实验数据描述

我们在 kaggle 官网上下载了官方的数据集进行实验。这个数据集大约有 13000 个样本，13 个特征参数，1 个标签（待预测项）。这 13 个参数均会对乘客的去向产生影响。数据集分为训练集和测试集合，训练集包含大约三分之二（~8700）的数据（train.csv），测试集包含大约三分之一（~4300）的数据（test.csv）。这些数据中，每一个数据元素包含 14 个属性，1 个是标签（即是否被传送到其他星球），13 个是特征参数，均会对结果产生影响。其中，

passangerID 是每一位乘客的 ID，每个 ID 的形式都表示乘客与一个团体一起旅行，小组的成员通常是家庭成员。比如 0013_01，表示是 0013 组 01 号成员。

HomePlanet 是指乘客离开的星球，通常是他们永久居住的星球。

CryoSleep 指的是乘客是否在航行期间保持假死的状态。处于假死状态的乘客将被限制在他们的客舱内。

Cabin 指的是乘客所住的仓号。采用的形式是 deck/num/side(P/S)。

Destination 表示乘客将要前往的星球。

Age 表示乘客的年龄。

VIP 表示乘客是否为 VIP(TRUE/FALSE)。

RoomService, FoodCourt, ShoppingMall. Spa 表示乘客在飞船上许多豪华设施中为购买服务交纳的金额。

Name 表示乘客的名字。

Transported 表示乘客是否被传送到另一个维度，也是整个数据集中的标签列。

缺失值和数据类型分析

为了保证训练的准确性，首先要做的便是对训练数据集的缺失值分析和数据类型分析。

使用 isna().sum()函数统计出，整个训练数据集中一共有 2324 个缺失的数据。针对每一个特征参数进行分析，得出的结果如下表所示：

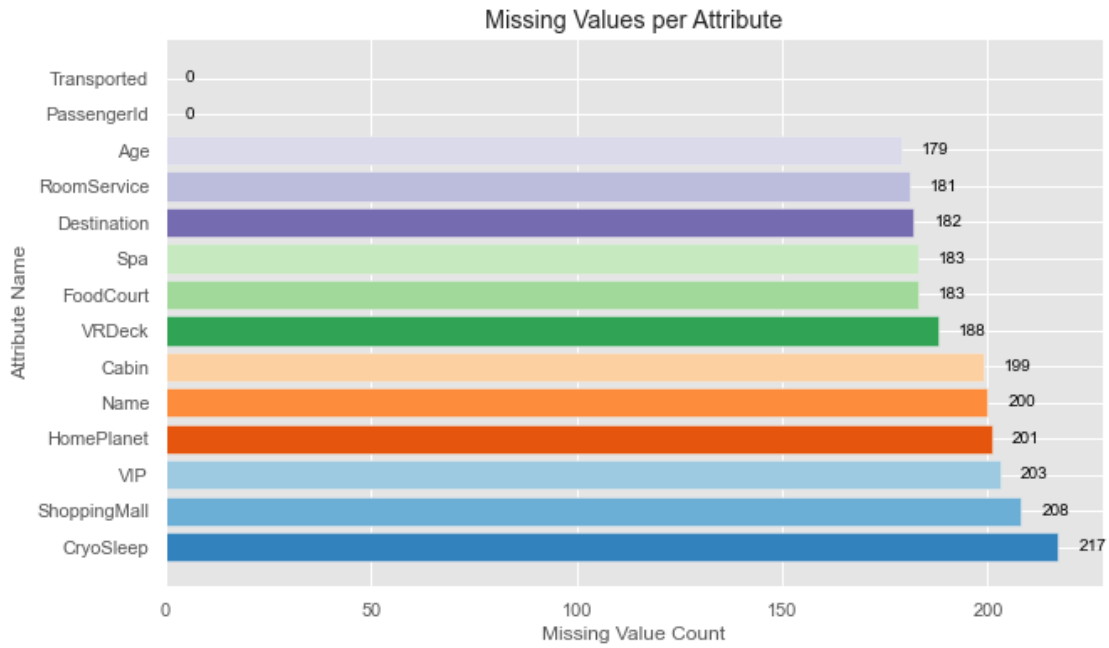

图 1 各属性缺失值数量

从缺失值数量可以看出，部分属性的缺失值比较多，其中 Age 属性缺失值最多，达到了 179 个，这可

能意味着该旅游航班的乘客年龄不是必填项，或者在采集数据时出现了一些问题。其他缺失值较多的属性包括 CryoSleep、ShoppingMall、VIP、HomePlanet 等，这些属性的缺失值数量可能需要进一步了解其缺失原因和对数据分析的影响。基本所有的属性均含有缺失值，所以处理缺失值是一个至关重要的问题。

接着本文又对数据集进行了重复值检查，幸运的是，所有的训练集和测试集均没有重复的数据。

针对实验数据的描述我们可以发现，类似于 PassangerID 这种属于字符型变量，应该是属于描述性属性，而对于金额这一类数据，应该是连续性浮点型或者整形，对于他们的处理方式也该是不同的，所以本文还做了如下的分析：

首先对于每一种属性进行唯一值统计：结果如下表所示：

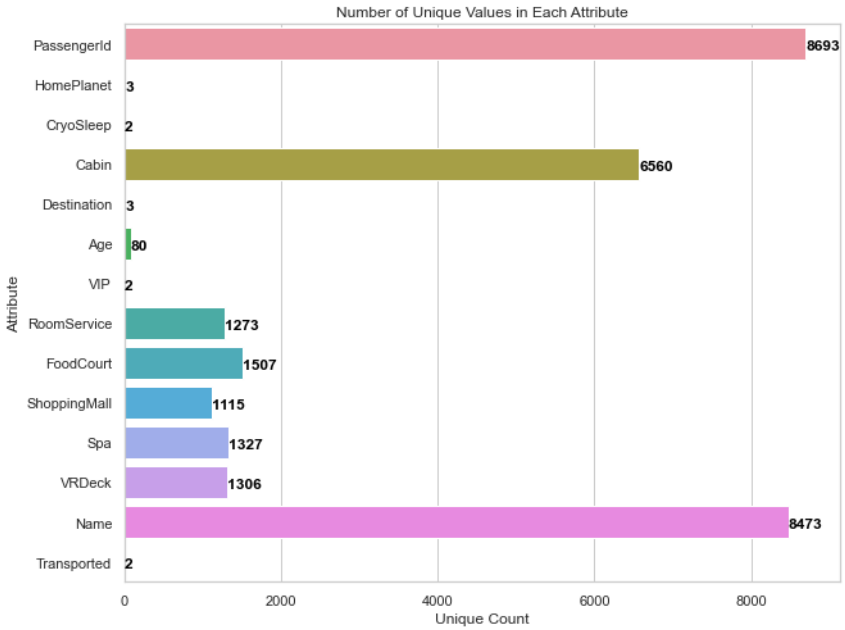

图 2 各属性的唯一值数量

从唯一值数量范围可以看出，不同属性的唯一值数量范围差异较大。其中，PassengerId 和 Name 属性的唯一值数量最大，超过了 8000 个，而其他属性的唯一值数量范围则较小。这可能意味着 PassengerId 和 Name 属性具有更加细致的区分度，而其他属性则存在较大的重复值。

通过分析每个属性的唯一值数量范围，我们可以发现 HomePlanet、CryoSleep、Destination、VIP 和 Transported 等属性的唯一值数量范围都比较小，这说明这些属性只包含了少量的离散值，可能是限定在了某种范围内的选项。而 Age、Cabin、RoomService、FoodCourt、ShoppingMall、Spa 和 VRDeck 等属性的唯一值数量范围更大，可能包含了更多的连续值或离散值。

对于每一个属性的属性类型，如下表所示：

object	PassengerId, HomePlanet, CryoSleep, Cabin, Destination, VIP, Name
float64	Age, RoomService, FoodCourt, ShoppingMall, Spa, VRDeck,
bool	Transported

表 1 各属性的属性类型

PassengerId、HomePlanet、CryoSleep、Cabin、Destination、VIP 和 Name 等属性的数据类型为 object，这可能意味着这些属性包含了文本或字符串值，例如乘客的姓名、舱位、目的地等。Age、RoomService、FoodCourt、ShoppingMall、Spa 和 VRDeck 等属性的数据类型为 float64，这可能意味着这些属性包含了浮点数或数字值，例如乘客的年龄、房间服务次数、食品广场次数等。最后，Transported 属性的数据类型为 bool，这可能意味着这个属性只包含了两个可选值，例如乘客是否已经被传送到另一星球。

对于属性不同的值仅有 1-10 的属性，显然他们类别特征属性的变量，即他们的取值种类数目特别少，像是否选择假死睡眠、VIP、是否被传送，只有是和否两种选择；而对于超过 8000 的属性，即每条记录有

一个不同的属性，即每个人独一无二的属性，类似于人的名字和乘客编号。而对于年龄，用户的付费服务（RoomService，FoodCourt，ShoppingMall，Spa，VRdeck）这些数据记录的是连续的数值型，年龄较为特殊，都是整数且均在 0 到 80 之间。这里最需要注意的是甲板（Deck）类型，这里分成三组数据，分别是 deck/num/side，deck 表示甲板号，num 表示编号，side 表示两侧（仅有两个取值），对于这个数据处理要采用特别的方式。数据的更详细的分析如下文所示。由于姓名，乘客编号和仓号这三个参数可以唯一确定一个乘客，所以我们称之为定性特征。

模型建立的初步尝试

缺失值处理

在数据类型的分析中，可以发现数据集的数据可以分为数值型和非数值型。对于这两种数据类型，本文尝试以两种不同的填补方式对其进行填补。对于数值型变量，经过分析发现他们都是连续型变量，因此采用均值填充的方法。而对于非数值型的 object 变量，除去 passengerID 和 name 这两个没有缺失值的属性，其余均使用众数填补法进行填充，具体各个属性的填补方法见下表：

填补方法	属性
均值填补	Age, RoomService, FoodCourt, ShoppingMall, Spa, VRDeck
众数填补	Destination, HomePlanet, CryoSleep, VIP, Cabin

表 2 缺失值填补方法

经过这样的填补，无论是测试集还是训练集，所有的缺失值全都被填补完毕了。

数据预处理

同时，经分析可得，Name 列是乘客的姓名，而姓名对于是否传送到另外一个领域是没有影响的，所以本文将这一列做删除的处理。

对于非数值型变量，本文采用 LabelEncoder 类的 fit_transform()方法对其进行转换，转换的属性列分别为 Destination, HomePlanet, Cabin。同时，将 true 和 false 类的 bool 属性列转换为 0 和 1，方便后续的处理。具体方法分类见下表所示：

填补方法	属性
fit_transform()	Destination, HomePlanet, Cabin
$(true, false) \rightarrow (1,0)$	VIP, CryoSleep, transported

表 3 非数值型变量转换方法

分类器训练

这里使用 lazypredict 方法将 27 种分类器模型均进行训练，其中的模型有 LGBMClassifier, RandomForestClassifier, ExtraTreesClassifier 等各种 sklearn 包中的分类器方法，得到的分类准确度如下图所示：

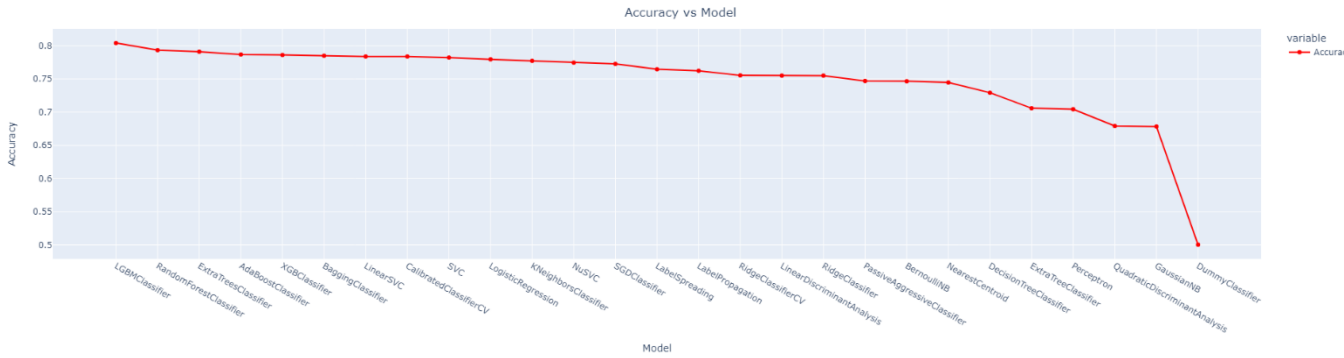

图 3 各种分类器结果对比

从表中可以看出，最好的模型是 LGBMClassifier，它在所有指标上的表现都最好，准确度、平衡准确度、ROC AUC 和 F1 分数均为 0.80，所花费的时间也比其他模型少。

其次，RandomForestClassifier、ExtraTreesClassifier 和 AdaBoostClassifier 在所有指标上的表现也很接近，均为 0.79，但所花费的时间略有不同。

LogisticRegression 和 KNeighborsClassifier 的表现也很不错，它们的准确度、平衡准确度、ROC AUC 和 F1 分数均为 0.78。此外，这两个模型所花费的时间也相对较短。

最后，DummyClassifier 表现最差，准确度、平衡准确度、ROC AUC 和 F1 分数都很低，只有 0.50，而所花费的时间则相对较短。