任务分析：

本次任务聚焦于比较 Logistic Regression, SVM 与 XGBoost 的分类性能，讨论在这个问题中为什么一些算法表现得更好？

根据 NFL（No Free Lunch）定理：

假设一个算法为 a，而随机胡猜的算法为 b，假设样本空间为γ和假设空间为 H 都是离散的。令 P(h|X,a)表示算法 a 基于训练数据 X 产生假设 h 的概率，再令 f 代表希望的真实目标函数。a 的训练集外误差，即 a 在训练集之外的所有样本上的误差为:

$$E_{\text{ote}}(a \mid X,f) = \sum_{h} \sum_{x \in \gamma - X} P(x) I(h(x) \neq f(x)) P(h \mid X,a)$$

其中$I(\cdot)$为指示函数，若括号内为真取 1 否则取 0

考虑二分类问题，且真实目标函数可以是任何函数$\gamma \mapsto$ {0,1},函数空间为$\{0,1\}^{|\gamma|}$(|γ|指样本空间中元素个数，对所有可能的 f 按均匀分布对误差求和，有：

$$\begin{aligned}
&\sum_{f} E_{\text{ote}}(a \mid X,f) \\
&= \sum_{f} \sum_{h} \sum_{x \in \gamma - X} P(x) I(h(x) \neq f(x)) P(h \mid X,a) \\
&= \sum_{x \in \gamma - X} P(x) \sum_{h} P(h \mid X,a) \frac{1}{2} 2^{|\gamma|} \\
&= 2^{|\gamma|-1} \sum_{x \in \gamma - X} P(x) \cdot 1
\end{aligned}$$

可以看到总误差与算法无关，对于任何两个算法 a 和 b 都有

$$\sum_{f} E_{ote}(a \mid X,f) = \sum_{f} E_{ote}(b \mid X,f)$$

因此如果没有对实际问题的假设和指标选择，这些模型都是一样好的，不同的前提和指标下也会得到不同的结论。因此本文的分析基于以下的假设和选择：

我们需要预先假设**样本数据之间服从 i.i.d（独立同分布）**

可以选择①准确率(Accuracy),②基于召回率(Recall)和精确率(Precise)的 F1-Score③只对于二分类问题适用的 ROC 曲线和 AUC 值作为我们的模型指标。它们从不同的目标需求对模型性能进行了评测。考虑到本次任务并没有对正类负类样本有所偏好，我选择准确率，F1-Score,AUC 三个指标对模型的性能做出综合评测。

一.数据集分析

本次实验的训练集有正类样本 1000 个：

X,Y, Z 方向均值为: [0.00771333 0.00103692 -0.01089041]

X,Y, Z 方向方差为: [1.08476788 0.74020916 0.74236275]

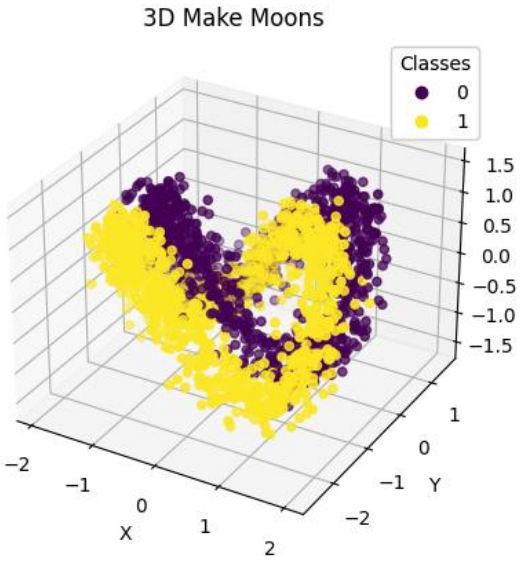
负类样本 1000 个：

X,Y, Z 方向均值为: [0.00422131 -0.99423378 -0.00642876]

X,Y, Z 方向方差为: [1.08153784 0.73402093 0.72859756]

可以看出，数据集**未出现类不均衡问题**，正负类在**均值上差异较大**，但是**方差的差异很小**，正负类数据集的分布具有一定的相似性。

下面是训练集的打印情况：



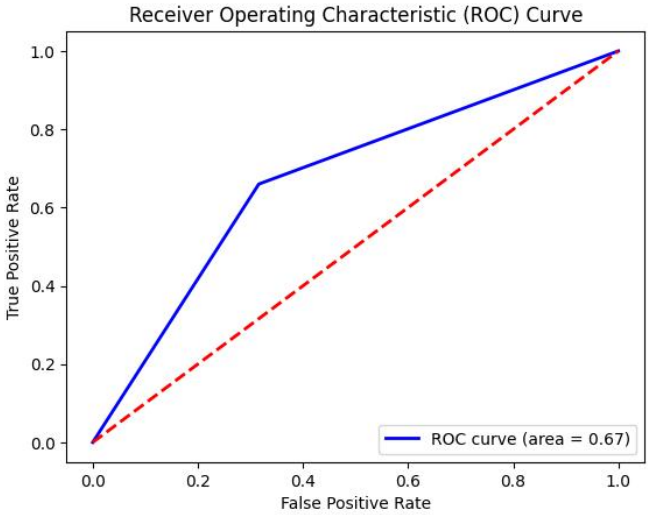
实际结果与我们的分析相符

测试集（正类 250，负类 250）与训练集使用同一代码随机生成，分析过程类似。

二. 模型预测结果

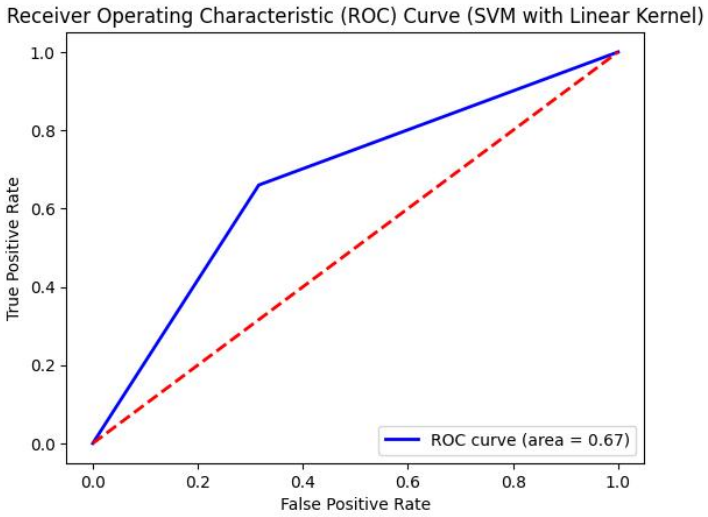
1.Logistic Regression

性能指标	数值
准确率	0.672
F1-Score	0.67
AUC	0.672



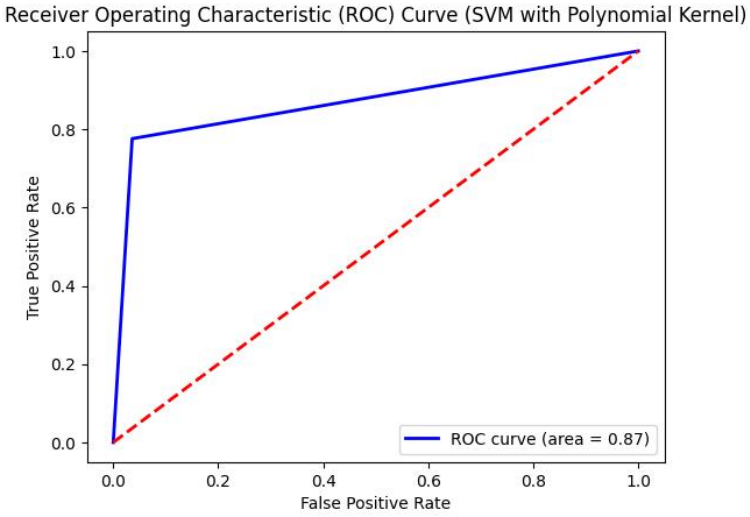
2.SVM With Linear kernel

性能指标	数值
准确率	0.672
F1-Score	0.67
AUC	0.67



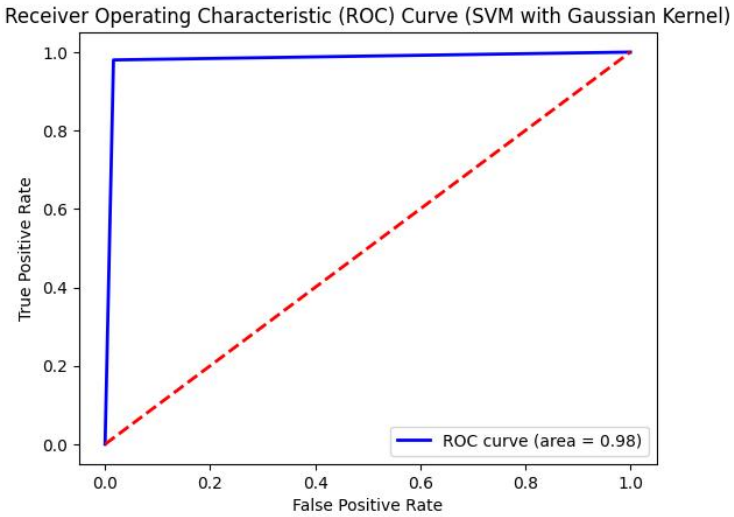
3.SVM With Polynomial kernel

性能指标	数值
准确率	0.870
F1-Score	0.87
AUC	0.87



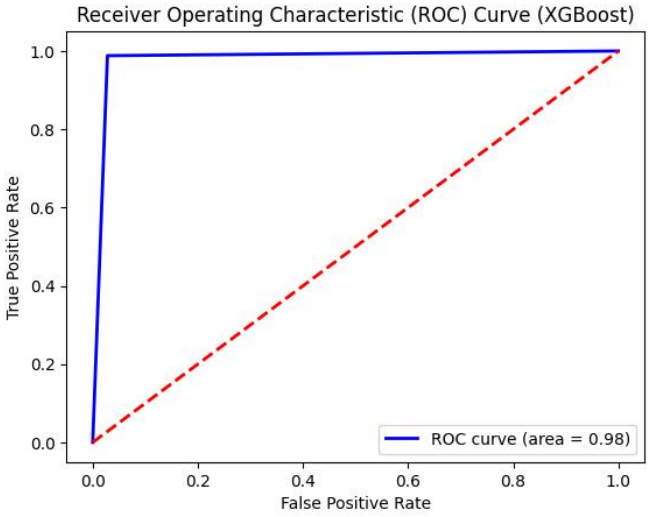
4.SVM With Gaussian kernel

性能指标	数值
准确率	0.982
F1-Score	0.98
AUC	0.98



5.XGboost

性能指标	数值
准确率	0.980
F1-Score	0.98
AUC	0.98



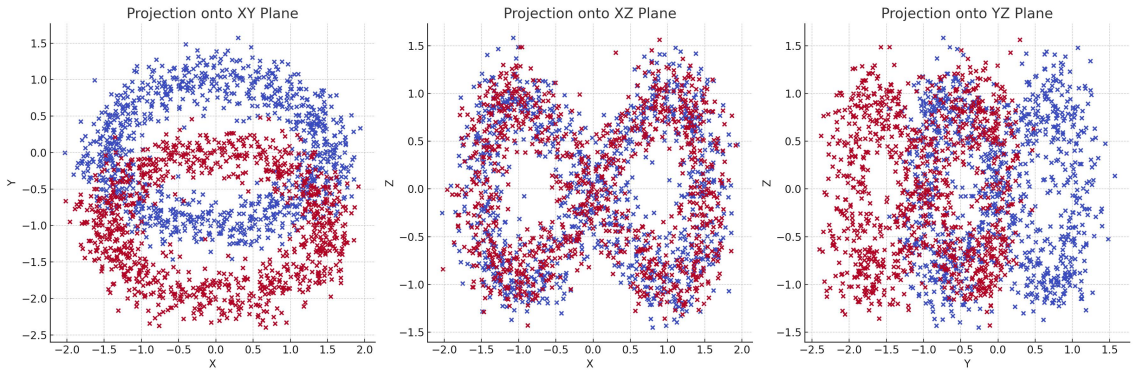
三. 结果分析

可以看出性能的优劣如下：

XGBoost≈高斯核的 SVM 支持向量机>多项式核的 SVM 支持向量机>线性核的支持向量机≈Logistic 回归

1）我们先分析高斯核，多项式核比线性核核 Logistic 回归模型性能好的原因：

对于这个问题，正负类样本无法通过通过简单的线性分界面分开，原因是在数据生成过程中，每个数据都加入了一定的噪声，这导致在分界面区域正负类样本有非常严重的交叉：



如果把原始三维数据投影在 XY,XZ,YZ 平面时候，我们会发现在交界区域，正负类样本交叉非常严重。因此我们无法通过超平面，简单的 Logistic 回归将这两种样本分开。因此线性核和 Logistic 回归模型对于该问题的效果都并不好。

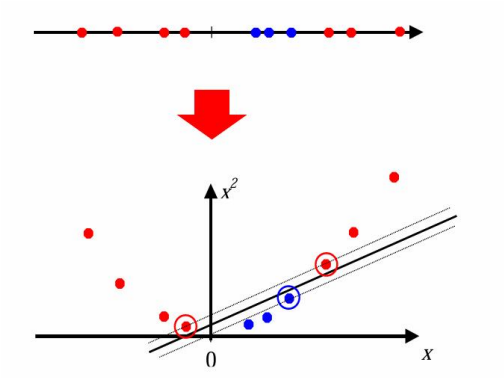
我们需要更复杂的分类器。

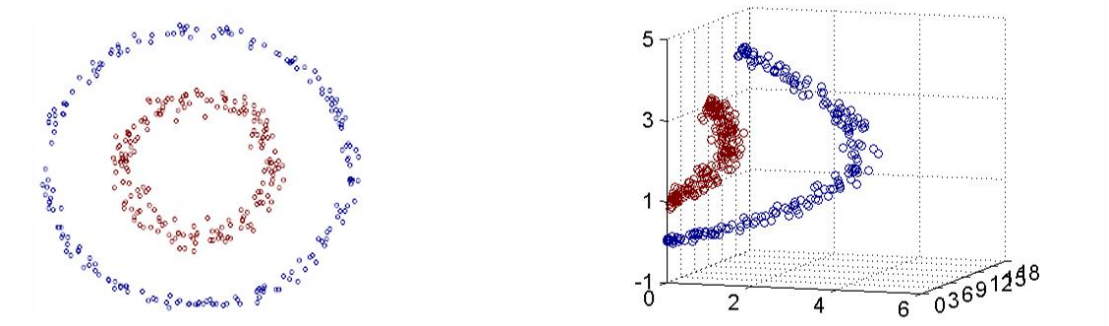
$$y(x)=w^Tx+b \quad \Longrightarrow \quad y(x)=w^T\phi(x)+b$$

$$y(x)=\sum_{n=1}^{N}a_nt_nx_n^Tx+b \Rightarrow y(x)=\sum_{n=1}^{N}a_nt_nk(x,x_n)+b$$

对于线性的 SVM 支持向量机，我们可以设置一个映射利用核函数 $k(x,x_n)=\phi(x_n)^T\phi(x)$ 将低维度的特征映射到高维度时，便可能寻找到一个分界线将正负类样本分类。

这两个例子便很好地反映了这个思想：





线性不可分　　高维下线性可分

利用一个固定的非线性映射将数据映射到特征空间学习的线性分类器等价于基于原始数据学习的非线性分类器。我们采用了多项式内核和高斯内核将原始数据点映射到更高的维度上，可以看出，准确率，F1-Score，AUC 值都都有明显的提升。而高斯核可以将样本映射到一个无穷维的空间中，可以应对更加复杂的数据，多项式内核只是将样本映射到有限维度的空间可能还不足以寻找到一个最佳的分类界面。因此高斯内核的指标是最好的。

2）下面分析 XGBoost 模型性能好的原因：

XGBoost 将 CART 回归树作为基分类器，不断生成新的 CART 回归树，每生成一颗树即是在学习一个新的函数，这个函数将每个样本映射到唯一确定的一个叶子节点中，同一叶子节点中的所有样本共享相同的预测值，函数的目标则是去拟合所有叶子节点中样本的历史残差；损失函数可以是与 CART 回归树相同的均方误差，也可以是交叉熵. XGBoost 可以生成多个 CART 回归树，经过一段时间的训练后，可以找到更为复杂精确的分界面。因此分类效果非常不错。

3）为什么我们没有深究过拟合？

不用太考虑过拟合问题也是 XGBoost 等分类器性能遥遥领先的原因。因为这个问题的训练集和测试集非常特殊，测试集和训练集公用了一套数据点生成函数，两者分布非常近似。

因此如果模型能在训练集上找到更加细致的分类界面（即使出现了过拟合），在测试集上表现也不会差。