的空间中分布较为紧凑，且与非抑郁症样本能够较好地分离。此外，LDA 模型能够有效地识别出对抑郁症诊断最具影响力的特征，如学业压力、自杀倾向、经济压力等，这些特征与抑郁症的临床表现高度相关，进一步验证了 LDA 模型在特征选择和解释方面的有效性。

针对样本不平衡问题，我们尝试了类加权和 SMOTE 两种方法来改善模型性能。实验结果表明，这两种方法均能显著提高模型在少数类上的分类效果，尤其是召回率的提升，对于抑郁症诊断具有重要的现实意义。通过平衡样本数量，模型的泛化能力得到了显著增强，同时训练效率也得到了提高。这表明，解决样本不平衡问题对于提高模型性能至关重要，尤其是在处理少数类样本时。

然而，尽管 LDA 模型在本研究中表现优异，但我们也注意到其在某些方面仍存在局限性。例如，LDA 假设数据在低维空间中是线性可分的，对于非线性可分的数据，其性能可能会受到限制。此外，LDA 模型对特征的解释能力虽然较强，但其解释结果仍需结合领域知识进行进一步验证和解释。未来的研究可以考虑探索更复杂的非线性方法，如深度学习模型，以进一步提高抑郁症诊断的准确性和泛化能力。同时，结合多种模型的优势，开发混合模型或集成学习方法，可能会为

抑郁症诊断提供更可靠的解决方案。

总之，本研究通过对比和分析多种分类模型，深入探讨了 LDA 模型在抑郁症诊断中的优势，并提出了有效的解决方案来应对样本不平衡问题。这些发现不仅为抑郁症的早期诊断提供了新的视角和工具，也为未来的研究提供了有价值的参考和方向。

参考文献（References）